

Tworzenie stron i aplikacji internetowych oraz baz danych i administrowanie nimi

INF.03



Podręcznik do nauki zawodu

- TECHNIK INFORMATYK
- TECHNIK PROGRAMISTA

Autorzy: Agnieszka Klekot, Tomasz Klekot

Typ szkoły: **technikum**

Zawód: **technik informatyk, technik programista**

Kwalifikacja: INF.03. Tworzenie i administrowanie stronami i aplikacjami internetowymi oraz bazami danych

© Copyright by Wydawnictwa Szkolne i Pedagogiczne
Warszawa 2020

Wydanie I (2020)

ISBN 978-83-02-19146-6 (część 1) (podręcznik w postaci elektronicznej – flipbook)

ISBN 978-83-02-19148-0 (całość) (podręcznik w postaci elektronicznej – flipbook)

ISBN 978-83-02-19076-6 (część 1) (podręcznik w postaci papierowej)

ISBN 978-83-02-19078-0 (całość) (podręcznik w postaci papierowej)

Opracowanie merytoryczne i redakcyjne: **Tomasz Kompanowski** (redaktor koordynator),
Sebastian Jabłoński (redaktor merytoryczny)

Redakcja językowa: **Tomasz Karpowicz**

Redakcja techniczna: **Elżbieta Walczak**

Projekt okładki: **Dominik Krajewski**

Ilustracje: **Roman Kirilenko**

Skład i łamanie: **Ledor**

Wydawnictwa Szkolne i Pedagogiczne Spółka Akcyjna

00-807 Warszawa, Aleje Jerozolimskie 96

KRS 0000595068

Tel.: 22 576 25 00

Infolinia: 801 220 555

www.wsip.pl

Druk i oprawa: Drukarnia Pozkał

Publikacja, którą nabyłaś / nabyłeś, jest dziełem twórcy i wydawcy. Prosimy, abyś przestrzegała / przestrzegał praw, jakie im przysługują. Jej zawartość możesz udostępnić nieodpłatnie osobom bliskim lub osobiście znanym. Ale nie publikuj jej w internecie. Jeśli cytujesz jej fragmenty, nie zmieniaj ich treści i koniecznie zaznacz, czyje to dzieło. A kopiując jej część, rób to jedynie na użytek osobisty.

prawolubni


Szanujmy cudzą własność i prawo.

Więcej na www.legalnakultura.pl

Polska Izba Książki

I. Bezpieczeństwo i higiena pracy na stanowisku komputerowym

- 1.** Zagrożenia związane z wykonywaniem zadań zawodowych 6

II. Podstawowe pojęcia dotyczące stron i aplikacji internetowych

- 2.** Podstawy komunikacji w sieci internet 14
3. Technologie webowe 20

III. Tworzenie stron internetowych w języku HTML

- 4.** Struktura dokumentu HTML 28
5. Podstawowe znaczniki i atrybuty w języku HTML 34
6. Barwa jako wartość atrybutów HTML 50
7. Podział strony i obsługa danych w języku HTML 55
8. Grafika i multimedia w dokumentach HTML 78

IV. Kaskadowe arkusze stylów CSS

- 9.** Kaskadowość stylów w dokumentach HTML 90
10. Selektory, właściwości oraz wartości w CSS 98
11. Model blokowy w CSS 134
12. Pozycjonowanie elementów w CSS 140
13. Praktyczne zastosowania CSS i HTML 143

V. Systemy zarządzania treścią – CMS

- 14.** Publikacja stron i aplikacji internetowych 156
15. Rodzaje systemów zarządzania treścią 161
16. Instalacja CMS w usłudze hostingowej 165
17. Instalacja CMS na lokalnym serwerze 182
18. Personalizacja i zarządzanie systemem CMS Joomla! 190

VI. Projektowanie grafiki komputerowej

- 19.** Obraz cyfrowy 206
20. Edytor grafiki rastrowej GIMP 212
21. Edytor grafiki wektorowej Inkscape 220
22. Edycja gotowych obrazów cyfrowych 231

VII. Multimedia na stronach internetowych

- 23.** Animacja na stronach internetowych 242
24. Video na stronach internetowych 258
25. Dźwięk na stronach internetowych 264

VIII. Edytory WYSIWYG

- 26.** Rodzaje edytorów WYSIWYG 272
27. Tworzenie dokumentów w edytorach WYSIWYG 275

IX. Wykonywanie stron internetowych zgodnie z projektem

- 28.** Zasady projektowania stron internetowych 282
29. Projekt strony internetowej 285
30. Wykonywanie stron zgodnie z projektem 301

X. Testowanie i optymalizowanie stron oraz aplikacji internetowych

- 31.** Weryfikacja oraz walidacja stron internetowych 306
32. Pozycjonowanie, optymalizacja oraz dostępność stron i aplikacji internetowych 314
33. Bezpieczeństwo stron internetowych 322

Załącznik 328

Wykaz podstawowych pojęć w językach: polskim, angielskim i niemieckim 333

I. Bezpieczeństwo i higiena pracy na stanowisku komputerowym

- Środowisko pracy informatyka / programisty
- Zagrożenia związane z wykonywaniem zadań zawodowych
- Ergonomia na stanowisku komputerowym
- Zasady bhp na stanowisku komputerowym

Na organizację każdego stanowiska pracy wpływają m.in. zadania zawodowe realizowane przez pracownika na danym stanowisku, a także odpowiednie akty prawne. Zależnie od rodzaju zadań i specyfiki wyposażenia stanowiska pracy określa się również pozycję pracy pracownika. Wszystkie zadania zawodowe informatyka / programisty wyraźnie wskazują na to, że podstawową pozycją pracy jest pozycja siedząca, a stanowisko pracy zbliża się specyfiką do stanowiska biurowego wyposażonego w komputer.

1

Zagrożenia związane z wykonywaniem zadań zawodowych

ZAGADNIENIA

- Czynniki tworzące środowisko pracy
- Wpływ czynników środowiskowych na uciążliwość pracy
- Zagrożenia zdrowia i życia na stanowisku pracy
- Indywidualne i zbiorowe środki ochrony przed zagrożeniami

Na każde środowisko pracy oddziałuje grupa czynników występujących w jego najbliższym otoczeniu, z którymi styka się każdy pracownik wykonujący zadania zawodowe. Na stanowisku biurowym, wyposażonym w komputer, można wyróżnić czynniki: rzeczowe, fizyczne, biologiczne i psychologiczno-społeczne. Wszystkie czynniki danego środowiska pracy mogą być niebezpieczne lub szkodliwie wpływać na samopoczucie i zdrowie pracownika.

Czynniki rzeczowe to przede wszystkim wyposażenie stanowiska i pomieszczenia, w którym się ono znajduje. Mają one szczególny wpływ na komfort i bezpieczeństwo pracy. Dotyczy to systemu komputerowego i urządzeń peryferyjnych orazumeblowania stanowiska i całego pomieszczenia. Szerzej czynniki rzeczowe opisano w rozdziale dotyczącym organizacji stanowiska pracy.

Czynniki fizyczne są związane z ergonomią pracy. Na stanowisku pracy z komputerem zalicza się do nich: oświetlenie, hałas, mikroklimat i promieniowanie elektromagnetyczne.

Czynniki biologiczne występują na każdym stanowisku pracy. Tworzą je bakterie i wirusy roznoszone przez ludzi i zwierzęta.

Czynniki społeczne wynikają z organizacji zarządzania zespołem lub projektem oraz ze stosunków międzyludzkich związanych z relacjami między pracownikami.

Czynniki tworzące środowisko pracy mogą wywierać pozytywny, neutralny lub negatywny wpływ na pracownika, jego wydajność oraz jakość wykonywanej przez niego pracy. Ze względu na stopień oddziaływania wyróżnia się następujące czynniki negatywne:

- uciążliwe, które bezpośrednio nie oddziałują na zdrowie pracownika, ale utrudniają wykonanie zadań zawodowych i mogą wpływać na wydajność pracy;
- szkodliwe, które mogą stwarzać bezpośrednie zagrożenie dla zdrowia pracownika;
- niebezpieczne, które stanowią zagrożenie dla życia i zdrowia pracownika.

Praca na stanowisku pracy z komputerem i monitorem ekranowym w warunkach oddziaływania czynników negatywnych może prowadzić do powstania wad zdrowotnych, chorób oraz wypadków. Celem nadrzędnym jest właściwa organizacja stanowiska pracy z optymalizacją wszystkich czynników tworzących środowisko pracy.

Do najczęstszych schorzeń będących wynikiem długotrwałej pracy w pozycji siedzącej z komputerem i monitorem ekranowym w warunkach działania niekorzystnych czynników środowiskowych należą:

- choroby układu mięśniowo-szkieletowego (kończyny górne, kręgosłup, stawy);
- wady wzroku i schorzenia oczu (niedowidzenie, wady refrakcji, zespół suchego oka i ogólne zmęczenie wzroku);
- zwyrodnienia w obrębie nadgarstka, wywołujące przewlekłą chorobę obwodowego układu nerwowego;
- choroby układu oddechowego i alergię;
- kłopoty z krążeniem kończyn dolnych;
- wypalenie zawodowe oraz depresja;
- choroby psychiczne związane z obciążeniem pracą i stresem.

Do najczęstszych zagrożeń występujących na stanowisku pracy wyposażonym w system komputerowy z monitorem ekranowym, będących przyczyną urazów i schorzeń, zalicza się:

- uderzenia i skaleczenia o przedmioty;
- porażenie prądem elektrycznym;
- działanie promieniowania elektromagnetycznego monitora ekranowego;
- oparzenia o elementy utrwalające drukarek i kserokopiarek laserowych;
- przeciążenia układu ruchu i narządu wzroku;
- zagrożenie czynnikami biologicznymi (bakterie i wirusy).

Przed zagrożeniami na stanowisku pracy mają chronić indywidualne i zbiorowe środki ochrony. Zbiorowe środki ochrony muszą zabezpieczać systemowo cały budynek lub biuro. Zalicza się do nich:

- wyłączniki nadmiarowoprądowe – chronią poszczególne obwody przed przeciążeniem (odcinają zasilanie w razie zwarcia);
- wyłączniki różnicowoprądowe – reagują na upływ prądu w chronionym obwodzie (np. porażenie pracownika prądem); wyłączają wtedy jego zasilanie;
- uziemienie – chroni np. przed pojawieniem się niebezpiecznego napięcia na metalowej obudowie uszkodzonego urządzenia elektrycznego;
- systemy przeciwpożarowe z detektorami dymu, ognia oraz siatki rur i dysz – zraszają wodą pomieszczenia w razie pożaru;
- systemy wentylacyjne i klimatyzacyjne;
- gaśnice itp.

Indywidualne środki ochrony ograniczają niekorzystne czynniki panujące na stanowisku pracy, bezpośrednio oddziałujące na pracownika. Zalicza się do nich:

- wyposażenie ergonomiczne stanowiska (zostanie opisane w następnym rozdziale);
- okulary do pracy przy komputerze;
- szczepienia ochronne i leki antyalergiczne.

1.1. Organizacja stanowiska pracy w pozycji siedzącej z komputerem

Przepisy prawne obowiązujące pracodawcę, które dotyczą organizacji stanowisk pracy z systemem komputerowym, zawarto w *Rozporządzeniu Ministra Pracy i Polityki Socjalnej z dnia 1 grudnia 1998 r. w sprawie bezpieczeństwa i higieny pracy na stanowiskach wyposażonych w monitory ekranowe* (DzU z 1998 r. nr 148, poz. 973). Mają one zastosowanie w sytuacji, gdy pracownik użytkuje monitory ekranowe przez co najmniej cztery godziny dziennie (nie dotyczą one np. ekranów sterujących przy maszynach przemysłowych czy kas sklepowych). W tym rozporządzeniu szczegółowo określono wymagania dotyczące bezpieczeństwa i higieny pracy, ergonomii oraz organizacji pracy dla stanowisk wyposażonych w monitory ekranowe.

Stanowisko pracy to przestrzeń pracy wraz z wyposażeniem w środki i przedmioty pracy obejmujące: wyposażenie podstawowe (system komputerowy z monitorem ekranowym i oprogramowaniem), wyposażenie dodatkowe (urządzenia peryferyjne), wyposażenie pomocnicze (stół, krzesło, uchwyt na dokument, podnózek).

Celem stworzenia optymalnego stanowiska pracy z monitorem ekranowym jest uniknięcie wad lub chorób zawodowych wynikających z wykonywanych czynności. Jak wspomniano w poprzednim podrozdziale, najczęstszymi zagrożeniami związanymi z wykonywaniem zadań na stanowisku wyposażonym w monitor ekranowy są:

- zwyrodnienie kręgosłupa,
- wady słuchu,
- alergie,
- wady wzroku,
- bóle mięśni i zwyrodnienia stawów,
- kłopoty z krążeniem kończyn dolnych.

Pracodawca ma obowiązek przygotować optymalne stanowisko, na które będą się składały takie elementy, jak:

- odpowiednia powierzchnia pracy;
- ergonomiczne umeblowanie, dopasowane do charakterystyki psychofizycznej pracownika;
- ergonomiczne rozmieszczenie jednostki centralnej, urządzeń peryferyjnych i akcesoriów;
- właściwe oświetlenie;
- obniżenie poziomu hałasu;
- zapewnienie neutralnej aranżacji wnętrza, zapewniającej każdemu atmosferę odprężenia, domowego ciepła i relaksu psychicznego;
- zastosowanie zabezpieczeń stanowiących ochronę przed porażeniem prądem elektrycznym;
- stworzenie optymalnego mikroklimatu.

Dodatkowo pracodawca jest zobowiązany do stworzenia organizacji pracy zapewniającej pracownikom szkolenia stanowiskowe, cykliczne przerwy w pracy (minimum 5 min na każdą godzinę pracy) oraz profilaktyczną opiekę zdrowotną (np. refundację kosztów korekcji wzroku).

Znacznie rozszerzone wymagania w stosunku do pracodawcy występują wtedy, gdy ma on przygotować stanowisko pracy dla osoby z niepełnosprawnością. W zawodach **technik informatyk** i **technik programista** istnieją warunki do pracy dla osób z niepełnosprawnościami lub z przewlekłymi chorobami, choć możliwość wykonywania zadań wiąże się wtedy często z koniecznością zapewnienia dodatkowego wyposażenia lub innych parametrów technicznych urządzeń techniki komputerowej. Przykładowo: osoby z chorobami narządu wzroku lub niewidzące powinny mieć na swoim stanowisku pracy (zależnie od stopnia utraty wzroku):

- monitor komputerowy o przekątnej powyżej 19”;
- klawiaturę brajlowską;
- oprogramowanie czytające tekst;
- wypukłe napisy i tablice informacyjne;
- wytyczone specjalne ciągi komunikacyjne bez zbędnych przeszkód;
- przywoływacz dźwiękowy.

Należy pamiętać, że wyposażenie stanowiska pracy najlepiej dopasować indywidualnie do pracownika, który będzie z niego korzystał. Tylko wtedy spełni ono zasady ergonomii pracy i pozytywnie wpłynie na jakość i wydajność zatrudnionego.

Pomieszczenie

Stanowiska pracy z monitorem ekranowym powinny być tak usytuowane, aby zapewniały pracownikowi swobodny do niego dostęp. Odległość między sąsiednimi monitorami musi wynosić ok. 60 cm, a między użytkownikiem a tyłem sąsiedniego monitora – ok. 80 cm. Należy też zapewnić swobodny dostęp do innych elementów umeblowania, przy czym szerokość przejścia na jego całej długości powinna być nie mniejsza niż 60 cm.

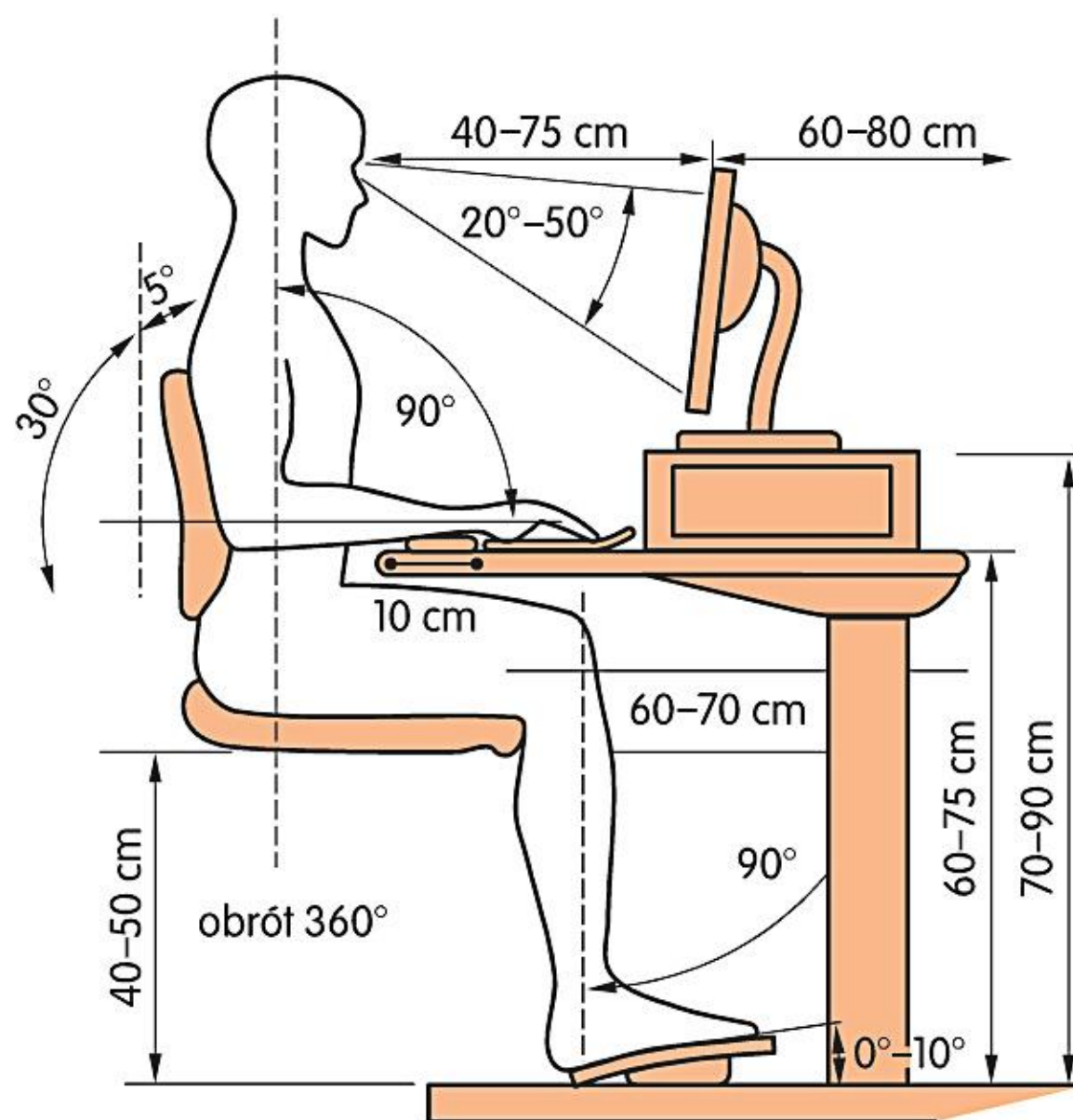
Wyposażenie

Wyposażenie stanowiska pracy oraz sposób rozmieszczenia jego poszczególnych elementów nie może powodować podczas pracy nadmiernego obciążenia układu mięśniowo-szkieletowego oraz wzroku pracownika. Dobrą praktyką jest korzystanie z dodatkowych elementów wyposażenia, takich jak podnóżek.

Monitor ekranowy powinien umożliwiać konfigurację parametrów pracy (jasność, kontrast i częstotliwość odświeżania). Monitor musi mieć certyfikat CE, który gwarantuje kompatybilność elektromagnetyczną. Pracownika należy wyposażyć w taki monitor, który może pochylić co najmniej 20° do tyłu i 5° do przodu oraz obrócić wokół jego osi co najmniej o 120° (po 60° w obu kierunkach). Ustawienie ekranu monitora względem źródeł światła powinno ograniczać olśnienie i odbicia światła oraz zapewniać wyraźną ekspozycję znaków.

Zależnie od efektu, jaki użytkownik chce osiągnąć, na stanowisku pracy z monitorem ekranowym można zastosować różnego rodzaju zabezpieczenia filtrujące oglądany obraz. Jeżeli chcemy poprawić wyrazistość i widoczność wyświetlonego obrazu, a jednocześnie zapewnić warstwę chroniącą przed kurzem i zarysowaniami, stosujemy filtry polaryzacyjne rozpraszające światło odbijające się w ekranie. Jeżeli natomiast zależy nam na bezpieczeństwie danych wyświetlanych na ekranie, to powinniśmy zastosować filtr prywatyzujący, skutecznie zawężający kąt widzenia monitora.

Klawiatura komputerowa powinna umożliwiać pracownikowi przyjęcie pozycji, która nie powoduje zmęczenia mięśni kończyn górnych podczas pracy. Klawiatura musi mieć możliwość regulacji kąta nachylenia w zakresie od 0° do 15° . Trzeba ją tak umieścić, by zachować odległość nie mniejszą niż 10 cm między klawiaturą a przednią krawędzią stołu. Podczas wykonywania zadań wymagających długotrwałego pisania tekstu (np. kodu programu) można rozważyć korzystanie z klawiatur ergonomicznych, dopasowanych do budowy anatomicznej kończyn górnych i zmniejszających zmęczenie użytkownika.



Rys. 1.1. Schemat organizacji pracy w pozycji siedzącej na stanowisku z komputerem i monitorem ekranowym

Meble

Na stanowisku wyposażonym w komputer z monitorem ekranowym podstawowe umeblowanie stanowią biurko i ergonomiczny fotel. Przepisy określają, że wysokość stołu oraz siedziska fotela powinna być taka, aby zapewniała naturalne położenie kończyn górnych podczas obsługi klawiatury, z zachowaniem co najmniej kąta prostego między

ramieniem a przedramieniem. Powierzchnia blatu biurka powinna być matowa, najlepiej barwy jasnej. Jeżeli w trakcie pracy trzeba skorzystać z dokumentów, stanowisko pracy należy wyposażyć w uchwyt na dokument, z regulacją ustawienia wysokości, pochylenia oraz odległości od pracownika.

Ergonomiczny fotel powinien mieć:

- dostateczną stabilność (powinien być wyposażony w podstawę co najmniej pięciopodporową, z kółkami jezdnyymi);
- wymiary oparcia i siedziska zapewniające wygodną pozycję ciała i swobodę ruchów;
- regulację wysokości siedziska w zakresie 40–50 cm, licząc od podłogi;
- regulację wysokości oparcia oraz regulację pochylenia oparcia: 5° do przodu i 30° do tyłu;
- wyprofilowanie płyty siedziska i oparcia (właściwe dla naturalnego wygięcia kręgosłupa i odcinka udowego kończyn dolnych);
- możliwość obrotu wokół osi pionowej o 360°;
- regulowane podłokietniki.

Hałas

Technik informatyk i technik programista nie są narażeni na hałas generowany przez maszyny przemysłowe, budowlane lub pojazdy. Na stanowiskach wyposażonych w systemy komputerowe nie stosuje się indywidualnych środków ochrony słuchu, takich jak wkładki lub nauszniki przeciwhałasowe. Zalecany poziom hałasu dla stanowisk komputerowych w razie wykonywania prac o dużym stopniu trudności powinien wynosić nie więcej niż 45–55 dB. Głównym źródłem hałasu są wentylatory, umieszczone w systemie komputerowym, dlatego ważne są ich okresowa konserwacja lub wymiana. Można też zastosować systemy chłodzące pasywne, pozbawione wentylatorów. Przepisy określają, że systemy komputerowe muszą zapewniać przekazywanie pracownikom informacji zwrotnej o ich działaniu – dlatego warto kontrolować poziom hałasu i temperaturę wewnątrz jednostki centralnej.

Mikroklimat

Mikroklimat wpływa bezpośrednio na samopoczucie na stanowisku pracy. Temperatura w pomieszczeniu, w którym znajdują się stanowiska pracy wyposażone w systemy komputerowe, powinna wynosić: zimą 20–24°C, a latem 23–26°C, wilgotność natomiast nie może być mniejsza niż 40%. Źródła światła dzieli się na: naturalne, ogólne i miejscowe. Stanowisko do pracy z komputerem powinno być oświetlone, według norm, światłem o natężeniu ok. 500 lx.

Urządzenia zabezpieczające

Sprzęt elektroniczny jest narażony na awarie spowodowane przepięciami, zasilaniem prądem o niestabilnych parametrach lub całkowitym brakiem zasilania. Każdy z wymienionych czynników może uszkodzić wyposażenie stanowiska pracy lub zniszczyć dane będące efektem pracy wielu osób. Z tego powodu do wyposażenia stanowiska pracy

z systemem komputerowym zalicza się również urządzenia zabezpieczające przed negatywnymi skutkami przepięć lub podtrzymujące napięcie w razie braku zasilania. W dużych przedsiębiorstwach lub korporacjach stosuje się rozwiązania systemowe. W mniejszych organizacjach te zabezpieczenia dotyczą każdego stanowiska pracy.

Najpopularniejszymi urządzeniami zabezpieczającymi przed przepięciami są listwy antyprzepięciowe. Do kluczowych parametrów tego typu urządzenia zalicza się:

- absorpcję energii (co najmniej 150 J);
- czas reakcji układu przeciwprzepięciowego (maks. 25 ns);
- maksymalny prąd impulsu (ok. 6 kA);
- napięcie progowe (ok. 1 kV).



Rys. 1.2. Biurowy zasilacz awaryjny dla jednego stanowiska komputerowego

Kiedy dochodzi do nagłego, całkowitego braku zasilania, idealnie sprawdzają się zasilacze awaryjne UPS (ang. *Uninterruptible Power Supply*). Zasilacz UPS jest wyposażony w akumulatory. W razie przerwy lub zakłóceń w dostawie energii elektrycznej z sieci energetycznej przełącza się na pracę z akumulatora. Czas podtrzymania napięcia wynosi od kilku minut do kilkunastu godzin, ale zależy to od obciążenia zasilacza i pojemności akumulatorów.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Jakie wymagania powinno spełnić stanowisko pracy przewidziane dla niedowidzącego programisty?
2. Jakie zadanie spełniają zasilacze awaryjne UPS?
3. Oceń prawdziwość poniższych zdań.

	Prawda	Fałsz
Stanowisko pracy z komputerem i monitorem ekranowym powinno być wyposażone w ergonomiczny fotel z regulacją podłokietników.		
Blat biurka musi mieć ciemną barwę.		
Każdy pracownik powinien być wyposażony w nauszники przeciwhałasowe.		
Klawiatura musi mieć możliwość regulacji kąta nachylenia w zakresie od 0° do 15°.		

4. Jakie elementy zalicza się do środowiskowych czynników rzeczowych w miejscu pracy?

II. Podstawowe pojęcia dotyczące stron i aplikacji internetowych

- Komunikacja w sieci internet
- Technologie webowe
- Pojęcia dotyczące stron i aplikacji internetowych

2

Podstawy komunikacji w sieci internet

ZAGADNIENIA

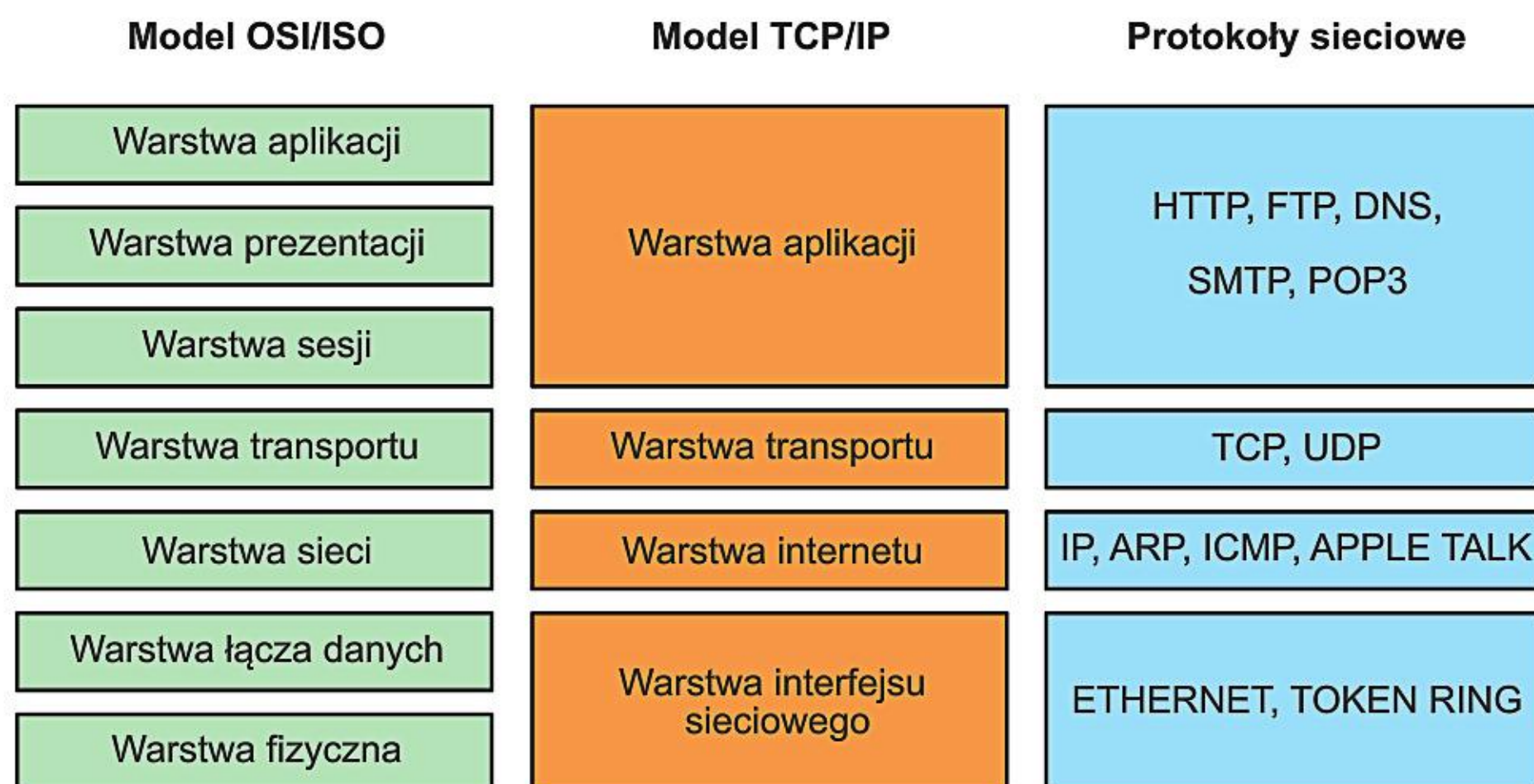
- Komunikacja w sieci internet
- Protokoły wykorzystywane przez aplikacje internetowe
- Zasady działania protokołu HTTP

Wszystko, co wiąże się z transmisją danych w sieci komputerowej, wymaga wstępnego sformułowania zasad, stworzenia schematów i wypracowania standardów komunikacji oraz przesyłania i prezentacji danych w sieci.

Jednymi z pierwszych standardów porządkujących ruch danych w sieci są modele komunikacji sieciowej OSI/ISO oraz TCP/IP. Są to modele odniesienia dla użytkowników i twórców sieciowych rozwiązań komunikacyjnych, zapewniające kompatybilność komunikacyjną między różnymi protokołami odpowiedzialnymi za przesyłanie i prezentację danych w sieci. Dzięki temu można np. otwierać stronę internetową w różnych systemach operacyjnych i na dowolnym urządzeniu (laptop, PC czy smartfon). Międzynarodowa Organizacja Normalizacyjna **ISO** (ang. *International Organization for Standardization*) opracowała model łączenia tzw. systemów otwartych **OSI** (ang. *Open Systems Interconnection Reference Model*) złożony z siedmiu warstw. Każda warstwa opisuje oddzielny etap komunikacji między dwoma komputerami w sieci. Warstwy wymieniają się jednostkami danych i tworzą narastające zbiory, zwane segmentami, ramkami lub pakietami danych. Każda z poszczególnych warstw definiuje również konkretne zadania związane z komunikacją sieciową, realizowane przez protokoły i urządzenia sieciowe pracujące na poziomie danej warstwy.

Dynamiczny rozwój sieci Internet skłonił międzynarodowe organizacje tworzące standardy sieciowe do opracowania nowego modelu odniesienia, który trafniej opisze strukturę tej sieci. Model przyjęty jako powszechny standard opracowano w latach 80. XX w.

Model sieci Internet nazywa się też modelem odniesienia **TCP/IP** lub stosem protokołów TCP/IP. Na rys. 2.1 pokazano różnice między modelami OSI/ISO oraz TCP/IP, a także to, że w każdej warstwie modelu TCP/IP wykorzystuje się konkretne protokoły sieciowe.

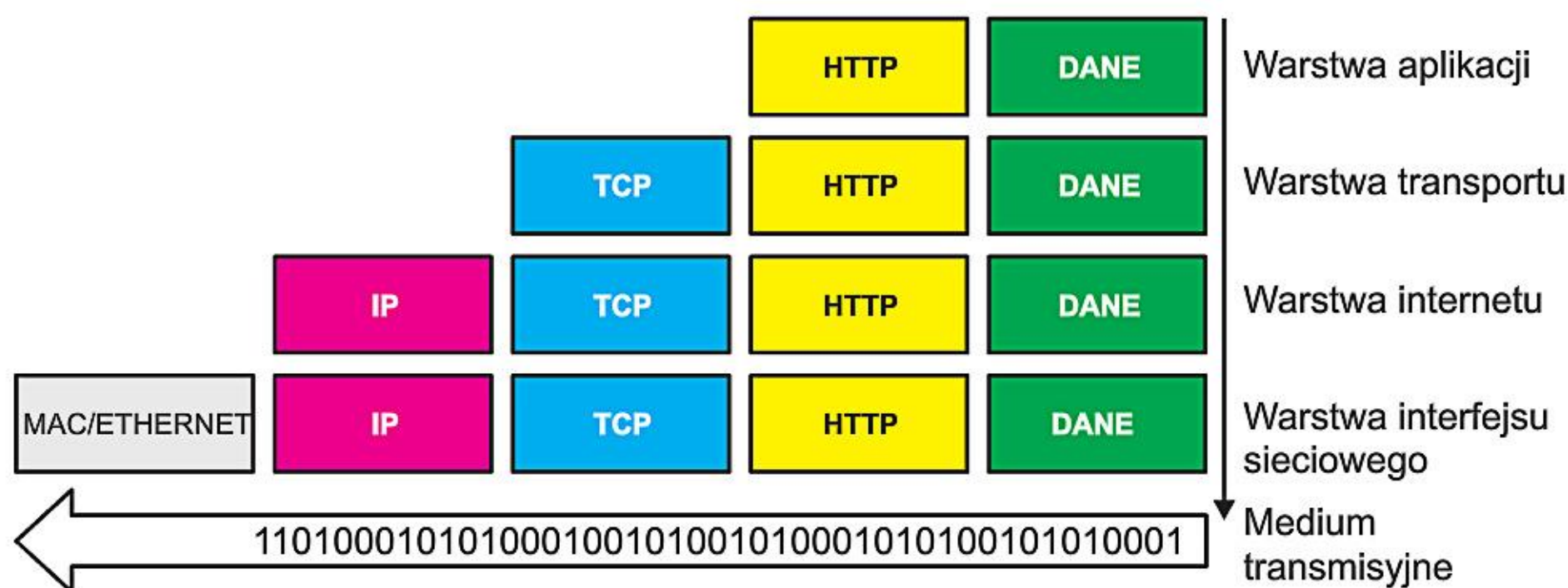


Rys. 2.1. Porównanie modeli OSI/ISO oraz TCP/IP

Przygotowanie danych do przesłania w sieci komputerowej wygląda następująco: po zatwierdzeniu wysłania danych uruchamia się działanie w warstwie aplikacji. Dane otrzymują odpowiedni format, zgodny z usługą, którą wykorzystywaliśmy. Następnie nasze dane są przesyłane do warstwy transportu. Tutaj zostanie określony port pozwalający na zestawienie sesji. Przykładowo: dla strony internetowej i protokołu HTTP będzie to port 80. Dane zostają tutaj podzielone na segmenty i wysłane do następnej warstwy. W warstwie internetu są tworzone pakiety, w których do danych dodaje się adresy komputera nadawcy i odbiorcy. Odpowiada za to protokół IP. Dodatkowo planuje się najlepszą trasę dla tych pakietów. Tak przygotowana informacja jest przekazywana do warstwy interfejsu sieciowego. W tej warstwie dane są dzielone na ramki i przygotowywane do transportu przez określone medium transmisyjne w danym standardzie sieciowym (np. ETHERNET). W ramce określa się fizyczny adres karty sieciowej (MAC adres) oraz koduje się dane do postaci bitowej (w postaci zer i jedynek). Cały opisany proces nazywa się **enkapsulacją** (ang. *encapsulation*).

Tak zakodowane dane są transportowane w postaci sygnału elektrycznego modulowanego wartościami napięcia w taki sposób, aby łatwo można było rozróżnić logiczne 0 oraz 1.

Na komputerze docelowym informacja jest rozkodowywana i przechodzi poszczególne warstwy w odwrotnej kolejności. Taki proces nazywa się **dekapsulacją** (ang. *decapsulation*). W aplikacji odbiorcy zostaje odczytane zapytanie dotyczące adresu strony internetowej i zwrótnie są wysyłane pliki tworzące tę stronę. Cały proces rozpoczyna się od nowa.



Rys. 2.2. Proces wysyłania danych w poszczególnych warstwach

W każdej warstwie modelu TCP/IP czy OSI/ISO działają określone protokoły sieciowe.

Protokół komunikacji sieciowej (ang. *network communication protocol*) to zbiór reguł i procedur postępowania, które są stosowane w ramach określonej warstwy modelu sieciowego przez urządzenia sieciowe i aplikacje.

W pracy przy stronach i aplikacjach internetowych wykorzystujemy opisane protokoły komunikacji sieciowej.

HTTP (ang. *HyperText Transfer Protocol*) to podstawowy protokół wykorzystywany do transportu i prezentacji danych w postaci strony internetowej. Obecnie protokół zarówno pozwala na przesyłanie stron w postaci plików, jak i może obsługiwać np. strumieniową transmisję danych. Aplikacją obsługującą ten protokół jest każda przeglądarka internetowa. Kluczowy dla pobierania treści i ich prezentacji jest adres **URL** (ang. *Uniform Resource Locator*), który definiuje lokalizację pliku i usługę protokołu.

PRZYKŁAD 2.1

<https://sportowefakty.wp.pl/pilka-nozna>

HTTPS (ang. *HyperText Transfer Protocol Secure*) to protokół umożliwiający przesyłanie szyfrowanych plików stron internetowych. Zwiększa to znacznie bezpieczeństwo pracy, np. na formularzach wykorzystywanych w aplikacjach internetowych. Do szyfrowania używa się protokołu **SSL** (ang. *Secure Socket Layer*).

FTP (ang. *File Transfer Protocol*) to protokół umożliwiający przesyłanie plików. W aplikacjach internetowych pozwala na pobieranie i wgrywanie plików między użytkownikiem aplikacji a serwerem oraz na publikowanie na serwerze plików stron internetowych przez administratora. Dostępny jest również protokół oferujący szyfrowanie danych **SFTP** (ang. *Secure File Transfer Protocol*).

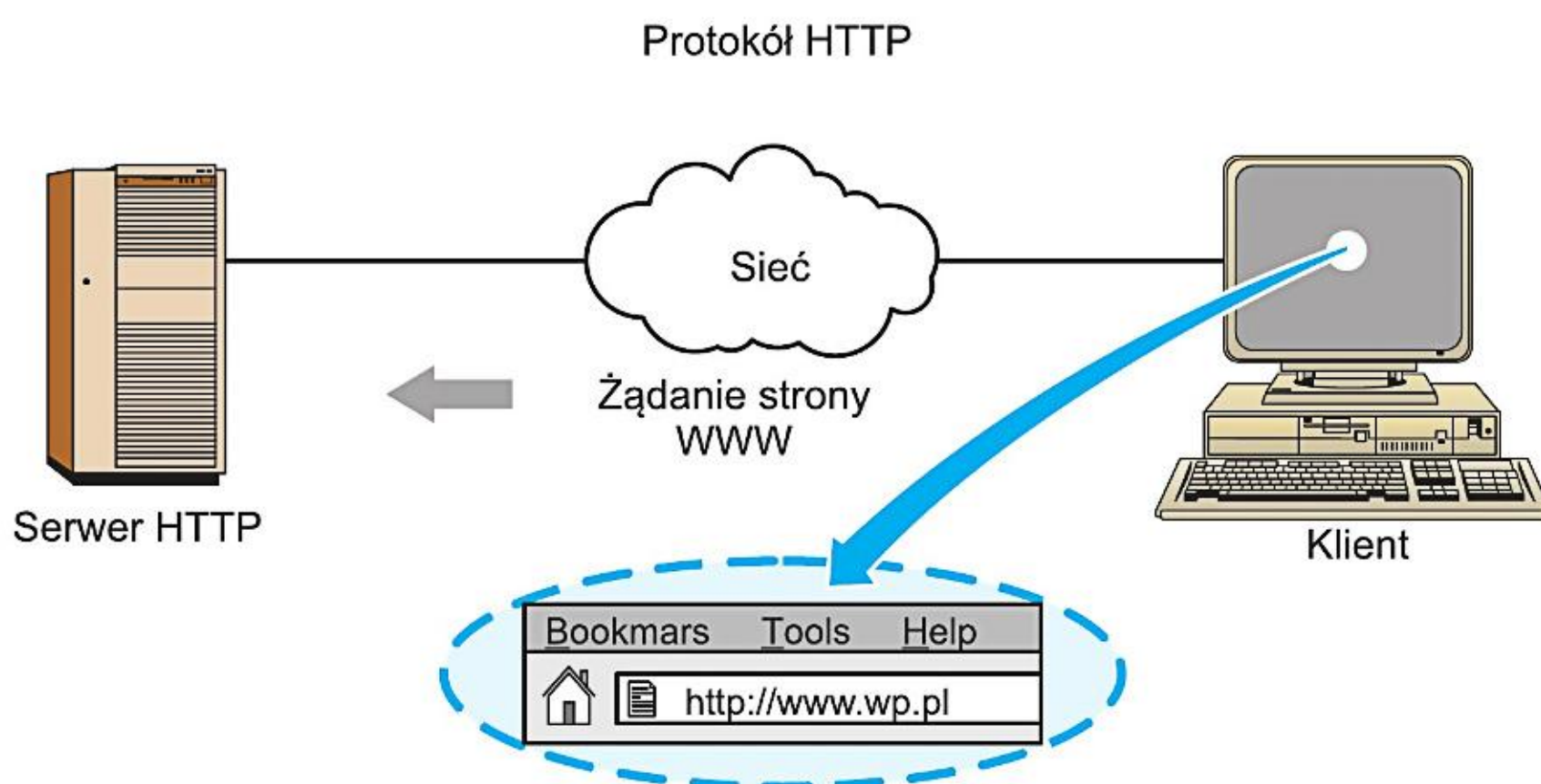
DNS (ang. *Domain Name System*) to protokół pozwalający na nadawanie użytkownikom internetu alfanumerycznych nazw adresów logicznych. Protokół tłumaczy je na odpowiadający im adres IP. Dzięki temu użytkownik łatwo zapamięta adres naszej strony, nie przejmując się zapisem binarnym adresu IP, pod którym strona się znajduje.

Wszystkie opisane protokoły zalicza się do protokołów publikowanych w standardzie otwartym. Oznacza to, że ich specyfikacja nie jest ograniczona prawnie i każdy może na ich podstawie tworzyć własne rozwiązania sieciowe.

PRZYKŁAD 2.2

Przykład działania protokołu http podczas wczytywania strony WWW

Użytkownik, wpisując adres URL strony internetowej, wysyła do sieci żądanie pobrania i wyświetlenia danego zasobu. Jeżeli zostanie odnaleziony odpowiedni serwer http, pliki strony zostaną przesłane na komputer użytkownika. Ten proces jest powszechnie nazywany ładowaniem strony. Jeżeli po domyślnym okresie nie zostanie znaleziony określony w żądaniu plik, przeglądarka wyświetli komunikat błędu numer 404 (brak pliku lub zła lokalizacja pliku).



Rys. 2.3. Działanie protokołu HTTP

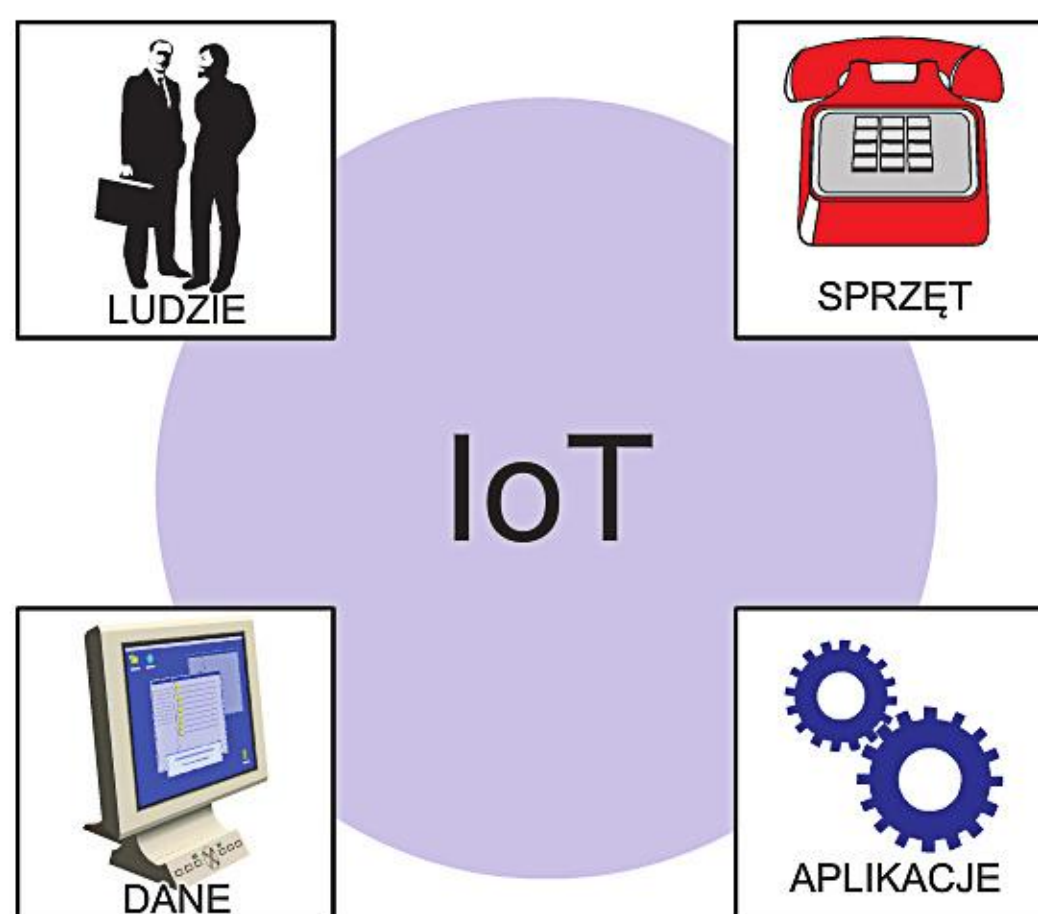
W systemach informatycznych, w których informacja w formie danych jest przetwarzana za pomocą systemów komputerowych, głównym kanałem komunikacyjnym jest sieć komputerowa wykorzystująca różne media transmisyjne – przewodowe i bezprzewodowe. Nowoczesne technologie oparte na sieci komputerowej, które łączą różne urządzenia fizyczne po to, aby swobodnie gromadzić, wymieniać i przetwarzać między sobą dane, tworzą coraz częściej zintegrowane środowiska zwane **IoT**, czyli internet rzeczy (ang. *Internet of Things*). Podstawowym celem IoT jest stworzenie inteligentnych przestrzeni, takich jak:

- inteligentne domy,
- inteligentne miasta,
- inteligentne sieci zdrowia,
- inteligentne przedsiębiorstwa,
- inteligentny przemysł,
- inteligentne systemy alarmowe i ostrzegawcze.

Nową technologią coraz częściej występującą w rozwiązaniach sieciowych jest usługa chmury obliczeniowej (ang. *cloud computing*).

Chmura obliczeniowa to zasób sieciowy (np. przestrzeń dyskowa, moc obliczeniowa lub aplikacja), który może zostać udostępniony użytkownikom w celu jego wykorzystania.

Idea udostępniania zasobów nie jest niczym nowym w świecie internetu, ale dopiero wysokie przepustowości łącz abonenckich pozwoliły rozszerzyć zakres działania chmury i jej możliwości. Dodatkowymi czynnikami wspierającymi rozwój chmury w ostatnich latach są: technologia wirtualizacji, stale rosnąca moc obliczeniowa systemów komputerowych oraz zwiększanie przepustowości sieci bezprzewodowych (np. technologie 4G i 5G).



Rys. 2.4. Idea Internetu rzeczy

 SPRAWDŹ SWOJĄ WIEDZĘ

1. Jakie protokoły występują w warstwie aplikacji modelu TCP/IP?
2. Co to jest adres URL? Wyjaśnij, czym różni się adres URL od bezwzględnej ścieżki dostępu do zasobu w systemie Windows.
3. Oceń prawdziwość poniższych zdań.

W stosie protokołów TCP/IP	Prawda	Falsz
są dwie warstwy: IP i TCP.		
warstwa transportu odpowiada za przygotowanie danych do przesłania w sieci.		
warstwa internetu jest odpowiednikiem warstwy sieci w modeli OSI/ISO.		

3

Technologie webowe

ZAGADNIENIA

- Rodzaje technologii webowych
- Historia języka HTML
- Podstawowe pojęcia dotyczące stron WWW
- Strony i aplikacje mobilne

Aplikacja internetowa (ang. *web application*), zwana również aplikacją webową, to program komputerowy, który pracuje na zdalnym serwerze i komunikuje się przez sieć komputerową z komputerem personalnym lub urządzeniem mobilnym użytkownika z wykorzystaniem przeglądarki internetowej. Do tworzenia aplikacji oraz nadzorowania ich przesyłania i prezentacji używa się wielu różnych technologii zwanych technologiami webowymi.

Rodzaje technologii webowych

Jedną z najpopularniejszych usług sieciowych jest tworzenie i publikowanie w sieci dokumentów hipertekstowych zwanych powszechnie stronami **WWW** (ang. *World Wide Web*). Do tworzenia stron WWW używa się różnych technologii nazywanych webowymi. Technologie webowe to wszystkie języki programowania i techniki multimedialne oraz sieciowe pozwalające na jak najlepsze, kreatywne i uniwersalne tworzenie stron i aplikacji internetowych, interpretowanych zarówno na urządzeniach stacjonarnych, jak i na mobilnych.

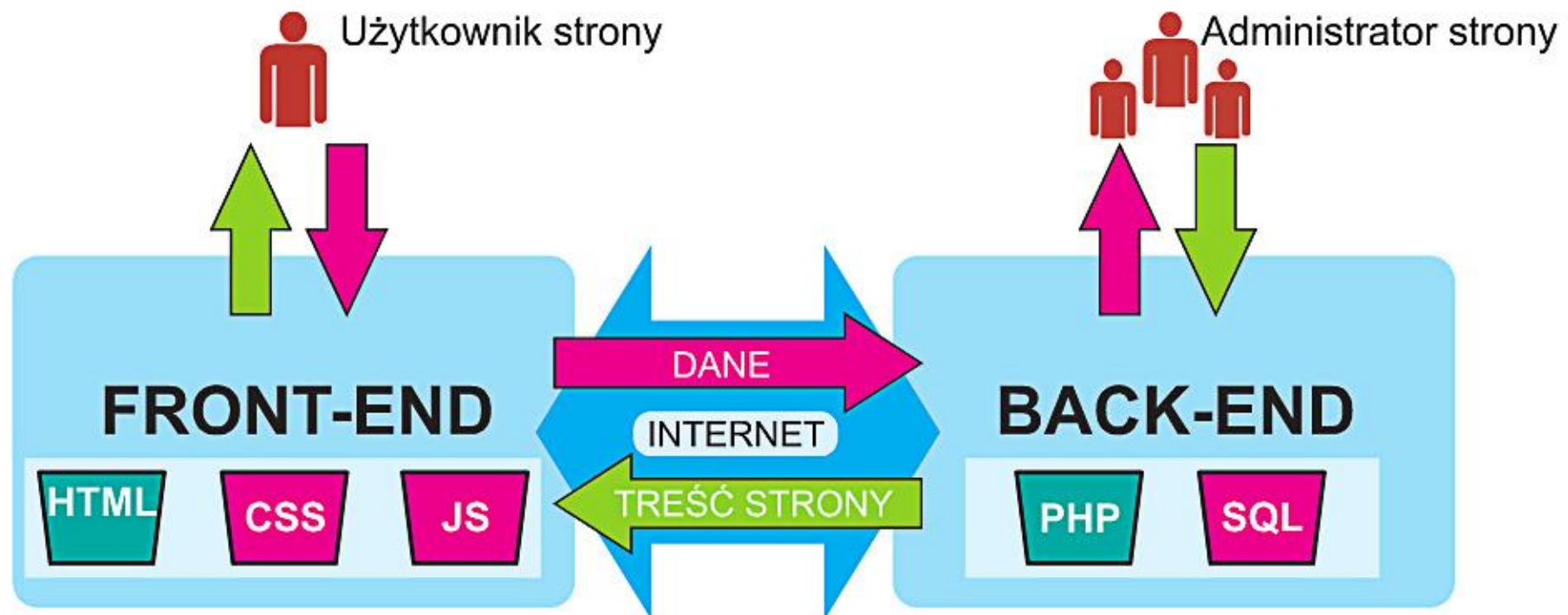
Początkowo strony internetowe zawierały głównie tekst, hiperłącza oraz grafikę statyczną i były tworzone wyłącznie na podstawie języka HTML.

HTML (ang. *HyperText Markup Language*) to hipertekstowy język znaczników wykorzystywany do tworzenia dokumentów hipertekstowych, interpretowanych przez przeglądarkę internetową.

Wraz ze wzrostem przepustowości sieci i możliwości przeglądarek zaczęto tworzyć witryny internetowe korzystające z różnych technologii. Obecnie wszystkie technologie webowe dzieli się na obszary działania:

- front-end – technologie wykorzystujące kod jawny, wykonywany po stronie przeglądarki klienta / użytkownika;
- back-end – technologie wykorzystujące kod ukryty (tajny), wykonywany po stronie serwera.

Do głównych technologii front-end zalicza się: HTML, CSS, JavaScript, grafikę wektorową uruchamianą po stronie klienta **SVG** (ang. *Scalable Vector Graphics*). Natomiast back-end opiera się głównie na frameworkach: .NET, Java czy Python oraz skryptach PHP współpracujących z sieciowymi bazami danych SQL.



Rys. 3.1. Podział technologii webowych

Wybór technologii strony, którą zamierzamy stworzyć, jest bardzo ważny. Określa możliwości zastosowania poszczególnych obiektów, interaktywności z użytkownikiem, szybkości działania oraz funkcjonalności. Może w zasadniczy sposób wpłynąć również na koszt całego projektu. Wybór technologii powinien się odbyć na początku prac nad projektem strony WWW. Musi to być wspólna decyzja zleceniobiorcy i zleceniodawcy.

Historia i rozwój języka HTML

Od początków lat 80. XX w. trwały prace nad stworzeniem standardu przesyłania dokumentów w sieci Internet. Zniesienie zakazu wykorzystywania zasobów sieci Internet do celów komercyjnych w 1991 r. przyspieszyło rozwój tej usługi. W tym samym roku Tim Berners-Lee zamieścił w sieci dokument *HTML Tags*, w którym opisał 24 znaczniki do zapisu dokumentów udostępnianych w internecie. Każdy z nich zawierał słowo, które było otoczone nawiasami ostrymi. Powstał on na podstawie standardu **SGML** (ang. *Standard Generalized Markup Language*), który był metajęzykiem umożliwiającym definiowanie dowolnych znaczników i zasad ich użycia. Można uznać, że był to standard HTML 1, chociaż nie pojawiła się oficjalna, pierwsza specyfikacja tego języka.

W 1993 r. Mark Andreessen udostępnił pierwszą graficzną przeglądarkę stron internetowych – **Mosaic**. Pierwszą, oficjalną specyfikację języka **HTML 2.0** opublikowała **IETF** (ang. *Internet Engineering Task Force*) w 1995 r. Niedługo potem do czuwania nad rozwojem standardów pisania stron internetowych powołano organizację **W3C** (ang. *World Wide Web Consortium*).

Po opublikowaniu specyfikacji **HTML 3.2** w 1997 r. przystąpiono do prac nad specyfikacją języka HTML związanego z kaskadowymi arkuszami stylów **CSS**, które przejmowały główny ciężar formatowania stron internetowych. Tym samym oddzielono strukturę

logiczną tekstu od struktury prezentacji. W 1999 r. ukazała się specyfikacja **HTML 4**. Dopiero w 2001 r. udało się zakończyć pracę nad stabilną, interpretowaną przez wszystkie przeglądarki specyfikacją **HTML 4.1**.

Następnym krokiem w rozwoju języka HTML było podporządkowanie składni standardom wprowadzonym przez metajęzyk **XML** (ang. *Extensible Markup Language*). XML wykorzystuje się do budowania strukturyzowanych form prezentacji danych w przeglądarkach internetowych i aplikacjach bazodanowych. W ten sposób powstała specyfikacja języka **XHTML 1.0** (ang. *Extensible HyperText Markup Language*), która jest prostą ewaluacją HTML 4.1 umożliwiającą współpracę z dokumentami XML.

Dynamiczny rozwój przeglądarek internetowych stał się przyczyną rozłamu w planowaniu przyszłości języków hipertekstowych. Powstała druga organizacja zajmująca się rozwojem stron WWW – **WHATWG** (ang. *Web Hypertext Application Technology Working Group*). Tworzyli ją głównie przedstawiciele Apple, Mozilli i Opery. Jej głównym celem był rozwój technologii umożliwiających tworzenie na podstawie kodu HTML bardziej zaawansowanych aplikacji sieciowych.

Po kilku latach trwania równoczesnych projektów W3C ogłosiła chęć współpracy z organizacją WHATWG w rozwijaniu języka HTML. Efektem współpracy jest specyfikacja języka **HTML 5**, którą zaleca się do tworzenia stron internetowych (będzie podstawową specyfikacją przedstawianą w tym podręczniku). Opublikowano ją w 2014 r.

Historia i rozwój SQL i PHP

W latach 70. ubiegłego stulecia nastąpił dynamiczny rozwój relacyjnych baz danych. Pojawiła się wówczas konieczność opracowania języka do manipulacji, wyciągania i obsługi danych w bazach. Pierwszym oficjalnym językiem relacyjnych baz danych był SEQUEL (ang. *Structured English Query Language*), opracowany przez pracowników firmy IBM. Nazwa z czasem się zmieniła na SQL (ang. *Structured Query Language*). W 1986 r. SQL stał się oficjalnym standardem, wspieranym przez Międzynarodową Organizację Normalizacyjną ISO. SQL jest przede wszystkim strukturalnym językiem zapytań używanym do tworzenia, modyfikowania baz danych na sieciowych serwerach oraz do umieszczania na nich i pobierania z nich danych w formie zapytań. Po przejęciu licencji SQL przez firmę Oracle można spotkać się z projektem na otwartej licencji (ang. *open source*) o nazwie MariaDB. Szczegółowo język SQL omówiono w drugiej części podręcznika.

Język HTML nie nadaje się do współpracy z bazami danych. Dlatego niezbędne stało się zastosowanie języka, którego skrypty mogą być interpretowane przez środowiska webowe po stronie serwera WWW i współpracować z bazami danych SQL. Najczęściej stosowanym językiem skryptowym w aplikacjach internetowych jest PHP (ang. *PHP Hypertext Preprocessor*). Pierwszą wersję PHP stworzył Rasmus Lerdorf w 1994 r. jako zestaw skryptów w języku programowania Perl (jako *Personal Home Page Tools*). Obecnie wykorzystuje się wersję PHP 7. Zastosowanie języka PHP pozwala na obsługę formularzy, zapytań baz danych, wykonywania obliczeń arytmetycznych na stronach

internetowych, obsługi ciasteczek (ang. *cookies*) i wiele innych funkcji. Zastosowanie określonej wersji standardu SQL i PHP może wpłynąć na funkcjonalność i bezpieczeństwo tworzonych aplikacji internetowych. Szczegółowo język PHP omówiono w drugiej części podręcznika.

Baza danych obsługiwana przez język SQL, usługa serwera WWW oraz język PHP to obecnie główne elementy większości pakietów serwerowych do budowy oraz obsługi stron i aplikacji internetowych.

Podstawowe pojęcia dotyczące stron WWW

W gąszczu pojęć odnoszących się do stron internetowych warto wyjaśnić te, które przydadzą się w dalszej nauce:

- **statyczna strona internetowa** – strona z określonymi na stałe parametrami formatowania jej zawartości; użytkownik nie ma wpływu na prezentację treści;
- **dynamiczna strona internetowa** – strona generowana dynamicznie przez serwer na podstawie parametrów przekazywanych przez przeglądarkę internetową; użytkownik ma wpływ na prezentację i zawartość strony;
- **witryna internetowa** – grupa stron internetowych tworząca określoną strukturę oraz jednolita tematycznie;
- **portal internetowy** – serwis internetowy zawierający informacje na różne tematy i z różnych dziedzin;
- **wortal internetowy** – wyspecjalizowany serwis internetowy;
- **system zarządzania treścią CMS** (ang. *Content Management System*) – system tworzenia i zarządzania witryną internetową, sklepem lub blogiem oparty na infrastrukturze klient-serwer z wykorzystaniem bazy danych, języka HTML i skryptów PHP;
- **serwer WWW** – usługa (rola) oferowana przez komputery-serwery z sieciowym systemem operacyjnym; najczęściej takie usługi są oferowane przez firmy hostingowe, które dysponują farmami serwerów;
- **blog** (ang. *web log*) – dziennik internetowy, najczęściej o określonej tematyce; może przybierać różne formy, np. fotobloga, wiedeobloga lub audiobloga.
- **serwer bazy danych** – usługa oferująca zasoby do zarządzania sieciowymi bazami danych; w technologii webowej istotne jest realizowanie przez serwer zapytań SQL; przykładowymi serwerami wykorzystywanymi w tworzeniu stron i aplikacji internetowych są MySQL i MSSQL;
- **hosting** – udostępnianie przez dostawcę usług internetowych zasobów i usług serwerowych;
- **cyberbezpieczeństwo** – zespół zagadnień związanych z zapewnianiem ochrony w obszarze cyberprzestrzeni.

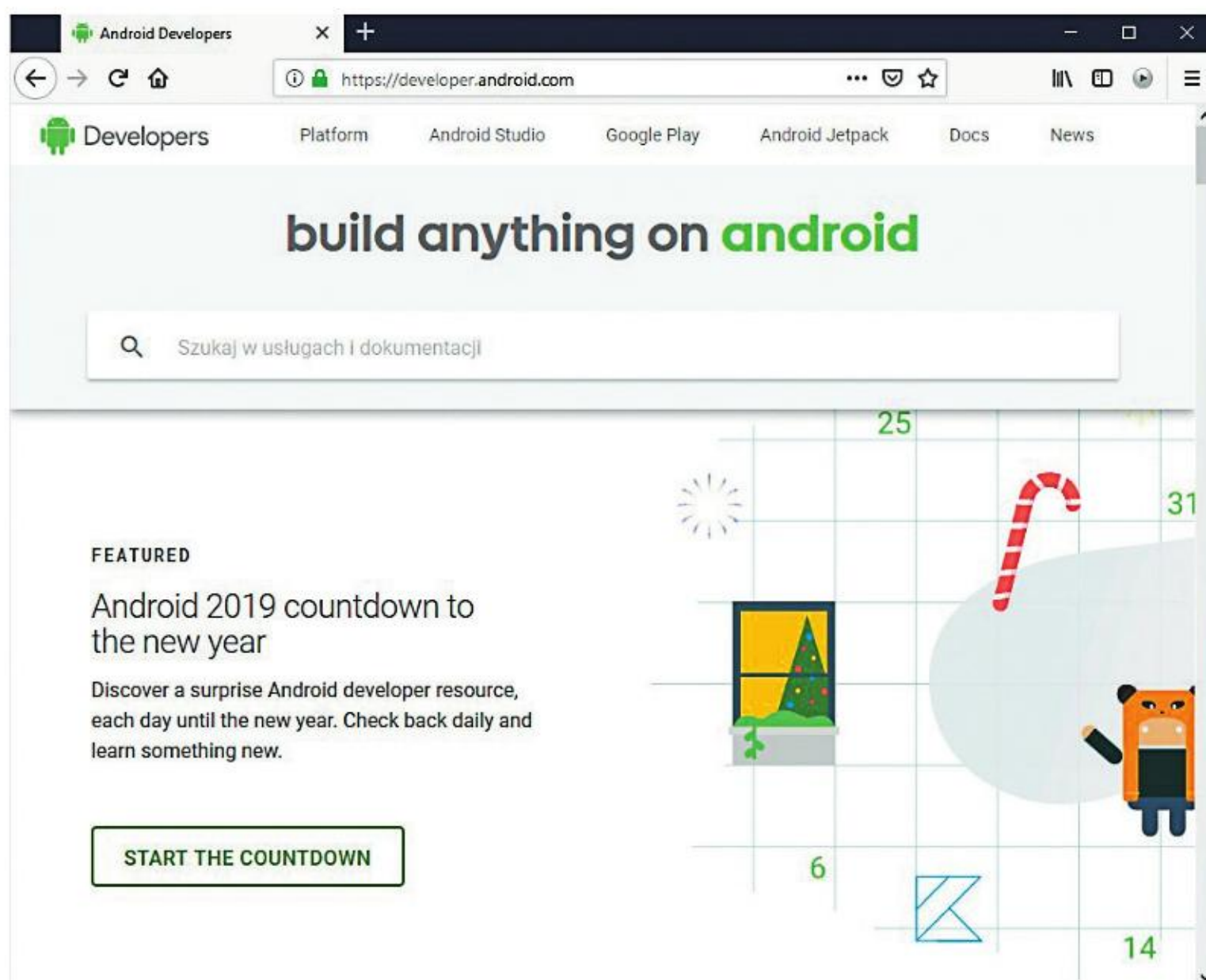
Technologie webowe na urządzenia mobilne

Aplikacje mobilne to specyficzne oprogramowanie użytkowe do wykorzystania w urządzeniach przenośnych, takich jak tablety, notebooki lub smartfony.

Aplikacje mobilne mają inne cechy niż aplikacje desktopowe czy aplikacje internetowe. Jest to uwarunkowane przede wszystkim ograniczonymi wymaganiami technicznymi, mniejszym rozmiarem pliku bądź prostszym interfejsem użytkownika. Inną cechą aplikacji mobilnych jest ich brak kompatybilności między różnymi platformami systemowymi urządzeń przenośnych.

Na rynku urządzeń mobilnych przodują systemy operacyjne iOS oraz Android. Nie wielki udział w rynku mają również urządzenia z systemem Windows Phone. Istotnym problemem podczas tworzenia aplikacji dla tych systemów jest wybór narzędzi programistycznych. Obecnie wykorzystywane narzędzia programistyczne do tworzenia aplikacji mobilnych można podzielić na trzy generacje:

- **generacja 1** – rozwiązania dedykowane konkretnej platformie sprzętowej. Podczas pisania aplikacji na system iOS możemy zastosować języki Objective-C lub Swift oraz narzędzie w postaci środowiska programistycznego XCode. W systemie Android jedynym słusznym wyborem będzie język Java, a środowiskiem programistycznym – Android Studio. Dla systemów Windows Phone dobrym rozwiązaniem będzie C#;
- **generacja 2** – rozwiązania cross-platformowe, hybrydowe i wykorzystujące język HTML oraz JavaScript. Najbardziej popularnymi rozwiązaniami są tutaj PhoneGap i Cordova – pozwalają na generowanie z kodu HTML aplikacji działających na wszystkich systemach;
- **generacja 3** – rozwiązania natywne – cross-platformowe. To połączenie zalet dwóch poprzednich generacji (po wyeliminowaniu większości wad). Mamy tu dostępne dwa narzędzia – Xamarin oraz React Native.



Rys. 3.2. Witryna platformy deweloperskiej Android

Oprócz tworzenia aplikacji, twórcę czeka też proces jej dystrybucji. Każdy użytkownik urządzeń przenośnych wie, że podstawowym źródłem pozyskiwania aplikacji mobilnych są sklepy internetowe nadzorowane przez firmy produkujące systemy operacyjne. Autor aplikacji mobilnej najczęściej chce ją wprowadzić do dystrybucji w dedykowanym sklepie. Taką możliwość mają tylko zarejestrowani deweloperzy, co wiąże się z cykliczną opłatą członkowską. Poza tym wszystkie aplikacje muszą przejść proces certyfikacji, zwłaszcza wtedy, gdy warunkiem ich pracy jest korzystanie z danych o użytkowniku czy ingerencji w konfigurację urządzenia.

W procesie tworzenia i dystrybucji aplikacji mobilnych nieocenioną rolę odgrywają platformy deweloperskie. Najważniejsze z nich to:

- Apple: <https://developer.apple.com>,
- Android: <http://developer.android.com>,
- Windows: <https://dev.windows.com>.

Rozwój urządzeń mobilnych narzuca projektowanie i tworzenie stron WWW obsługiwanych w wielu przeglądarkach i wielu rozdzielczościach. Popularne stało się tworzenie **stron responsywnych**, czyli takich, których wygląd i układ dostosowywały się automatycznie do rozmiaru okna przeglądarki, na której były wyświetlane.



Rys. 3.3. Zasada działania stron responsywnych

 SPRAWDŹ SWOJĄ WIEDZĘ

1. Wejdź na stronę organizacji W3C (<http://www.w3.org>) i odszukaj specyfikacje dotyczące języka HTML. Zapoznaj się ze strukturą opisu specyfikacji języka stosowanej przez W3C.
2. Wyjaśnij, dlaczego korzysta się z tworzenia stron responsywnych.
3. Jakie technologie webowe zastosujesz do tworzenia prostej strony informacyjnej zakładu zegarmistrzowskiego oraz sklepu internetowego? Swoje propozycje zapisz w zeszycie w formie tabeli.

Rodzaj strony	Proponowane technologie
strona informacyjna	
sklep internetowy	

4. Wyszukaj w zasobach sieci Internet informacje dotyczące usług chmury obliczeniowej. Sporządź w zeszycie notatkę zawierającą zestawienie ofert dwóch wybranych dostawców tej usługi.

III. Tworzenie stron internetowych w języku HTML

- Struktura dokumentu HTML
- Podstawowe znaczniki i atrybuty języka HTML
- Barwa i grafika na stronie internetowej
- Podział strony w języku HTML
- Grafika i multimedia w dokumentach HTML

4

Struktura dokumentu HTML

ZAGADNIENIA

- Składnia języka HTML
- Struktura dokumentu HTML
- Zasady edycji plików HTML

Każda strona internetowa jest dokumentem tekstowym. W związku z tym można ją utworzyć w dowolnym edytorze tekstu, takim jak Notatnik, NotePad++, MS Word, Brackets czy Aptana Studio. Strony internetowe są plikami z rozszerzeniem .html lub .php.

Dokument utworzony w języku HTML ma określoną strukturę, czyli kod dokumentu zawiera elementy niezbędne do poprawnej interpretacji przez przeglądarkę internetową. Ponadto cały dokument musi być napisany zgodnie z regułami składni języka HTML.

Składnia języka HTML

Strukturę strony w języku HTML tworzy się za pomocą tzw. **znaczników** (z ang. **tagów**). Każdy znacznik jest ograniczony symbolami `<` oraz `>`. Większość znaczników ma atrybuty, a każdy atrybut – swoje wartości. Znaczniki powinny być pisane małymi literami.

PRZYKŁAD 4.1

```
<znacznik atrybut="wartość"> tekst </znacznik>
```

Znaczniki dzieli się na:

- **sparowane**, czyli parę znaczników, z których jeden otwiera element składni języka HTML, np. `<title>`, a drugi służy do jego zamknięcia, np. `</title>`;
- **niesparowane**, składające się tylko ze znacznika otwierającego, np. `<b>` lub `<hr>`.

Dodatkowo w składni języka HTML stosuje się **znaki specjalne**, np. ` ` (pozwalający na wstawienie twardej spacji). Składnia języka HTML dopuszcza również stosowanie komentarzy. Budowa komentarza wygląda następująco:

```
<!--komentarz -->
```

Składnia języka HTML dopuszcza zagnieżdżanie znaczników. Należy jednak pamiętać, że zamykamy je w odwrotnej kolejności do tej, w której je otwieraliśmy.

PRZYKŁAD 4.2

```
<p><b><i><font color="green">przykład zagnieżdżenia znaczników
</font></i></b></p>
```

Struktura dokumentu HTML

W strukturze prawidłowego dokumentu HTML należy wyróżnić następujące elementy:

- deklarację typu dokumentu – **DOCTYPE**;
- definicję dokumentu – **HTML**;
- nagłówek dokumentu – **HEAD**;
- treść dokumentu, tzw. ciało strony – **BODY**.

```
1 <!DOCTYPE html>
2 <!--deklaracja typu dokumentu HTML5-->
3 <html>
4 <!--znacznik początku dokumentu HTML-->
5 <head>
6 <!--nagłówek zawierający parametry dotyczące wyświetlania strony
7 oraz dołączania skryptów i plików zewnętrznych-->
8 <meta charset="utf-8">
9 <!--sposób kodowania polskich znaków-->
10 <link rel="stylesheet" href="plik.css">
11 <!--sposób dołączania zewnętrznego arkusza stylów do formatowania strony-->
12 <title> Tytuł strony </title>
13 <!--tytuł strony wyświetlany w zakładce przeglądarki-->
14 </head>
15 <!--zamknięcie nagłówka strony-->
16 <body>
17 <!--znacznik początku treści strony-->
18 Treść strony internetowej wyświetlany w oknie głównym przeglądarki
19 </body>
20 <!--zamknięcie treści strony-->
21 </html>
22 <!--zamknięcie dokumentu HTML-->
23
```

Rys. 4.1. Struktura dokumentu HTML 5

Deklarację typu dokumentu definiuje się przez określenie standardu **DTD** (ang. *Document Type Definition*), w jakim tworzy się stronę internetową. Specyfikacja języka HTML 4.01 ma trzy typy dokumentów: **Strict**, **Transitional** i **Frameset**.

PRZYKŁAD 4.3

Wersja **Strict** definicji typu dokumentu (DTD) wyklucza wszelkie elementy prezentacyjne, które specyfikacja HTML 4.01 uznaje za zdeprecjonowane, czyli przestarzałe. Występuje ona często z dodanym adresem sieciowym wersji DTD, pomagającym w pobraniu definicji przez przeglądarkę.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.
w3.org/TR/html4/strict.dtd">
```


Wersja **Transitional** (przejściowa lub luźna) zawiera wszystko, co w Strict, a ponadto elementy i atrybuty zdeprecjonowane oraz element `<iframe>`.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

Wersja **Frameset** (obsługująca ramki) zawiera wszystko, co w Transitional, a ponadto elementy odnoszące się do ramek: `<frame>`, `<frameset>` oraz `<noframes>`.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
```

W specyfikacji HTML 5 zapis typu dokumentu jest znacznie uproszczony.

```
<!DOCTYPE html>
```

W XHTML nie ma żadnego typu deklaracji.

Definicja dokumentu określa język, w jakim napisano stronę. Najczęściej używanym znacznikiem jest `<html>`. Możliwe jest także zastosowanie dodatkowych parametrów do określenia typu języka lub ustawień regionalnych.

PRZYKŁAD 4.4

```
<html lang="pl-PL">
```

Nagłówek dokumentu zawiera informacje o stronie, które nie są wyświetlane w oknie przeglądarki. Najczęściej stosowanymi znacznikami w nagłówku są:

- `<title>` – określa tytuł strony;
- `<style>` – definiuje wewnętrzne arkusze stylów;
- `<link>` – odsyłacz do innego dokumentu hipertekstowego;
- `<meta>` – zawiera szczegółowe informacje o stronie, np. kodowanie, słowa kluczowe, dane autora.

PRZYKŁAD 4.5

Znacznik `<meta>` pozwala umieścić w nagłówku wiele istotnych informacji. W tym znaczniku korzysta się z następujących atrybutów: **http-equiv**, **name** oraz **content**. Na rys. 4.2 przedstawiono przykłady wykorzystania tych atrybutów.


```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <!--sposób kodowania polskich znaków-->
6      <meta http-equiv="Content-Language" content="pl">
7      <!--sposób definiowania języka dla zawartości strony-->
8      <meta http-equiv="reply-to" content="bb@pl.pl">
9      <!--sposób podania adresu e-mail autora strony-->
10     <meta http-equiv="refresh" content="30">
11     <!--sposób zdefiniowania odświeżania strony np. co 30 sek.-->
12     <meta http-equiv="creation-date" content="2017-09-14T12:15:38Z">
13     <!--sposób zdefiniowania daty i czasu utworzenia strony-->
14     <meta http-equiv="default-style" content="preferowane stylesheet">
15     <!--sposób określenia preferowanych stylów dla strony-->
16     <meta name="description" content="przykładowa strona WWW">
17     <!--sposób opisu treści strony-->
18     <meta name="keywords" content="html, strona, informatyka">
19     <!--sposób określania słów kluczowych-->
20     <meta name="author" content="Bartilini Babacki">
21     <!--sposób na podanie autora strony-->
22     <meta name="generator" content="Notepad++">
23     <!--sposób na określenie edytora kodu-->
24     <meta name="robots" content="index">
25     <!--sposób na indeksowanie strony-->
26     <meta name="copyright" content="(c) 2005 Jan Nowak">
27     <!--sposób na zdefiniowanie praw autorskich-->
28 </head>
29 <body>
30     Treść strony internetowej wyświetlany w oknie głównym przeglądarki
31 </body>
32 </html>

```

Rys. 4.2. Przykład stosowania znacznika `<meta>`

Kodowanie polskich znaków

Podczas tworzenia dokumentu w języku HTML należy pamiętać, aby treść wyświetlała się w przeglądarce prawidłowo. Istotnymi etapami tworzenia strony są deklaracja języka dla strony oraz określenie sposobu kodowania znaków diakrytycznych (np.: ż, ą, ź). Dokonuje się tego w znaczniku `<meta>`. Najczęściej używane kodowania polskich znaków to:

- iso-8859-2,
- utf-8,
- windows-1250.

Kodowanie utf-8 jest obecnie najczęściej stosowanym standardem na witrynach internetowych.

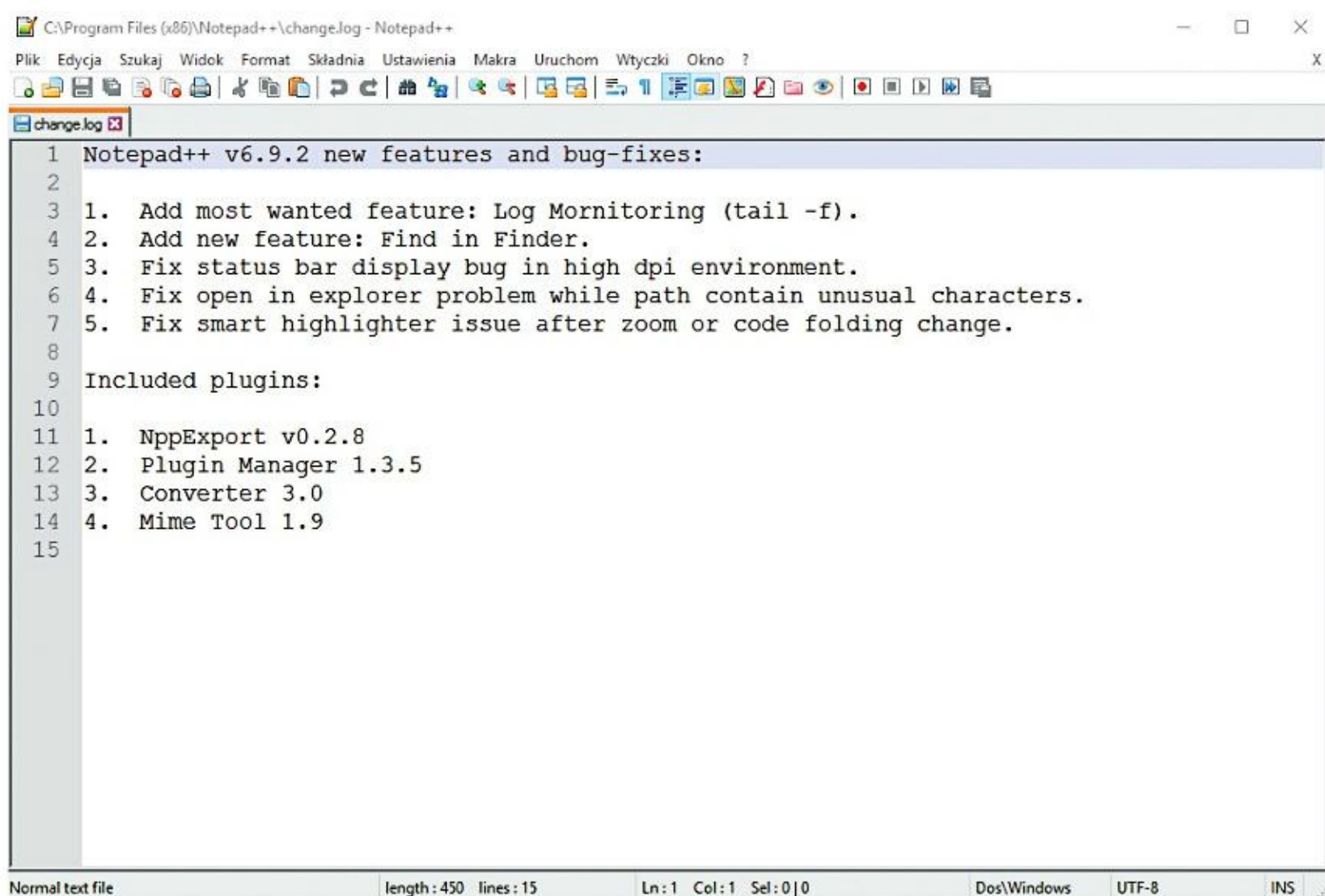
Treść dokumentu to obszar strony znajdujący się między znacznikami `<body>` i `</body>`. Zawiera treść dokumentu w postaci tekstu, obrazów, multimediiów itp., wyświetlaną w obszarze roboczym przeglądarki internetowej.

Narzędzia do edycji dokumentów HTML

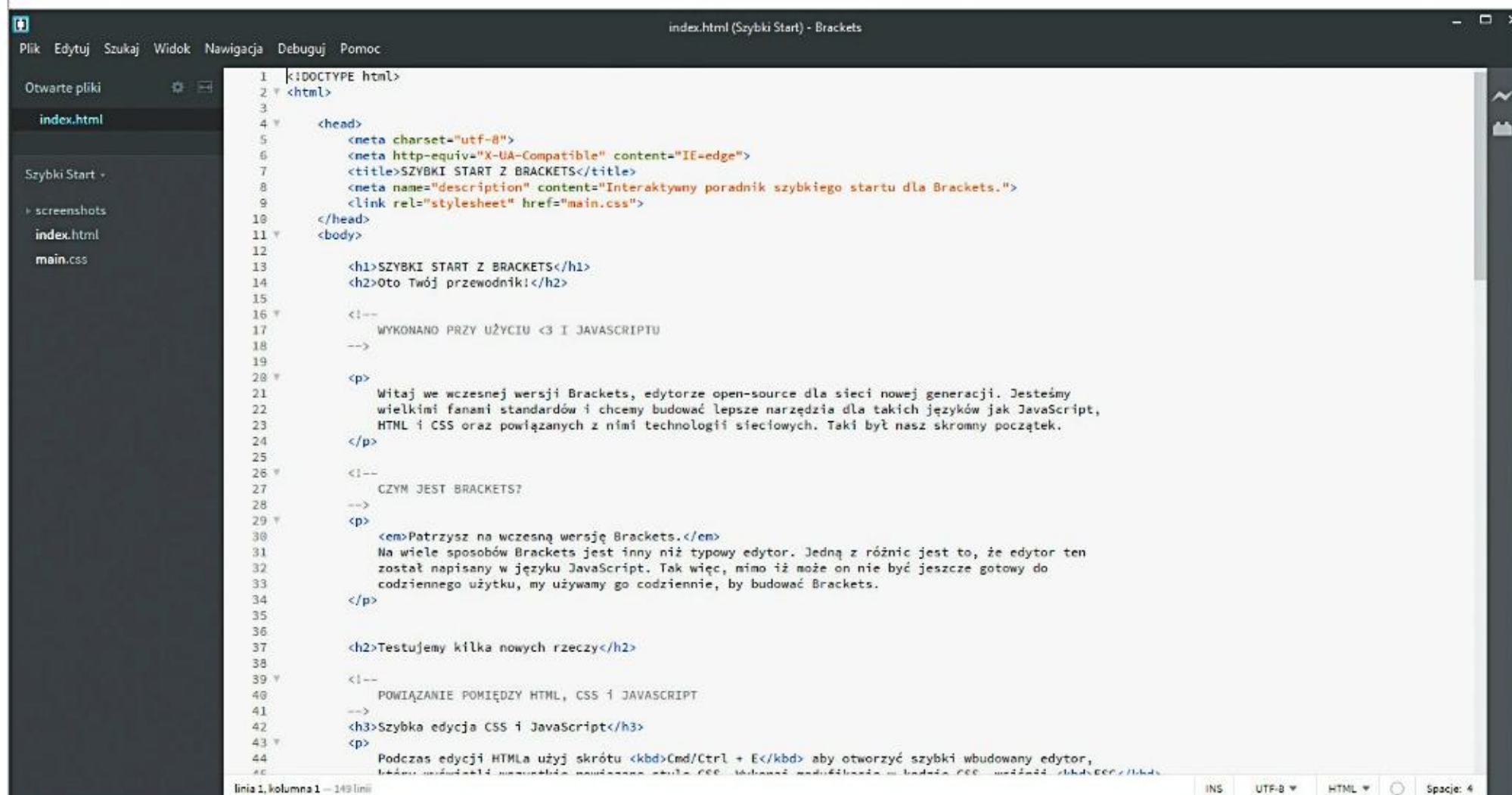
Domyślnym plikiem otwierającym dowolną witrynę internetową jest plik o nazwie **index.html** lub **index.php**.

Dokument w języku HTML można edytować w dowolnym edytorze, zaczynając od najprostszego notatnika systemowego, który jest jednak rzadko stosowanym programem. Najczęściej programiści korzystają z aplikacji: NotePad++, Brackets bądź Aptana Studio oraz edytorów spełniających wymagania WYSIWYG.

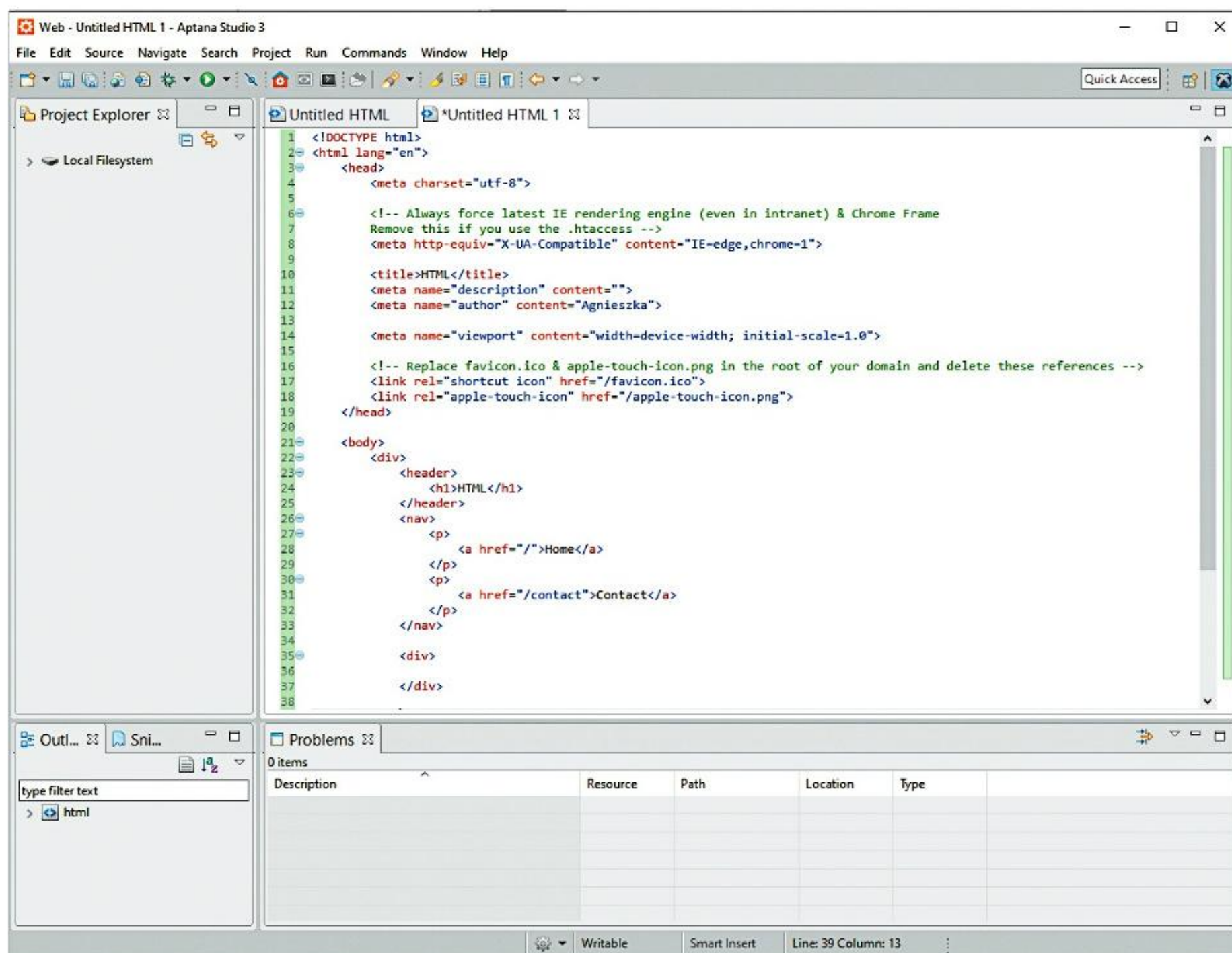
W podręczniku będziemy korzystać z edytora Brackets. Notepad++, który można pobrać, korzystając ze strony <https://notepad-plus-plus.org/>, natomiast Brackets – ze strony <http://brackets.io/index.html>.



Rys. 4.3. Okno programu Notepad++



Rys. 4.4. Okno programu Brackets



Rys. 4.5. Okno programu Aptana Studio

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Co oznacza skrótowiec DTD?
2. Scharakteryzuj znane ci edytory języka HTML.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Pobierz, z podanej w treści rozdziału lokalizacji, program Notepad++ i zainstaluj go na swoim stanowisku komputerowym. Po instalacji uruchom aplikację i zapoznaj się z jej budową oraz wyglądem.
2. Skorzystaj z dowolnego edytora kodu HTML i utwórz dokument zawierający prawidłową strukturę w języku HTML 5. W nagłówku strony zdefiniuj kodowanie polskich znaków w standardzie iso-8859-2, autora strony, odświeżanie strony co 65 sekund oraz umieść dane dotyczące daty i czasu utworzenia strony. Utwórz również nazwę strony „Pierwsza strona”, która będzie wyświetlana w zakładce przeglądarki. W treści strony napisz pełną nazwę swojej szkoły. Zapisz dokument pod nazwą index.html, a następnie otwórz go w dowolnej przeglądarce internetowej. Sprawdź, czy polskie znaki wyświetlają się właściwie. Jeżeli nie, zastosuj inny standard kodowania.

5

Podstawowe znaczniki i atrybuty w języku HTML

ZAGADNIENIA

- Składnia poleceń w języku HTML
- Formatowanie tekstu
- Elementy liniowe i blokowe
- Znaczniki formatowania tekstu
- Określanie barwy w kodzie HTML

Składnia poleceń w języku HTML

Jak wspomniano w poprzednim rozdziale, elementy języka HTML nazywa się znacznikami lub tagami. Opis wszystkich znaczników, atrybutów oraz ich wartości jest dostępny na stronach internetowych W3C.

Niemal każdy znacznik ma przynajmniej jeden atrybut, któremu można przypisać określone wartości. Atrybut to jeden z podstawowych elementów składni języka HTML. Atrybuty określają cechy wybranego znacznika. Umieszcza się je zawsze w znaczniku otwierającym. W celu nadania danemu tagowi wybranego atrybutu stosuje się następujący zapis:

```
<nazwa atrybutu="wartość atrybutu">
```

PRZYKŁAD 5.1

```
<p align="center">Moja pierwsza strona internetowa</p>
```

Polecenie spowoduje utworzenie akapitu, w którym zostanie wyśrodkowany tekst.

W tabeli 5.1 podano najczęściej stosowane atrybuty znaczników języka HTML, które opisano wcześniej.

Tabela 5.1. Wybrane atrybuty HTML

Atrybut	Znacznik, z którym jest stosowany	Opis funkcji
charset	<code><meta></code>	określa typ kodowania znaków
content	<code><meta></code>	służy do określania wartości związanej z atrybutami name i http-equiv
download	<code><a></code>	określa, że rozpocznie się pobieranie zasobu po naciśnięciu przez użytkownika tego linku
href	<code><a></code>	opisuje adres URL dla danego odnośnika
lang	atrybut globalny	określa język dla danego elementu
name	<code><meta></code> i inne	określa nazwę dla danego elementu strony
rel	<code><a></code> , <code><link></code>	definiuje powiązanie z określonym zasobem lokalnym lub sieciowym

Listę wszystkich atrybutów dla znaczników HTML można znaleźć na stronie W3C (https://www.w3schools.com/tags/ref_attributes.asp).

Wszystkie znaczniki języka HTML dzieli się na grupy:

- znaczniki edycji i formatowania tekstu;
- znaczniki do tworzenia list i wykazów;
- znaczniki do tworzenia odsyłaczy;
- znaczniki podziału strony;
- znaczniki tworzące tabele;
- znaczniki tworzące formularze;
- znaczniki do wstawiania elementów multimedialnych oraz API.

Co nowego w HTML 5

W najnowszej specyfikacji języka HTML 5 wprowadzono zmiany w stosunku do HTML 4.01. Do najważniejszych z nich zalicza się dopisanie nowych elementów dotyczących:

- podziału strony, np.: `<header>`, `<footer>`, `<nav>`, `<article>` oraz `<section>`;
- atrybutów znaczników do tworzenia formularzy, np.: `number`, `date`, `time`, `calendar` oraz `range`;
- obsługi grafiki na stronie, np.: `<svg>` oraz `<canvas>`;
- obsługi multimedii, np.: `<audio>` oraz `<video>`.

Zmiany dotyczą również składni języka, deklaracji typu dokumentu oraz poszerzenia listy elementów zdeprecjonowanych (ang. *deprecated*), które wychodzą z użycia.

PRZYKŁAD 5.2

W HTML 5 wyszły z użycia znaczniki tworzące ramki: **<frame>**, **<frameset>**, **<noframes>**. W związku z tym obecnie nie można dokonywać podziału strony według poniższego przykładu.



Rys. 5.1. Niepoprawne użycie znaczników **<frameset>** oraz **<frame>** w HTML5.

HTML 5 jest bardziej otwarty na współpracę w ramach **API** (ang. *Application Programming Interface*). Pozwala to na wykorzystanie nowych technologii, np. geolokalizacji. Dodatkowo HTML 5 nie ma żadnych elementów prezentacyjnych. Prezentacją treści strony zajmuje się całkowicie **CSS** (ang. *Cascading Style Sheets*), czyli kaskadowe arkusze stylów, o których będzie mowa w dalszej części podręcznika.

Znaczniki formatowania tekstu

Znaczniki do wyświetlania i formatowania tekstu używa się w sekcji **<body>**. Mimo że większość znaczników prezentacji treści wycofano ze specyfikacji HTML 5, nadal można korzystać z elementów określających proste formatowanie tekstu w oknie przeglądarki internetowej. W tabeli 5.2 przedstawiono podstawowe znaczniki do formatowania tekstu.

Tabela 5.2. Znaczniki formatowania tekstu

Znaczniki	Opis
tekst	pogrubienie
<i>tekst</i>	kursywa
<u>tekst</u>	podkreślenie
<s>tekst</s>	przekreślenie

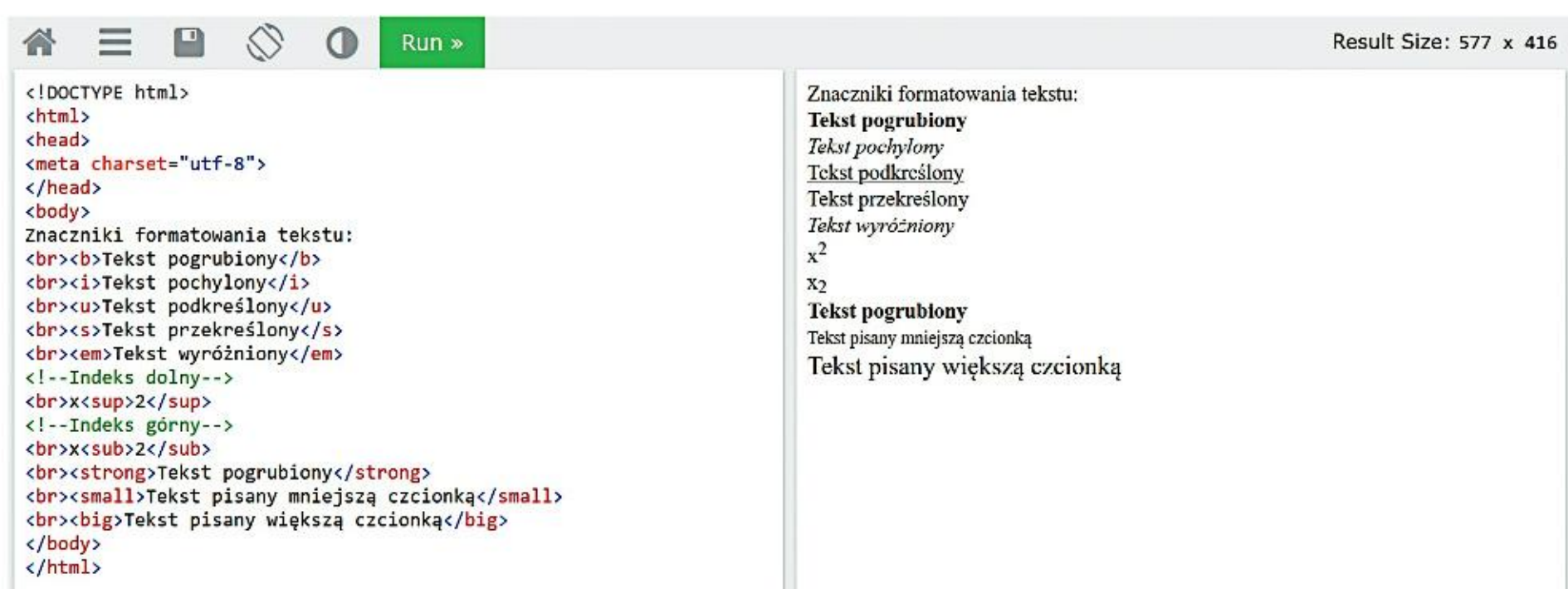
Znaczniki	Opis
<code><mark>tekst</mark></code>	podświetlenie
<code>tekst</code>	definiowanie czcionki
<code>tekst</code>	wyróżnienie tekstu
<code><sub>tekst</sub></code>	indeks dolny
<code><sup>tekst</sup></code>	indeks górny
<code>tekst</code>	czcionka mocno pogrubiona
<code><dfn>tekst</dfn></code>	znacznik używany do oznaczenia definicji
<code><small>tekst</small></code>	tekst pisany czcionką zmniejszoną o 1
<code><big>tekst</big></code>	tekst pisany czcionką zwiększoną o 1

! UWAGA

Przedstawione znaczniki pozwalają na proste formatowanie wyświetlanego tekstu na stronie internetowej. Wraz z rozwojem języka HTML liczba tych znaczników systematycznie się zmniejsza. Wiąże się to z założeniem, że za prezentację treści strony powinny być odpowiedzialne kaskadowe arkusze stylów.

PRZYKŁAD 5.3

Na rys. 5.2 przedstawiono działanie poszczególnych znaczników do formatowania tekstu.

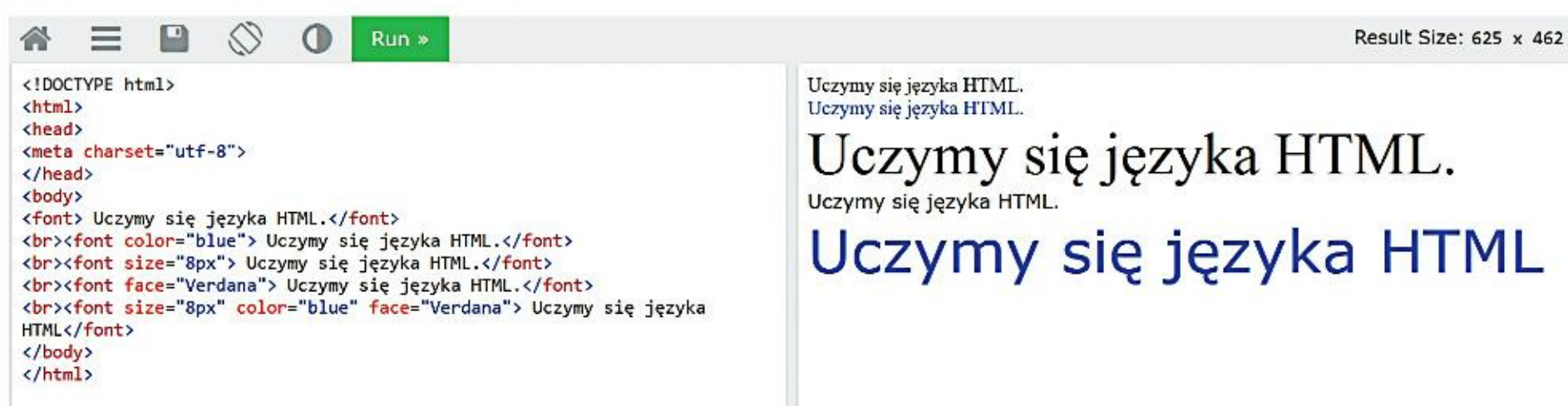


Rys. 5.2. Zastosowanie znaczników do formatowania tekstu

Znacznik `<font>` bez użycia atrybutów nie sformatuje wybranego tekstu. Aby sformatować czcionkę danego obszaru tekstu, trzeba określić kolor, styl, a także rozmiar czcionki.

PRZYKŁAD 5.4

Jeśli zastosuje się znacznik **** bez żadnych atrybutów, dany tekst zostanie wyświetlony za pomocą domyślnych ustawień czcionki. Nadanie tagowi atrybutu **color="blue"** spowoduje, że wybrany tekst zostanie napisany niebieską czcionką. Jeżeli zamierza się określić rozmiar czcionki, należy zastosować atrybut **size** i przypisać mu np. wartość w pikselach. Aby określić rodzaj kroju pisma, trzeba użyć atrybutu **face**. Można oczywiście wszystkie cechy wpisać dla jednego znacznika – oddzielić je za pomocą spacji i zapisać w dowolnej kolejności.



Rys. 5.3. Zastosowanie znacznika ****

Znaki specjalne

Składnia języka HTML wymaga czasem wprowadzenia dodatkowych znaków, zwanych znakami specjalnymi. Można je wprowadzać do dokumentu hipertekstowego w postaci kodu dziesiętnego, szesnastkowego oraz nazwy własnej.

Tabela 5.3. Znaczniki edycji znaków specjalnych

Nazwa	Opis	Kod dziesiętny	Kod szesnastkowy	Przykład
"	znak zapytania (ang. <i>quotation mark</i>)	"	"	„
&	ampersand	&	&	&
 	spacja niełamaalna (ang. <i>non-breaking space</i>)	 	 	
±	plus minus	±	±	±
§	paragraf	§	§	§
€	znak euro	€	€	€
<	mniejszy niż	<	<	<
>	wiekszy niż	>	>	>

PRZYKŁAD 5.5

Przykładem elementu liniowego jest znacznik `<a>`, który służy do wstawiania odsyłacza do innej strony.

```
<a href="kontakt.html">Przejdź dalej </a>
```

Innymi elementami liniowymi są znaczniki: `<b>` – pogrubienie tekstu na stronie, `<i>` – pochylenie tekstu (kursywa) oraz `<u>` – podkreślenie tekstu.

```
<b>Ten tekst zostanie pogrubiony</b>
```

```
<i>Ten tekst zostanie pochyłony</i>
```

```
<u>Ten tekst zostanie podkreślony</u>
```

Przykładem elementu blokowego jest znacznik `<p>`, który powoduje umieszczenie na stronie nowego paragrafu (akapitu).

```
<p>
```

```
to jest treść pierwszego
```

```
akapitu tekstu strony
```

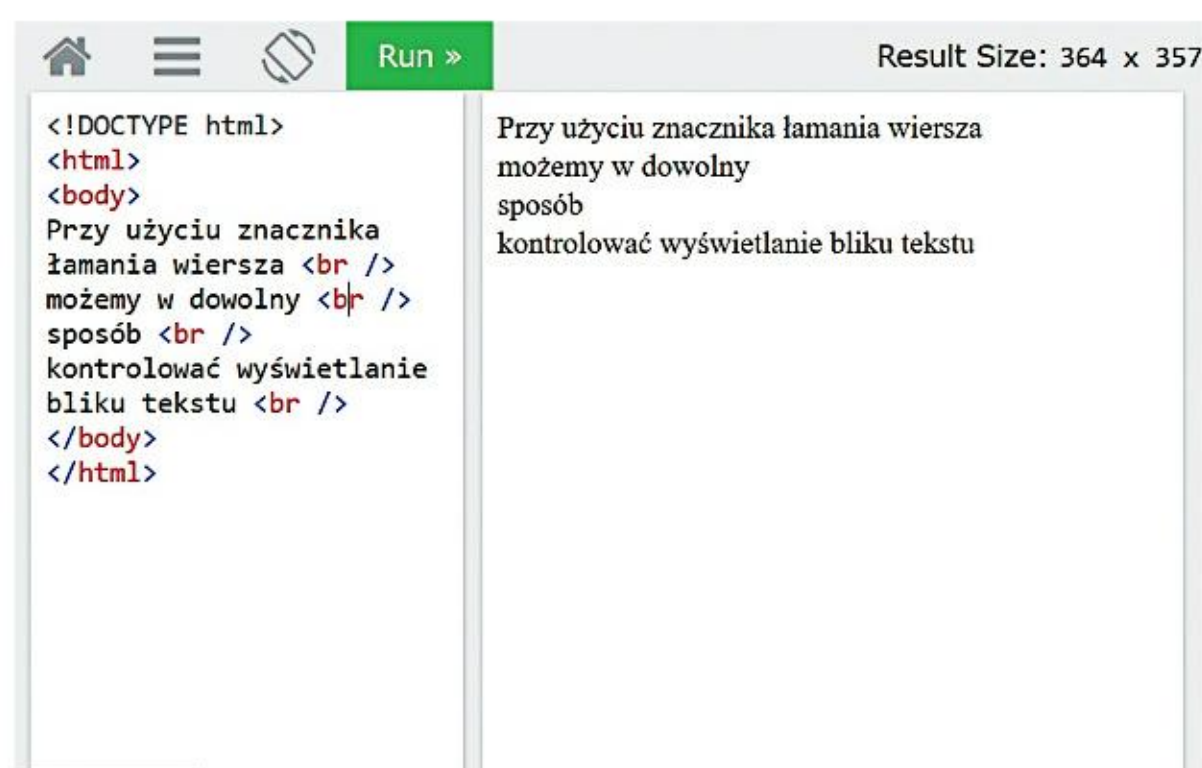
```
</p>
```

```
<p>Teraz umieszczamy nowy akapit</p>
```

Podczas tworzenia strony internetowej trzeba zadbać o jej czytelną zawartość. Oprócz zaprojektowania szablonu strony, należy też podzielić treść na określone **bloki tekstu**. Do tego służy celu grupa znaczników, które zostaną przedstawione.

Przełamanie wiersza

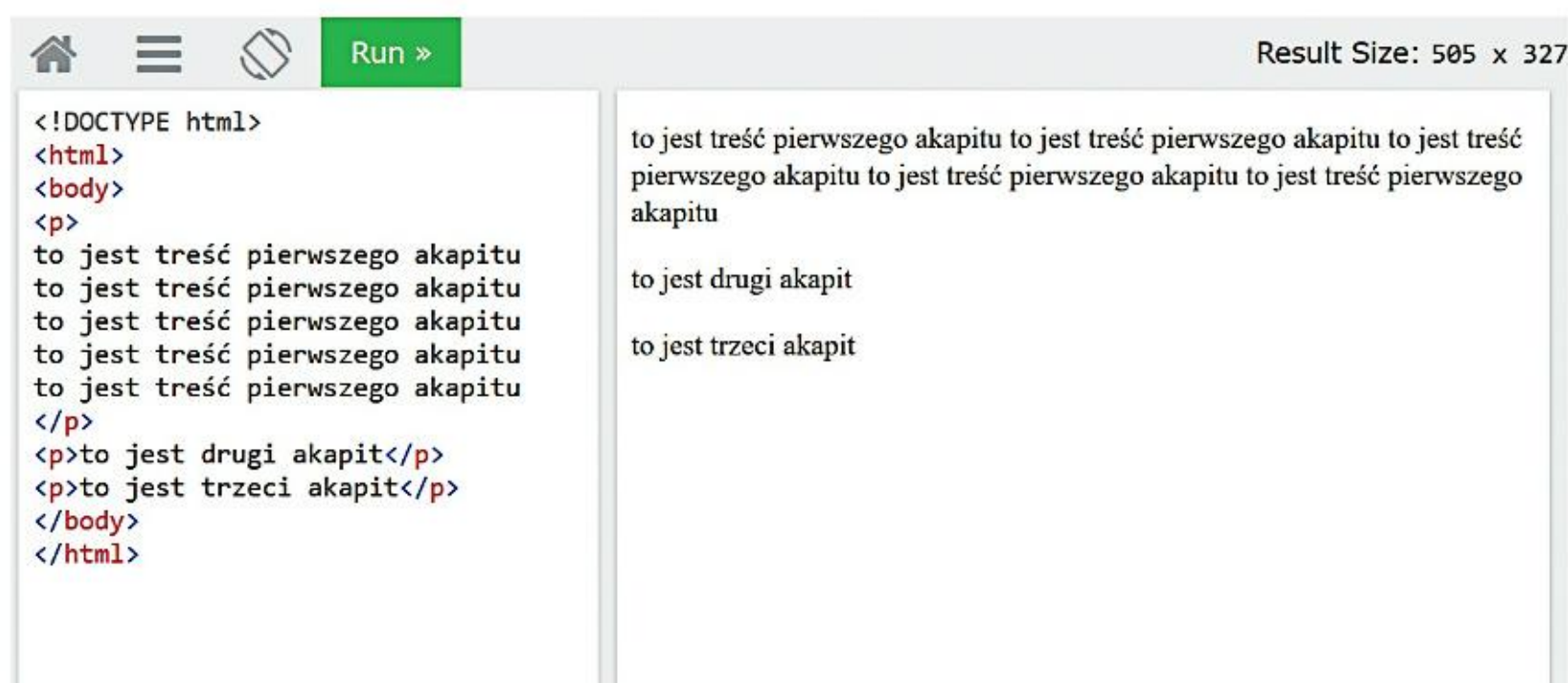
Znacznik przełamania wiersza `<br>` spełnia prostą funkcję przeniesienia wyświetlania tekstu do nowej linii. Dzięki niemu można kontrolować szerokość wiersza tekstu.

PRZYKŁAD 5.6

Rys. 5.5. Przykład użycia znacznika `<br>`

Akapit

Jest to część dłuższego tekstu, składająca się z jednego lub wielu zdań stanowiących pewną logiczną całość treści lub myśl. Tekst tworzący akapit umieszcza się wewnątrz znacznika `<p>` oraz `</p>`. Kolejne akapity zawsze umieszcza się jeden pod drugim.



Rys. 5.6. Przykład tworzenia akapitu

Nagłówki tekstu

Nagłówek dzieli treść strony na logicznie wyodrębnione części: nadaje im określone formatowanie, zależne od użytego znacznika. Znacznik nagłówka ma sześć stopni: od `<h1>` do `<h6>`. Należy jednak pamiętać, że nagłówek pierwszego stopnia definiuje największą czcionkę, a szóstego stopnia – najmniejszą.

PRZYKŁAD 5.7



Rys. 5.7. Nagłówki w dokumencie HTML

Znacznik linii poziomej

Linia pozioma jest linią ciągnącą się przez całą szerokość okna lub bloku, w którym została umieszczona. Atrybuty tego znacznika, tj. **noshade** (bez cieniowania), **width** (szerokość), **size** (grubość) oraz **align** (poziome wyrównanie), stosowano jeszcze w wersji HTML 4, natomiast w HTML 5 formatowanie przeniesiono do CSS.

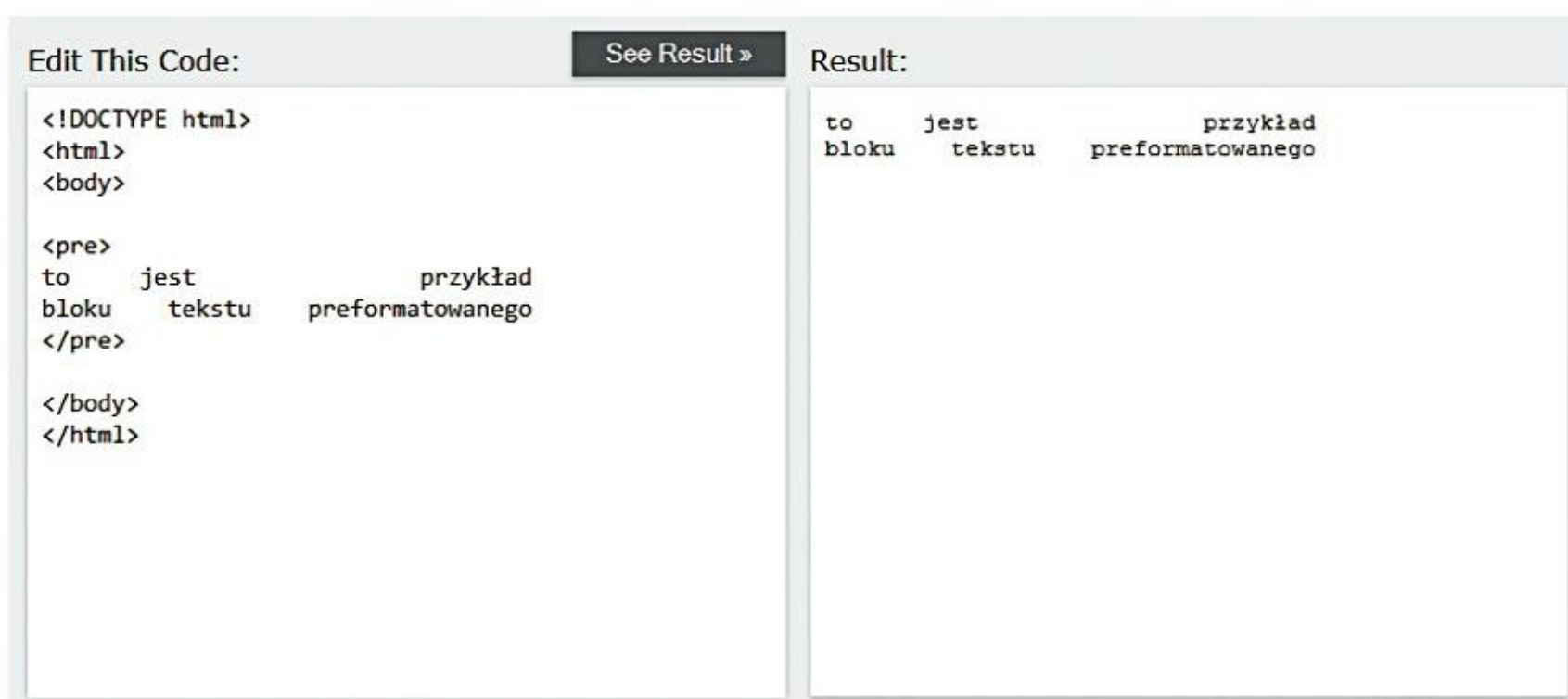
PRZYKŁAD 5.8

Przykład utworzenia linii poziomej o grubości 1 px, szerokości 400 px oraz wyrównanej do lewej krawędzi strony:

```
<hr size="1px" width="400px" align="left">
```

Bloki preformatowane

Tekst wpisany wewnątrz znaczników `<pre>` i `</pre>` zachowuje podczas wyświetlania w przeglądarce internetowej wszystkie spacje, tabulacje i złamania wiersza. Umożliwia to wymuszenia formatowania dla danego bloku tekstu.

PRZYKŁAD 5.9

Rys. 5.8. Przykład tekstu preformatowanego

Wykazy

Wykaz umożliwia tworzenie usystematyzowanych grup informacji. Można tworzyć **wykazy nieuporządkowane** (wypunktowane) oraz **uporządkowane** (numerowane).

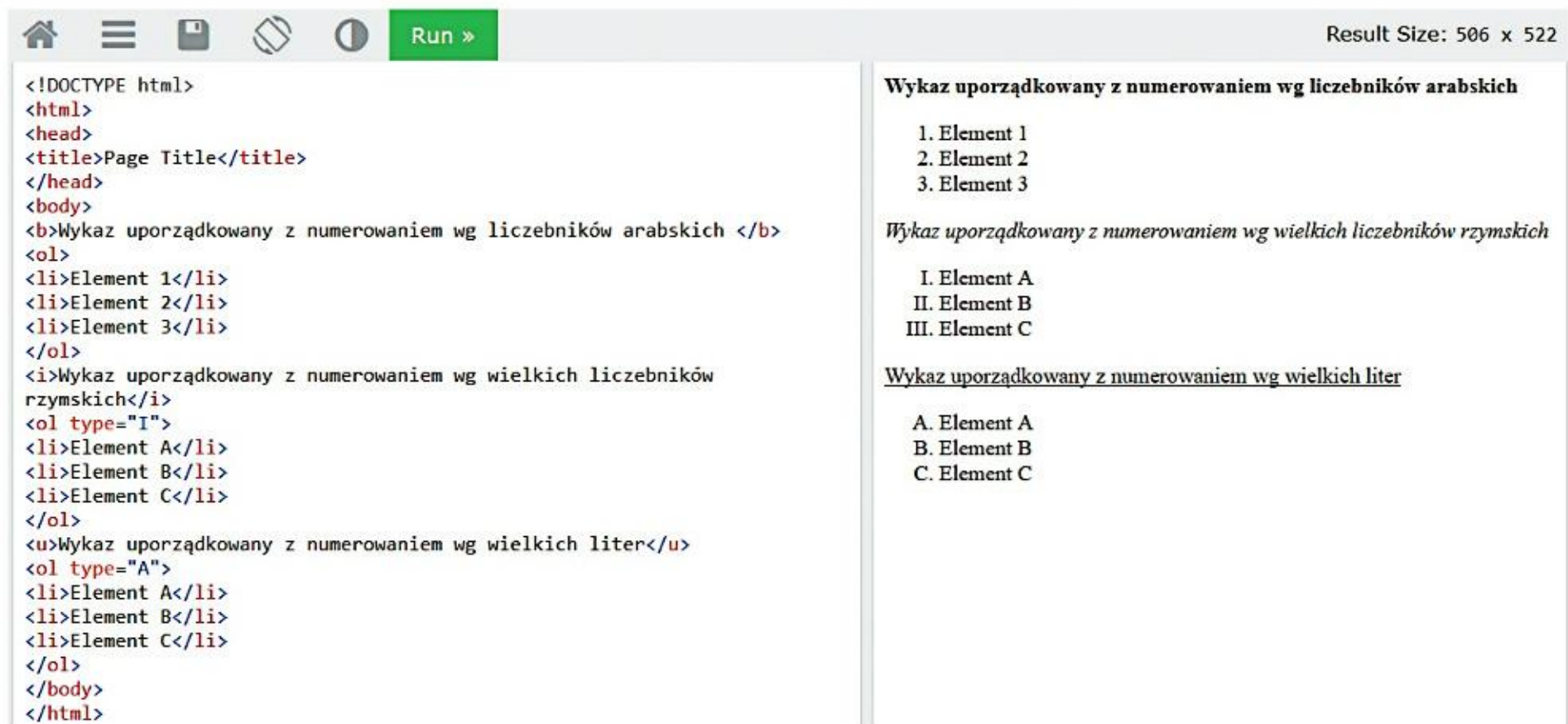
Aby utworzyć wykaz uporządkowany, należy zastosować znacznik `<ol> ...</ol>`, który spowoduje otwarcie nowej listy elementów. Dodanie elementów listy odbywa się za pomocą znacznika `<li> ...</li>`.

Znacznik `<ol>` ma atrybut **type**, który umożliwia określenie rodzaju stosowanego punktora w wykazie. Istnieją następujące wartości parametru **type**:

- **A** – numerowanie według wielkich liter;
- **a** – numerowanie według małych liter;
- **I** – numerowanie według wielkich liczebników rzymskich;
- **i** – numerowanie według małych liczebników rzymskich;
- **1** – numerowanie według liczebników arabskich.

Domyślną wartością jest numerowanie według liczebników arabskich.

PRZYKŁAD 5.10

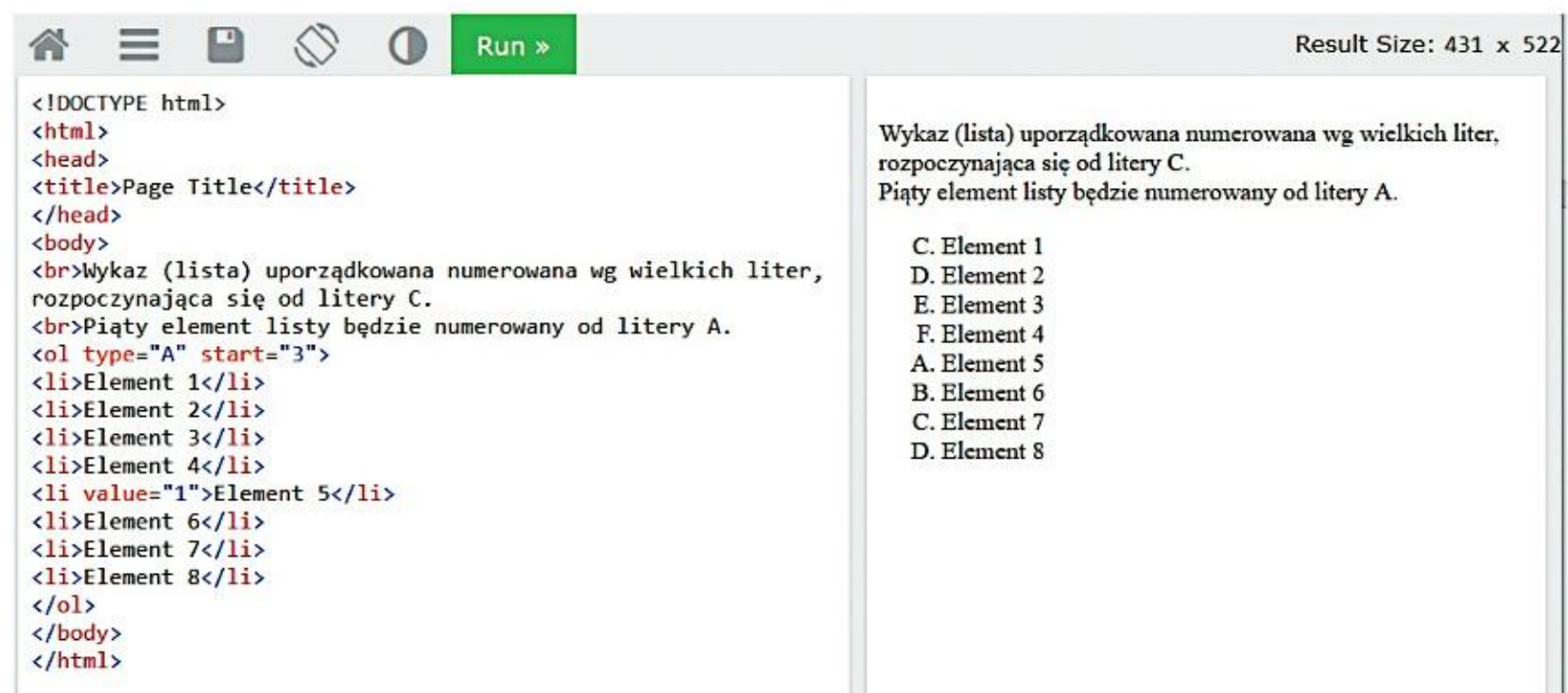


Rys. 5.9. Przykład wykazu uporządkowanego

Dla znacznika `<ol>` można również zastosować atrybut **start**, który określa wartość, od której ma się rozpocząć numerowanie listy. Ponadto znacznik `<li>` ma atrybut **value**, który umożliwia określenia wartości dla wybranego elementu listy.

PRZYKŁAD 5.11

Na rys. 5.10 pokazano przykład zastosowania wykazu uporządkowanego.

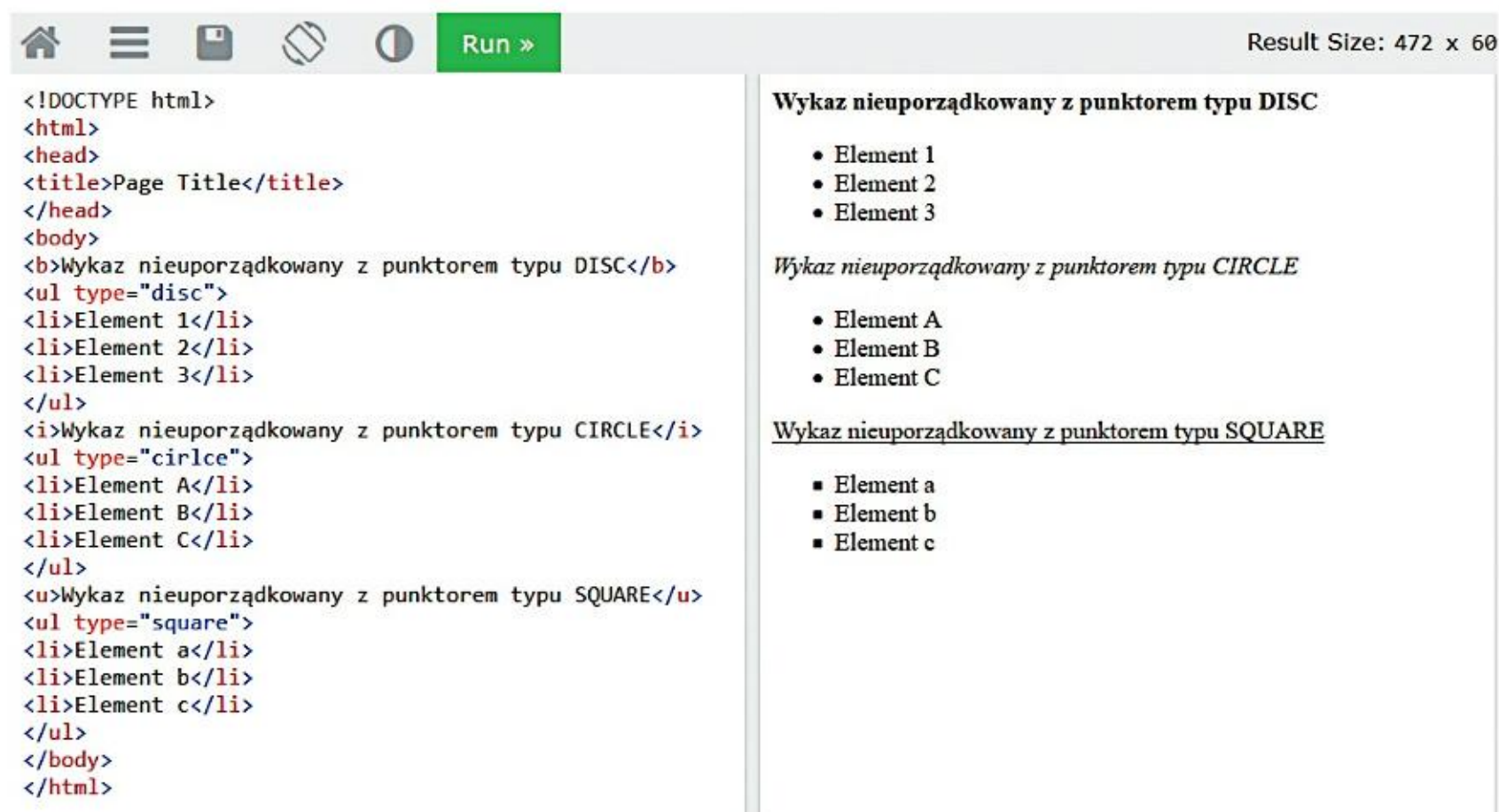


Rys. 5.10. Zastosowanie atrybutów start oraz value dla wykazu uporządkowanego

Aby utworzyć listę nieuporządkowaną, należy zastosować znacznik `<ul>...</ul>`. Kolejne elementy listy dodaje się za pomocą znacznika `<li>...</li>`. Znacznik `<ul>`, podobnie jak znacznik `<ol>`, ma atrybut **type**, który może przyjmować wartość **disc**, **circle** lub **square**. Domyślną wartością jest wartość **disc**.

PRZYKŁAD 5.12

Na rys. 5.11 pokazano przykład zastosowania wykazu nieuporządkowanego.



Rys. 5.11. Przykład wykazu nieuporządkowanego

Oprócz wykazów uporządkowanych i nieuporządkowanych, można tworzyć wykazy zagnieżdżone, czyli listy punktów z podpunktami. Podpunkty będą przesunięte bardziej w prawo w stosunku do punktów nadrzędnych, dzięki czemu wykaz będzie czytelny i zgodny z zasadami edycji tekstu.

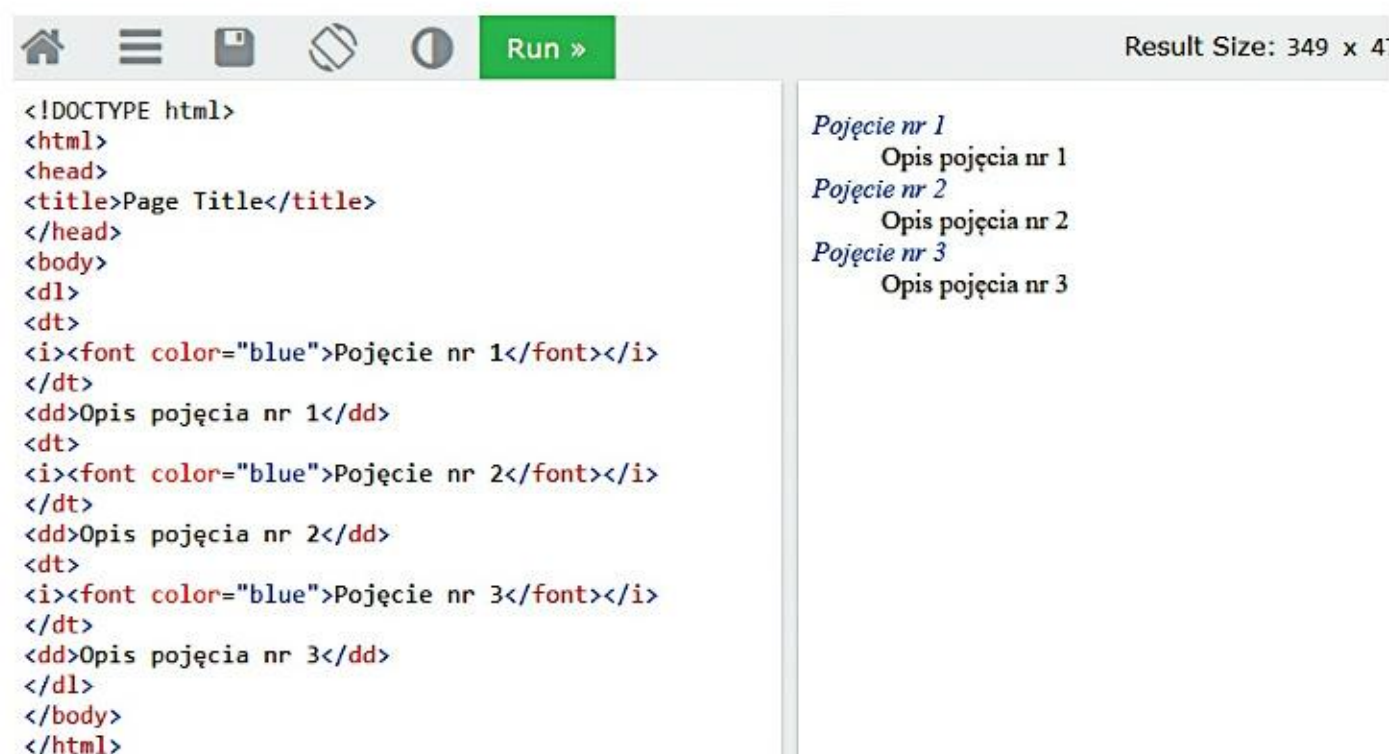
PRZYKŁAD 5.13



Rys. 5.12. Przykład wykazu zagnieżdżonego

Innym rodzajem wykazów są wykazy definicyjne. Umieszcza się w nich opisy określonych pojęć. Wykaz definicji rozpoczyna się znacznikiem `<dl>` (ang. *definition list*). Pojęcia w tego rodzaju liście wstawia się za pomocą znacznika `<dt>` (ang. *definition term*), natomiast opis pojęcia umieszcza się za pomocą znacznika `<dd>` (ang. *definition description*).

PRZYKŁAD 5.14



Rys. 5.13. Przykład wykazu definicyjnego

Wszystkie omówione wykazy: uporządkowany, nieuporządkowany i definicyjny, można w dowolny sposób zagnieżdżać, ale trzeba pamiętać o zachowaniu prawidłowej składni każdego z nich. Taki rodzaj wykazu nazywa się **mieszanym**.

PRZYKŁAD 5.15



Rys. 5.14. Przykład wykazu mieszanego

Odsyłacze (linki)

Zadaniem odsyłaczy jest utworzenie połączenia umożliwiającego przejście do innej strony internetowej, podstrony lub pliku. Rozróżnia się **odsyłacze absolutne** – do adresu internetowego w postaci formatu URL, oraz **odsyłacze względne** – do zasobu określonego ścieżką dostępu. Znacznikiem odsyłacza jest `<a>` (ang. *anchor*).

Przykład odsyłacza absolutnego:

```
<a href="http://www.wsip.pl"> Wydawnictwo Szkolne I Pedagogiczne  
</a>
```

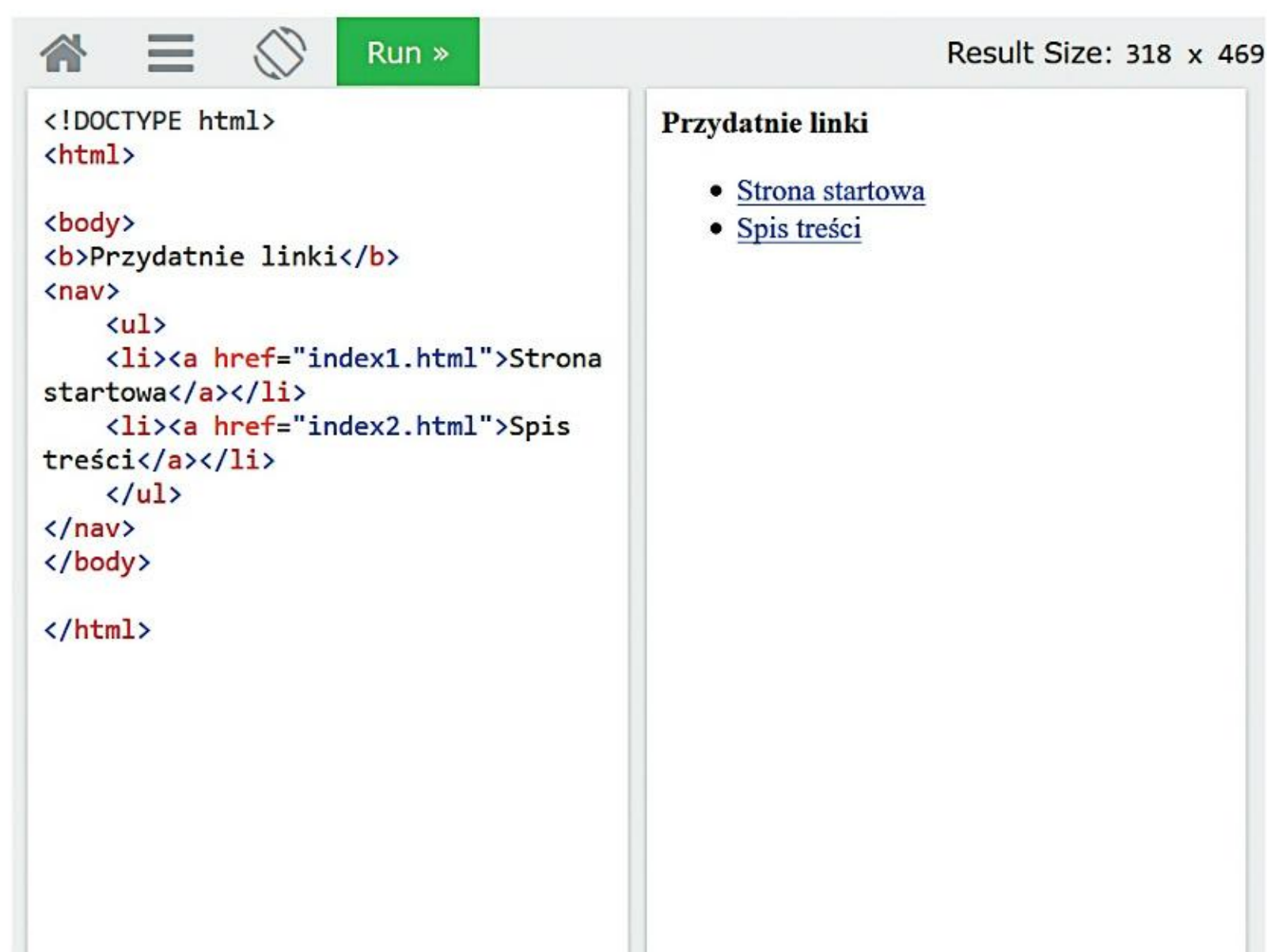
Przykład odsyłacza względnego:

```
<a href="C:\folder strony\index3.html"> Aktualności</a>
```

Znacznik `<nav>treść</nav>` reprezentuje sekcję strony, która zawiera odnośniki nawigacyjne do innych stron albo do określonych fragmentów na tej samej stronie. Nie wszystkie linki na stronie powinny być umieszczane w tym znaczniku. Przewidziano go przede wszystkim do oznaczania nim podstawowej nawigacji witryny.

PRZYKŁAD 5.16

Na rys. 5.15 przedstawiono przykład zastosowania omawianych znaczników do tworzenia wykazów i odsyłaczy.



Rys. 5.15. Przykład zastosowania odsyłaczy

Znacznik `<a>` umożliwia również tworzenie odsyłacza do skonfigurowanej skrzynki e-mail. W takiej sytuacji atrybut `href` uzupełnia się wyrażeniem „mailto:”, a po nim wpisuje adres e-mailowy, na który zamierza się wysłać wiadomość e-mail.

PRZYKŁAD 5.17

```
<a href="mailto:akowalski@wp.pl">Napisz do nas</a>
```

Po kliknięciu w link na ekranie pojawi się okno umożliwiające wysłanie wiadomości. Odsyłacz pozwala także na przejście między kolejnymi elementami na tej samej stronie. W tym celu należy umieścić tzw. kotwicę w treści strony, a odsyłacz tam, gdzie chcemy, aby kliknięcie w niego spowodowało przejście do danego miejsca na stronie.

PRZYKŁAD 5.18

Ustawienie kotwicy będzie wynikiem polecenia:

```
<a name="nazwakotwicy">Słowo kluczowe</a>
```

Umieszczenie odsyłacza odbywa się przez wpisanie nazwy kotwicy poprzedzone znakiem `#` do atrybutu `href`.

```
<a href="#nazwa_kotwicy">Opis odsyłacza</a>
```

Grupowanie elementów liniowych

Znacznik `<span>` służy do zgrupowania elementów liniowych, takich jak tekst bądź grafika. Grupowanie elementów ułatwia przypisanie do nich określonego stylu.

PRZYKŁAD 5.19



Rys. 5.16. Zastosowanie grupowania elementów liniowych

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Utwórz dokument HTML wyświetlający w przeglądarce internetowej pogrubiony i pochylony tekst:

Zegarek męski "A.Lange&Söhne" ©2017 Model Lange1 cal.L901.0™ Cena ±31000 €

2. W dowolnym edytorze utwórz dokument HTML o nazwie index.html, który pokaże działanie znaczników formatujących tekst zgodnie ze wzorem pokazanym na rysunku. Po zakończeniu pracy wyświetl gotowy plik w przeglądarce i porównaj wynik ze wzorem.

To jest poprawnie sformatowany **bardzo ważny tekst**

To jest poprawnie sformatowany tekst *wyróżniony*

To jest poprawnie sformatowany indeks dolny

To jest poprawnie sformatowany indeks górny

To jest poprawnie sformatowany tekst ~~przekreślony~~

To jest poprawnie sformatowana część *definicji*

To jest poprawnie sformatowany przykładowy **tekst pogrubiony**

To jest poprawnie sformatowany tekst podkreślony

To jest poprawnie sformatowany tekst *pochylony*

3. Utwórz dokument HTML wyświetlający wykaz w postaci spisu treści według wzoru. Zastosuj wykaz zagnieżdżony.

A. Punkt 1

a. Punkt 1.1

b. Punkt 1.2

c. Punkt 1.3

i. Punkt 1.3.1

ii. Punkt 1.3.2

B. Punkt 2

C. Punkt 3

a. Punkt 3.1

i. Punkt 3.1

1. Punkt 3.1.1

4. Utwórz dokument HTML wyświetlający wykaz w postaci spisu treści według wzoru. Zastosuj wykaz mieszany.

A. Logika

- Teoria modeli
- Teoria dowodu

B. Algebra

- Teoria liczb

Co to jest liczba pierwsza?

Jeśli poza dzielnikami trywialnymi liczba naturalna N większa od jedności nie ma innych dzielników naturalnych, to nazywamy ją liczbą pierwszą.

- Teoria ciał i wielomianów
- Teoria grup

C. Analiza

- a. Teoria funkcji rzeczywistych
- b. Teoria miary i całkowania
- c. Równania różniczkowe zwyczajne
- d. Analiza funkcjonalna

D. Geometria

- Geometria płaszczyzny i przestrzeni

E. Statystyka

6

Barwa jako wartość atrybutów HTML

ZAGADNIENIA

- Model RGB
- Model HSV
- Model CMYK
- Kodowanie barw w HTML 5 i CSS 3

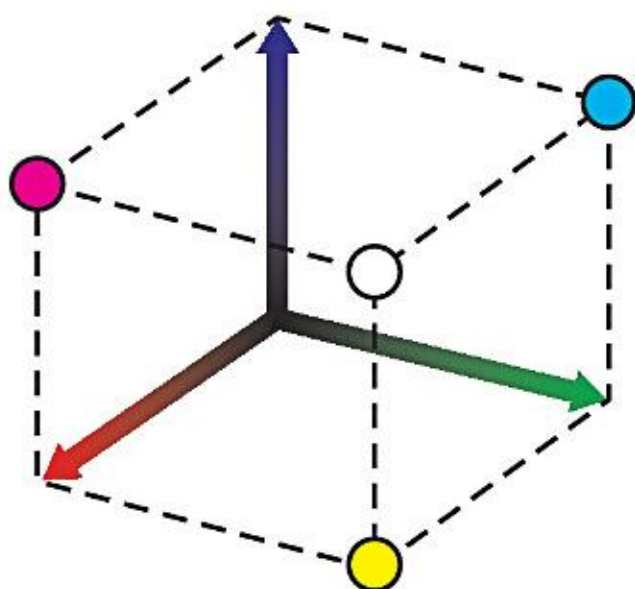
Barwy są wykorzystywane niemal na każdym etapie tworzenia dokumentu hipertekstowego oraz cyfrowego przetwarzania obrazów. Prezentacja treści strony, a co za tym idzie – barwy jej różnych elementów, definiuje się wewnątrz arkuszy stylów (CSS). Nie jest jednak zabronione używanie wartości barwy w atrybutach znaczników języka HTML. W tym rozdziale pokażemy, w jaki sposób z tego korzystać.

Barwa jest wrażeniem wzrokowym powstałym na skutek oddziaływania fal elektromagnetycznych z pasma częstotliwości światła widzialnego (między 700 nm a 400 nm).

Modele barw

Aby jednoznacznie określić daną barwę na stronach internetowych, stosuje się określone modele barw oraz metody ich zapisu. **Modele barw** służą do opisu określonych palet kolorów. Do podstawowych modeli barw stosowanych w grafice komputerowej oraz tworzenia stron internetowych zalicza się:

- **model RGB** – barwa jest tworzona przez złożenie z trzech podstawowych kolorów: R – czerwony (*red*), G – zielony (*green*) oraz B – niebieski (*blue*). Model jest stosowany podczas prezentacji na ekranie monitora bądź obrazu z projektora. Modelu RGB używa się zarówno do zapisu barwy w kodzie HTML 5, jak i CSS 3. W specyfikacji CSS 3 dopuszczono do stosowania również pokrewny model **RGBA** z dodatkowym współczynnikiem kanału alfa. Zostanie on dokładnie opisany w rozdziale poświęconym CSS 3;

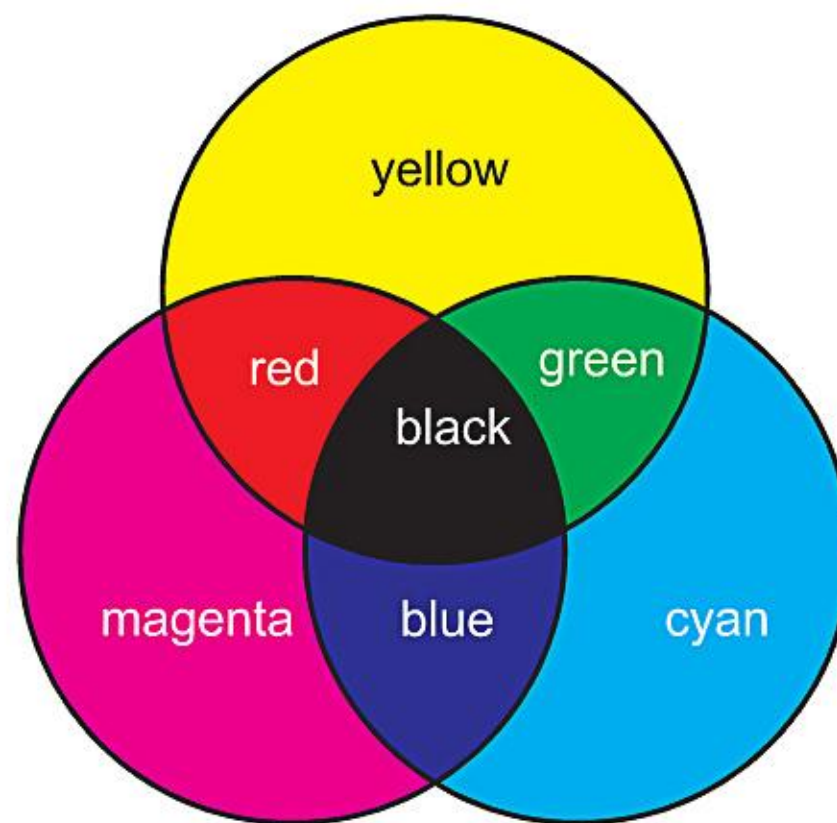


Rys. 6.1. Sześciątą w przestrzeni parametrów R, G, B

- **model HSL** – każdą barwę w tym modelu można opisać za pomocą trzech atrybutów:
 - **odcienia (H)** – powiązanego z określoną długością fali;
 - **nasycenia (S)** – określającego moc barwy;
 - **jasności (L)** – intensywności światła w danej barwie.

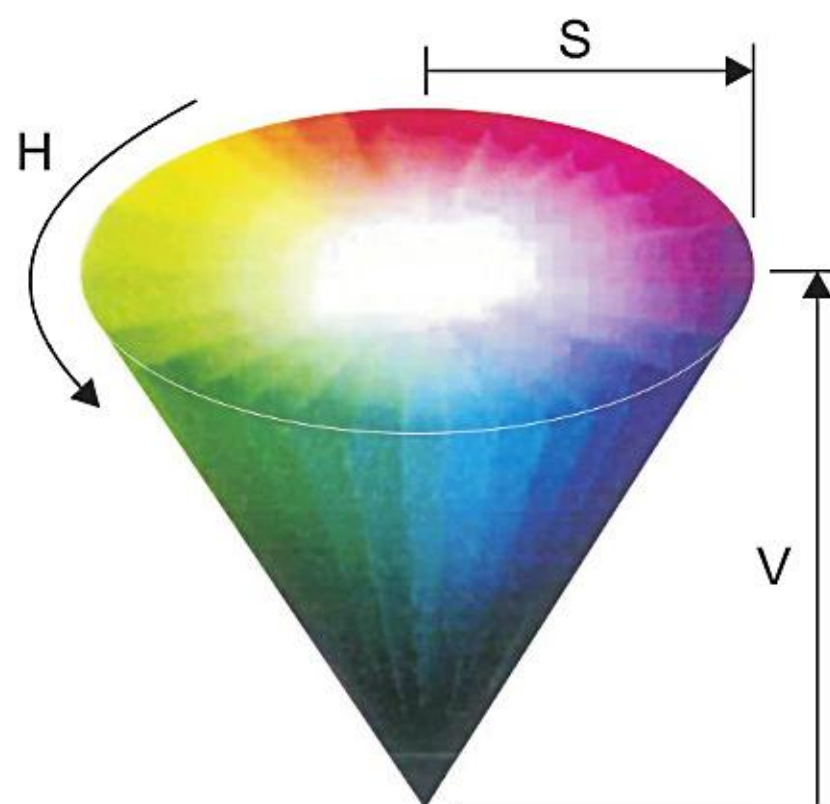
Ten model powstał na potrzeby telewizji. Używa się również modelu **HSLA**, który jest rozwinięciem omawianego wcześniej modelu barw HSL (uzupełniony o obsługę przezroczystości). Oba modele są akceptowane przez arkusze CSS 3;

- **model CMYK** – barwa jest tworzona z czterech kolorów: C – cyjan, M – magenta, Y – żółty (*yellow*), oraz K – czarny (*black*). Model stosowany do przygotowania wydruków;



Rys. 6.2. Model CMYK

- **model HSV** – oparty na trzech parametrach: H – odcień (*Hue*), S – nasycenie (*Saturation*), V – wartość (*Value*). Model jest rozpatrywany jako stożek, którego podstawą jest koło barw.



Rys. 6.3. Stożek przestrzeni barw HSV

Kodowanie barw w HTML 5 i CSS 3

Zapis kodów kolorów w dokumentach HTML 5 i CSS 3 przeprowadza się za pomocą określonych notacji. W języku HTML stosuje się zapisy nazw barw oraz wartości zapisane kodem szesnastkowym lub kodowaniem palety kolorów RGB. W arkuszach stylów CSS 3 dodatkowo dopuszcza się stosowanie zapisów w notacji palety RGBA, HSL oraz HSLA.

PRZYKŁAD 6.1

W języku HTML dopuszcza się zapis barwy w następujący sposób:

1. Zapis słowny

```
<body bgcolor="red"> treść strony </body>
```

2. Zapis barwy w kodzie szesnastkowym

```
<body bgcolor="#FF0000"> treść strony </body>
```

3. Zapis w kodzie RGB

```
<body bgcolor="rgb(255,0,0)"> treść strony </body>
```

4. Zapis w kodzie RGBA

```
<div style="background-color:rgba(0,255,0,0.1)"> treść bloku</div>
```

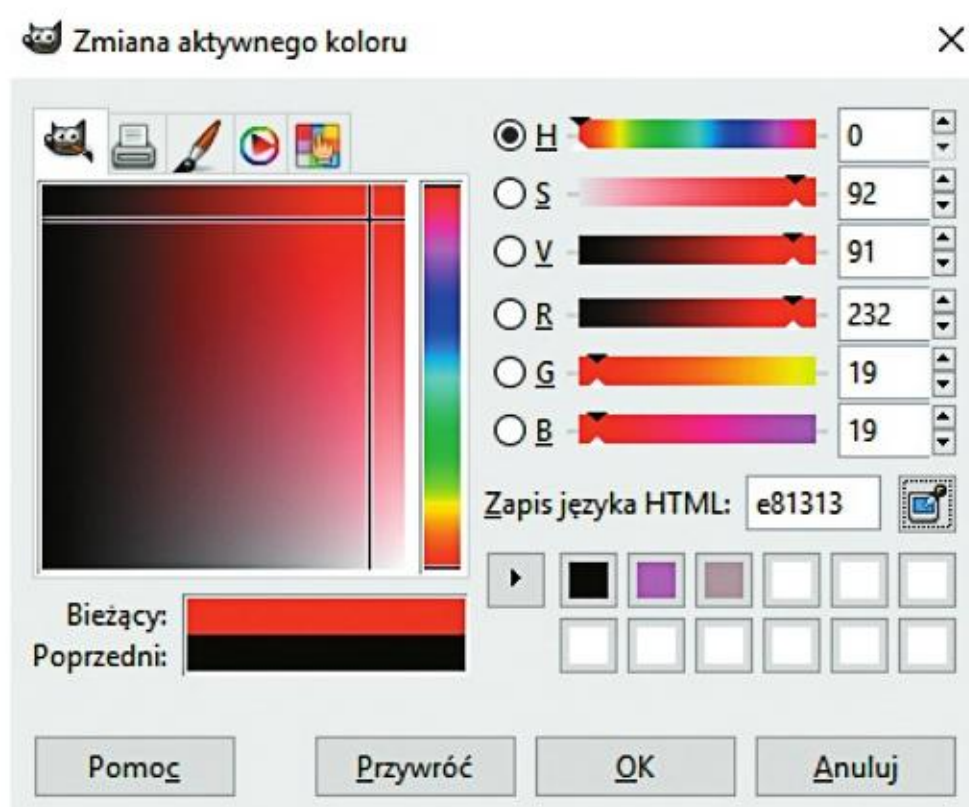
5. Zapis w kodzie HSL

```
<div style="background-color:hsl(0,100%, 50%)">treść bloku</div>
```

! UWAGA

Zapis w notacji szesnastkowej można czasami skrócić – podać np. wartość #0F0 zamiast wartości #00FF00.

Do sprawdzenia wartości kodowej określonej barwy można użyć dowolnego edytora graficznego. Na rys. 6.4 przedstawiono okno edytora koloru programu GIMP, w którym można odczytać kod wybranej barwy.



Rys. 6.4. Okno edycji barw w edytorze GIMP

Specyfikacja HTML 4.01 zawiera zestawienie 16 barw według nazw i kodu szesnastkowego. Na rys. 6.5 przedstawiono tabelę zawierającą możliwe notacje tych barw.

Kolor	Nazwa po angielsku	Zapis szesnastkowy	Kolor	Nazwa po angielsku	Zapis szesnastkowy
cyjan	aqua	#00FFFF	granatowy	navy	#000080
czarny	black	#000000	oliwkowy	olive	#808000
niebieski	blue	#0000FF	fioletowy	purple	#800080
fukcja	fuchsia	#FF00FF	czerwony	red	#FF0000
zielony	green	#008000	srebrny	silver	#C0C0C0
szary	grey	#808080	morski	teal	#008080
limonka	lime	#00FF00	biały	white	#FFFFFF
wiśniowy	maroon	#800000	żółty	yellow	#FFFF00

Rys. 6.5. Barwy w HTML 4.01

W specyfikacji HTML 5 dostępność barw jest o wiele większa. Na stronie organizacji W3C (https://www.w3schools.com/html/html_colors.asp) jest dostępny mikser barw RGB. Każdy może tam utworzyć własny kod, który będzie najlepiej pasował do danego elementu strony internetowej.

W dokumencie HTML definiuje się barwę – wpisuje jej nazwę lub kod jako wartość określonego atrybutu znacznika.

PRZYKŁAD 6.2

Run »

Result Size: 410 x 327

```

<!DOCTYPE html>
<html>
<body bgcolor="silver" text="#3300FF"
link="black" vlink="blue">
Przykład wykorzystania barwy jako wartości
atrybutów znaczników:<br />
<font color="red">czerwona czcionka</font><br
/>
tekst w kolorze zdefiniowanym w znaczniku body.
</body>

</html>

```

Przykład wykorzystania barwy jako wartości atrybutów znaczników:

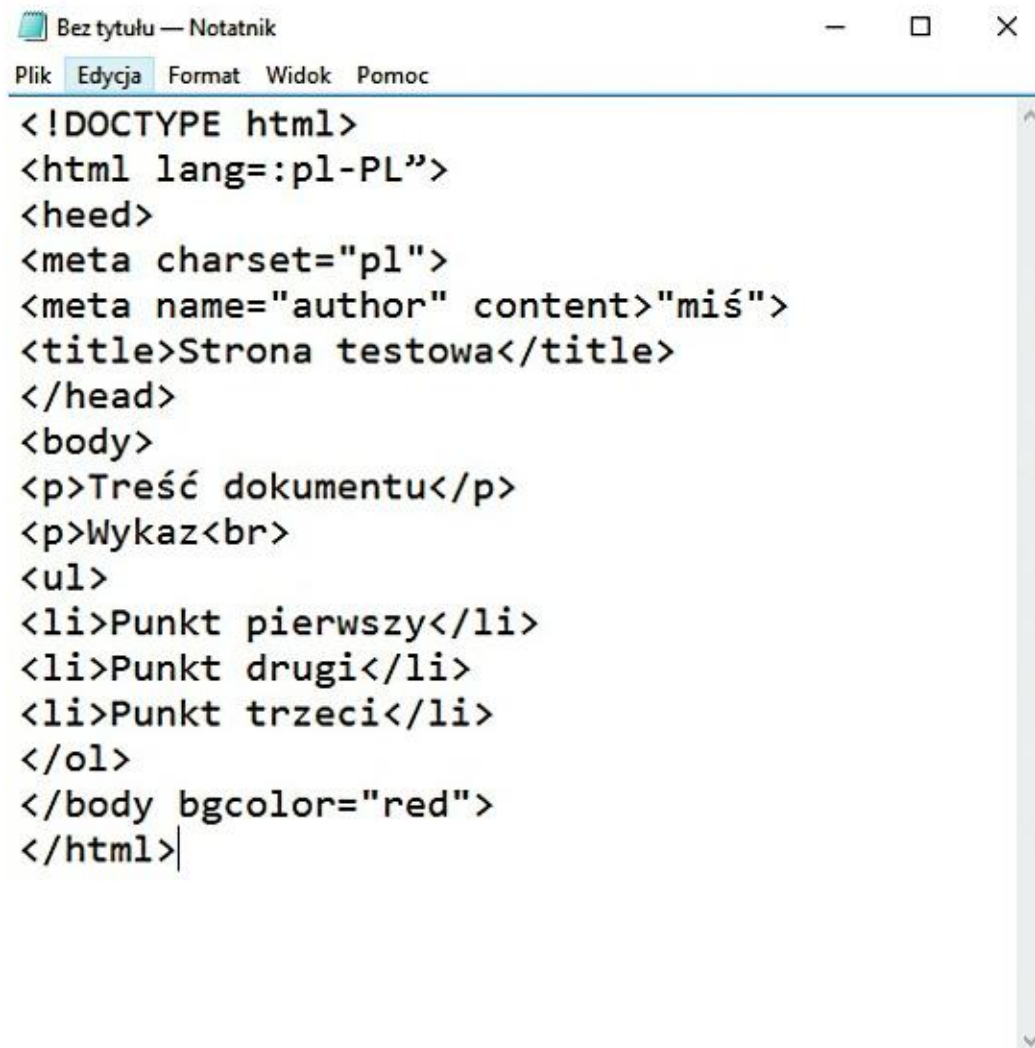
czerwona czcionka

tekst w kolorze zdefiniowanym w znaczniku body.

Rys. 6.6. Przykład zastosowania kodowania barw w dokumencie HTML 4.01

SPRAWDŹ SWOJĄ WIEDZĘ

1. Przeanalizuj załączony kod i wskaż ewentualne błędy w składni języka oraz poprawności użycia znaczników.



```

Bez tytułu — Notatnik
Plik Edycja Format Widok Pomoc
<!DOCTYPE html>
<html lang=:pl-PL">
<heed>
<meta charset="pl">
<meta name="author" content="miś">
<title>Strona testowa</title>
</head>
<body>
<p>Treść dokumentu</p>
<p>Wykaz<br>
<ul>
<li>Punkt pierwszy</li>
<li>Punkt drugi</li>
<li>Punkt trzeci</li>
</ol>
</body bgcolor="red">
</html>

```

2. Poszukaj w dostępnych źródłach (np. <https://www.w3schools.com>) informacji na temat poprawności stosowania atrybutów color, bgcolor, align w składni języka HTML 5. W zeszycie przedmiotowym przygotuj zestawienie najważniejszych danych na ich temat.
3. W zeszycie przedmiotowym narysuj i wypełnij tabelę. Podaj nazwę modelu barw zapisanego w kodzie HTML.

Zapis barwy w kodzie	Nazwa modelu barw
(0,100%, 50%)	
#FF00FF	
(255,0,0)	
(0,255,0,0.1)	

4. Czym charakteryzuje się odsyłacz względny?
5. Co to są znaczniki zdeprecjonowane?

7

Podział strony i obsługa danych w języku HTML

ZAGADNIENIA

- Podział strony na bloki i sekcje
- Tabele w dokumentach hipertekstowych
- Obsługa danych za pomocą formularzy

Istotnym etapem tworzenia strony internetowej jest zbudowanie jej struktury z wydzielaniem poszczególnych bloków, kontenerów lub części spełniających określoną funkcję. W obecnych czasach, kiedy dużą część witryn projektuje się na podstawie filozofii zarządzania **CMS** (ang. *Content Management System*), stworzenie jednolitej struktury, z której korzystają wszystkie strony danej witryny, jest kluczowe dla sukcesu całego projektu.

W starszych standardach HTML do podziału strony używano najczęściej ramek. Z czasem stały się one przestarzałe, a obecnie znaczniki tworzenia ramek znalazły się w grupie elementów zdeprecjonowanych. Do podziału strony na części tematyczne stosuje się znacznik **<div>** (ang. *division* – ‘oddział, sekcja’). W HTML 5.0 znacznik **<div>** zastąpiono znacznikami: **<header>**, **<article>**, **<aside>**, **<nav>**, **<section>**, **<footer>**.

Bloki

W HTML 4 do podziału strony na części funkcjonalne stosowano elementy zwane blokami, pudełkami lub kontenerami. Elementy wpisane między znacznikami **<div>** i **</div>** łatwo można wypełnić treścią – załadować określoną podstronę oraz zastosować wewnętrzne lub zewnętrzne formatowanie za pomocą stylów (tworzy się wtedy tzw. kontenery CSS).

Elementy **<div>** są elementami blokowymi, tak jak np. znacznik **<p>**. Zajmują one całą szerokość strony. Ponadto są to elementy, które domyślnie nie są sformatowane, tj. nie są nadane żadne obramowanie, kolor wypełnienia czy szerokość.

Aby określić format znacznika **<div>**, można skorzystać z jego następujących atrybutów:

Atrybuty	Opis / znacznik
class	klasa CSS
id	identyfikator elementu
style	styl elementu
title	tekst do wyświetlenia po najechnięciu myszką

Atrybuty	Opis / znacznik
align	wyrównanie zawartości elementów blokowych w poziomie; nie jest wspierany przez standard HTML 5
dir	określa kierunek tekstu w bloku; wartość ltr oznacza kierunek od lewej krawędzi strony do prawej, a wartość rtl – odwrotnie

PRZYKŁAD 7.1

Zastosowanie znacznika **<div>** bez atrybutów nie przynosi żadnych spektakularnych efektów. Jego działanie jest wówczas zbliżone do tego po użyciu znacznika **<p>**.



Rys. 7.1. Zastosowanie znacznika **<div>** bez atrybutów

Za pomocą atrybutu **style** określa się wygląd poszczególnych bloków. Każdy z nich może mieć inny kolor tła (właściwość **background-color**), szerokość (właściwość **width**), wysokość (właściwość **height**) bądź inną dowolną cechę.

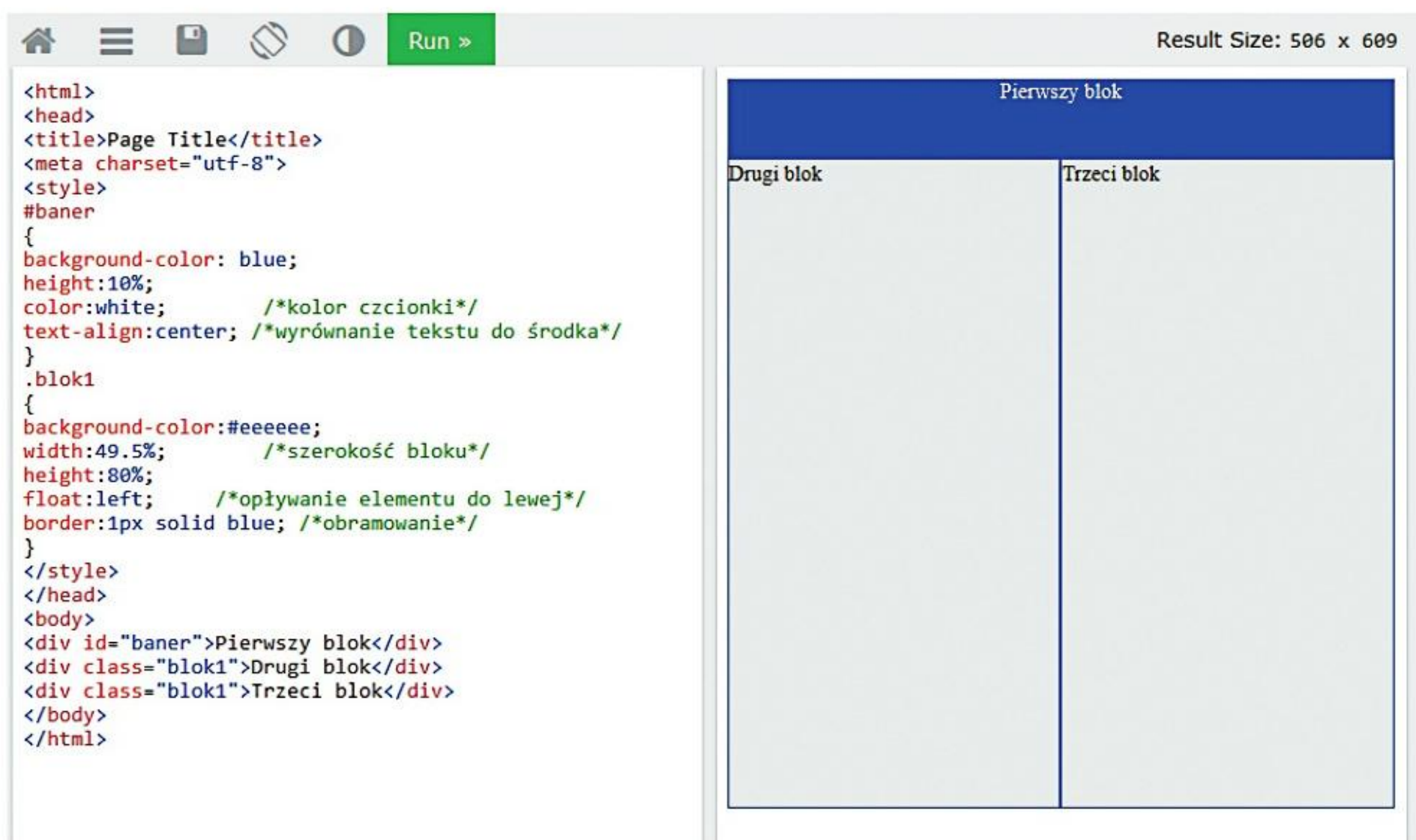


Rys. 7.2. Przykład formatowania bloków za pomocą atrybutu **style**

Nie jest to jednak najlepszy sposób formatowania bloku. Kod źródła strony staje się mało czytelny. Lepszym sposobem jest stosowanie atrybutów **id** oraz **class**, czyli wyodrębnianie poszczególnych bloków przez nadanie im identyfikatora lub klasy.

PRZYKŁAD 7.2

Do tworzenia layoutu współczesnej witryny internetowej stosuje się tzw. znaczniki strukturalne, takie jak **<div>**. Domyślnie element blokowy układa się pod poprzedzającym go elementem. Jeżeli chcielibyśmy umieścić dwa bloki obok siebie, nie wystarczy określić ich szerokości na 50%. Mimo że zmieściłyby się obok, to i tak ułożą się jeden pod drugim. Możemy jednak spowodować, że bloki będą układać się jeden obok drugiego. W blokach, które chcemy umieścić obok siebie, należy zastosować metodę **float:left**.



Rys. 7.3. Przykład zastosowania metody **float:left**

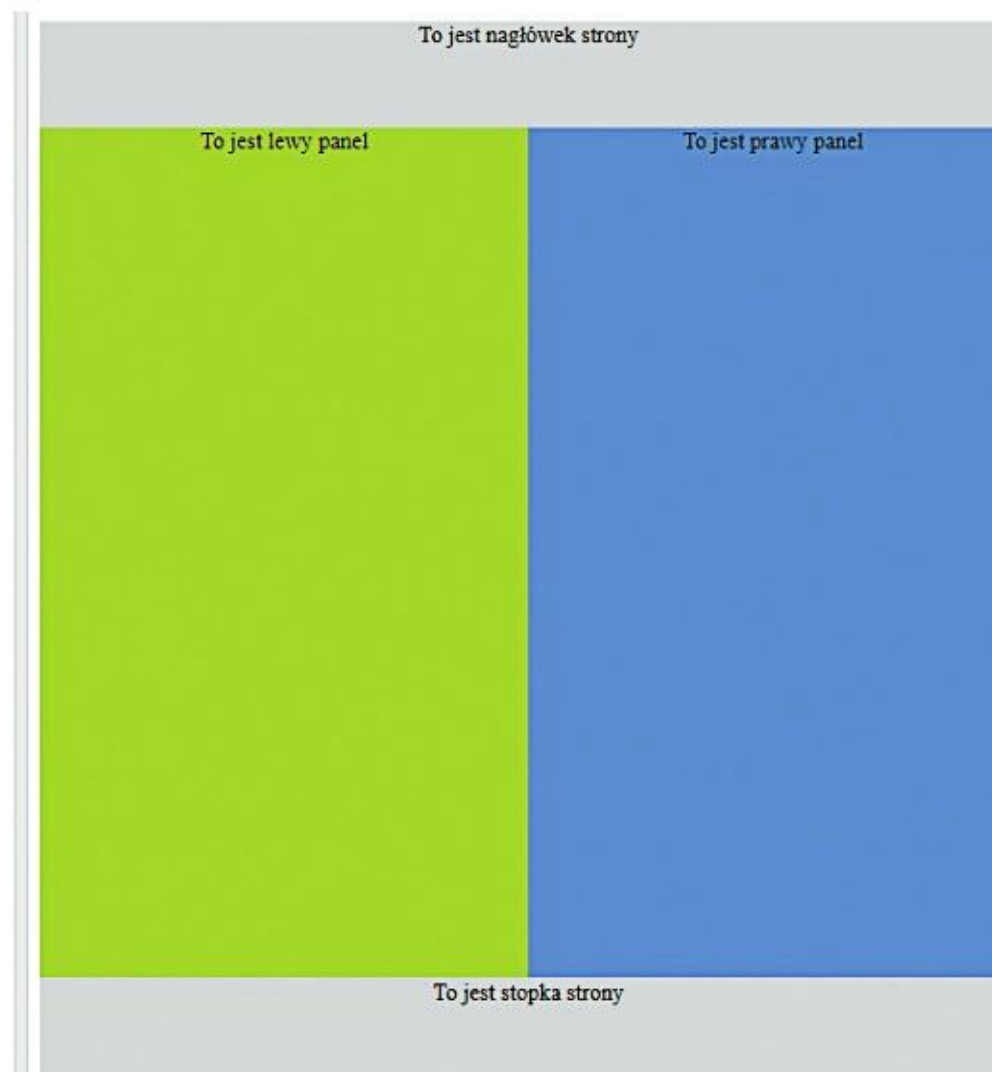
W przykładzie 7.2 zastosowano identyfikator i klasę. Różnica między nimi polega na tym, że atrybut **id** o takiej samej wartości (w tym wypadku – nazwie) można wykorzystać tylko raz w obrębie jednej strony. Natomiast atrybut **class** o danej nazwie można zastosować wielokrotnie w obrębie jednej strony internetowej.

PRZYKŁAD 7.3

```

<html>
<head>
<title>Page Title</title>
<meta charset="utf-8">
<style>
header
{
background-color:#ddddd;
height:10%;
text-align:center;
}
section
{
background-color:#ccff33;
height:80%;
width:50%;
text-align:center;
float:left;
}
aside
{
background-color:#6699ff;
height:80%;
width:50%;
text-align:center;
float:left;
}
footer
{
background-color:#ddddd;
height:10%;
text-align:center;
clear:both;
}
</style>
</head>
<body>
<header>To jest nagłówek strony</header>
<section>To jest lewy panel</section>
<aside>To jest prawy panel</aside>
<footer>To jest stopka strony</footer>
</body>
</html>

```



Rys. 7.4. Przykład podziału strony za pomocą `<div>` z zastosowaniem metody `clear:both`

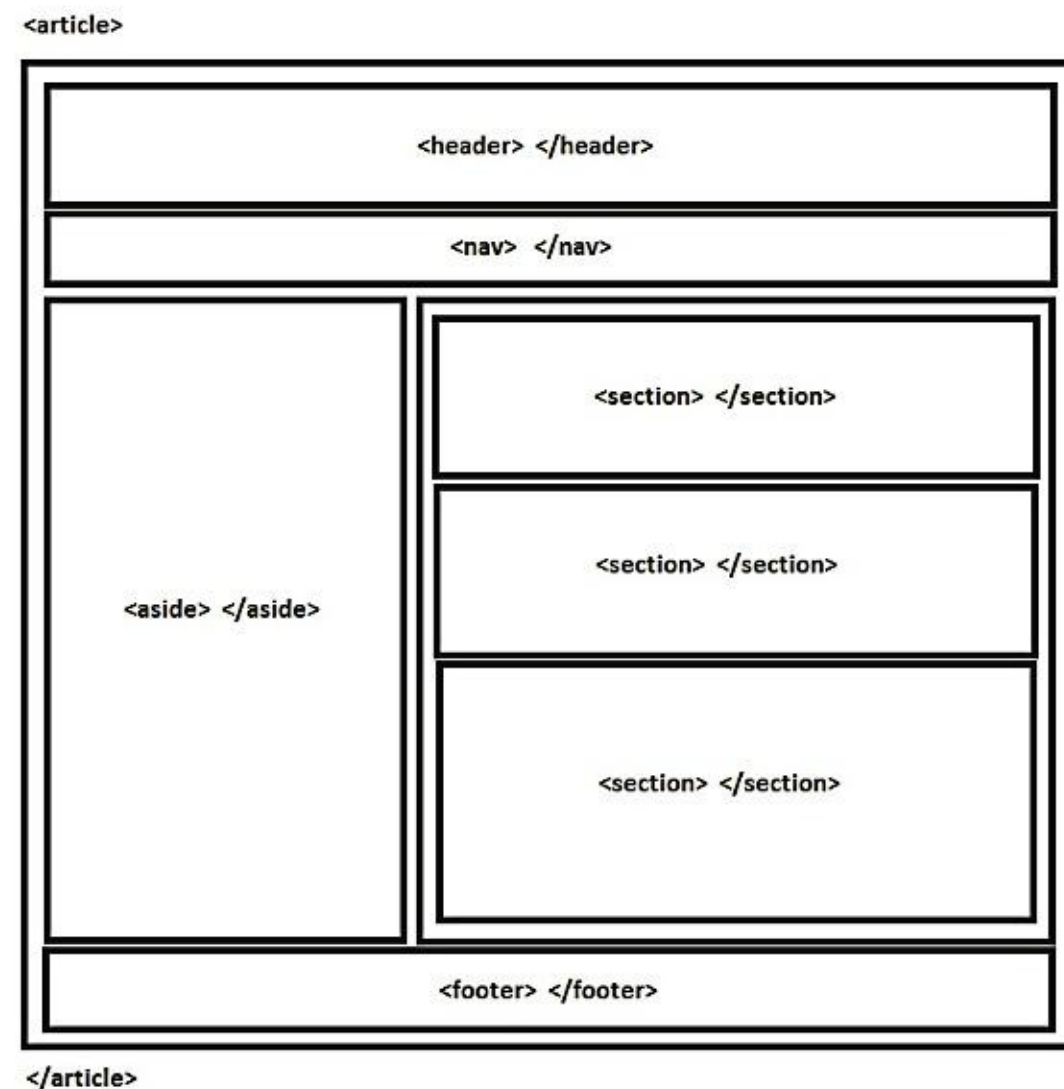
Wszystkie bloki, które mają być ułożone obok siebie, będą miały w swoich stylach CSS atrybut `float:left`. Blok, który już ma się nie układać obok, musi mieć zapis – tzw. blokowe opływanie: `clear:both`.

Sekcje

Wraz z opublikowaniem specyfikacji HTML 5 wprowadzono nowe elementy blokowe, które mają zastąpić opisywany wcześniej znacznik `<div>` i sprecyzować rolę określonych fragmentów dokumentu hipertekstowego. Tego typu elementy języka HTML nazywa się znacznikami semantycznymi. Są one jednoznacznie interpretowane przez przeglądarkę i nie wymagają użycia dodatkowych atrybutów definiujących ich rolę.

Do najczęściej wykorzystywanych znaczników sekcji zalicza się: `<article>`, `<header>`, `<nav>`, `<section>`, `<aside>` oraz `<footer>`. Na rys. 7.5 zilustrowano funkcję poszczególnych znaczników sekcji w strukturze strony.

Opis poszczególnych znaczników sekcji możliwych do wykorzystania w standardzie HTML 5 znajduje się w tabeli 7.1.



Rys. 7.5. Rola poszczególnych znaczników sekcji

Tabela 7.1. Opis znaczników sekcji

Nazwa znacznika	Opis znacznika
<header>	znacznik nagłówka; może być wielokrotnie stosowany na stronie; może być zarówno nagłówkiem strony, jak i nagłówkiem określonej części strony
<section>	znacznik sekcji; część strony, która zazwyczaj zawiera grupę innych znaczników
<footer>	znacznik stopki; może być stosowany wielokrotnie na stronie
<article>	znacznik artykułu
<aside>	znacznik elementu pobocznego; najczęściej umieszcza się w nim coś dodatkowego, np. dodatkowe menu
<main>	znacznik głównej części strony; umieszcza się w nim najważniejszą treść na stronie
<nav>	znacznik nawigacji; umieszcza się w nim tylko główne menu strony
<details>	definiuje interaktywne sekcje widoczne, np. po naciśnięciu na daną pozycję menu
<summary>	definiuje widoczny element dla sekcji <details>
<time>	definiuje blok daty i czasu
<figure>	znacznik sekcji dla obiektów graficznych
<mark>	zaznacza lub uwydatnia tekst w jednym dokumencie, by odwołać się do niego w innym kontekście

PRZYKŁAD 7.4

Definiowanie stopki za pomocą znacznika `<div>`:

```
<div id="stopka"> Stopka strony </div>
```

Definiowanie stopki za pomocą znacznika `<footer>`:

```
<footer> Stopka strony </footer>
```

PRZYKŁAD 7.5

Na rys. 7.6 przedstawiono przykład zastosowania znacznika `<nav>`.



Rys. 7.6. Zastosowanie znacznika `<nav>`

PRZYKŁAD 7.6

Identyczny układ jak w przykładzie 7.3 można uzyskać za pomocą znaczników sekcji zgodnych z HTML 5.0.

```
<html>
```

```
<head>
```

```
<title>Podział na sekcje</title>
```

```
<meta charset="utf-8">
```

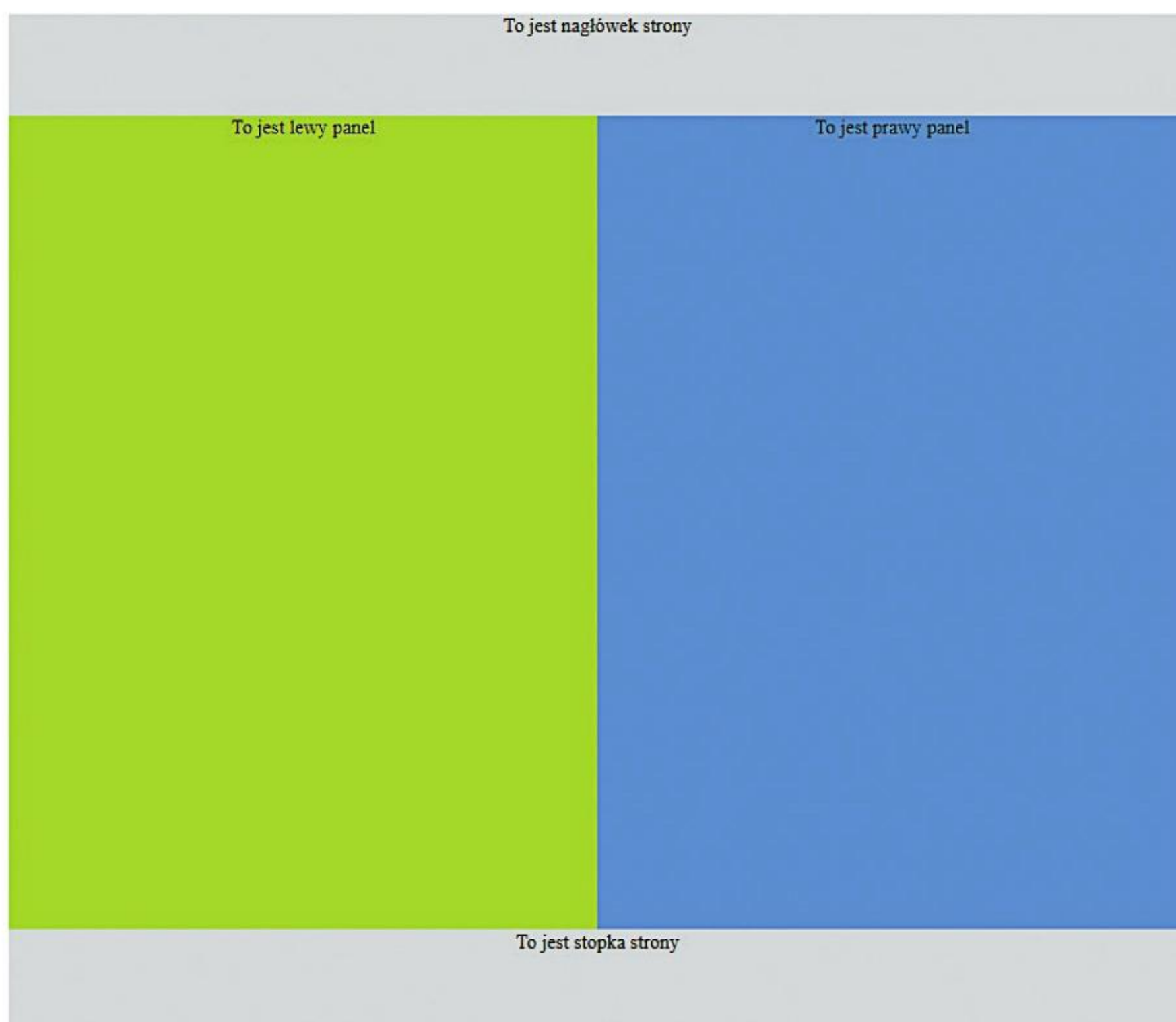
```
<style>
```

```
header {background-color:#dddddd;height:10%;text-align:center;}
```

```
section {background-color:#ccff33; height:80%; width:50%;
```



```
text-align:center; float:left;}
aside {background-color:#6699ff; height:80%; width:50%;
text-align:center;
float:left;}
footer { background-color:#dddddd; height:10%; text-align:center;
clear:both;}
</style>
</head>
<body>
<header>To jest nagłówek strony</header>
<section>To jest lewy panel</section>
<aside>To jest prawy panel</aside>
<footer>To jest stopka strony</footer>
</body>
</html>
```



Rys. 7.7. Przykład podziału strony za pomocą znaczników sekcji

Tabele

Tabele pozwalają na łatwą i czytelną prezentację danych, obrazów, tekstu i innych elementów na stronach internetowych. Służą do przedstawiania danych w postaci tabelarycznej. Informacje umieszczane na stronie stają się czytelniejsze i łatwiejsze do analizy. W tabelach można umieszczać takie elementy, jak: obrazy, teksty, linki itp.

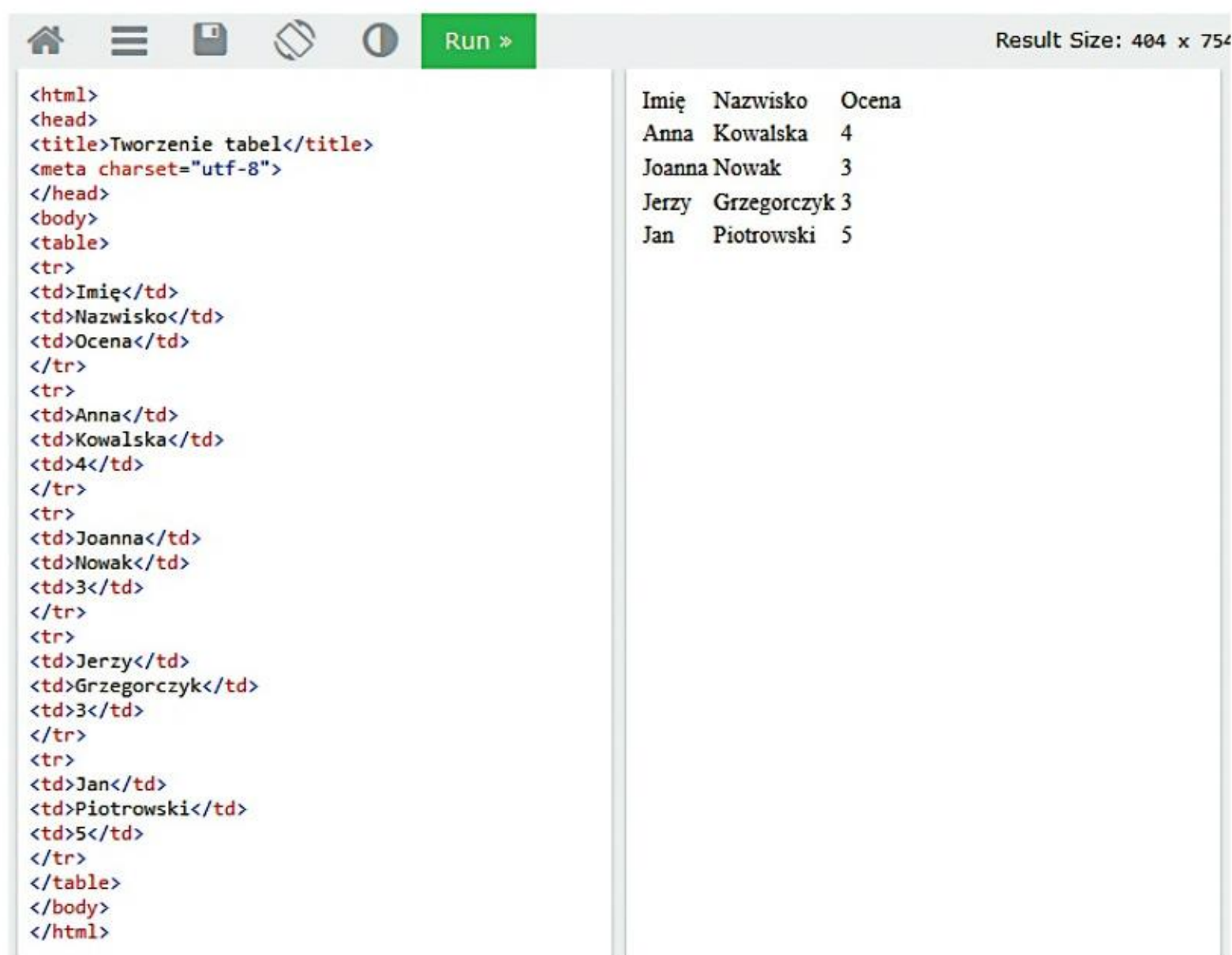
Do tworzenia tabeli wykorzystuje się znaczniki opisane w tabeli 7.2.

Tabela 7.2. Znaczniki tworzenia tabel

Znaczniki	Opis
<code><table></code> <code></table></code>	znaczniki określające obszar tabeli
<code><tr></code> <code></tr></code>	znaczniki wiersza tabeli
<code><td></code> <code></td></code>	znaczniki komórki tabeli
<code><th></code> <code></th></code>	komórka nagłówkowa w tabeli (zawiera się wewnątrz wierszy)
<code><caption></code> <code></caption></code>	tytuł tabeli

Każdą tabelę należy rozpocząć od znacznika `<table>`, a następnie wstawić w niej poszczególne wiersze, a w każdym z nich – komórki.

PRZYKŁAD 7.7



Rys. 7.8. Tworzenie tabeli w HTML

Zastosowanie znaczników tworzenia tabeli nie zapewniło wstawienia obramowania tej tabeli. Aby dodać obramowanie tabeli bądź też określić jej inne style formatowania, należy zastosować odpowiednie atrybuty.

Najważniejsze atrybuty znaczników tworzących tabele opisano w tabeli 7.3.

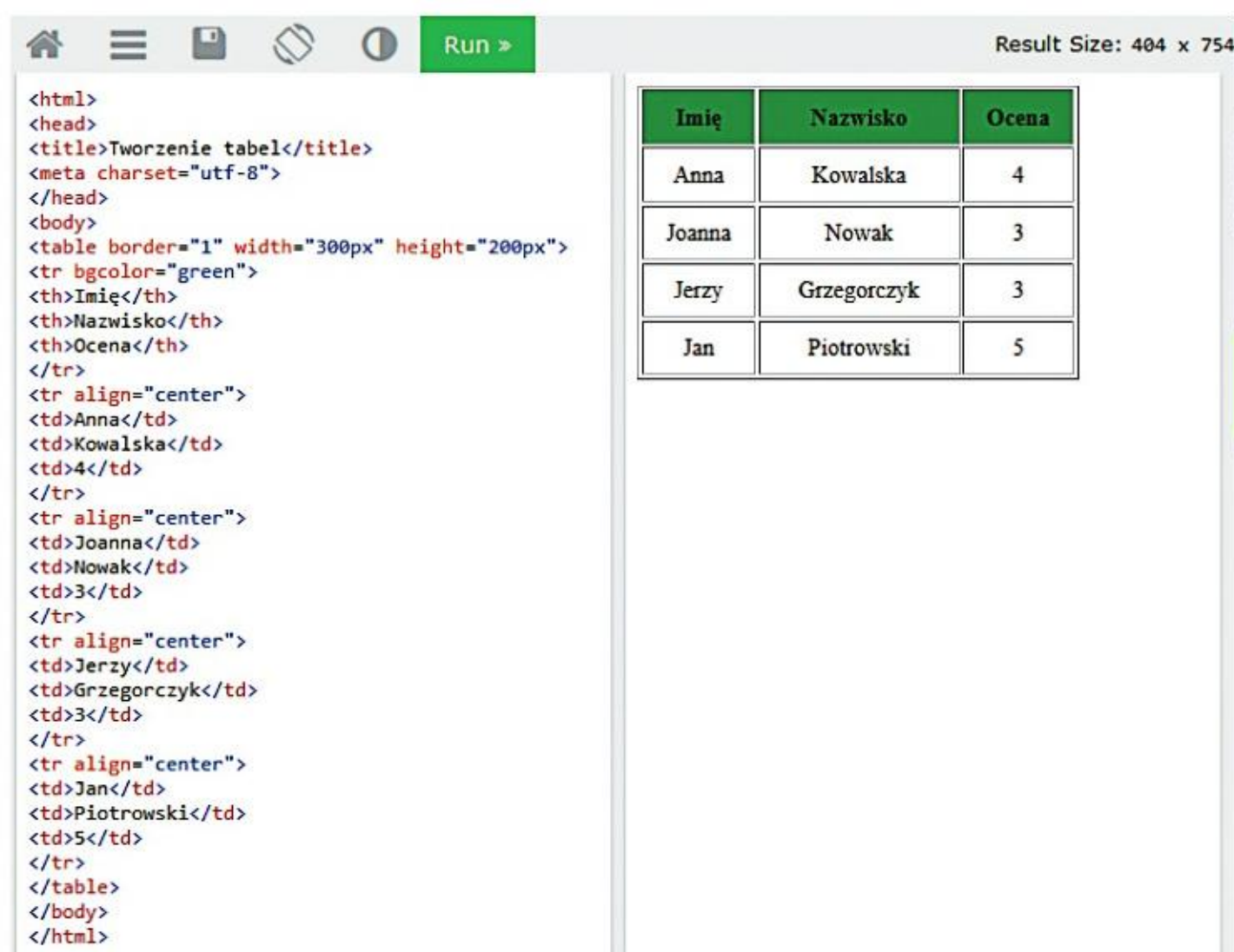
Tabela 7.3. Atrybuty znaczników do tworzenia tabel

Atrybuty	Opis / znacznik
colspan	scalenie kolumn
rowspan	scalenie wierszy
cellpadding	marginesy komórek
cellspacing	rozmiar odstępów między komórkami
border	rozmiar obramowania <table>, <tr>, <td>
width	szerokość elementu <table>, <tr>, <td>
height	wysokość elementu tabeli <table>, <tr>, <td>
valign	wyrównanie tekstu w pionie w komórkach w wierszu
align	wyrównanie tekstu w poziomie w komórkach w wierszu
bgcolor	kolor wypełnienia

Wygląd tabeli można określać także za pomocą stylów wewnętrznych lub zewnętrznych, co zostanie opisane w następnych rozdziałach.

PRZYKŁAD 7.8

Na rys. 7.9 przedstawiono przykład tabeli prostej złożonej z 3 kolumn i 5 wierszy, z których jeden jest wierszem nagłówkowym. W wierszu nagłówkowym elementy są domyślnie wyśrodkowane, a tekst jest pisany pogrubioną czcionką. Ponadto w tym wierszu określono zielony kolor tła. Całą tabelę obramowano, jej szerokość ustawiono na 300 pikseli, a wysokość – na 200 pikseli. Każdy kolejny wiersz ma ustawione wyśrodkowanie tekstu.



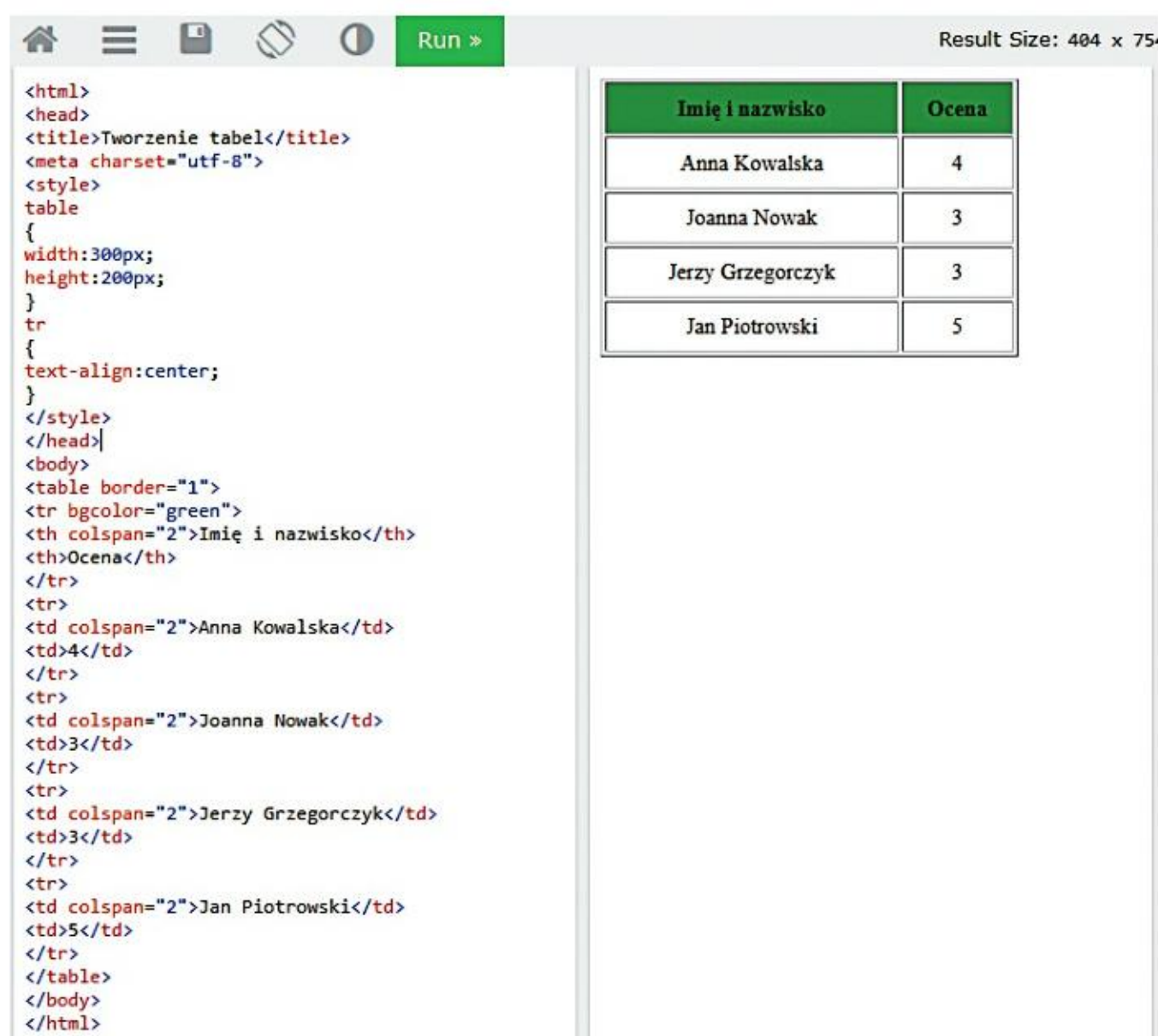
Rys. 7.9. Przykład formatowania tabeli w HTML

Oczywiście, przedstawiony sposób formatowania tabeli nie jest zgodny z dzisiejszymi standardami tworzenia stron internetowych. Należy pamiętać, że wygląd poszczególnych elementów strony należy formatować za pomocą stylów CSS.

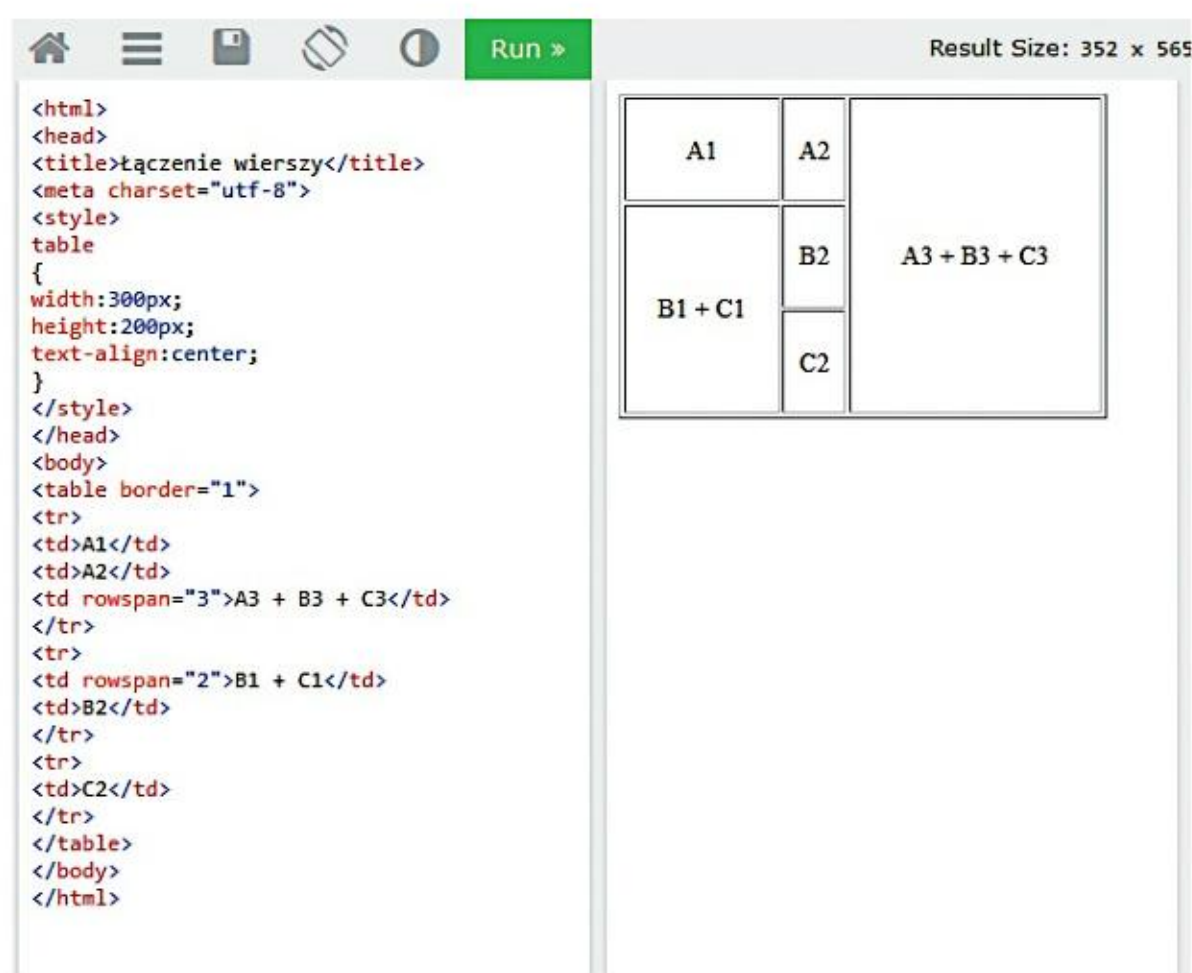
Atrybuty **colspan** i **rowspan** są właściwościami, które stosuje się bezpośrednio w kodzie HTML. Służą one do łączenia określonych komórek w jedną większą.

PRZYKŁAD 7.9

Aby połączyć kolumnę Imię z kolumną Nazwisko w tabeli z przykładu 7.8, należy zastosować zapis **colspan="2"** w pierwszej komórce każdego wiersza. Ponadto szerokość i wysokość tabeli oraz wyśrodkowanie tekstu w każdym wierszu ustawiono w stylach wewnętrznych.

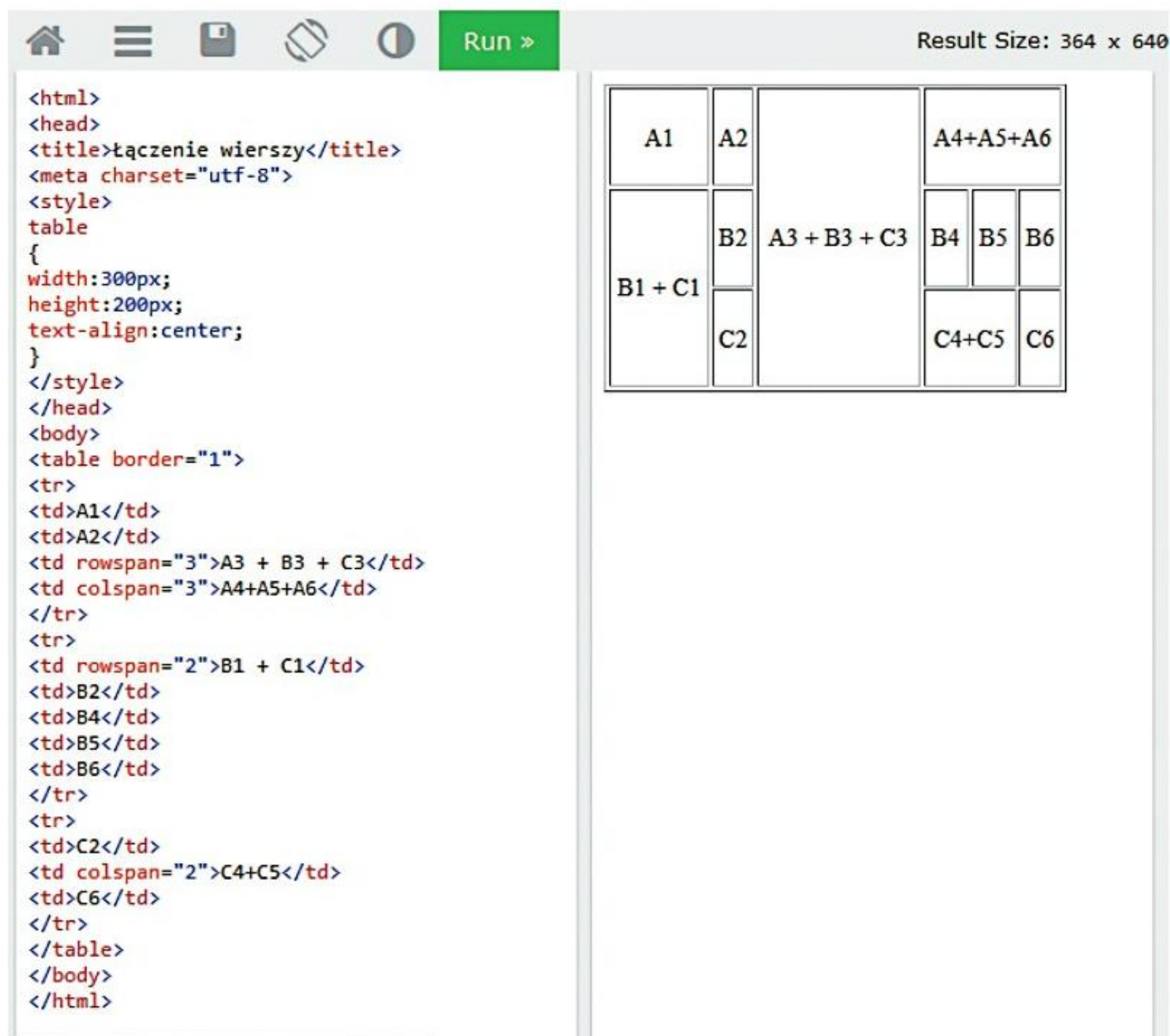
Rys. 7.10. Przykład zastosowania atrybutu **colspan****PRZYKŁAD 7.10**

Oprócz łączenia komórek w poziomie, można połączyć kilka komórek w pionie za pomocą atrybutu **rowspan**.

Rys. 7.11. Przykład zastosowania atrybutu **rowspan**

Atrybuty **colspan** i **rowspan** można stosować jednocześnie i wielokrotnie w obrębie jednej tabeli.

PRZYKŁAD 7.11



Rys. 7.12. Przykład wielokrotnego zastosowania atrybutu **colspan** i **rowspan**

Możliwe jest także zagnieżdżanie tabel w komórkach nadrzędnej tabeli. W tym celu należy zdefiniować tabelę w wybranej komórce tabeli nadrzędnej.

PRZYKŁAD 7.12

```
<html>
<head>
<title>Kalendarz</title>
<meta charset="utf-8">
<style>
table
{
text-align:center;
}
```



```

img
{
width:150px;
height:150px;
}
</style>
</head>
<body>
<table border="1">
<tr>
<td><font color="green"><b>LIPIEC</b></font><br>
<table bgcolor="#eeeeee">
<tr>
<td>Pn</td><td>Wt</td><td>Śr</td><td>Czw</td><td>Pt</
td><td>Sb</td>
<td><font color="red">Nd</font></td>
</tr>
<tr>
<td></td><td></td><td>1</td><td>2</td><td>3</td><td>4</td><t-
d><font color="red">5</font></td>
</tr>
<tr>
<td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</
td><td><font color="red">12</font></td>
</tr>
<tr>
<td>13</td><td>14</td><td>15</td><td>16</td><td>17</
td><td>18</td><td>
<font color="red">19</font></td>
</tr>
<tr>
<td>20</td><td>21</td><td>22</td><td>23</td><td>24</
td><td>25</td><td>
<font color="red">26</font></td>
</tr>
<tr>
<td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td></
td><td></td>
</tr>
</table>

```



```

<td><font color="green"><b>SIERPIEŃ</b></font><br>
<table bgcolor="#FFFF99">
<tr>
<td>Pn</td><td>Wt</td><td>Śr</td><td>Czw</td><td>Pt</
td><td>Sb</td><td>
<font color="red">Nd</font></td>
</tr>
<tr>
<td></td><td></td><td></td><td></td><td></td><td>1</td><td>
<font color="red">2</font></td>
</tr>
<tr>
<td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</
td><td>
<font color="red">9</font></td>
</tr>
<tr>
<td>10</td><td>11</td><td>12</td><td>13</td><td>14</
td><td>15</td><td>
<font color="red">16</font></td>
</tr>
<tr>
<td>17</td><td>18</td><td>19</td><td>20</td><td>21</
td><td>22</td><td>
<font color="red">23</font></td>
</tr>
<tr>
<td>24</td><td>25</td><td>26</td><td>27</td><td>28</
td><td>29</td><td>
<font color="red"></font></td>
</tr>
</table>
</tr>
</table>
</body>
</html>

```


LIPIEC							SIERPIEŃ						
													
Pn	Wt	Śr	Czw	Pt	Sb	Nd	Pn	Wt	Śr	Czw	Pt	Sb	Nd
		1	2	3	4	5						1	2
6	7	8	9	10	11	12	3	4	5	6	7	8	9
13	14	15	16	17	18	19	10	11	12	13	14	15	16
20	21	22	23	24	25	26	17	18	19	20	21	22	23
27	28	29	30	31			24	25	26	27	28	29	

Rys. 7.13. Przykład zagnieżdżania tabel

Formularze

Jednym z elementów interaktywnych w dokumentach HTML jest **formularz**. Na stronach internetowych służy on do zbierania i wysyłania określonych danych w celu założonym przez autora. Wykorzystuje się go również do zamawiania różnych towarów w sklepach internetowych, ale wtedy trzeba zadbać o bezpieczeństwo przesyłanych danych.

Wprowadzone dane można przesłać na adres poczty mailowej lub wykorzystać w uruchomionym skrypcie zarówno po stronie użytkownika (klienta), jak i zdalnego serwera.

Każdy formularz umieszcza się między znacznikami **<form>** i **</form>**. Podstawowe znaczniki wprowadzające dane przedstawiono w tabeli 7.4.

Tabela 7.4. Znaczniki tworzenia formularza

Znaczniki	Opis
<input>	tworzy pola formularza
<select>	tworzy listy rozwijane
<textarea>	tworzy pola opisowe
<fieldset>	obramowanie pól formularza
<legend>	opis pola formularza
<label>	tworzy etykietę pola dla znacznika <input>

Najczęściej wykorzystywane atrybuty znaczników do tworzenia formularzy umieszczono w tabeli 7.5.

Tabela 7.5. Lista atrybutów znaczników do tworzenia formularzy

Atrybuty	Opis
<code>type="text"</code>	pole tekstowe
<code>type="password"</code>	pole na hasło (*****)
<code>type="radio"</code>	pole wyboru jednej opcji
<code>type="checkbox"</code>	pole wyboru wielu opcji
<code>type="submit"</code>	pole tworzące przycisk do wysyłania danych
<code>type="reset"</code>	pole tworzące przycisk czyszczący pola formularza
<code>type="file"</code>	pole wyboru pliku do przesłania
<code>type="image"</code>	pole tworzące przycisk pod postacią zewnętrznego rysunku z podaniem źródła <code>src=„ścieżka dostępu do pliku”</code>
<code>name="nazwa"</code>	nazwa pola
<code>value="wartość"</code>	wartość początkowa pola
<code>action="adres URL"</code>	miejsce docelowe przetwarzania danych, np. adres e-mail lub skrypt
<code>method="get/post"</code>	metoda przesyłania danych: get (domyślna) lub post
<code>maxlength="x"</code>	służy do ograniczania długości tekstu wprowadzonego w pole tekstowe
<code>enctype="typ kodowania"</code>	sposób kodowania informacji przed przesłaniem
<code>rows="x"</code>	wysokość liczona liczbą wierszy w znaczniku <code><textarea></code>
<code>cols="x"</code>	szerokość liczona liczbą kolumn w znaczniku <code><textarea></code>
Atrybuty wprowadzone w HTML 5.0	
<code>type="email"</code>	pole służy do wprowadzania adresu e-mail, wprowadzany tekst musi zawierać znak @, nie może się kończyć ani zaczynać kropką
<code>type="url"</code>	pole służy do wprowadzania adresu URL, adres musi zaczynać się protokołem (np. <code>http://</code> , <code>https://</code> , czy <code>ftp://</code>)

Atrybuty	Opis
<code>type="number"</code>	pole służy do wprowadzania liczb; zakres liczb można określić za pomocą atrybutów <code>max.</code> i <code>min.</code> , ponadto atrybut <code>step</code> pozwala określić zwiększenia wartości liczby w polu
<code>type="date"</code>	pole służy do wprowadzania daty
<code>type="time"</code>	pole służy do wprowadzania godziny
<code>type="week"</code>	pole służy do wyboru tygodnia w danym roku (wybiera się numer tygodnia i rok)
<code>type="month"</code>	pole służy do wyboru miesiąca roku (wybiera się numer miesiąca i rok)
<code>type="tel"</code>	pole służy do wprowadzania numeru telefonu, format wprowadzania numeru można określić za pomocą atrybutu <code>pattern</code>
<code>type="search"</code>	pole służy do wprowadzania szukanego wyrażenia na danej stronie; tego typu pola używa się podczas budowy wyszukiwarki
<code>type="color"</code>	pole służy do wybierania koloru
<code>type="range"</code>	pole służy do wybierania liczby z podanego zakresu
<code>required</code>	umożliwia wymuszenie – na użytkowniku – wypełnienia lub zaznaczenia pola
<code>pattern</code>	pozwała określić wzorzec, według którego użytkownik będzie musiał wypełnić dane pole; wzorzec tworzy się za pomocą tzw. wyrażeń regularnych

Nie wszystkie przeglądarki wspierają podane pola formularza. Brak wsparcia ze strony przeglądarki spowoduje, że dane pole zostanie wyświetlone jako zwykłe tekstowe

Podczas przetwarzania danych wpisanych do formularza istotne jest, w jaki sposób prześlemy je do bazy danych czy aplikacji. W formularzach korzystamy z dwóch metod: **GET** i **POST**. **GET** jest metodą domyślną, wykorzystywaną do jawnego przesyłania informacji. Dlatego wykorzystuje się ją do przesyłania mniej istotnych, krótkich pakietów danych. Metodę **POST** przyjmujemy na potrzeby przesyłania danych uwierzytelniających, osobowych, drażliwych lub całych plików.

PRZYKŁAD 7.13

Na rys. 7.14 przedstawiono przykład prostego formularza z opcją przesłania danych na adres e-mail podany w kodzie.

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<form action="mailto:wspis@wsip.pl">
Imię:
<br><input type="text" name="imie">
<br>Nazwisko:
<br><input type="text" name="nazwisko">
<br>Płeć
<br><input type="radio" name="plec">Kobieta
<input type="radio" name="plec">Mężczyzna
<br>Wiek
<br><select name="wiek">
<option>0-17</option>
<option>18-40</option>
<option>41-60</option>
</select>
<br>Uwagi:
<br><textarea rows="6" cols="17">Wpisz uwagi
</textarea>
<br><input type="submit" value="Wyślij">
</form>
</body>
</html>

```

Rys. 7.14. Formularz w HTML

Do przesłania danych – co jest widoczne w przykładzie – tworzy się przycisk za pomocą typu **submit** z etykietą **wyślij**. Metodę i miejsce docelowe przesyłania danych z formularza definiuje się atrybutami **action** i **method** w znaczniku **<form>** tworzącym blok formularza.

Formularz w przykładzie 7.13 zawiera pole listy utworzone za pomocą znacznika **<select>**. Ten znacznik ma następujące atrybuty:

- **Required;**
- **multiple="multiple"** – umożliwia wybranie kilku pozycji z listy;
- **size="wartość"** – określa, ile pozycji listy ma być wyświetlane (domyślna wartość 0).

PRZYKŁAD 7.14

```

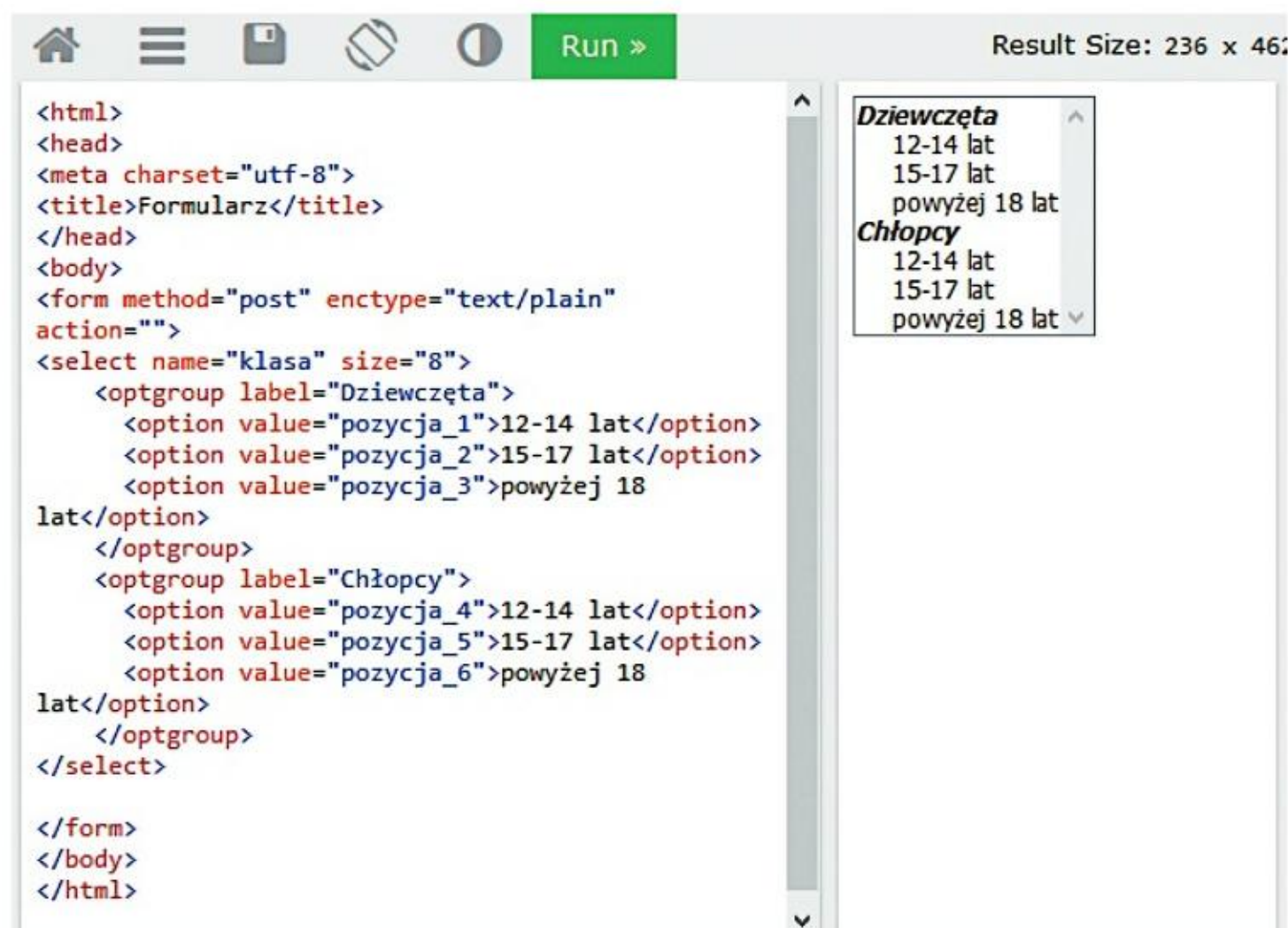
<html>
<head>
<meta charset="utf-8">
<title>Formularz</title>
</head>
<body>
<form method="post" enctype="text/plain" action="">
Wybierz język:
<select name="jezyki" multiple="multiple">
<option value="angielski">J.angielski</option>
<option value="niemiecki">J.niemiecki</option>
<option value="francuski">J.francuski</option>
<option value="hiszpanski">J.hiszpański</option>
<option value="rosyjski">J.rosyjski</option>
</select>
</form>
</body>
</html>

```

Rys. 7.15. Tworzenie listy **<select>** z możliwością dokonywania wyboru kilku elementów

Każdą listę można podzielić na grupy opcji. W tym celu należy zastosować znacznik `<optgroup>`.

PRZYKŁAD 7.15



Rys. 7.16. Przykład zastosowania znacznika `<optgroup>`

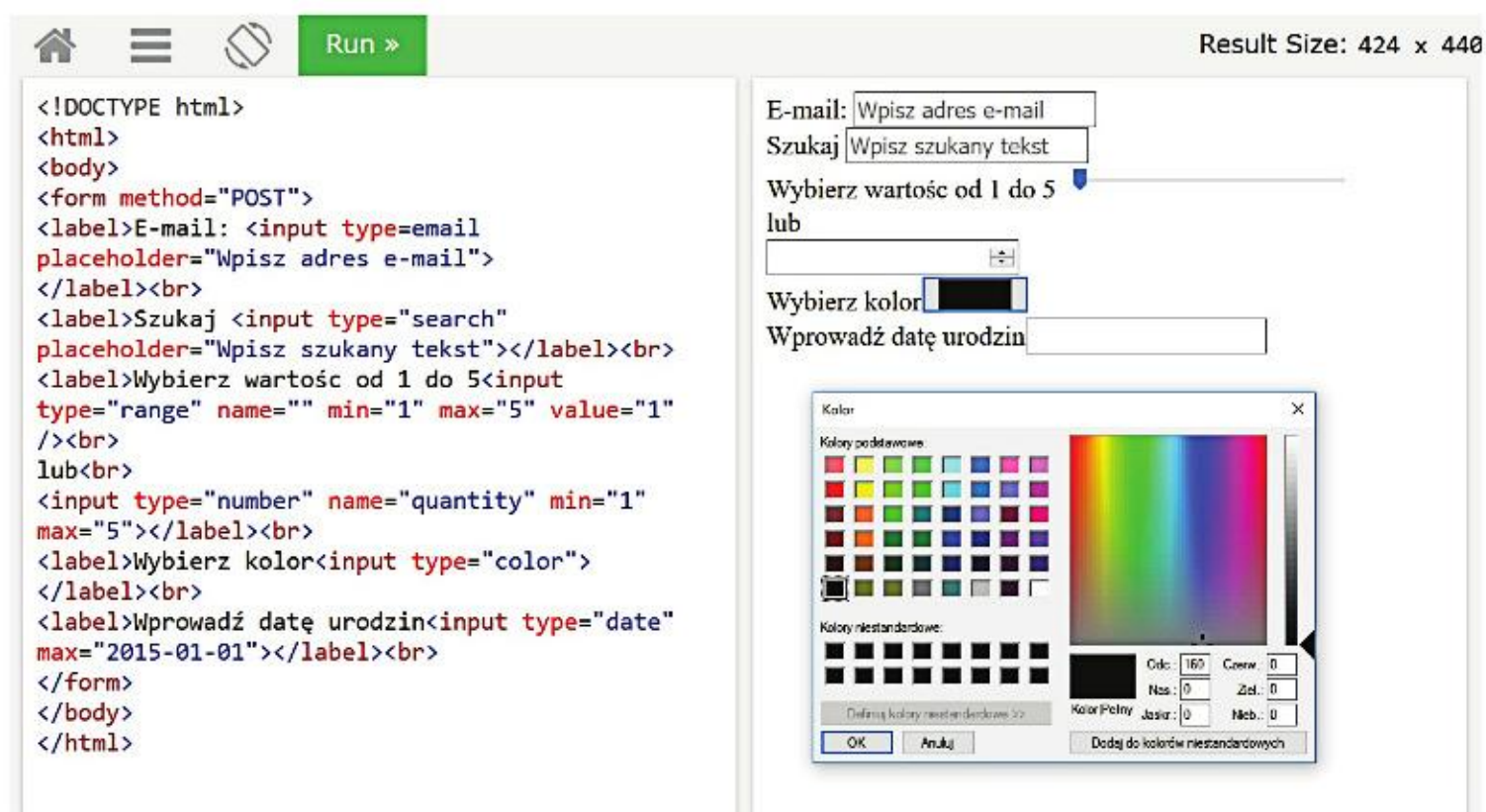
Jeśli nie zostaną zastosowane żadne metody sprawdzania wprowadzonych danych, do formularza należy załączyć atrybut **novalidate** (np. `<form action="kontakt.php" novalidate>`). Trzeba jednak pamiętać, żeby sprawdzić poprawność wprowadzanych danych. Brak sprawdzania poprawności danych po stronie serwera może spowodować wiele poważnych zagrożeń bezpieczeństwa dla danej aplikacji.

Oprócz atrybutów sprawdzających wprowadzanie danych istnieją też:

- atrybut **placeholder** – pozwala na wprowadzenie tekstu tymczasowego do danego pola formularza; ta wartość zostaje usunięta, gdy użytkownik wprowadzi daną do pola;
- atrybut **autocomplete** – pozwala na wyłączenie funkcji autouzupełniania dla danego pola;
- atrybut **autofocus** – pozwala, by wybrane pole było ustawiane jako aktywne zaraz po załadowaniu strony.

PRZYKŁAD 7.16

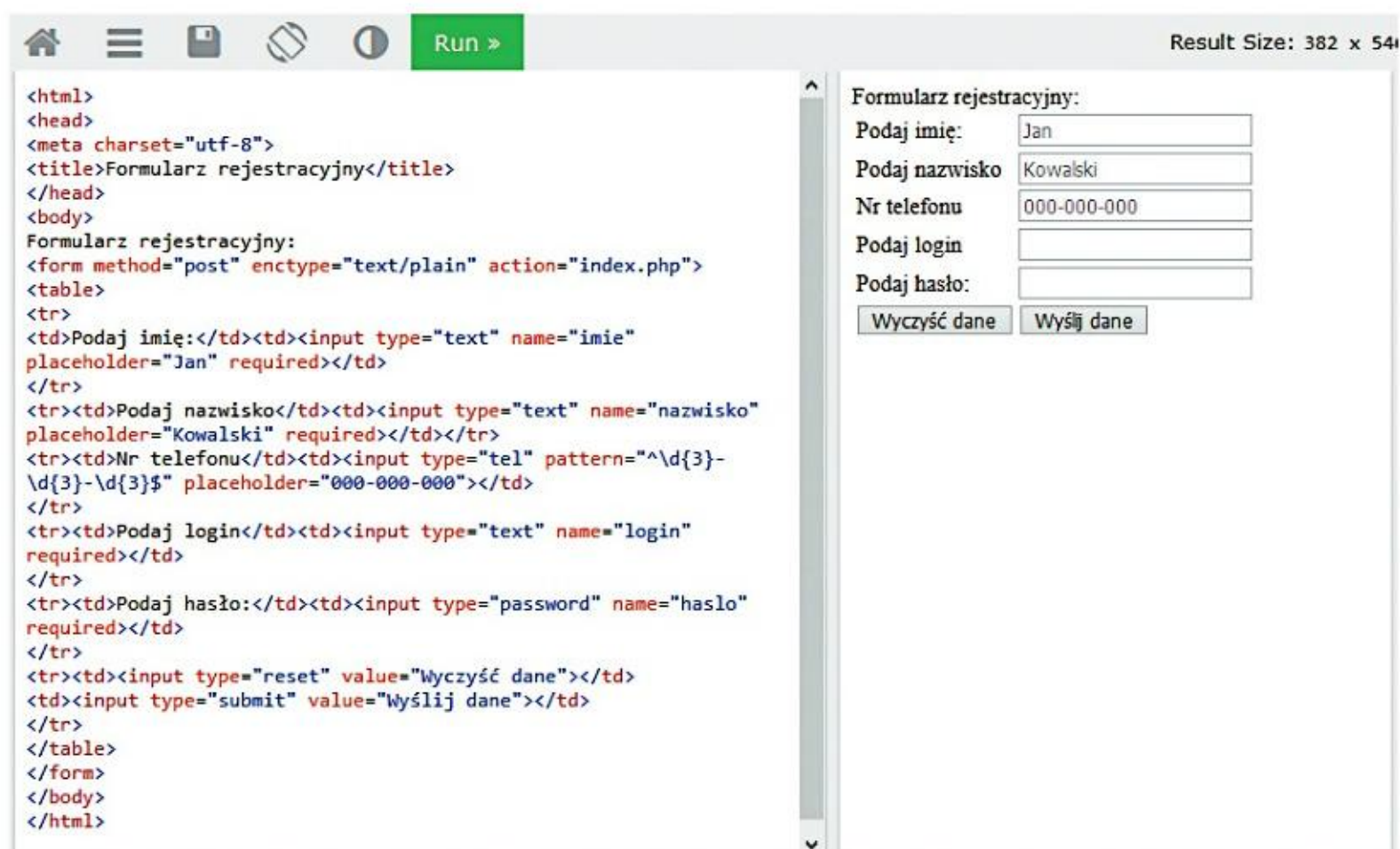
Na rys. 7.17 przedstawiono formularz wykorzystujący nowe typy znacznika `<input>` w HTML 5.



Rys. 7.17. Przykład typów znacznika `<input>` w HTML 5

PRZYKŁAD 7.17

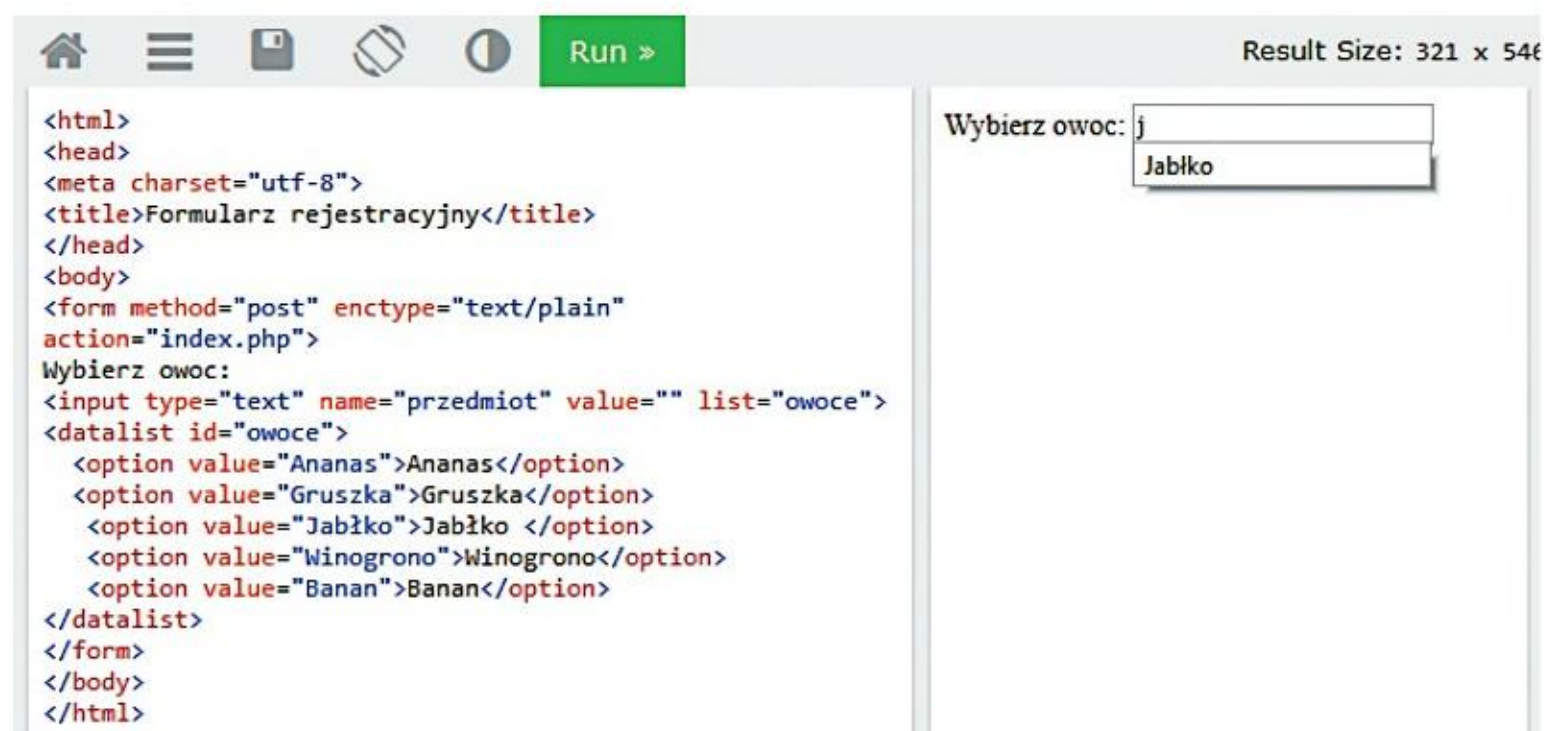
Przykładowy formularz rejestracyjny ma pola obowiązkowe (atrybut **required**), czyli pole, które musi zostać uzupełnione. Za pomocą atrybutu **placeholder** została wprowadzona wartość tymczasowa, która odgrywa rolę podpowiedzi dla użytkownika. Natomiast za pomocą atrybutu **pattern** wprowadzono wzór numeru telefonu. Pod formularzem znajdują się dwa przyciski: **submit** oraz **reset**. Pierwszy z nich spowoduje wysłanie danych do pliku `index.php`, a drugi wyczyści dane wprowadzane do formularza.



Rys. 7.18. Przykład formularza rejestracyjnego

Nowy element **datalist** umożliwia tworzenie listy podpowiedzi do wypełnianego pola typu **<input>**. Za pomocą tej listy ułatwia się użytkownikowi wypełnianie formularza. Nawet jeżeli użytkownik po raz pierwszy wypełnia formularz, od razu zobaczy podpowiedzi, które może wpisać w określonych polach.

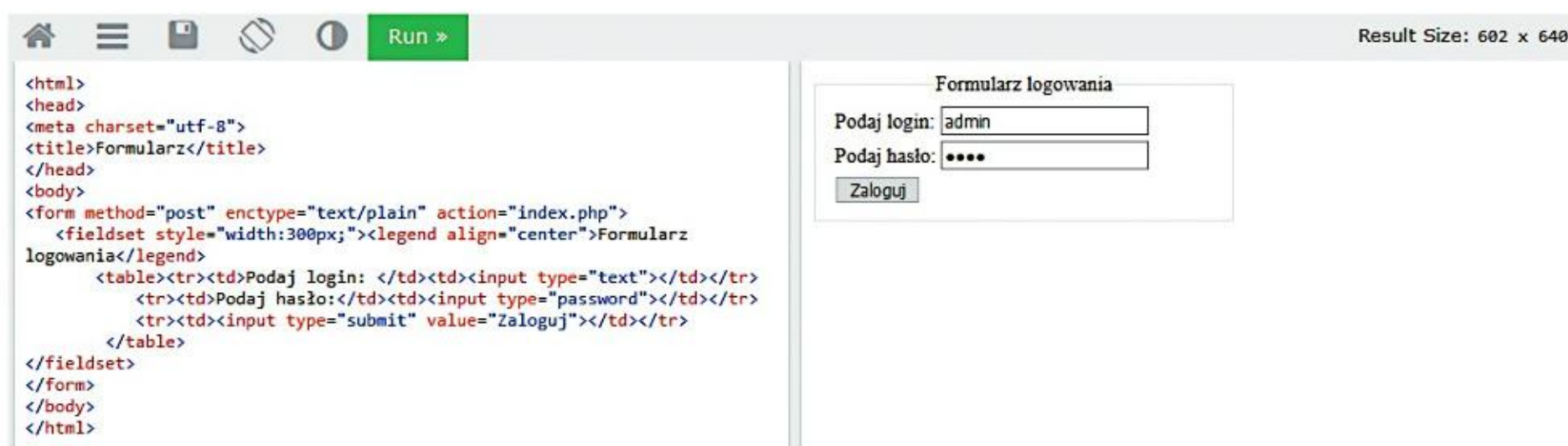
PRZYKŁAD 7.18



Rys. 7.19. Zastosowanie elementu **<datalist>**

Pola formularza można grupować dzięki nadaniu im określonej nazwy. Służy do tego znacznik **<fieldset>**. Aby nadać tytuł danej grupie pól, należy zastosować znacznik **<legend>**, który ma atrybut **align** określający położenie tytułu względem krawędzi ramki grupującej.

PRZYKŁAD 7.19



Rys. 7.20. Zastosowanie znacznika **<fieldset>**

SPRAWDŹ SWOJĄ WIEDZĘ

1. Znacznikami niesemantycznymi są np. `<div>` lub `<span>`, ponieważ nic nam nie mówią o roli, jaką mają odgrywać na stronie.
Do znaczników semantycznych zalicza się takie znaczniki, jak `<table>` czy `<header>`, ponieważ zarówno twórca strony, jak i przeglądarka wiedzą, jaką rolę odgrywają w strukturze strony.
Wypisz w zeszycie nazwy pięciu znaczników semantycznych i pięciu niesemantycznych. Uzasadnij na forum klasy ich wybór.
2. Utwórz dokument hipertekstowy, w którym zdefiniujesz tabelę zgodną ze wzorem podanym poniżej. Pamiętaj, że kolor obramowania tabeli ma być szary, a szerokość obramowania powinna wynosić 2 px. Gotowy plik zapisz pod nazwą `tabela.html` i otwórz w dowolnej przeglądarce internetowej.

Rozkład zajęć dodatkowych z informatyki				
rok szkolny	miesiąc	dzień	godzina	sala
2017/2018	wrzesień–styczeń	wtorek	8.00	116
	luty–kwiecień	czwartek	14.00	
	maj–czerwiec	wtorek	16.00	

3. Jakie zmiany w możliwościach tworzenia formularzy wprowadzono w specyfikacji języka HTML 5 w stosunku do poprzednich wersji?
4. Jaką rolę odgrywają znaczniki `<nav>`, `<input>`, `<td>`?
5. Utwórz dokument hipertekstowy, który zdefiniuje podział strony internetowej według podanego wzoru. Do wykonania zadania wykorzystaj znaczniki zgodne ze specyfikacją HTML 5. Gotowy plik zapisz pod nazwą `szablon.html`.

STRONA INTERNETOWA SZKOŁY		
WAŻNE LINKI: www.men.gov.pl www.ore.gov.pl www.wsip.pl www.dobraszkoła.pl	O NAS: Nasza szkoła jest najlepsza, najładniejsza i najlepiej położona	MENU: Dla uczniów Dla nauczycieli Dla kandydatów Dla rodziców
STRONA ZARZĄDZANA PRZEZ SAMORZĄD UCZNIOWSKI		

6. Utwórz plik o nazwie index.html, a w drugiej kolejności stwórz w nim następującą tabelę:

Wpisz swoje imię i nazwisko												
Element 1.1	Element 1.2	Element 1.3										
Element 2.1-2.3												
Element 3.1 <table><tr><td>4.1-5.1</td><td>4.2-4.3</td></tr><tr><td></td><td>5.2 5.3</td></tr></table>	4.1-5.1	4.2-4.3		5.2 5.3	Element 3.2	Element 3.3 <table><tr><td>A1</td><td>A2</td><td>A3</td></tr><tr><td>B1</td><td>B2</td><td>B2</td></tr></table>	A1	A2	A3	B1	B2	B2
4.1-5.1	4.2-4.3											
	5.2 5.3											
A1	A2	A3										
B1	B2	B2										

Tabela zewnętrzna ma szerokość 400 px i wysokość 400 px, natomiast dwie mniejsze mają szerokość i wysokość po 100 px. Pierwszy wiersz tabeli jest wierszem nagłówkowym. Umieść w nim swoje imię i nazwisko.

7. Utwórz poniższy formularz.

Dane osobowe

Imię:

Nazwisko:

Data urodzenia:

Miejsce urodzenia:

Pozostałe informacje

Wykształcenie: ☐ podstawowe ☐ średnie ☐ wyższe

Zawód wykonywany:

Dodatkowe umiejętności

Wyślij



Grafika i multimedia w dokumentach HTML

ZAGADNIENIA

- Wstawianie obrazów do dokumentów hipertekstowych
- Wstawianie obiektów multimedialnych do dokumentów hipertekstowych

Pierwsze strony internetowe służyły głównie do prezentacji treści tekstowych. Na początku lat 90. XX w. rozpoczęła się dyskusja nad wprowadzeniem znacznika, który wstawiłby na stronę bitmapę lub pixmapę (popularne wówczas formaty graficzne) przez pobranie przez przeglądarkę określonego pliku z podanej w atrybucie znacznika lokalizacji. Obecnie nie wyobrażamy sobie strony internetowej bez elementów grafiki statycznej, animacji, dźwięków bądź nawet krótkich filmów.

Właściwe zastosowanie znaczników multimedialnych w dokumentach hipertekstowych zależy od rodzaju i wersji przeglądarki internetowej. Dlatego przed użyciem konkretnych technologii jest konieczne ustalenie przeglądarek i urządzeń, na których ma pracować dana strona internetowa.

Grafika statyczna

Do osadzania obrazów statycznych (zdjęć, rysunków, symboli lub wykresów) służy znacznik ****. W dokumencie hipertekstowym nie wkleja się gotowego obrazu, tylko wskazuje ścieżkę URL do pliku, który otworzy się w oknie przeglądarki po jego uprzednim pobraniu. Dlatego też poza samym znacznikiem niezbędne jest zastosowanie atrybutu **src="adres URL do pliku"**.

```

```

Z czasem możliwość osadzania różnych formatów plików graficznych znacznie się poszerza, a stare, pocziwe bitmapy (**bmp**) coraz rzadziej wykorzystuje się w technologii tworzenia stron internetowych.

Do najczęściej używanych rozszerzeń plików dla statycznych obrazów cyfrowych zalicza się: **jpg**, **jpeg**, **png** oraz **gif**.

Aby zdecydować się na najlepszy format do stosowania na stronach internetowych, trzeba poznać różnice występujące między poszczególnymi formatami.

Format **JPG** ma możliwość kompresji obrazów z płynnymi przejściami tonalnymi. Nadaje się idealnie do osadzania na stronie wielobarwnych zdjęć, zawierających płynne przejścia między kolorami.

Format **GIF** jest dobry do prezentacji tworzonych rysunków, banerów bądź przycisków, korzystających z palety 256 kolorów. Warto pamiętać, że ten format obsługuje przezroczystość i animację.

Format **PNG** jest przeznaczony do zastosowań internetowych. To następca formatu GIF. PNG nie ma limitu 256 kolorów i dysponuje lepszą kompresją. Wspiera on także przezroczystość kanału alfa, czyli może być częściowo przezroczysty, całkowicie przezroczysty lub zupełnie nieprzezroczysty.

Znacznika **** używa się również z atrybutami. Jednym z nich jest atrybut **alt**, który pozwala na dodanie tekstu alternatywnego, wyświetlanego, gdy grafika nie może zostać załadowana na stronie. Dodatkowo atrybutami **width** (szerokość) oraz **height** (wysokość) można wykonać skalowanie obrazu do wymaganego na stronie rozmiaru.

PRZYKŁAD 8.1

Zastosujemy tu dwa przykłady osadzania obrazu w kodzie HTML. Wynik i różnice w prezentacji obrazu przedstawiono na rys. 8.1.

Fragment kodu:

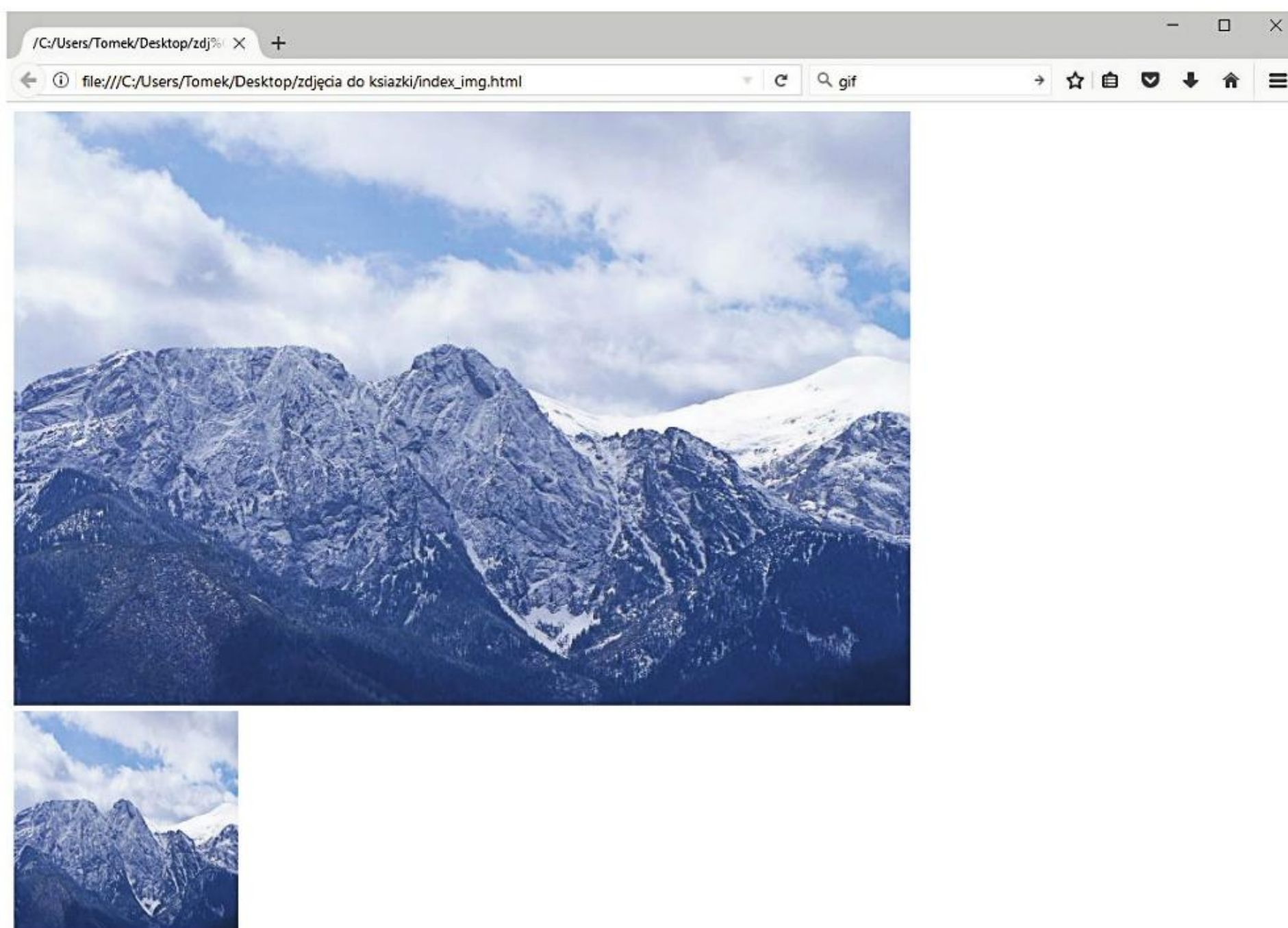
```

```

```

```

Wynik działania kodu:



Rys. 8.1. Zastosowanie znacznika ****

Mapa obrazu

Z grafiki umieszczonej na stronie można wykonać mapę odsyłaczy, czyli utworzyć odnośnik z wybranego obszaru grafiki. Danym obszarom obrazu przypisuje się określony odsyłacz i dzieli na wyznaczone części. Po najechaniu kursorem myszy na dany obszar można uruchomić odsyłacz zdefiniowany w kodzie.

Standardowe mapowanie ma następującą postać:

```

<map name="nazwa">
<area shape="rect" coords="1, 1,50,50" href="strona.html">
</map>
```

W tym przypadku opieramy się na znaczniku `<img>`; definiuje on nazwę mapy atrybutem `usemap="#nazwa"`. Znacznik `<map>` służy do stworzenia mapy obrazu po stronie przeglądarki internetowej. Znacznik `<area>` zawsze umieszcza się wewnątrz znacznika `<map>`. Służy on do tworzenia obszarów na mapie obrazu.

Atrybut **shape** określa kształt mapy odnośnika. Może przyjmować następujące wartości:

- rect – obszar prostokątny,
- circle – obszar w postaci okręgu,
- poly – obszar w postaci wielokąta,
- default – obszar obejmuje całą grafikę.

Atrybut **coords** służy do określenia współrzędnych aktywnego obszaru. Dla każdego z typów obszaru może przyjmować określone wartości:

- dla rect przyjmuje cztery wartości: od lewej, od góry, szerokość, wysokość;
- dla circle przyjmuje trzy wartości: od lewej, od góry, promień okręgu;
- dla poly wartościami będą wierzchołki wielokąta, czyli jego współrzędne.

PRZYKŁAD 8.2

Na rys. 8.2 przedstawiono kod HTML tworzący podział rysunku kwadrat.jpg na cztery obszary, dla których zdefiniowano odnośniki.

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     
5
6     <map name="kw">
7       <area shape="rect" coords="0,0,100,100" alt="górny lewy" href="1.htm">
8       <area shape="rect" coords="0,100,100,200" alt="dolny lewy" href="2.htm">
9       <area shape="rect" coords="100,0,200,100" alt="górny prawy" href="3.htm">
10      <area shape="rect" coords="100,100,200,200" alt="dolny prawy" href="4.htm">
11    </map>
12  </body>
13 </html>
14
```

Rys. 8.2. Przykład tworzenia mapy obrazu

Kontener grafiki SVG

Specyfikacja języka HTML 5 pozwala na zastosowanie znaczników, które nie tylko wstawiają pliki formatu svg na stronę internetową, lecz także umożliwiają dynamiczne rysowanie w oknie przeglądarki obrazu zdefiniowanego w kodzie za pomocą znacznika **<svg>**. Skalowalna grafika wektorowa – SVG (ang. *Scalable Vector Graphics*) jest językiem opracowanym na podstawie XML-a, służącym do opisu dwuwymiarowej grafiki wektorowej. Pliki SVG mogą być w HTML 5.0 zamieszczone na stronie.

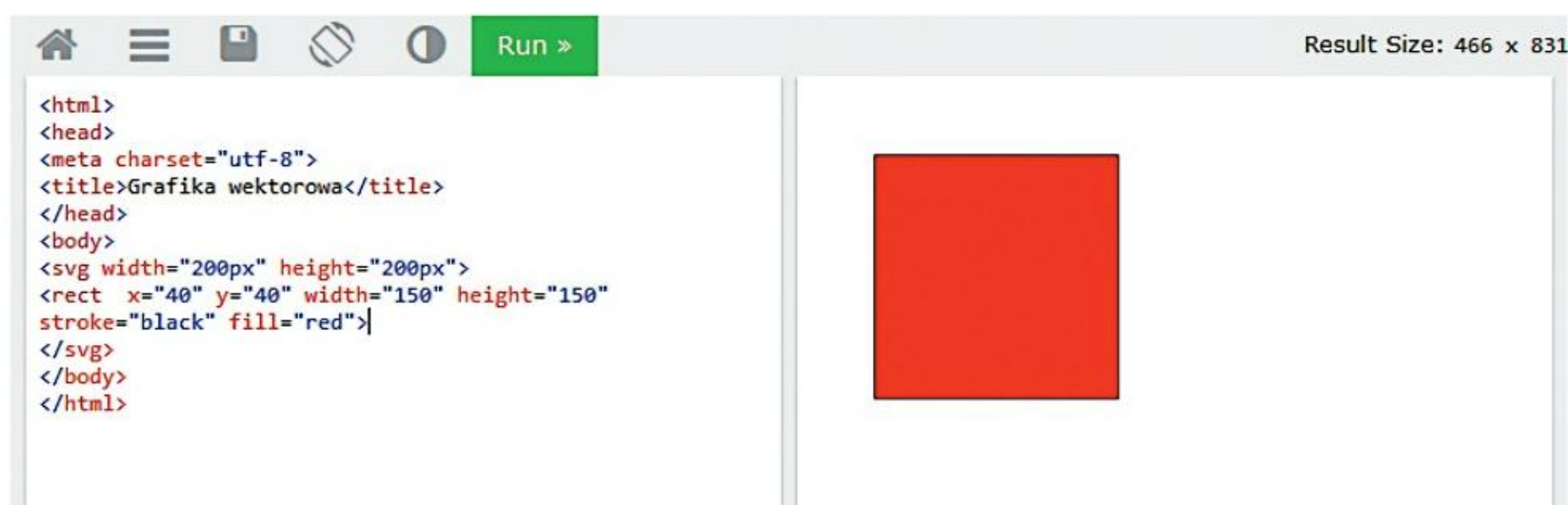
Znacznik **<svg>** określa obszar rysowania wektorowej na stronie i można w nim umieszczać następujące obiekty graficzne:

- **<rect>** – prostokąt,
- **<circle>** – okrąg,
- **<ellipse>** – elipsa,
- **<line>** – linia,
- **<polyline>** – linia łamana,
- **<polygon>** – wielokąt,
- **<path>** – ścieżka.

Podstawowymi atrybutami znacznika **<svg>** są własności **width** i **height**, które określają szerokość oraz wysokość grafiki svg. Te atrybuty można zastosować podczas tworzenia szczególnych obiektów graficznych, np. prostokąta.

PRZYKŁAD 8.3

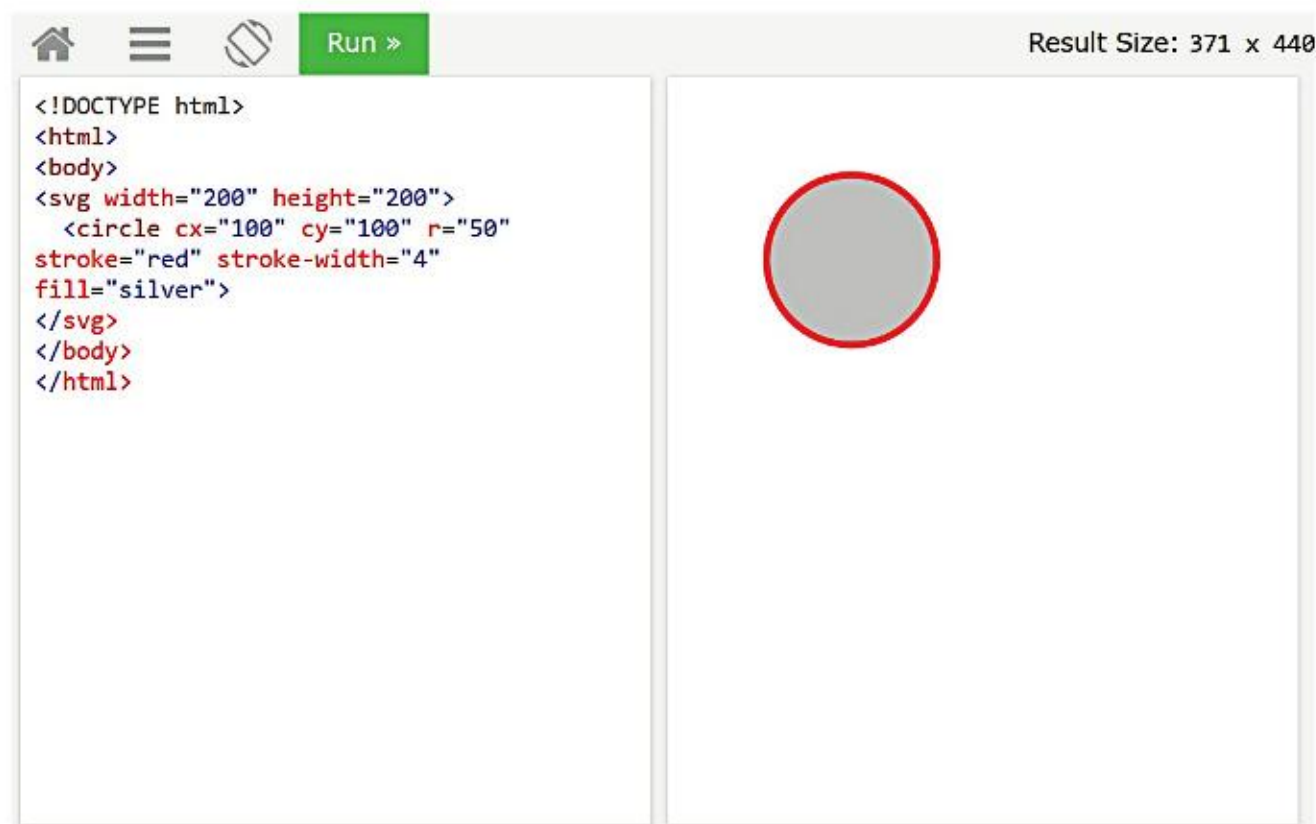
Na rys. 8.3 przedstawiono przykład rysowania czerwonego kwadratu z czarnym obramowaniem. W znaczniku **<rect>** użyto atrybutów **x** oraz **y**, które określają współrzędne górnego lewego wierzchołka rysowanego prostokąta, natomiast atrybuty **height** oraz **width** określają jego wymiary. Atrybut **stroke** określa kolor obramowania, a atrybut **fill** ustala kolor wypełnienia danej figury.



Rys. 8.3. Rysowanie kwadratu w **<svg>**

PRZYKŁAD 8.4

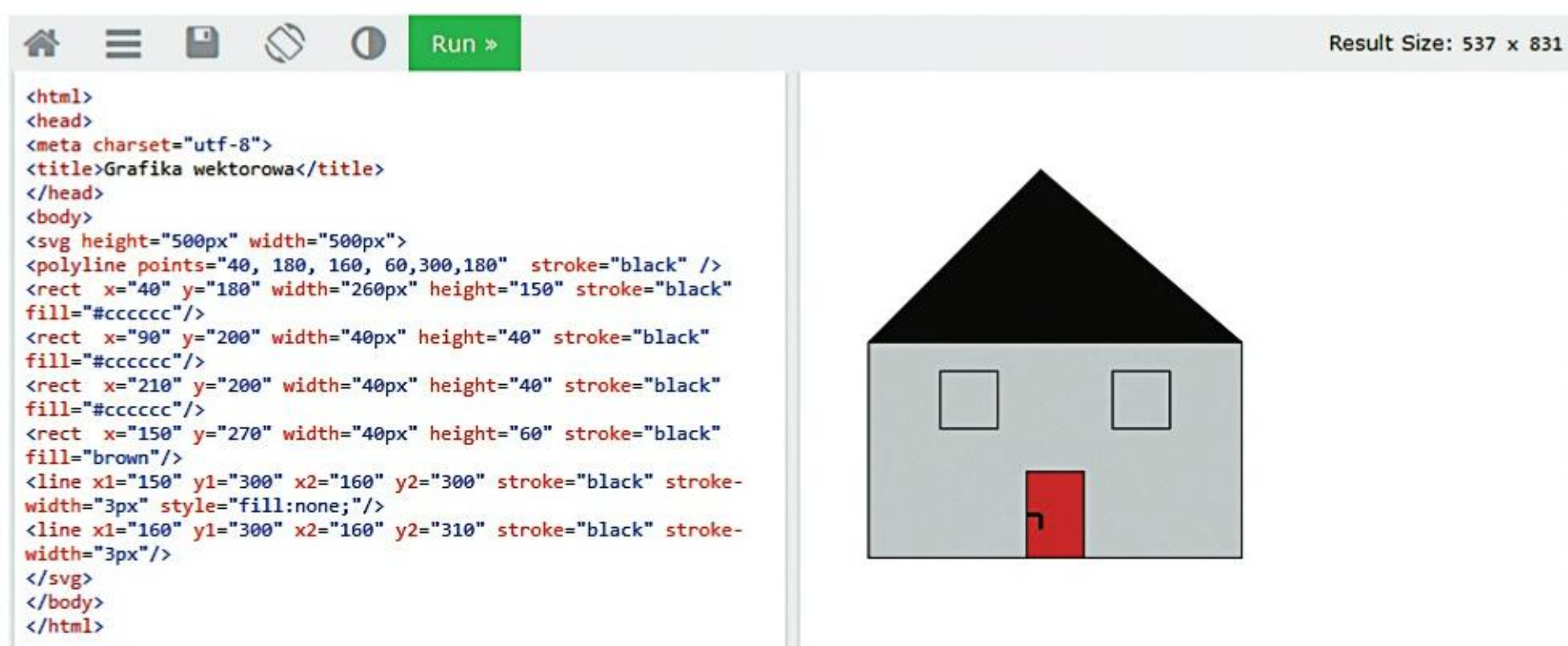
Na rys. 8.4 pokazano przykład zastosowania znacznika `<svg>`. W oknie przeglądarki zdefiniowano kontener `svg` o wymiarach 200 na 200 px, w którym narysowano okrąg o podanych parametrach `cx`, `cy` i `r` oraz nadanym kolorze wypełnienia i ustalonej krawędzi za pomocą atrybutu `stroke`. Ze wszystkimi atrybutami znacznika `<svg>` można zapoznać się na stronach W3C.



Rys. 8.4. Przykład rysowania okręgu w `<svg>`

PRZYKŁAD 8.5

Oprócz rysowania prostokątów bądź okręgów, jest też możliwe rysowanie linii łamanych i prostych. Dla linii łamanych określa się poszczególne współrzędne wierzchołków, wypisując je w atrybucie `points`. Linia prosta wymaga czterech atrybutów: `x1`, `y1`, `x2`, `y2`, określających odpowiednio początek i koniec rysowanej linii. Za pomocą elementów grafiki `<svg>` można utworzyć prosty rysunek składający się z podstawowych figur geometrycznych (rys. 8.5).



Rys. 8.5. Prosty rysunek w `<svg>`

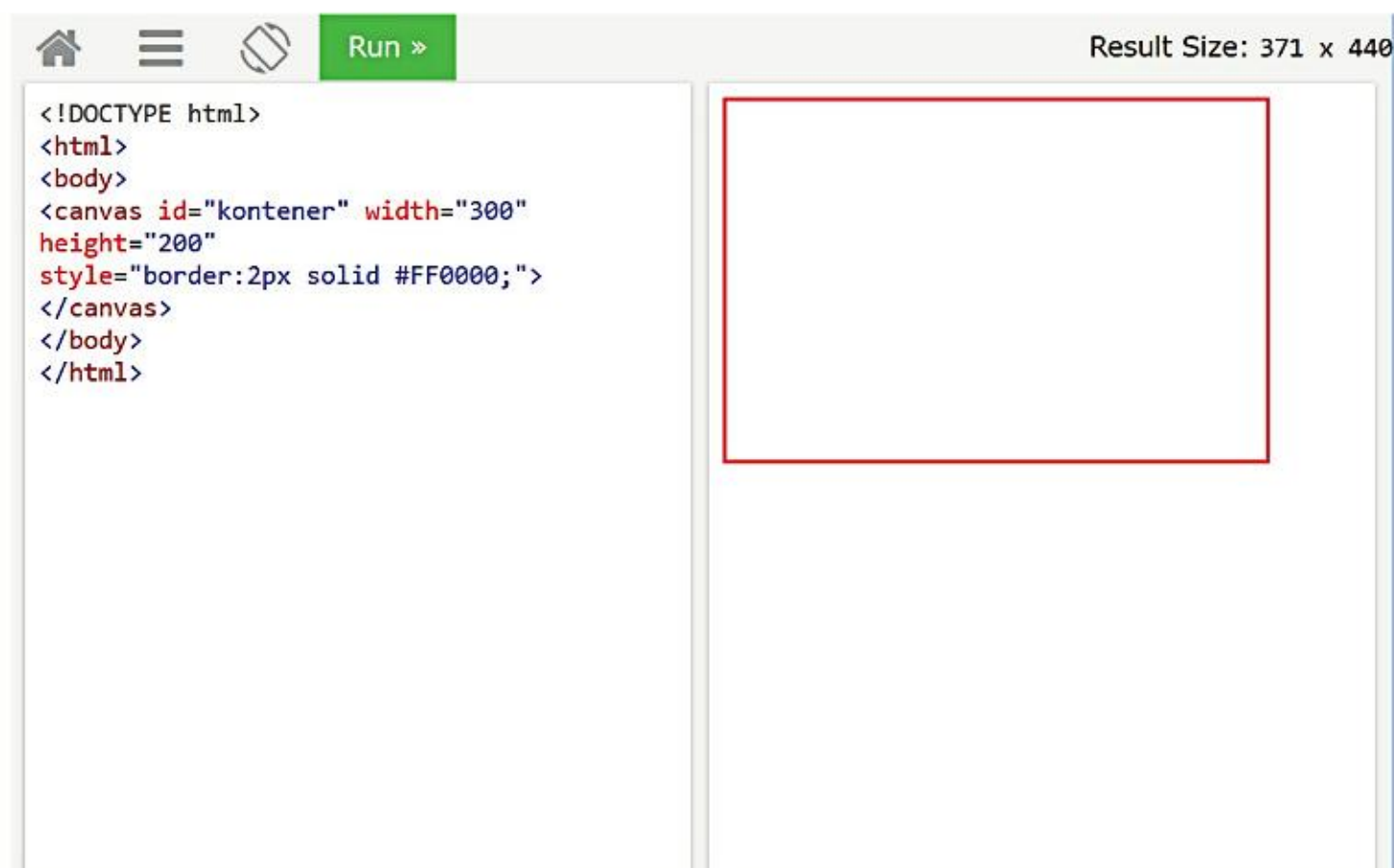
Element Canvas

Innym elementem umożliwiającym rysowanie grafiki bezpośrednio w oknie przeglądarki jest znacznik **<canvas>**. Definiuje on kontener, który możemy wypełnić kolorem, tekstem, figurami geometrycznymi lub innym obrazem.

Działanie znacznika **<canvas>** jest oparte na dynamicznym, skryptowym renderowaniu kształtów i obrazów bitmapowych. Dlatego musimy pamiętać, że znacznik **<canvas>** definiuje tylko kontener do rysowania grafiki. Dzięki temu znacznikowi za pomocą skryptów JavaScript można tworzyć zaawansowane aplikacje graficzne.

PRZYKŁAD 8.6

Językiem skryptowym zajmiemy się w dalszej części podręcznika, dlatego na rys. 8.6 pokazano tylko definiowanie kontenera do tworzenia grafiki.



Rys. 8.6. Zastosowanie znacznika **<canvas>**

Multimedia na stronach internetowych

Multimedia reprezentują grupę plików zawierających dźwięk i wideo. Najczęściej obsługują one następujące formaty:

- pliki dźwiękowe – wav, mid, midi, mp3, ogg;
- pliki wideo i animacji – avi, mpeg, mp4, asf, swf.

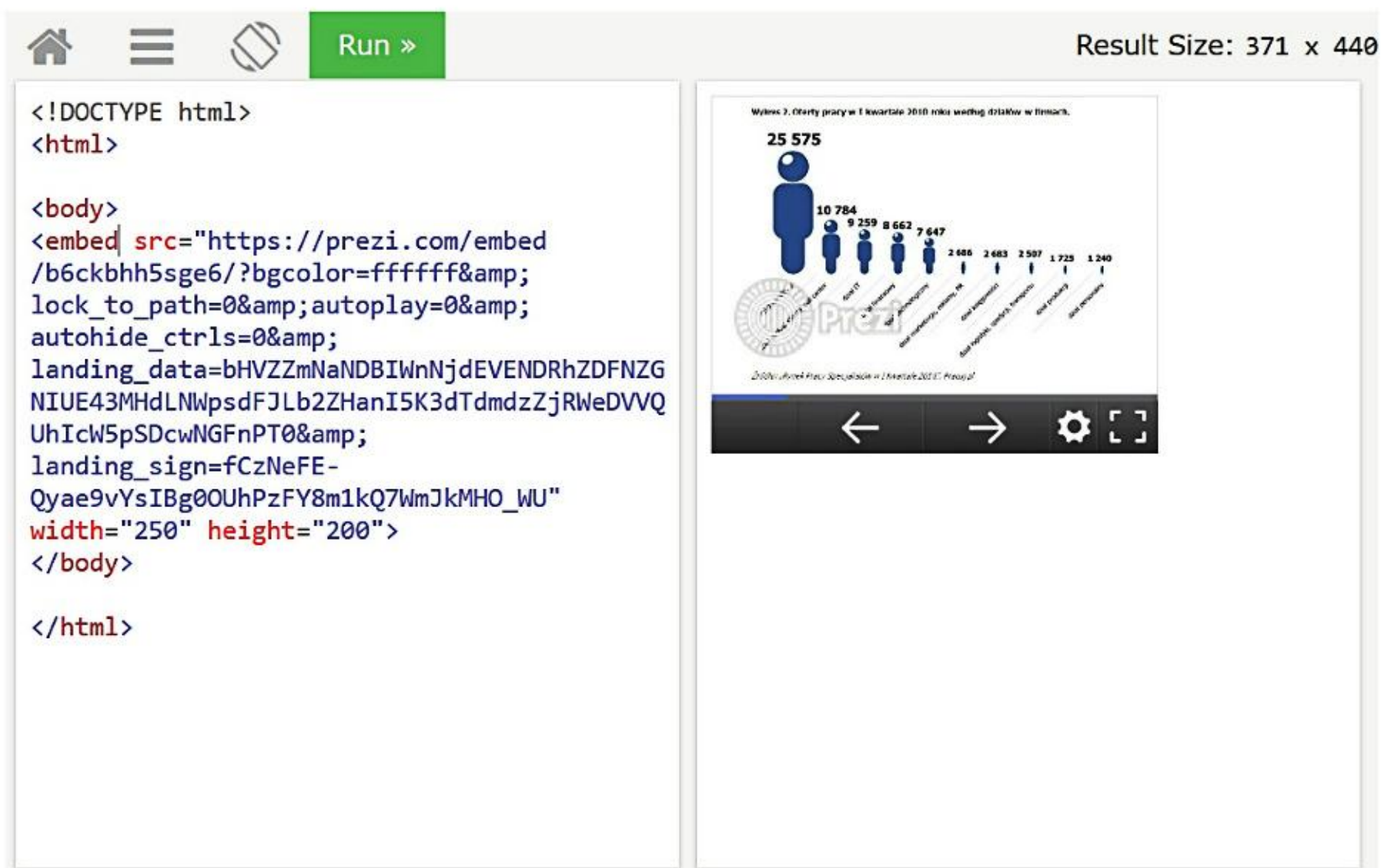
Niektóre z nich wymagają zainstalowania w przeglądarce specjalnego rozszerzenia zwanego plug-in lub wtyczką.

Do umieszczania na stronie internetowej plików multimedialnych służy znacznik **<embed>**. Niestety, nie jest on obsługiwany przez wszystkie przeglądarki internetowe. Jedynym wymaganym przez znacznik **<embed>** atrybutem jest **src**, który definiuje ścieżkę dostępu do pliku multimedialnego. Dodatkowo można wykorzystać standardowe atrybuty określające obszar wyświetlania.

PRZYKŁAD 8.7

```
<embed src="animacja.swf" width="100px" height="50px">
```

Na rys. 8.7 przedstawiono zastosowanie znacznika `<embed>` z zastosowaniem sieciowego pliku multimedialnego do wyświetlenia na stronie.



Rys. 8.7. Zastosowanie znacznika `<embed>`

Jeżeli przeglądarka nie obsługuje znacznika `<embed>`, można zastosować znacznik `<object>`. Pozwala on na osadzanie w dokumentach hipertekstowych plików audio, wideo, apletów Javy, animacji Flash, a nawet dokumentów PDF. Podczas wykorzystania znacznika `<object>` należy pamiętać, że większość jego atrybutów nie jest obsługiwanych przez HTML 5. W najnowszej specyfikacji języka HTML stosuje się atrybuty:

- **data** – definiujący ścieżkę URL do pliku;
- **form** – definiujący jedną lub więcej form należących do obiektu;
- **name** – określający nazwę obiektu;
- **width** oraz **height** – określające wymiary okna wyświetlania obiektu.

PRZYKŁAD 8.8

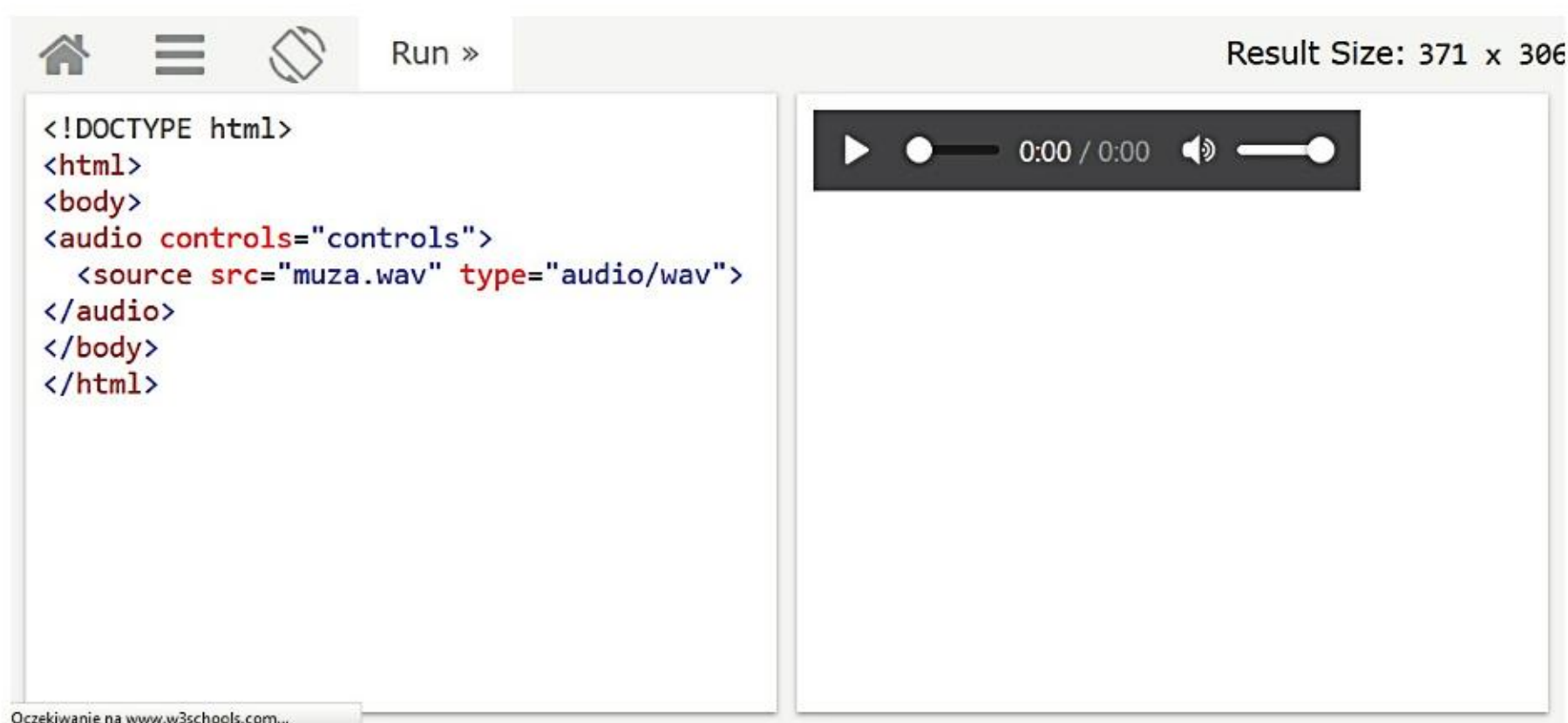
```
<object width="100px" height="50px" data="animacja.swf">
</object>
```


W języku HTML 5 występują dodatkowe znaczniki pozwalające na osadzanie plików multimedialnych w kodzie dokumentu hipertekstowego. Nie każda przeglądarka jednak obsługuje wszystkie formaty plików audio i wideo. Należy o tym pamiętać podczas projektowania witryny.

Do umieszczania plików audio służy znacznik **<audio>**. Może zawierać jedno lub kilka źródeł dźwięku reprezentowanych przez użycie atrybutu **src** lub elementu **<source>**. Określona przeglądarka wybierze wówczas najbardziej odpowiedni dla niej zapis. Dzięki użyciu MediaStream znacznik może służyć również jako destynacja dla mediów strumieniowych.

PRZYKŁAD 8.9

Na rys. 8.8 przedstawiono przykład zastosowania znacznika **<audio>**. W oknie przeglądarki jest widoczny panel sterowania odtwarzaniem pliku dźwiękowego.



Rys. 8.8. Zastosowanie znacznika **<audio>**

Do umieszczania plików wideo służy znacznik **<video>**. Do standardowych atrybutów tego znacznika zalicza się m.in.:

- **src** – definiuje ścieżkę URL do pliku;
- **poster** – definiuje ścieżkę do obrazka, który będzie wyświetlany w trakcie ładowania filmu;
- **autoplay** – automatyczne odtwarzanie podczas ładowania filmu;
- **controls** – wyświetlanie panelu sterowania;
- **loop** – pętla odtwarzania filmu.

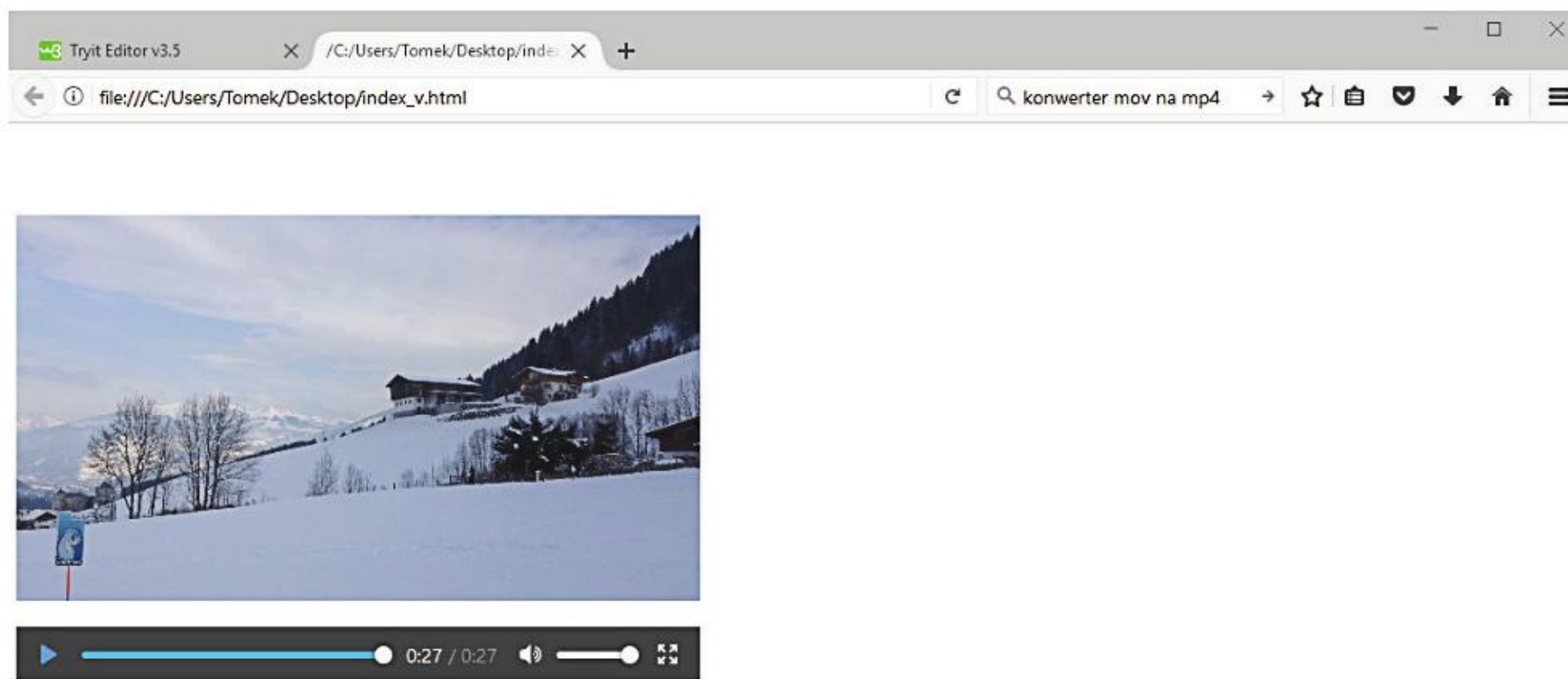
PRZYKŁAD 8.10

Na rys. 8.9 pokazano kod strony wykorzystującej publikację pliku wideo za pomocą znacznika **<video>**. Zastosowano tutaj definicję dwóch plików z różnymi rozszerzeniami, do wyboru przez przeglądarkę obsługiwanego rodzaju multimediiów.

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <video width=500" height="400" controls>
5       <source src="animacja.mp4" type="video/mp4">
6       <source src="animacja.mov" type="video/mov">
7     </video>
8   </body>
9 </html>
```

Rys. 8.9. Zastosowanie znacznika **<video>** w kodzie HTML

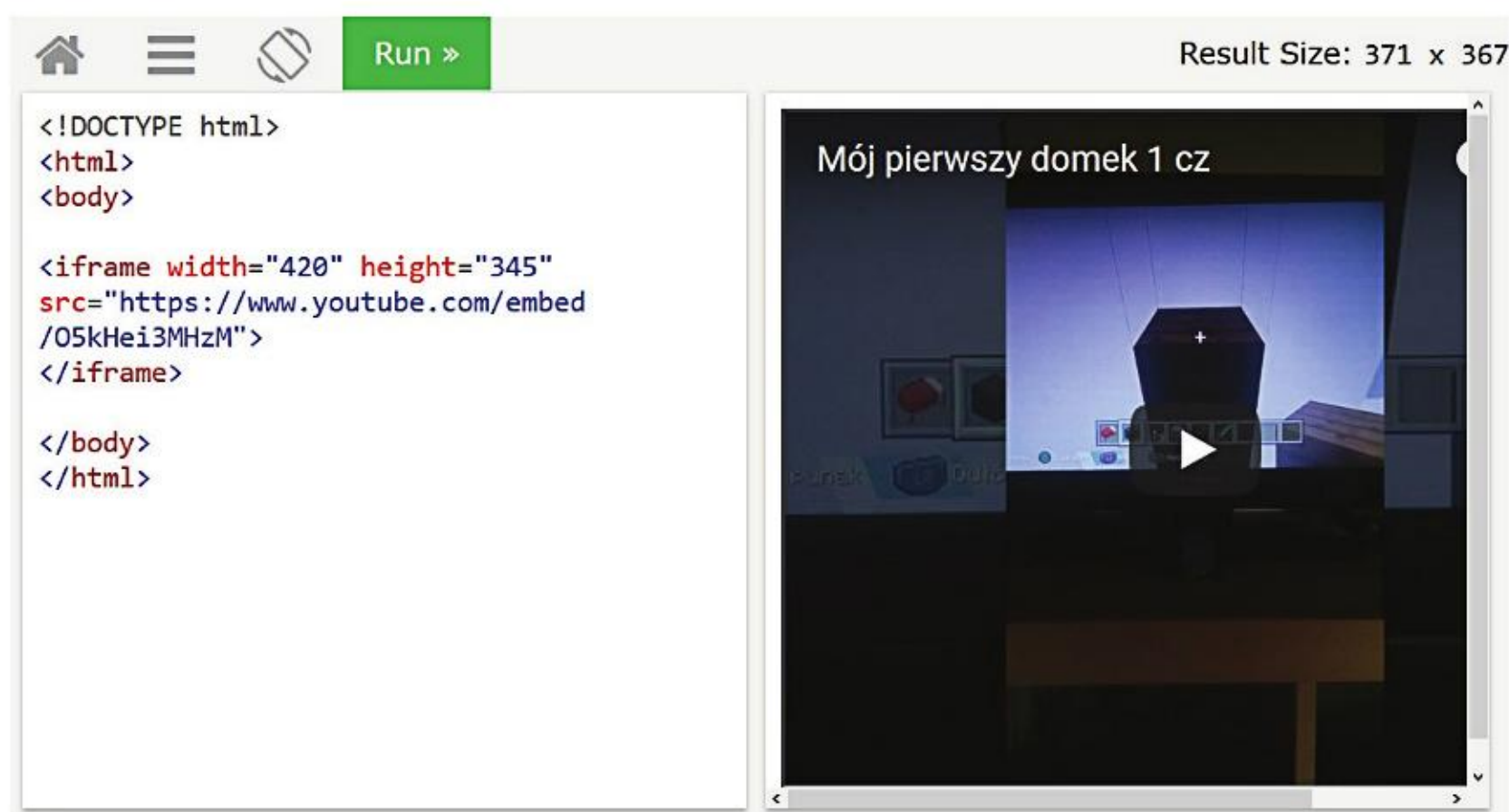
Efektem załadowania omawianego kodu jest widoczny na rys. 8.10 obszar z wyświetlanym plikiem multimedialnym oraz panelem sterowania wyświetlaniem.



Rys. 8.10. Efekt działania kodu ze znacznikiem **<video>**

Do odtwarzania filmów wideo przez przeglądarkę internetową są często niezbędne kodeki lub wtyczki. Pozwalają na sprawne przeprowadzenie procesu kodowania lub dekodowania plików multimedialnych. Do popularnych kodeków zalicza się Real Player i Adobe Flash Player.

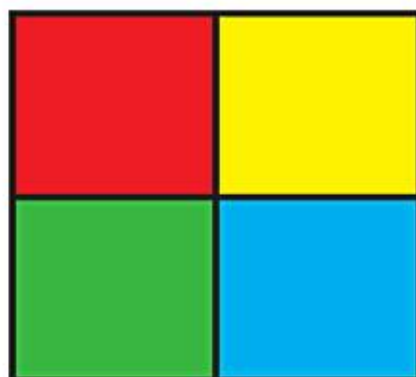
Zainstalowanie właściwych wtyczek umożliwia publikację na własnych stronach internetowych materiałów wideo z popularnych portali, takich jak YouTube. Wykorzystuje się wówczas znacznik **<iframe>**. W ścieżkę URL wystarczy wpisać identyfikator filmu (np. XGSy3_Czz8k) oraz atrybuty parametrów wyświetlania obrazu. Przykład zastosowania tego elementu pokazano na rys. 8.11.



Rys. 8.11. Przykład osadzania filmu w dokumencie hipertekstowym

SPRAWDŹ SWOJĄ WIEDZĘ

1. Przygotuj rysunek w aplikacji MS Paint według podanego wzoru. Zapisz go pod nazwą kwadrat.jpg. Następnie utwórz dokument hipertekstowy tworzący mapę obrazu (wykorzystaj przykład z rys. 8.2). Zmodyfikuj kod. Zamiast nazw 1.htm, 2.htm, 3.htm, 4.htm podaj adresy URL dowolnych stron internetowych. Efekt działania pliku przetestuj w dowolnej przeglądarce.



2. Skorzystaj z dowolnej metody osadzania plików multimedialnych i utwórz dokument hipertekstowy publikujący wybrany film z portalu społecznościowego YouTube. Efekt swojej pracy zapisz pod nazwą wideo.html. Otwórz swój plik w trzech przeglądarkach: Opera, IE, Firefox. Zwróć uwagę na poprawność i sposób wyświetlania filmu. Wnioski z wykonanego zadania omów na forum klasy z nauczycielem.
3. Sprawdź, które z wymienionych poniżej formatów plików dźwiękowych są obsługiwane w przeglądarce internetowej Internet Explorer. Informacji szukaj w dokumentacji przeglądarki i specyfikacji języka HTML5.
Formaty plików dźwiękowych: mp3, wav, ogg.
4. Do czego służy znacznik **<object>**?
5. Zmodyfikuj kod źródłowy z rys. 8.9 w taki sposób, aby w trakcie ładowania filmu wyświetlał się obraz o nazwie rys.jpg, który znajduje się w katalogu głównym strony internetowej.

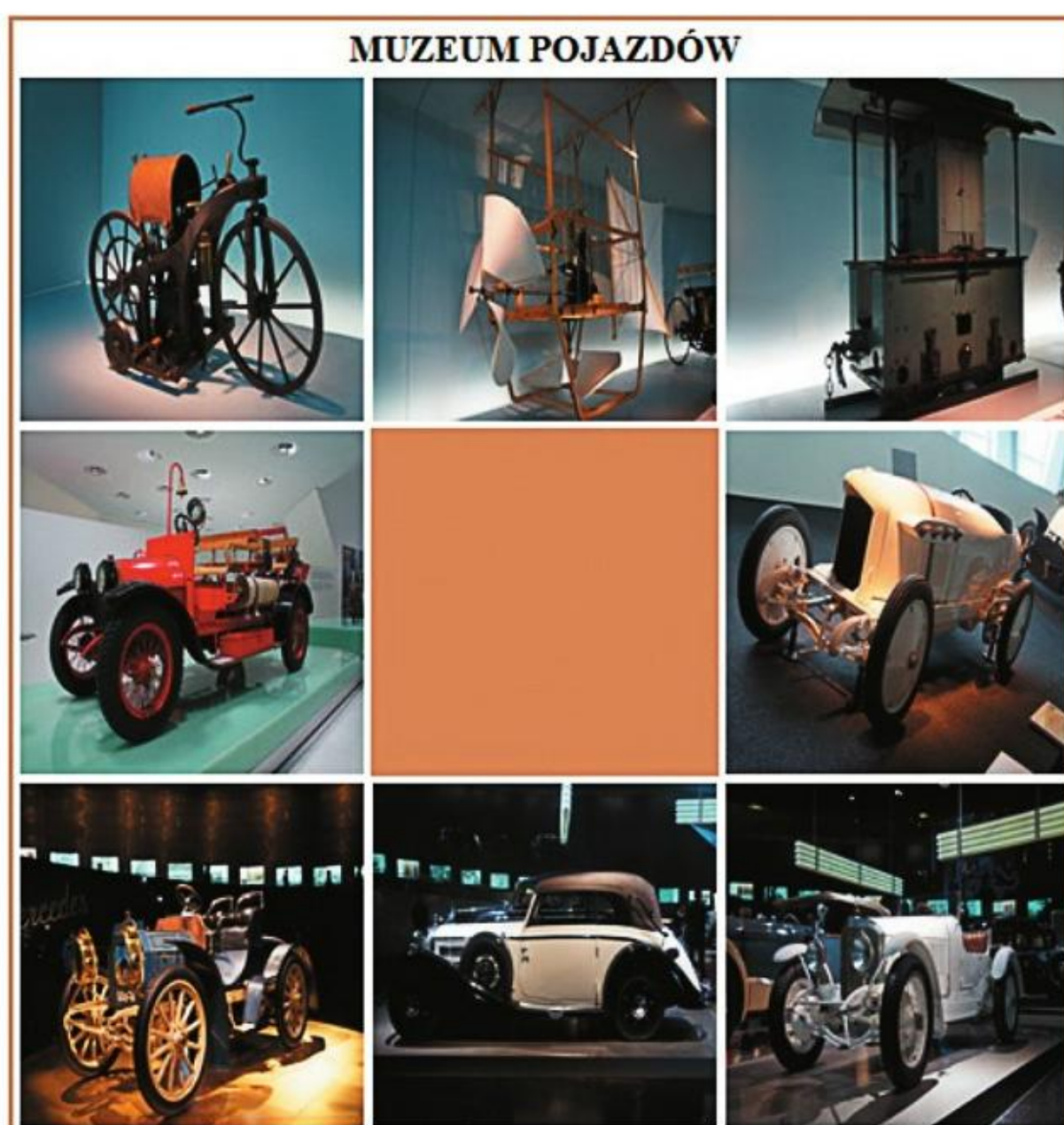
6. Utwórz formularz zgodny z poniższym rysunkiem.

Podaj swoje imię:		
Podaj swoje nazwisko		
Ile masz lat?		
Wybierz płeć:	☐ Kobieta ☐ Mężczyzna	
Jakiej muzyki słuchasz?	☐ Heavy Metal ☐ Rock ☐ Pop ☐ Techno	
Wyślij		

Pierwsze dwa pola są typu tekstowego, następne pole jest typu numerycznego, którego wartością minimalną jest 1. Wybór płci ma się odbywać za pomocą pola typu radio, w którym można wybrać tylko jedną moliwość. Ostatnie pole ma być typu checkbox. Formularz będzie miał przycisk z napisem Wyślij.

Cały formularz ma być wstawiony do tabeli o sześciu wierszach i trzech kolumnach, w ostatniej kolumnie tabeli wstaw grafikę przedstawiającą dowolną roślinę. Tło tabeli jest koloru #eeeeee. Tabela ma być szerokości 600 px i wysokości 300 px. Obrazek obramuj linią ciągłą grubości 3 px, koloru #993399.

7. Utwórz tabelę przedstawioną poniżej. Tabela ma cztery wiersze i trzy kolumny. Pierwszy wiersz tabeli jest wierszem nagłówkowym. Obramowanie tabeli jest koloru jasno-brązowego (dowolny odcień). Tabele wypełnij zdjęciami starych pojazdów (skorzystaj z zasobów internetowych).



IV. Kaskadowe arkusze stylów CSS

- Kaskadowość stylów w dokumentach CSS
- Selektory, właściwości oraz wartości w CSS
- Model blokowy w CSS
- Pozycjonowanie elementów w CSS
- Praktyczne zastosowania CSS i HTML

9

Kaskadowość stylów w dokumentach HTML

ZAGADNIENIA

- Definicja CSS
- Kaskadowość stylów
- Struktura arkusza stylów

Wraz z rozwojem budowy stron internetowych pojawiła się potrzeba rozdzielenia części kodu obejmującej strukturę i treść strony oraz części odpowiedzialnej za prezentację tej treści. Po powstaniu specyfikacji HTML 4 zaczęto rozdzielać kodowanie treści i jej prezentację. Rolę prezentacji treści przejęły kaskadowe arkusze stylów **CSS** (ang. *Cascading Style Sheets*).

CSS jest językiem przeznaczonym do tworzenia arkuszy (plików), za których pomocą określa się cechy poszczególnych znaczników, czyli po prostu wygląd strony. Od wersji 4.0 HTML służy tylko do określania zawartości strony, natomiast arkusze stylów stosuje się do określania tzw. layoutu strony.

Struktura arkusza stylów

Style CSS składają się z listy reguł (dyrektyw) określających, w jaki sposób zostanie wyświetlony wybrany element dokumentu HTML. Definiowanie stylów CSS podlega zasadom określonym w specyfikacji W3C. Styl składa się z **reguły** (sekcji), a każda reguła – z **selektora** i **deklaracji**. Deklaracja składa się z **właściwości**, którym przypisuje się określone **wartości**.

Poniżej przedstawiono składnię pojedynczej reguły kaskadowego arkusza stylów.

```
selektor {  
właściwość1: wartość1;  
właściwość2: wartość2;  
właściwość3: wartość3;  
}
```

Selektorem może być wybrany element strony lub grupa elementów, które zamierza się sformatować. Wewnątrz nawiasów klamrowych deklaruje się cechy danego elementu w postaci elementarnych właściwości. Każdej właściwości przyporządkowuje się wartość (wpisuje się ją po dwukropku). Zapis każdej deklaracji właściwości kończy się średnikiem.

PRZYKŁAD 9.1

Poniższy kod CSS spowoduje, że wszystkie akapity (paragrafy) na stronie będą wyśrodkowane i zapisane niebieską czcionką.

```
p{
  text-align:center;
  color:blue;
}
```

Zalety stosowania arkuszy stylów:

- w prosty i przejrzysty sposób można określić wygląd poszczególnych znaczników;
- każdemu użytkownikowi można zdefiniować stronę o tej samej treści, lecz innym wyglądzie;
- kod strony staje się bardziej czytelny, a strona szybciej się wczytuje.

Metody dołączania stylów do dokumentu HTML

Istnieją trzy metody dołączania kaskadowych arkuszy stylów do dokumentów HTML.

1. **Metoda inline** (styl wpisany, lokalny). W tej metodzie stosuje się atrybut **style** dla wybranego znacznika języka HTML.

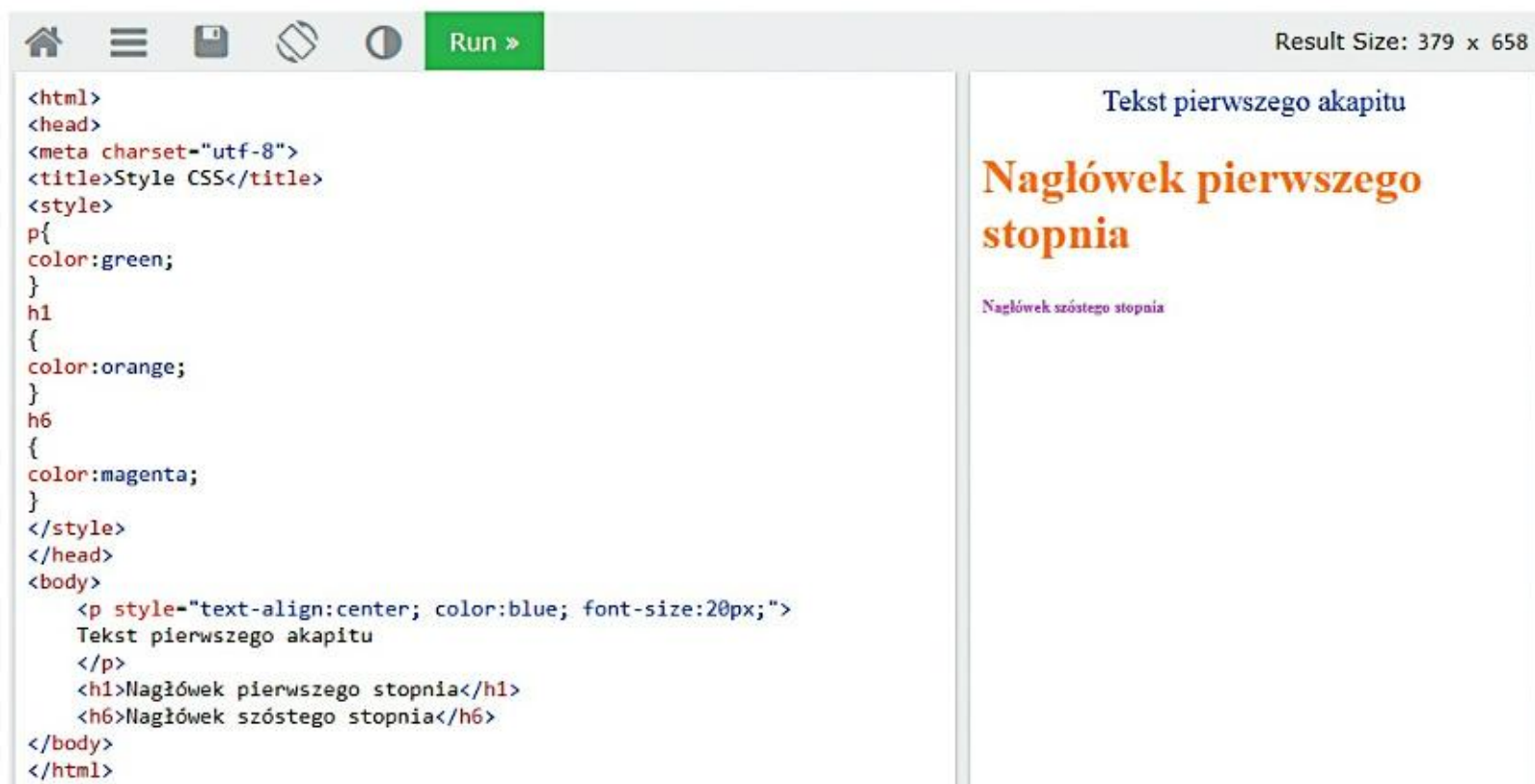
PRZYKŁAD 9.2

Rys. 9.1. Przykład zastosowania stylu w linii

2. **Metoda osadzenia w nagłówku <head>** (styl wewnętrzny). Taki arkusz stylów definiuje się w nagłówku dokumentu HTML, wewnątrz znaczników **<style>** oraz **</style>**. Przykład wykorzystania takiej metody pokazano na rys. 9.2.

PRZYKŁAD 9.3

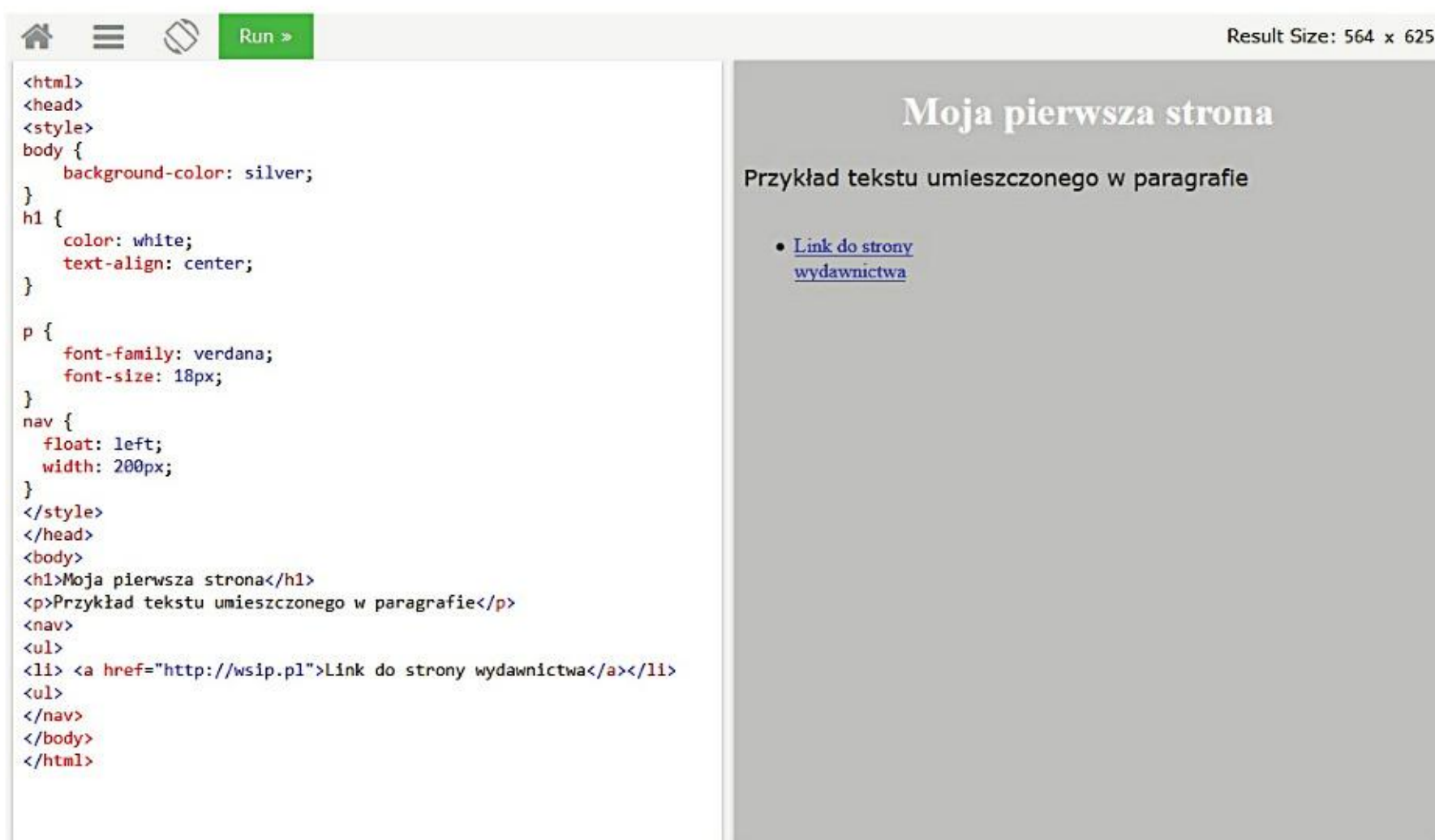
W poniższym przykładzie znacznik **p** sformatowano w stylach wewnętrznych oraz w linii. Mimo że kolor paragrafu określono w dwóch miejscach, to i tak na stronie wyświetlono tekst w kolorze niebieskim. Znaczniki **h1** oraz **h6** nie mają formatowania liniowego – ich wygląd określono w stylach wewnętrznych.



Rys. 9.2. Przykład zastosowania stylu wewnętrznego oraz w linii

PRZYKŁAD 9.4

W przykładzie jest widoczny styl wstawiony w nagłówku strony internetowej. Definiuje on cztery elementy strony: **body**, **h1**, **p** i **nav**, odnoszące się do znaczników języka HTML o tych samych nazwach. Oznacza to, że jeżeli w obrębie strony użyjemy znacznika **<h1>**, to formatowanie tego elementu będzie pobierane ze stylu CSS, a dokładniej – z deklaracji właściwości selektora **h1**. Efektem działania tego selektora będzie wyświetlenie tekstu w nagłówku pierwszego stopnia w kolorze białym z wyrównaniem do środka strony.



Rys. 9.3. Przykład zastosowania CSS

3. Metoda łączenia z plikiem zewnętrznym (styl zewnętrzny). Polega na utworzeniu oddzielnego pliku, który zawiera listę reguł. Plik zapisujemy z rozszerzeniem **.css**, a następnie definiujemy połączenie z nim za pomocą znacznika **<link>** w nagłówku dokumentu HTML.

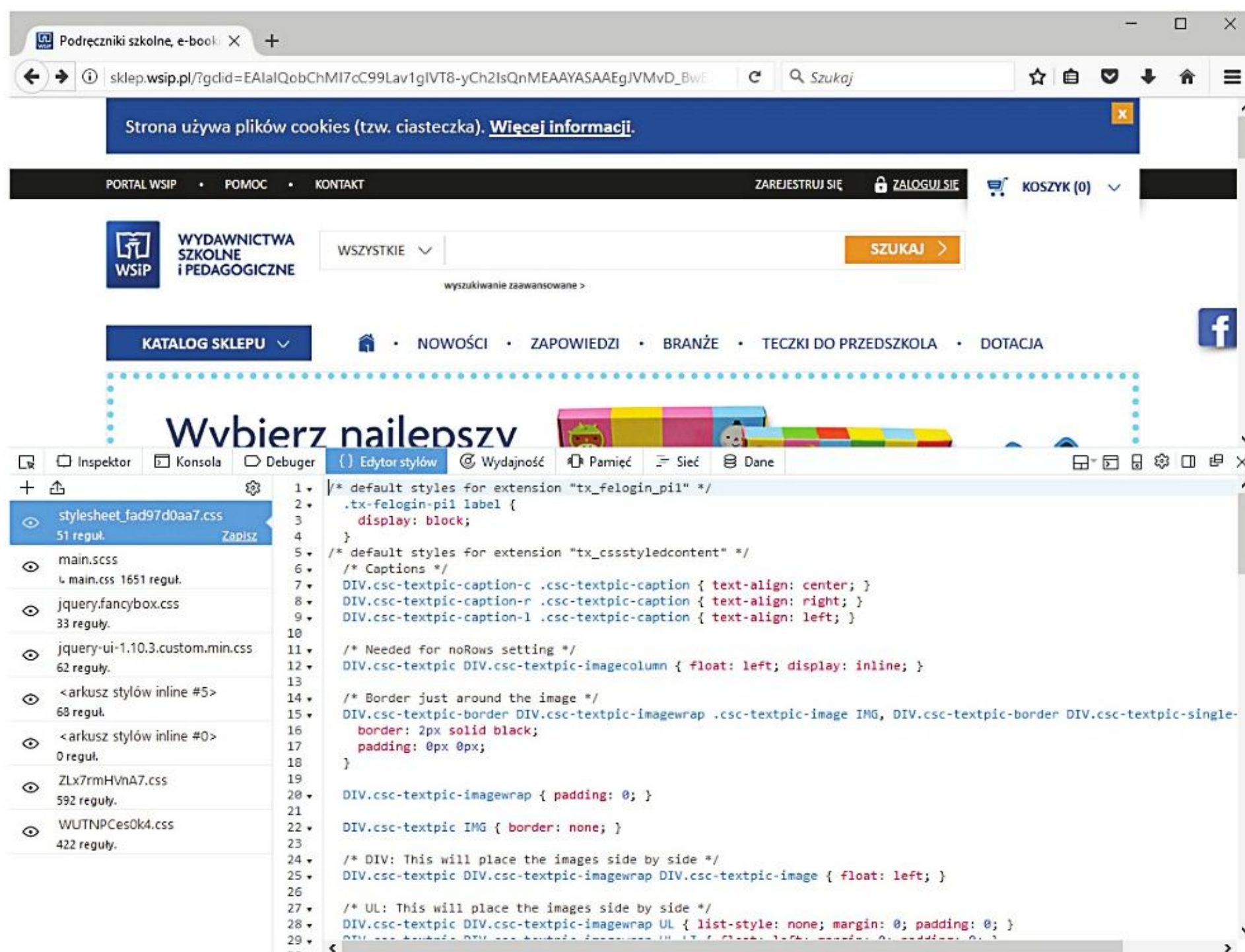
Ta metoda pozwala na definiowanie arkuszy stylów dla wielu dokumentów HTML i jest najpopularniejszym oraz najbardziej efektywnym sposobem zbiorczego formatowania wielostronicowych witryn internetowych i systemów zarządzania treścią (CMS).

Gdybyśmy w przykładzie 9.4 dopisali link do pliku css, to wszystkie cechy formatowania w nim zamieszczone byłyby zastosowane po użyciu stylów w linii oraz wewnętrznych. Na samym końcu zostają zastosowane style określone w danej przeglądarce.

Aby powiązać plik CSS z plikiem HTML, do nagłówka strony należy wpisać następujący kod:

```
<link rel="stylesheet" type="text/css" href="nazwa.css">
```

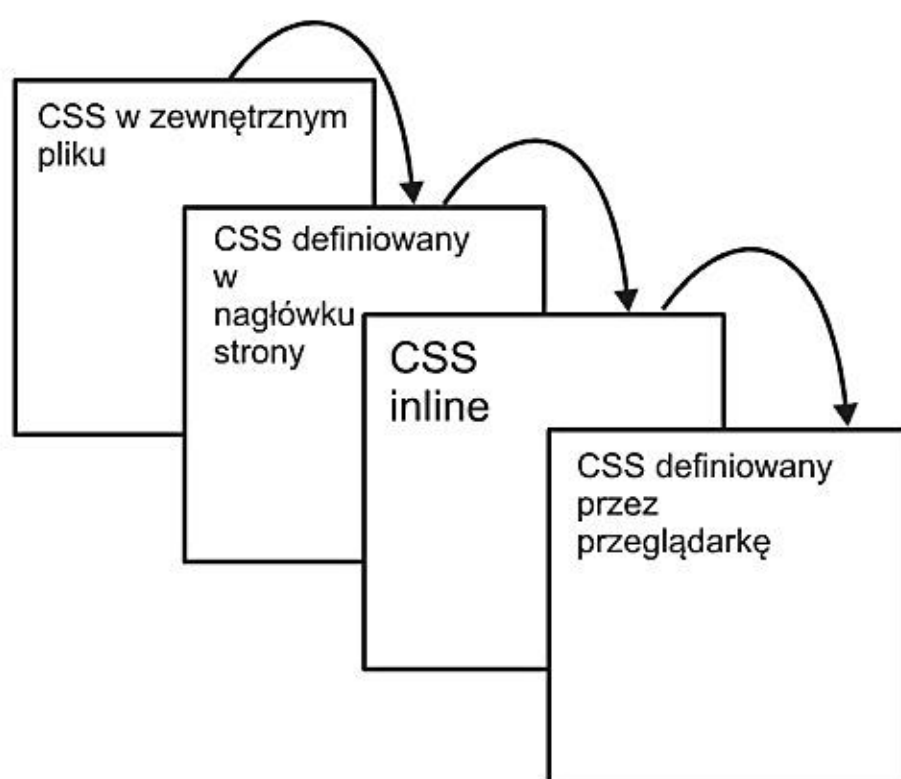
Oprócz tego każda z przeglądarek ma dodatkowo własne wbudowane arkusze stylów. Ponadto każdy użytkownik może dopisać własne style za pomocą narzędzia **inspektor**, wbudowanego w przeglądarkę.



Rys. 9.4. Widok narzędzia Inspektor w przeglądarce Firefox

Kaskadowość arkuszy stylów

Kaskadowość arkuszy stylów określa kolejność oddziaływania stylu zależnie od metody dołączenia jego kodu do formatowanego dokumentu HTML. Dlatego style o wyższym priorytecie ważności mają pierwszeństwo w modyfikowaniu elementów dokumentu HTML. Style o niższym priorytecie mogą zmienić formatowanie jakiegoś elementu tylko wtedy, jeżeli style o wyższym priorytecie nie definiują formatowania tego elementu. Dzięki temu można pracować z różnymi metodami tworzenia arkuszy stylów bez obawy, że będzie to powodować błędy formatowania elementów na stronie.



Rys. 9.5. Kaskadowość arkuszy stylów

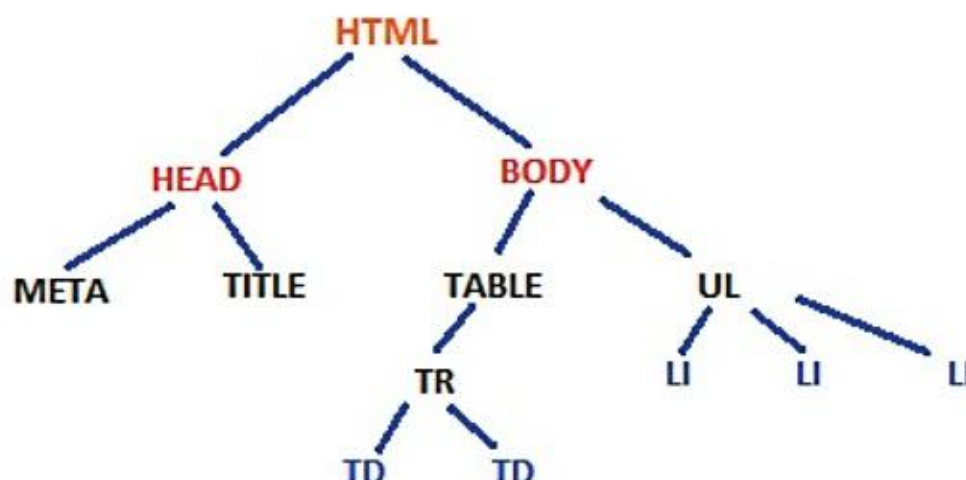
Komentarze w dokumentach CSS

Podobnie jak w dokumentach HTML, kaskadowe arkusze stylów można komentować za pomocą odpowiedniej składni kodu:

```
/* komentarz do arkusza CSS */
```

Dziedziczenie

Struktura każdego dokumentu hipertekstowego opiera się na hierarchicznej budowie **drzewa dokumentu**. Można to porównać do zależności występujących w hierarchii społecznej. Każdy element w drzewie dokumentu ma dokładnie **jednego rodzica**, oprócz elementu głównego, czyli korzenia (ang. *root*) drzewa, którym jest zawsze element **<HTML>** (rys. 9.6).



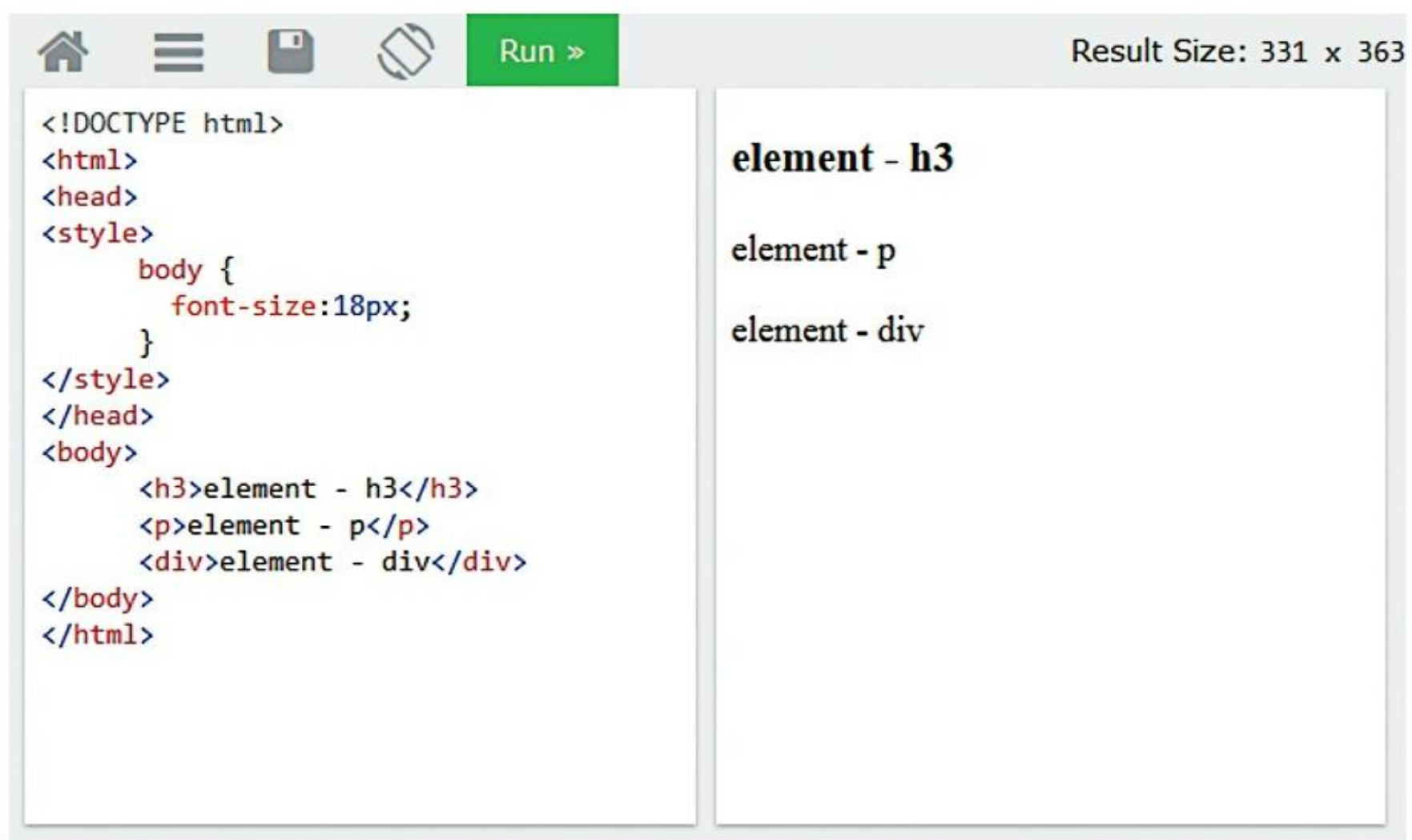
Rys. 9.6. Drzewo dokumentu HTML

Na rys. 9.6 jest widoczna struktura: przodkowie (rodzice, dziadkowie), bracia i potomkowie (dzieci). Pozycja w danej hierarchii decyduje również o **dziedziczeniu** pewnych cech (np. wyglądu) od elementów znajdujących się wyżej oraz o przekazywaniu cech elementom znajdującym się niżej w hierarchii. Hierarchię ustala się zawsze na podstawie miejsca, w którym znajduje się dany element dokumentu. Za prezentację dokumentu hipertekstowego odpowiadają kaskadowe arkusze stylów – dlatego elementy HTML mogą dziedziczyć również właściwości CSS od przodka lub przekazywać je swoim potomkom.

Należy jednak zwrócić uwagę na to, że nie wszystkie właściwości CSS są dziedziczone automatycznie. Niektóre elementy HTML mają domyślne właściwości (np. `<h4>`) lub nie obsługują wszystkich właściwości. W takim przypadku nie będą przyjmować formatowania od przodka.

PRZYKŁAD 9.5

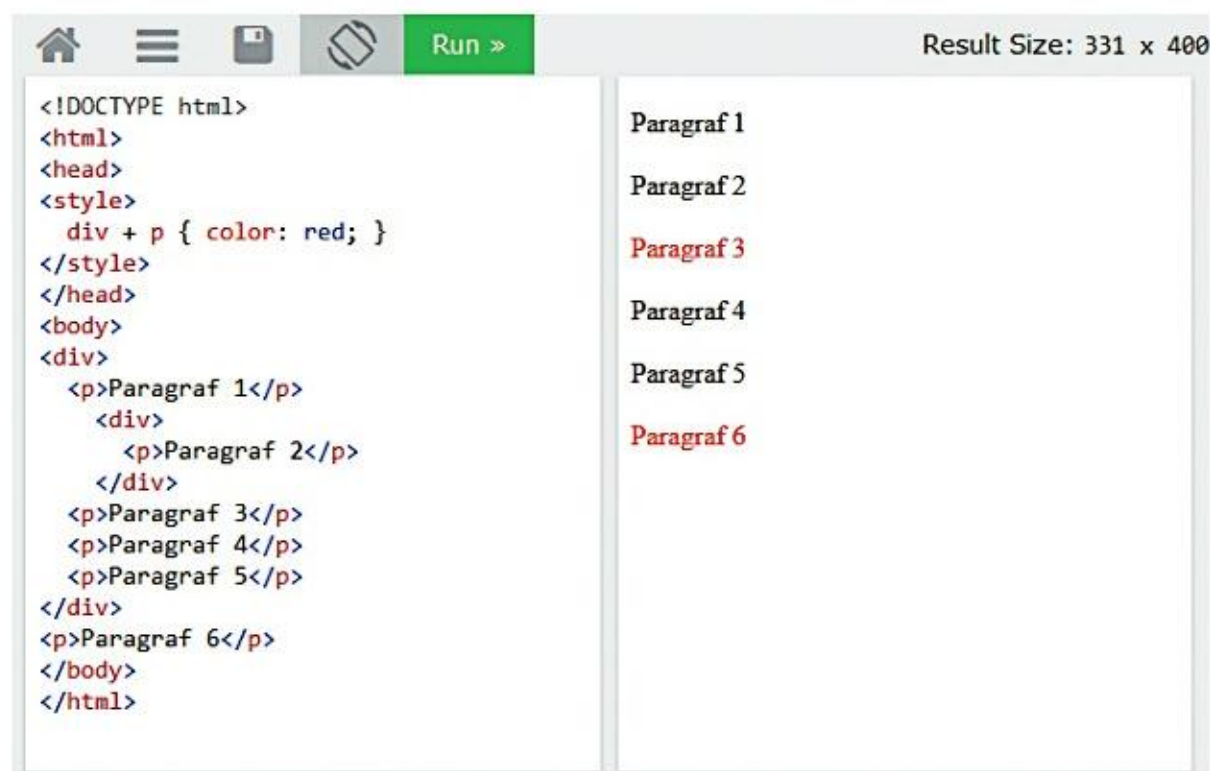
Dla elementów wyświetlanych w oknie strony internetowej rodzicem jest element `<body>`. Dlatego nadamy mu określone właściwości formatowania i zobaczymy, czy wszystkie elementy wewnątrz znacznika `<body>` odziedziczą te same cechy.



Rys. 9.7. Przykład dziedziczenia stylów CSS

Element `<h3>` nie odziedziczył formatu z rodzica (elementu `<body>`), ponieważ ma swój nadrzędny styl formatowania.

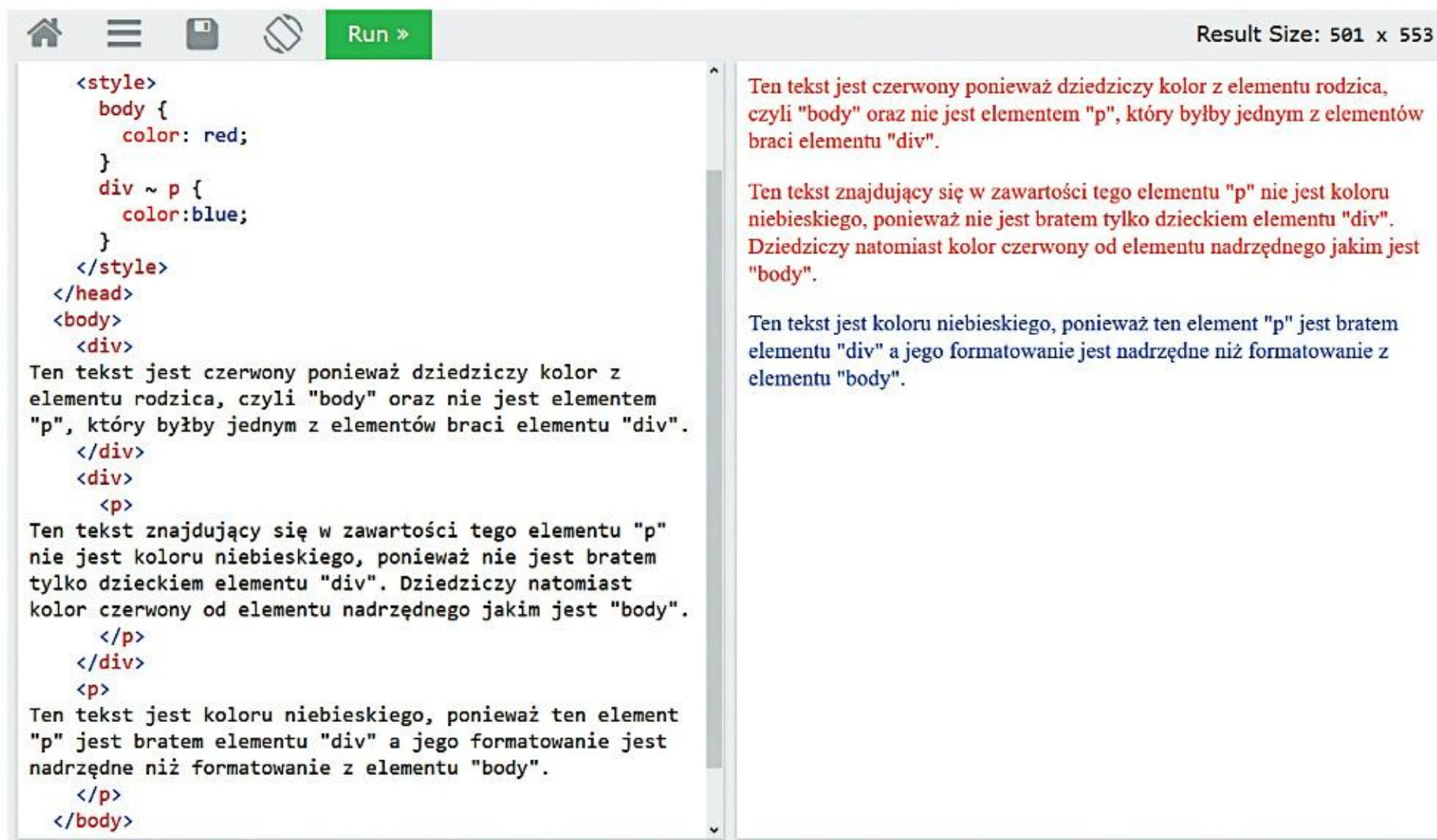
Dziedzicznie można łatwo zaobserwować w strukturze blokowej. Na rys. 9.8 pokazano przykład, jak można programować elementy blokowe, opierając się na tej przydatniej metodzie przejmowania cech w hierarchii drzewa dokumentu, jaką jest dziedziczenie.



Rys. 9.8. Przykład dziedziczenia stylów CSS przez elementy blokowe

Tego typu selektorów używa się często podczas tworzenia stylów opartych na zależnościach związanych z dziedziczeniem i hierarchią drzewa.

Innym przykładem selektora braci (braci – ponieważ leżą w tym samym rzędzie hierarchii) jest tylda. Dzięki temu symbolowi można nadać określone atrybuty wszystkim braciom, których nazwę w deklaracji podano jako drugą. Zwykły tekst znajdujący się między braćmi nie ma wpływu na działanie selektora, czyli jest ignorowany (rys. 9.9).



Rys. 9.9. Przykład zastosowania symbolu tylda

Podczas tworzenia tego stylu użyto selektora „+”. Spowodował on, że będą formatowane tylko elementy blokowe znajdujące się bezpośrednio w elemencie **<div>**. Dlatego efekt dziedziczenia ograniczono do paragrafów 3 i 6.

Każdy element HTML może dziedziczyć właściwości CSS od swojego „przodka”. Należy jednak pamiętać, że nie wszystkie właściwości CSS mogą być dziedziczone.

Właściwości, które nie są dziedziczone:

- **background,**
- **border,**
- **margin,**
- **padding,**
- **width,**
- **height.**

SPRAWDŹ SWOJĄ WIEDZĘ

1. Do czego służy selektor?
2. Omów metody dołączania do dokumentu HTML.
3. Na czym polega kaskadowość stylów CSS?
4. W jaki sposób wstawia się komentarze w CSS?

10

Selektory, właściwości oraz wartości w CSS

ZAGADNIENIA

- Rodzaje selektorów
- Podstawowe właściwości i ich wartości
- Przykłady zastosowań właściwości CSS

Selektory

Selektory CSS mogą się odnosić do elementów HTML, takich jak: konkretne znaczniki, identyfikatory (id), klasy (class) oraz inne atrybuty.

Jeżeli selektorem jest dowolny znacznik języka HTML, oznacza to, że mamy do czynienia z tzw. **selektorem prostym**. Formatowanie znaczników polega na definiowaniu selektorów za pomocą ich nazw.

PRZYKŁAD 10.1

```
h2 {  
    color: red;  
}
```

Można również grupować selektory znaczników, które mają być identycznie formatowane.

PRZYKŁAD 10.2

```
h1, p, i {  
    color: red;  
}
```

Do identyfikacji poszczególnych elementów w języku HTML używa się identyfikatorów, zapisywanych skrótem id. Często stosuje się je zwłaszcza w definiowaniu elementów blokowych **<div>**. Każdy identyfikator ma unikalną nazwę w obszarze jednego dokumentu. Tę nazwę można wykorzystać jako selektor do formatowania elementu, który ona identyfikuje. Należy pamiętać, że zapis selektora identyfikatora jest zawsze poprzedzony znakiem #.

PRZYKŁAD 10.3

Dla elementu `<div id="stopka"> </div>` definiowanie formatowania w arkuszu stylów CSS wygląda następująco:

```
#stopka {  
    text-align: center;  
    color: green;  
}
```

Inną metodą definiowania selektora jest użycie klasy, która również służy do rozróżniania elementów języka HTML. Klasy definiuje się atrybutem **class**. Selektor klasy odszukuje każdy element w dokumencie HTML, który ma ten atrybut o takiej samej nazwie. Ponieważ atrybutu klasy można używać wielokrotnie w obrębie jednego dokumentu, to dzięki wykorzystaniu go jako selektora jesteśmy w stanie formatować wiele elementów jednocześnie. Aby użyć atrybutu class jako selektora, należy przed jego nazwą postawić znak „.”.

PRZYKŁAD 10.4

Dla elementu `<div class="stopka"> </div>` definiowanie formatowania w arkuszu stylów CSS wygląda następująco:

```
.stopka {  
    text-align: center;  
    color: red;  
}
```

Jeżeli chcemy sformatować część takich samych znaczników na stronie internetowej, możemy również użyć atrybutu class. Pozwoli to na zróżnicowanie formatowania takich samych znaczników.

PRZYKŁAD 10.5

Dla elementu `<p class="inny"> Tekst </p>` definiowanie formatowania w arkuszu stylów CSS wygląda następująco:

```
p.inny {  
    color: red;  
}
```

W CSS stosuje się również **selektory uniwersalne**, oznaczone znakiem *. Mają one taką samą wagę jak definiowany styl elementu najwyższego poziomu, czyli **body**.

Hierarchiczny układ elementów HTML pozwala także na stosowanie selektorów zależnych od elementów umieszczonych wyżej w hierarchii, na tym samym poziomie lub niżej. Mówimy wtedy o **selektorach potomka** (ang. *descendant selector*), **selektorach dziecka** (ang. *child selector*) lub **selektorach braci** (ang. *adjacent sibling selector*).

PRZYKŁAD 10.6

```
ul li {color:red; font-size:20px;}
```

W tego typu zapisie wykorzystano selektor potomka. Oznacza to, że elementy wykazu na drugim poziomie zagnieżdżenia będą przedstawione w kolorze czerwonym oraz czcionką o wielkości 20 px, podczas gdy elementy wykazu pierwszego poziomu będą miały domyślne kolor i rozmiar czcionki.

Właściwości

Podczas doboru właściwości CSS pierwszą rzeczą, na jaką należy zwrócić uwagę, jest to, czy przeglądarki, na których będzie wyświetlana strona, poprawnie interpretują daną właściwość. Może się zdarzyć, że właściwości ze specyfikacji CSS 3 nie będą poprawnie interpretowane przez niektóre przeglądarki. Jest to spowodowane ciągłym rozwijaniem tej wersji specyfikacji.

Wszystkie właściwości CSS można podzielić na kilka kategorii. Każda z kategorii jest odpowiedzialna za zmianę wyglądu określonego elementu HTML. Wyróżnia się następujące kategorie właściwości CSS:

- tekst – właściwości odpowiedzialne za formatowanie cech wyglądu tekstu;
- czcionka – właściwości odpowiedzialne za formatowanie wyglądu czcionki tekstu;
- skalowanie – właściwości odpowiedzialne za formatowanie szerokości i wysokości elementu;
- tło – właściwości odpowiedzialne za formatowanie cech wyglądu tła w danym elemencie;
- obramowanie – właściwości odpowiedzialne za formatowanie cech wyglądu obramowania danego elementu;
- wykazy – właściwości formatujące listy;
- tabela – właściwości formatujące cechy komórek tabeli.

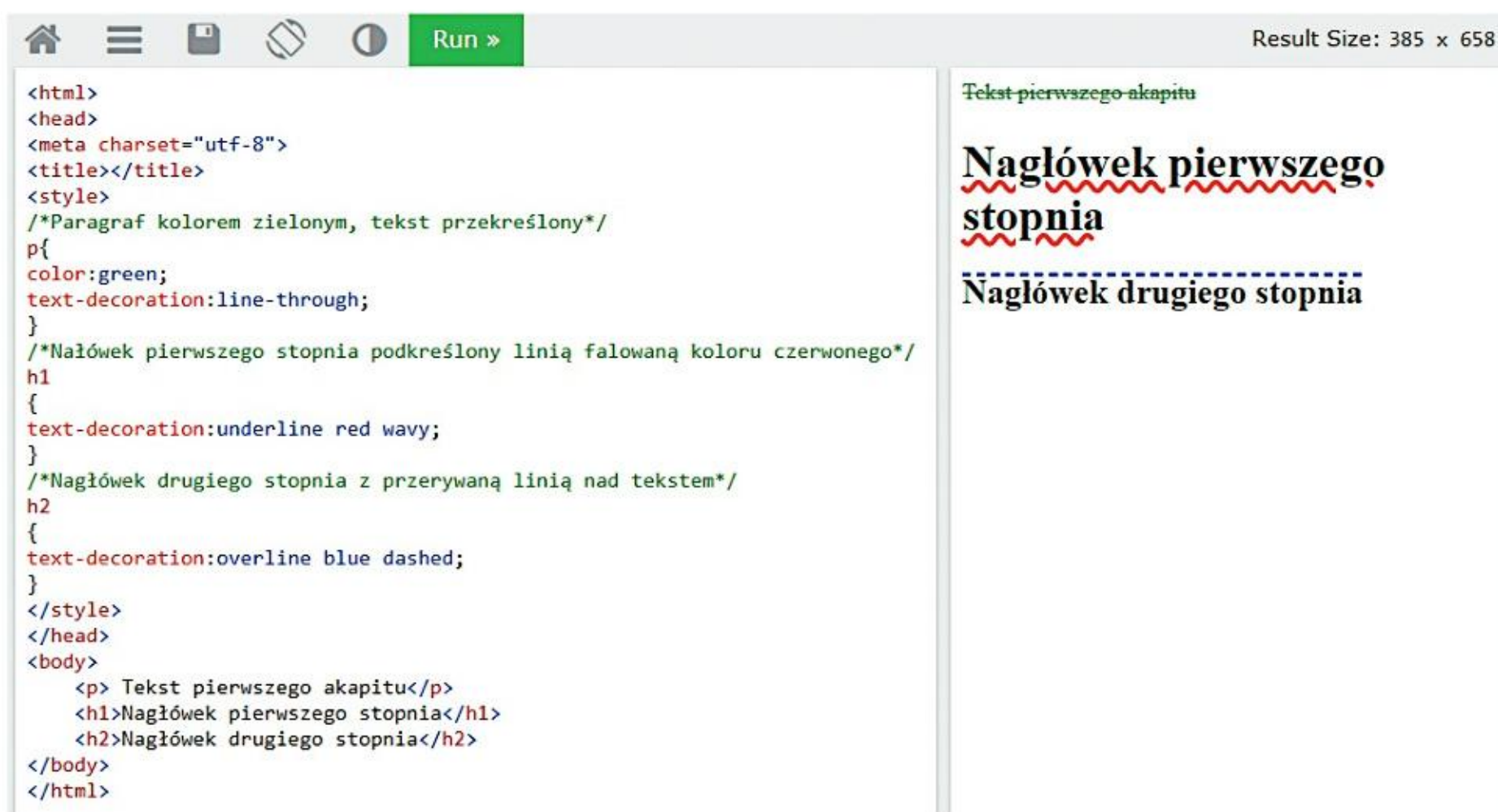
Ze względu na dużą liczbę właściwości możliwych do wykorzystania zostaną opisane tylko niektóre z nich.

Formatowanie tekstu w CSS

Do często używanych właściwości formatujących tekst zalicza się:

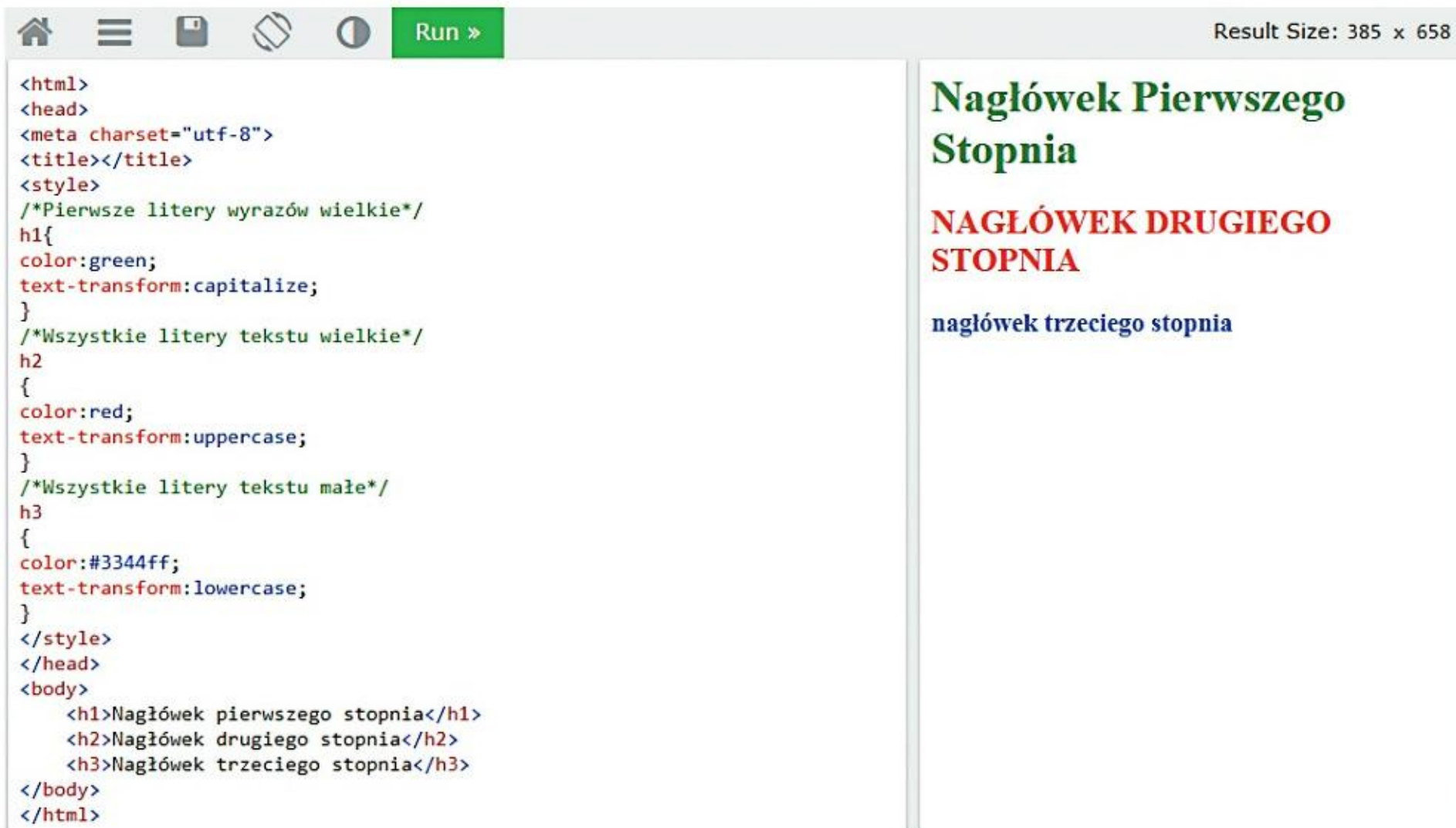
- **color** – pozwala na określenie koloru danego elementu; kolor czcionki określa się za pomocą nazwy koloru w język angielskim, kodu szesnastkowego lub rgb;
- **text-align** – definiuje wyrównanie tekstu; domyślnie tekst jest wyrównany do lewej krawędzi strony; pozostałe wartości tej właściwości to:
 - **right** – wyrównanie do prawej,
 - **center** – wyśrodkowanie tekstu,
 - **justify** – wyjustowanie tekstu;

- **text-decoration** – pozwala na dekorację (wyróżnienie) fragmentów tekstu; ta właściwość może przyjmować następujące wartości:
 - **none** – tekst bez dekoracji, wartość domyślna;
 - **blink** – migotanie tekstu (nie działa w Internet Explorer);
 - **underline** – podkreślenie tekstu;
 - **line-through** – przekreślenie tekstu;
 - **overline** – linia nad tekstem.
 Ponadto text-decoration ten może mieć przypisany kolor bądź rodzaj podkreślenia. Domyślnym rodzajem podkreślenia jest solid (linia pojedyncza ciągła), pozostałe to:
 - **dashed** – linia przerywana,
 - **dotted** – linia kropkowana,
 - **wavy** – linia falowana,
 - **double** – linia podwójna ciągła;

PRZYKŁAD 10.7**Rys. 10.1.** Zastosowanie właściwości **text-decoration**

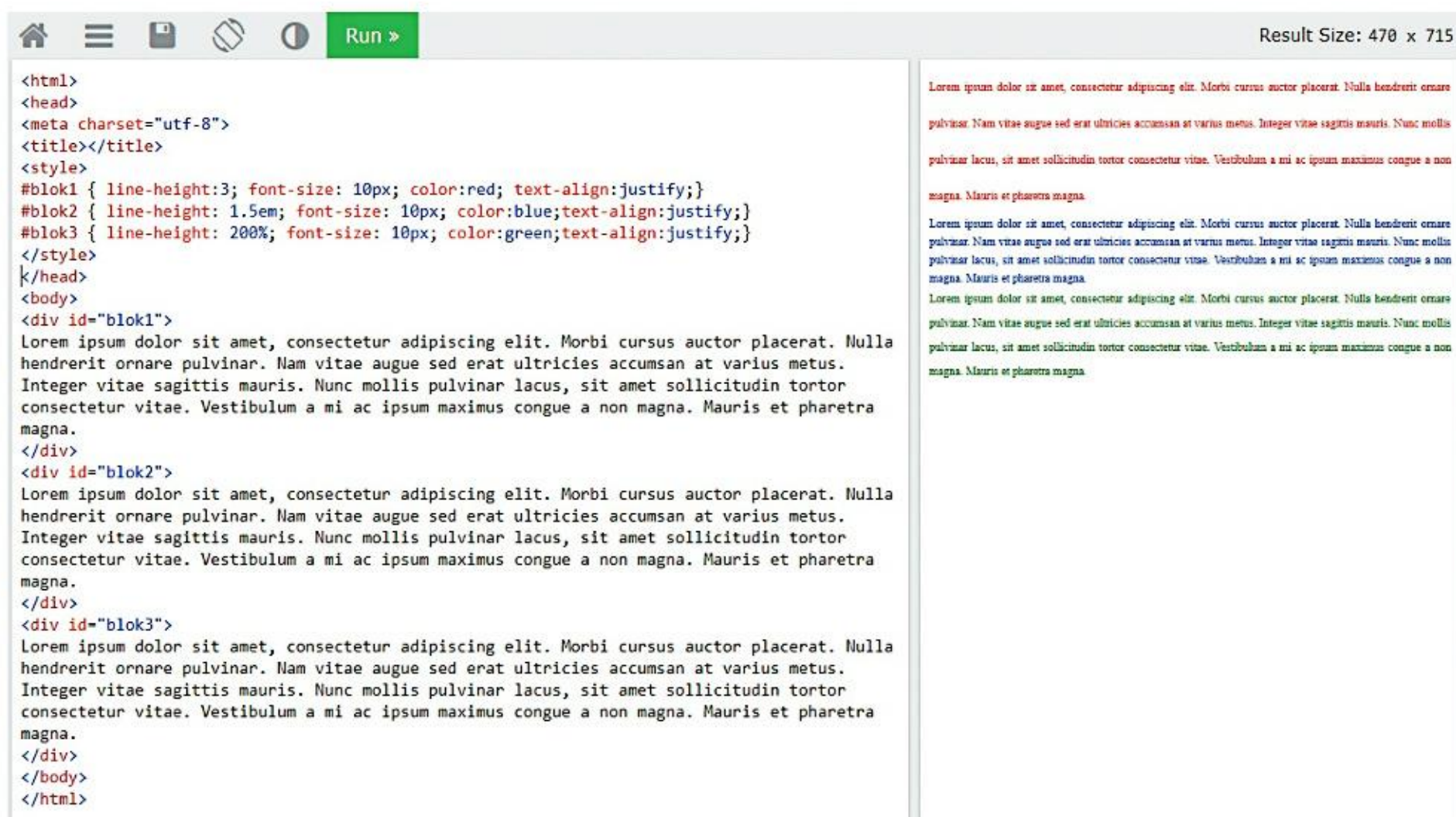
- **text-transform** – pozwala określić wielkości liter w tekście; ta właściwość może przyjmować następujące wartości:
 - **none** – wartość domyślna, tekst pozostanie niezmienny;
 - **capitalize** – wszystkie pierwsze litery wszystkich wyrazów tekstu zostaną zamienione na wielkie;
 - **lowercase** – wszystkie litery tekstu będą zamienione na małe litery;
 - **uppercase** – wszystkie litery tekstu będą zmienione na wielkie litery;

PRZYKŁAD 10.8

Rys. 10.2. Zastosowanie właściwości **text-transform**

- **direction** – określająca, od której krawędzi danego obszaru powinno się rozpocząć układanie wyrazów tekstu; ta właściwość może przyjmować następujące wartości:
 - **ltr** – tekst od lewej krawędzi obszaru do prawej; wartość domyślna;
 - **rtl** – tekst od prawej krawędzi obszaru do lewej;
- **line-height** – określa wysokość wierszy; właściwość ta może przyjmować następujące wartości:
 - **normal** – wartość domyślna;
 - **liczba** – odległość wyrażona liczbą względem rozmiaru czcionki;
 - **em** – odległość wyrażona w em względem rozmiaru czcionki;
 - **%** – odległość wyrażona w % względem rozmiaru czcionki;
 - **inherit** – wartość dziedziczona po rodzicu;

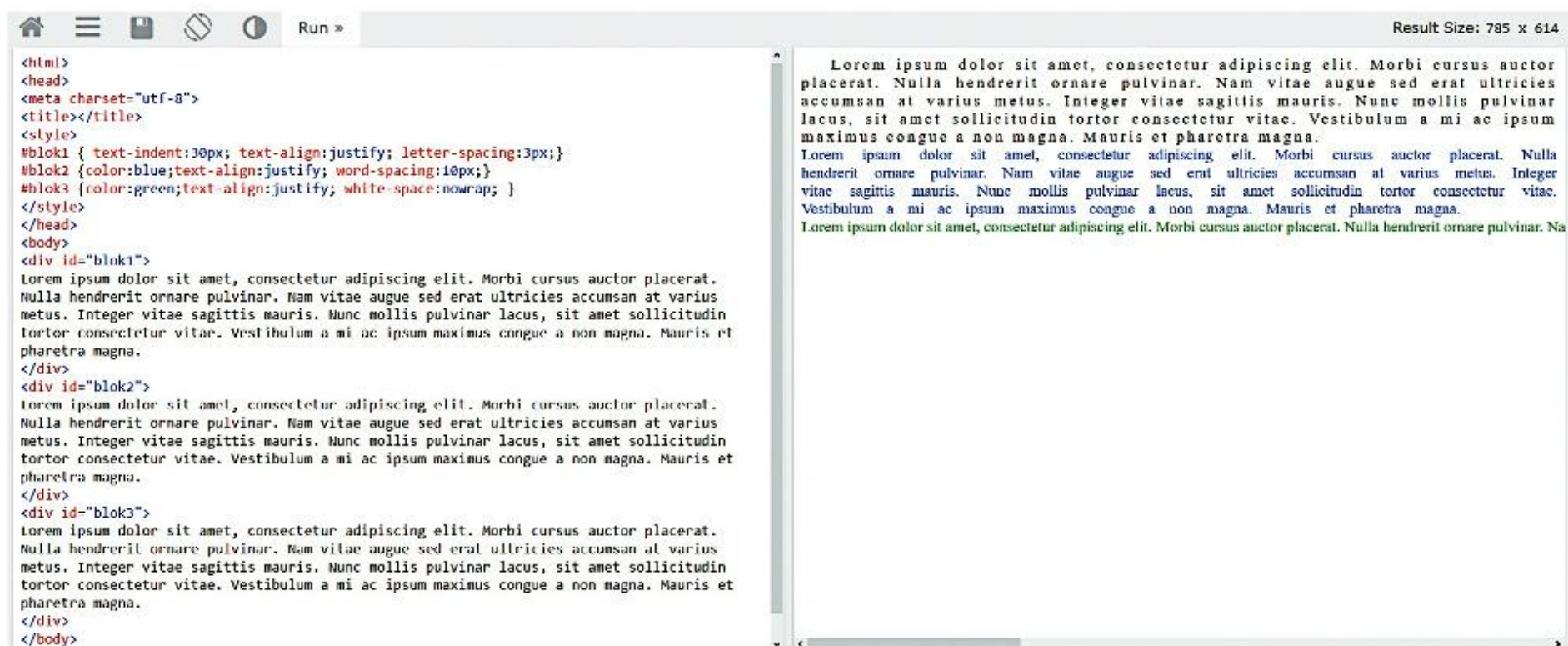
PRZYKŁAD 10.9

Rys. 10.3. Zastosowanie właściwości **line-height**

- **text-indent** – określa wielkość wcięcia w pierwszym wierszu; wartości podaje się w pikselach, w procentach lub jednostkach em;
- **letter-spacing** – określa odległość między literami;
- **word-spacing** – określa odległość między słowami;
- **white-space** – pozwala określić sposób traktowania przez przeglądarkę internetową białych znaków (spacja, tabulacja, znak przejścia do nowej linii – ENTER) w tekście; ta właściwość może przyjmować następujące wartości:
 - **normal** – wartość domyślna, kilka białych znaków nie może być wyświetlonych obok siebie, znak przejścia do nowej linii jest ignorowany, tekst jest automatycznie zawijany;
 - **pre** – kilka białych znaków może zostać wyświetlonych obok siebie, znak przejścia do nowego wiersza nie jest ignorowany, tekst w danym obszarze nie jest automatycznie zawijany;
 - **nowrap** – podobnie jak normal, ale tekst nie jest automatycznie zawijany;
 - **pre-wrap** – podobnie jak pre, ale tekst jest automatycznie zawijany;
- **text-overflow** – określa sposób wyświetlenia skróconego tekstu.

PRZYKŁAD 10.10

Poniżej przedstawiamy zastosowanie atrybutów **letter-spacing**, **word-spacing** i **white-space**. W pierwszych dwóch blokach **white-space** ma wartość domyślną, w związku z tym tekst został wyjustowany. W ostatnim bloku dla tego atrybutu została nadana wartość **nowrap**, co spowodowało brak zawijania tekstu.



Rys. 10.4. Przykład zastosowania odstępów między literami i słowami

Czcionka

Do najczęściej używanych właściwości formatujących tekst zalicza się:

- **font-style** – służy do określenia dodatkowego stylu czcionki wybranego tekstu; własność ta może przyjmować następujące wartości:
 - **normal** – wartość domyślna;
 - **italic** – tekst pisany kursywą;
 - **oblique** – tekst pisany czcionką ukośną.

Czcionka **italic** jest osobnym krojem czcionki, natomiast **oblique** może zostać wygenerowana przez pochylenie zwykłej czcionki;

- **font-family** – służy do określenia rodzaju czcionki danego tekstu; tę własność określa się według wzorca:

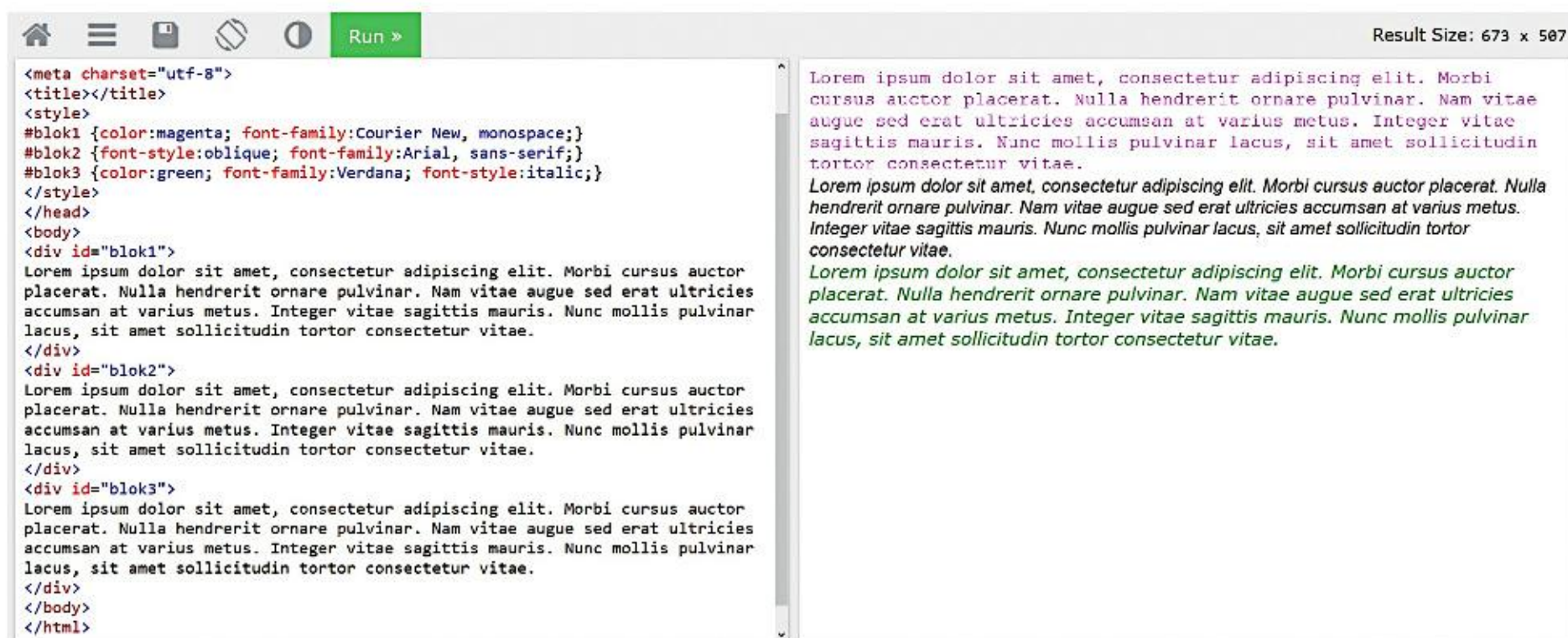
font-family: nazwa, generic;

Element **nazwa** określa nazwę rodzaju czcionki, którą chcemy przypisać do określonego obszaru. Element **generic** określa nazwę rodzaju ogólnego czcionki, która powinna być przypisana, gdy żadna z przypisanych przez nas czcionek nie będzie dostępna dla danego systemu operacyjnego.

Dostępne rodzaje ogólne czcionki: **serif**, **sans-serif**, **cursive** (czcionki pisane kursywą), **fantasy** (czcionki dekoracyjne), **monospace**;

PRZYKŁAD 10.11

W przykładzie utworzono trzy bloki, w których zastosowano równe rodzaje czcionek. Rodzaj ogólny **monospace** zawiera czcionkę Courier New. Są to czcionki, której wszystkie znaki mają taki sam rozmiar szerokości. Znaki czcionek z rodzaju **sans-serif** nie mają dodatkowych zakończeń. Należą do nich takie czcionki, jak: Arial, Verdana, Tahoma.

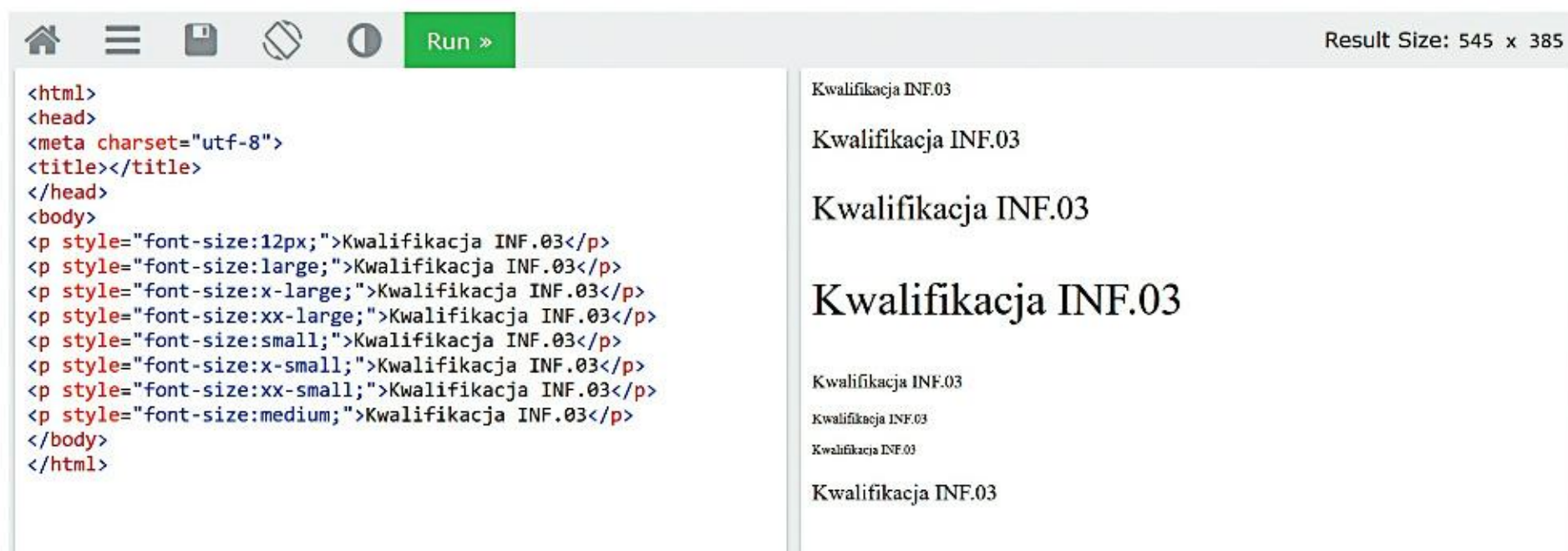


Rys. 10.5. Przykład zastosowania **font-style** oraz **font-family**

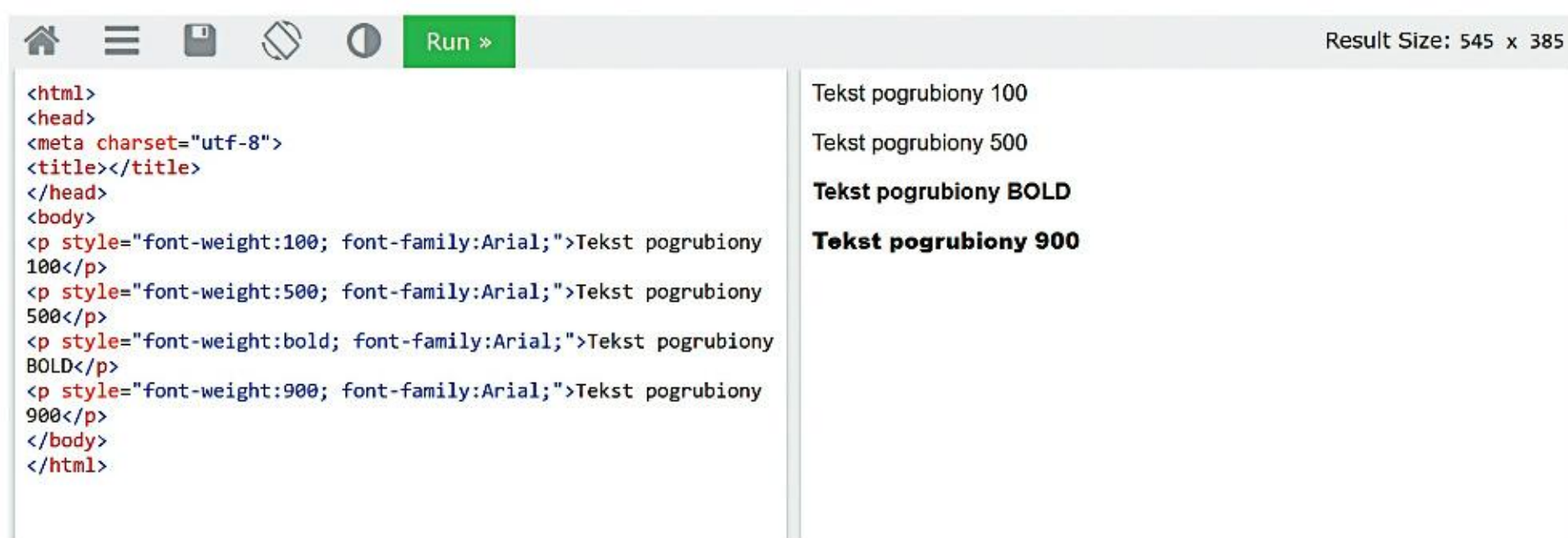
- **font-size** – służy do określenia rozmiaru czcionki; własność ta może przyjmować następujące wartości:
 - liczba podana za pomocą pikseli, procentów bądź jednostki em
 - **xx-small**
 - **x-small**
 - **small**
 - **medium**
 - **large**
 - **x-large**
 - **xx-large**
 - **smaller**
 - **larger**.

PRZYKŁAD 10.12

Poniższy przykład obrazuje zastosowanie poszczególnych rozmiarów czcionki. Pierwszy paragraf napisano czcionką 12 px. Rozmiar można również ustalić procentowo. Drugi akapit napisano dużą czcionką, trzeci – czcionką bardzo dużą, a czwarty akapit ma określony rozmiar jako bardzo bardzo duży. Analogiczne kolejne akapity mają ustalony rozmiar mały, bardzo mały i bardzo bardzo mały. Ostatni z akapitów napisano czcionką o średnim rozmiarze.

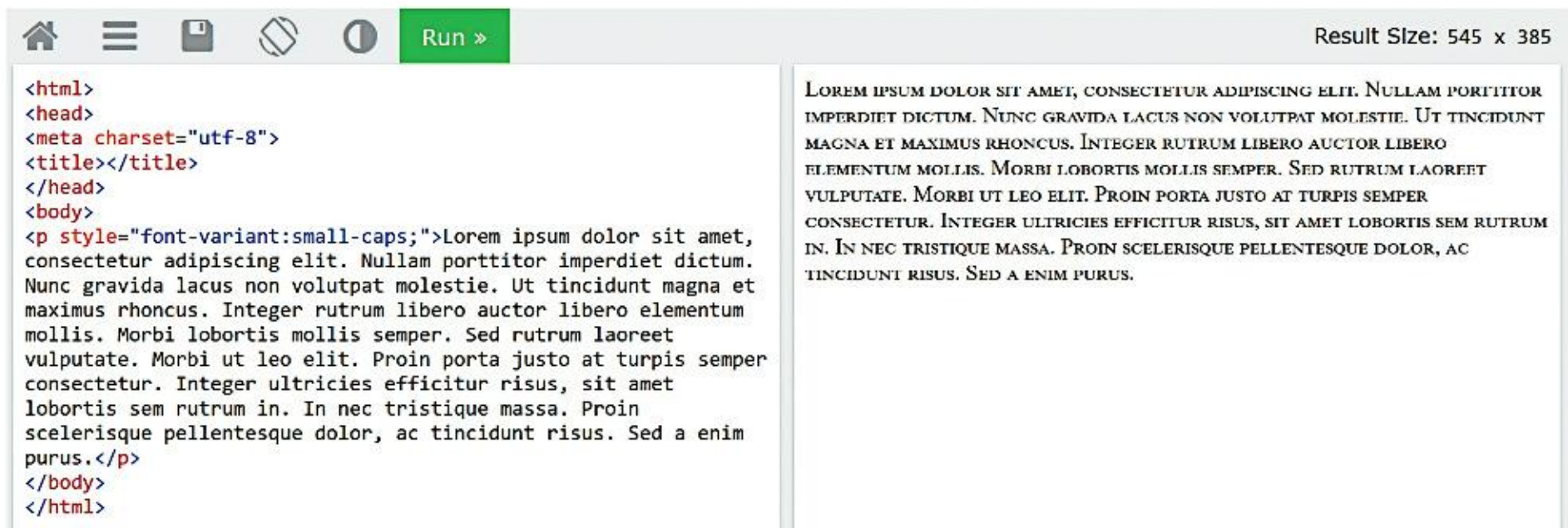
Rys. 10.6. Zastosowanie własności **font-size**

- **font-weight** – służy do określenia grubości danego tekstu; ta własność może przyjmować określone wartości:
 - **normal** – wartość domyślna;
 - **bold** – czcionka danego tekstu zostanie pogrubiona (jest równoważna wartości 700);
 - **lighter** – jeśli jest to możliwe, to czcionka danego tekstu będzie mieć mniejszą grubość niż grubość czcionki dziedziczonej od rodzica;
 - **bolder** – jeśli jest to możliwe, to czcionka danego tekstu będzie mieć większą grubość niż grubość czcionki dziedziczonej od rodzica;
 - stopień grubości: 100, 200, 300, 400, 500, 600, 700, 800, 900;

PRZYKŁAD 10.13Rys. 10.7. Zastosowanie własności **font-weight**

- **font-variant** – służy do określenia dodatkowego wariantu czcionki danego tekstu; ta własność może przyjmować określone wartości:
 - **normal** – wartość domyślna;
 - **small-caps** – małe litery tekstu są zamieniane na duże litery, ale w małym rozmiarze.

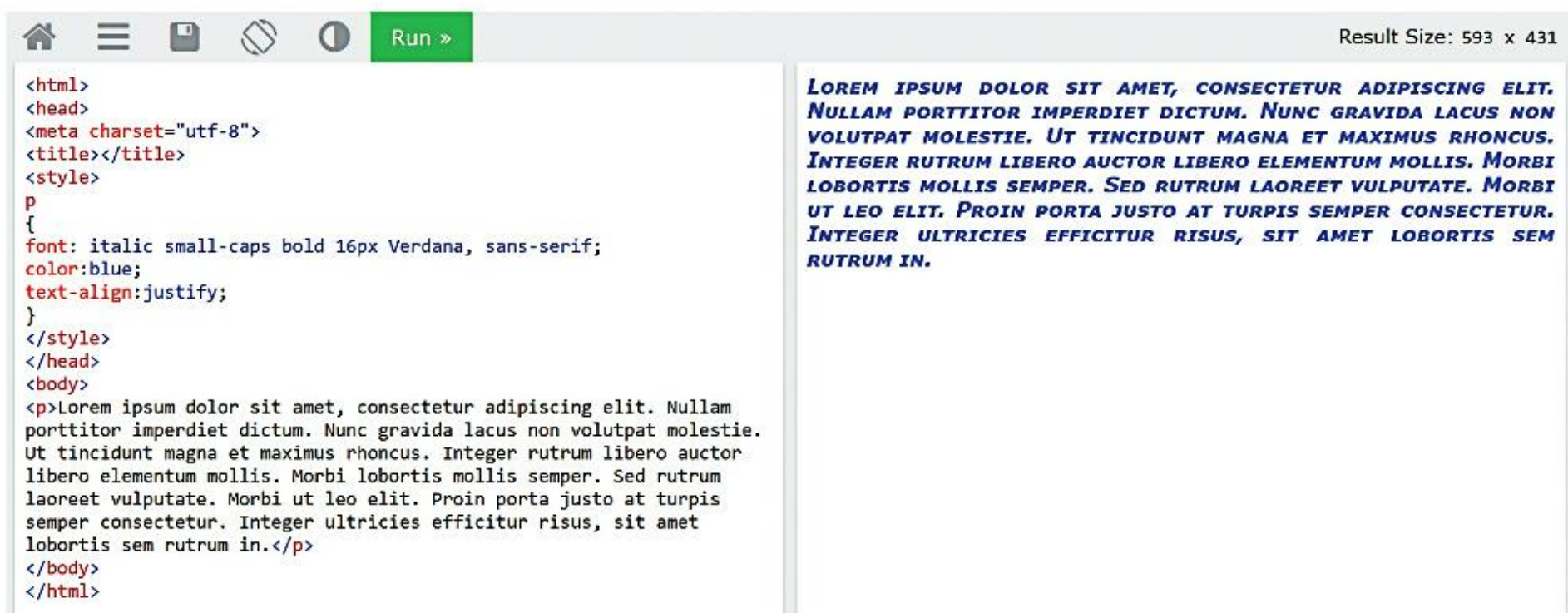
PRZYKŁAD 10.14

Rys 10.8. Zastosowanie własności **font-variant**

! UWAGA

Za pomocą właściwości font można określić kilka różnych cech czcionki wybranego tekstu.

PRZYKŁAD 10.15

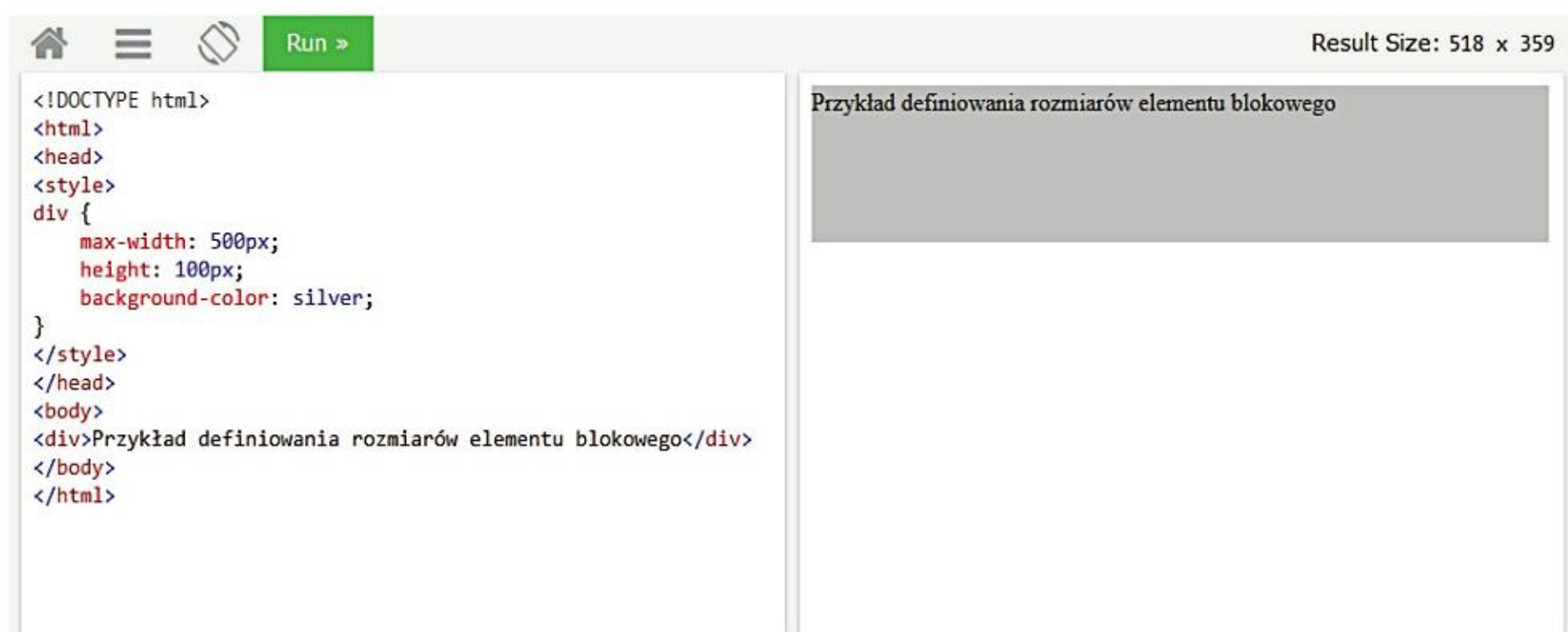
Rys. 10.9. Zastosowanie kilku cech właściwości **font**

Formatowanie rozmiaru

Do tej grupy zalicza się właściwości określające szerokość i wysokość danego elementu. Zalicza się do nich:

- **width** – służy do określenia szerokości wybranego elementu;
- **height** – służy do określenia wysokości wybranego elementu;
- **min-width** – służy do określenia minimalnej szerokości wybranego elementu;

- **max-width** – służy do określenia maksymalnej szerokości wybranego elementu;
- **min-height** – służy do określenia minimalnej wysokości wybranego elementu;
- **max-height** – służy do określenia maksymalnej wysokości wybranego elementu.

PRZYKŁAD 10.16

Rys. 10.10. Przykład formatowania rozmiaru obiektu

Tło

Do najczęściej używanych właściwości formatujących tło zalicza się:

- **background-color** – służy do określenia koloru tła; kolor można określić za pomocą:
 - nazwy w języku angielskim – np. red;
 - zapisu szesnastkowego – np. #CCDD33;
 - zapisu kolorów w rgb – np. rgb (12, 23, 45);
- **background-image** – służy do określenia grafiki jako tła wybranego elementu; ta własność może przyjmować następujące wartości:
 - **none** – wartość domyślna, brak grafiki w tle;
 - **url ()** – służy do określenia nazwy pliku, którego ma zostać utworzone tło;
 - **linear-gradient ()** – służy do określenia rodzaj gradientu liniowego, który będzie tłem obrazkowym;
 - **repeating-linear-gradient ()** – służy do określenia powtarzającego się gradientu liniowego tła obrazkowego;
 - **radial-gradient ()** – służy do określenia gradientu promienistego tła obrazkowego;
 - **repeating-radial-gradient ()** – służy do określenia powtarzającego się gradientu promienistego tła obrazkowego;

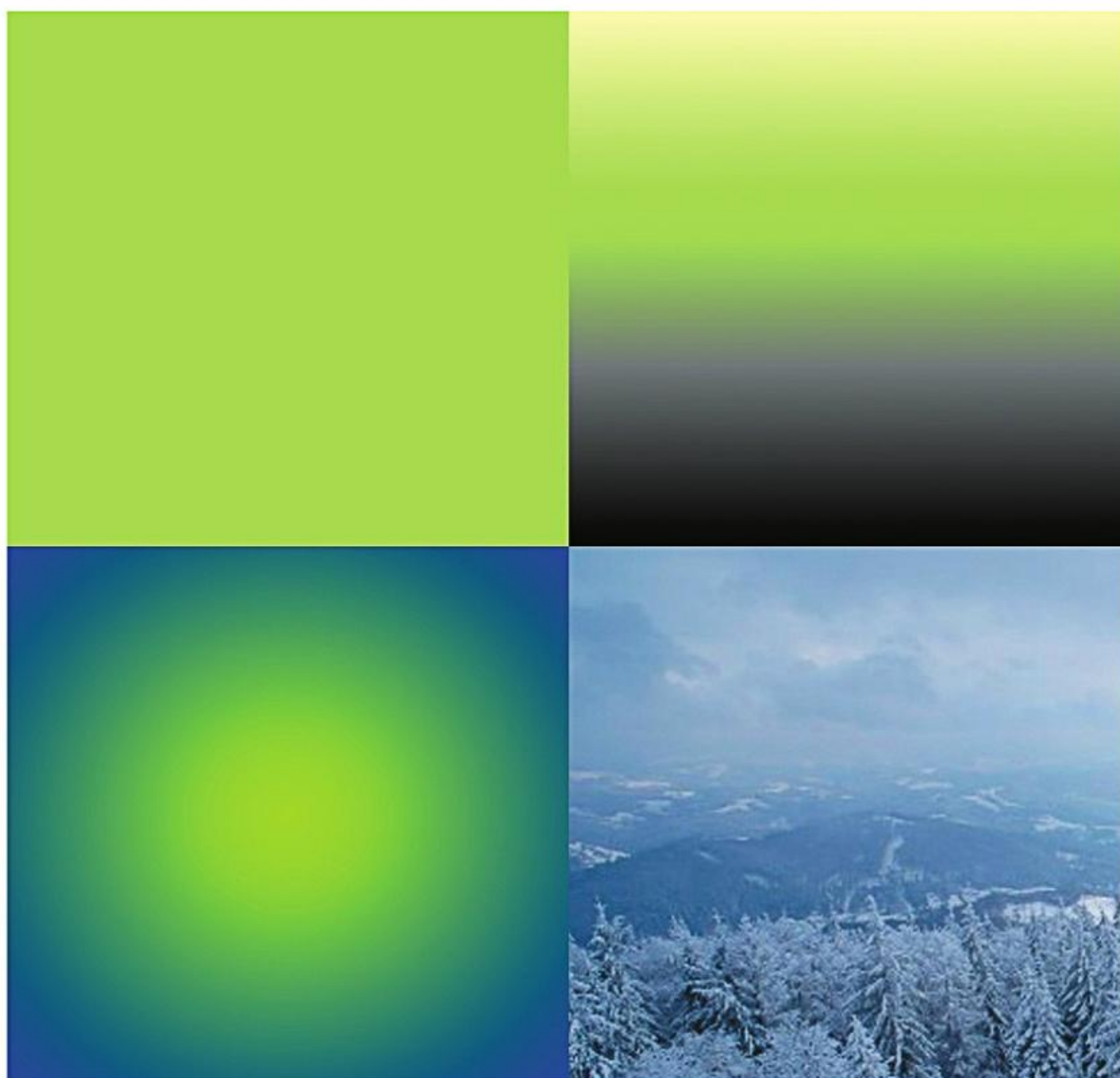
PRZYKŁAD 10.17

Poniżej przedstawiono zastosowanie właściwości **background-color** oraz **background-image**. Pierwsza z nich, zastosowana dla bloku nr 1, pozwala na określenie jednolitego koloru. W tym przypadku jest to kolor limonowy. W pozostałych trzech blokach użyto własności umożliwiającej określenie koloru gradientu liniowego (blok nr 2), promienistego (blok nr 3) oraz tła obrazkowego (blok nr 4).

```
<html>
<head>
<meta charset="utf-8">
<title></title>
<style>
#blokA, #blokB, #blokC, #blokD
{
width:50%;
height:50%;
float:left;
}
#blokA
{
background-color:#CCFF66;
}
#blokB
{
background-image:linear-gradient(#FFFFCC,#CCFF66,grey,black);
}
#blokC
{
background-image:radial-gradient(#CCFF33, #003399);
}
#blokD
{
background-image:url("zima.jpg");
}
</style>
</head>
<body>
<div id="blokA"></div>
<div id="blokB"></div>
```



```
<div id="blokC"></div>  
<div id="blokD"></div>  
</body>  
</html>
```



Rys. 10.11. Zastosowanie własności **background-color** oraz **background-image**

- **background-repeat** – służy do określenia sposobu powtarzania się tła obrazkowego; ta własność może przyjmować następujące wartości:
 - **repeat** – wartość domyślna, tło jest powtarzane zarówno w pionie, jak i w poziomie;
 - **repeat-x** – tło obrazkowe będzie powtarzane tylko w poziomie;
 - **repeat-y** – tło obrazkowe będzie powtarzane tylko w pionie;
 - **no-repeat** – tło obrazkowe nie będzie powtarzane;
 - **round** – tło obrazkowe będzie powtarzane zarówno w pionie, jak i w poziomie oraz będzie przeskalowane, tak aby wypełnić równomiernie w całości dany obszar;
 - **space** – tło obrazkowe będzie powtarzane zarówno w pionie, jak i w poziomie, zostanie rozmieszczone bez skalowania, tak aby wypełnić równomiernie w całości dany obszar;

PRZYKŁAD 10.18

Poniższy kod jest przykładem zastosowania własności **background-repeat**. Zaprezentowano efekt, gdy ta własność przyjmuje wartości **repeat-x** oraz **round**. W bloku A tło obrazkowe wypełniono tylko w poziomie, natomiast w bloku D tło obrazkowe wypełniło cały obszar.

```
<html>
<head>
<meta charset="utf-8">
<title></title>
<style>
#blokA, #blokB, #blokC, #blokD
{
width:50%;
height:50%;
float: left;
}

#blokA
{
background-image: url("gerbery.jpg");
background-color: #eeeeee;
background-repeat:repeat-x;
}

#blokD
{
background-image: url("gerbery.jpg");
background-color: #cccccc;
background-repeat:round;
}
</style>
</head>
<body>
<div id="blokA"></div>
<div id="blokB"></div>
<div id="blokC"></div>
<div id="blokD"></div>
</body>
</html>
```




Rys. 10.12. Zastosowanie własności **background-repeat**

- **background-position** – służy do określenia pozycji początkowej tła obrazkowego względem krawędzi danego obszaru. Pozycję tła można określić za pomocą wartości podanych w pikselach, procentach lub / i zastosować jedną z nazw wartości: **right**, **left**, **bottom**, **top**;
- **background-attachment** – służy do określenia obszaru, do którego ma zostać zaczepione tło obrazkowe. Własność ta może przyjmować określone wartości:
 - **scroll** – wartość domyślna;
 - **fixed** – tło obrazka zostanie zaczepione do obszaru tła okna przeglądarki;
 - **local** – tło obrazka zostanie zaczepione do obszaru zawartości danego elementu;
 - kombinacja powyższych wartości.

! UWAGA

Za pomocą właściwości **background** można określić kilka różnych cech tła wybranego obszaru strony.

Wykazy (listy)

Do najczęściej używanych właściwości formatujących listy zalicza się:

- **list-style-type** – służy do określenia typu punktora listy; ta własność może przyjmować bardzo wiele wartości, których lista znajduje się na stronie w3schools.com; oto niektóre z nich:
 - **disc** – wartość domyślna, punktor w kształcie koła;
 - **circle** – punktor w kształcie okręgu;
 - **square** – punktor w kształcie wypełnionego kwadratu;
 - **disclosure-closed** – punktor w kształcie wypełnionego trójkąta;
 - **none** – brak punktora;
 - **decimal** – numerowanie za pomocą dziesiętnego systemu liczbowego;
 - **lower-roman** – numerowanie za pomocą małych liter rzymskiego systemu zapisywania liczb;
 - **upper-roman** – numerowanie za pomocą wielkich liter rzymskiego systemu zapisywania liczb;
 - **lower-alpha**, **lower-latin** – numerowanie za pomocą małych liter alfabetu łacińskiego;
 - **upper-alpha**, **upper-latin** – numerowanie za pomocą wielkich liter alfabetu łacińskiego;
 - **lower-greek** – numerowanie za pomocą małych liter alfabetu greckiego;
- **list-style-image** – służy do określenia typu punktora w postaci obrazka; wartością domyślną jest **none**, wtedy żaden obrazek nie tworzy punktora w wykazie; aby wstawić punktor będący grafiką, należy zastosować funkcję **url()** w następującej postaci:


```
list-style-image: url(„nazwa_pliku_z_rozszerzeniem”);
```
- **list-style-position** – służy do określenia pozycji punktora; ta własność może przyjmować następujące wartości:
 - **outside** – wartość domyślna, punktor znajduje się na zewnątrz danego obszaru względem początkowej krawędzi liniowej elementu;
 - **inside** – punktor znajduje się wewnątrz danego obszaru względem początkowej krawędzi liniowej elementu.

PRZYKŁAD 10.19

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Formatowanie wykazów</title>
<style>
#lista1
```



```
{
list-style-type: lower-greek;
list-style-position: outside;
}
#lista2
{
list-style-type: disclosure-closed;
}
#lista3
{
list-style-type: lower-latin;
list-style-position: inside;
}
</style>
</head>
<body>
<ol id="lista1">
<li>Element 1</li>
<li>Element 2</li>
<li>Element 3</li>
</ol>
<hr>
<ul id="lista2">
<li>Element 1</li>
<li>Element 2</li>
<li>Element 3</li>
</ul>
<hr>
<ol id="lista3">
<li>Element 1</li>
<li>Element 2</li>
<li>Element 3</li>
</ol>
</body>
</html>
```

Powyższy kod spowoduje wyświetlenie trzech list. Pierwsza z nich numeruje elementy według małych liter greckich, druga wstawia punktory będące wypełnionym trójkątem, natomiast ostatnia numeruje elementy według alfabetu łacińskiego. Ponadto w pierwszej liście punktory ustawiono na zewnątrz, a w trzeciej – wewnątrz obszaru względem lewej krawędzi strony.

α. Element 1
β. Element 2
γ. Element 3

▶ Element 1
▶ Element 2
▶ Element 3

a. Element 1
b. Element 2
c. Element 3

Rys. 10.13. Formatowanie listy w CSS

Obramowanie

Do najczęściej używanych właściwości formatujących obramowanie elementów zalicza się:

- **border-width** – służy do określenia grubości obramowania; ta właściwość umożliwia jednocześnie określenie grubości krawędzi: górnej, prawej, dolnej i lewej; aby określić wartość dla jednej z wybranych krawędzi, należy wybrać jedno z czterech poleceń:
 - **border-top-width** – grubość krawędzi górnej;
 - **border-right-width** – grubość krawędzi prawej;
 - **border-bottom-width** – grubość krawędzi dolnej;
 - **border-left-width** – grubość krawędzi lewej.

PRZYKŁAD 10.20

Określenie grubości dolnej i górnej krawędzi na 2 px oraz lewej i prawej na 3 px może odbywać się za pomocą poleceń:

```
border-top-width: 2px;
border-bottom-width: 2px;
border-right-width: 3px;
border-left-width: 3px;
```

Powyższe cztery polecenia można zastąpić jednym:

```
border-width: 2px 3px 2px 3px;
```

Wartości podajemy zgodnie z ruchem wskazówek zegara, czyli wartość dla krawędzi: górnej, prawej, dolnej i lewej.

Poleceniem równoważnym jest zapis:

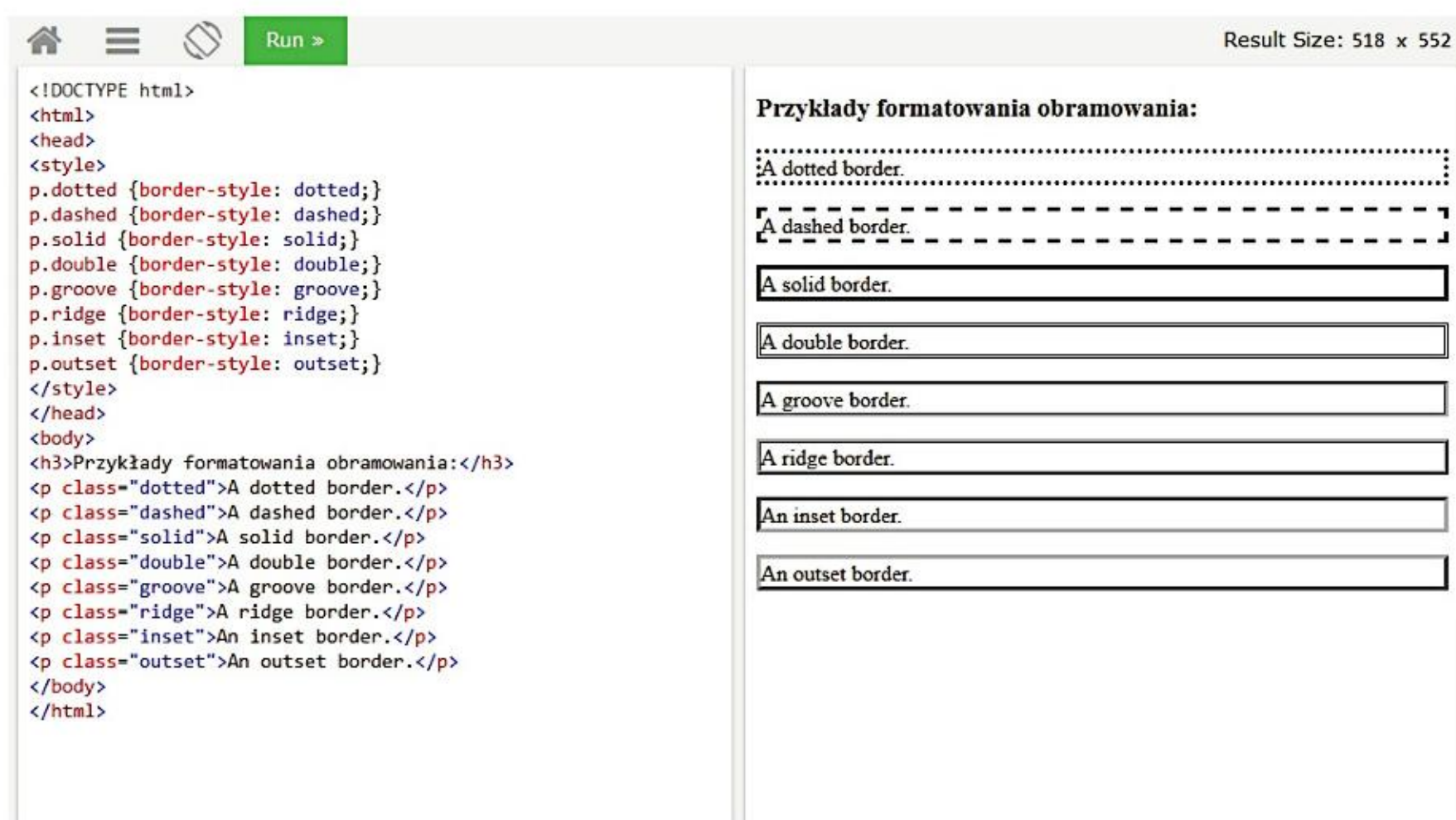
```
border-width: 2px 3px;
```

Wartość 2px odpowiada w tym przypadku krawędzi górnej oraz dolnej, natomiast 3px – krawędzi lewej i prawej.

- **border-style** – służy do określenia stylu obramowania; ta właściwość może przyjmować następujące wartości:
 - **solid** – linia ciągła pojedyncza
 - **double** – linia ciągła podwójna
 - **dotted** – linia kropkowana
 - **dashed** – linia przerywana
 - **groove** – linia z tzw. rowkiem
 - **ridge** – linia z tzw. grzbietem
 - **inset** – linia tzw. ramka
 - **outset** – linia tzw. przycisk.

Tak jak w przypadku poprzedniej własności, można analogicznymi poleceniami określać styl krawędzi: dolnej, górnej, prawej oraz lewej;

PRZYKŁAD 10.21



Rys. 10.14. Przykłady obramowania w CSS

- **border-color** – służy do określenia koloru obramowania. Kolor określa się za pomocą nazw w języku angielskim lub zapisu szesnastkowego. Ponadto, podobnie jak w przypadku dwóch poprzednich wartości, każda z czterech krawędzi może mieć inny kolor;
- **border-radius** – służy do określenia rozmiaru zaokrąglenia rogów obramowania danego obszaru.
Grupa wartości tej własności może składać się z jednego do ośmiu różnych wartości, przy czym zaokrąglenie w pionie poszczególnych rogów obramowania należy wpisać po znaku /;

PRZYKŁAD 10.22

```
<!DOCTYPE html>
<html>
<meta charset="utf-8">
<style>
#blok1
{
border-style:solid;
border-width:3px;
border-color:green;
margin-bottom:5px;
}
#blok2
{
border-style:solid;
border-width:3px;
border-color:red;
border-radius:12px;
margin-bottom:5px;
}
#blok3
{
border-style:solid;
border-width:3px;
border-color:blue;
border-radius:12px /24px;
margin-bottom:5px;
}
#blok3
{
border-style:solid;
border-width:3px;
border-radius:30px 30px 30px 30px /15px 15px 15px 15px;
margin-bottom:5px;
}
</style>
<body>
<div id="blok1">
Pierwszy blok obramujemy linią ciągłą grubości 3px i kolorem
zielonym.
</div>
```



```

<div id="blok2">
Drugi blok obramujemy linią ciągłą grubości 3px i kolorem
czerwonym i zaokrąglonymi rogami.
</div>
<div id="blok3">
Czwarty blok obramujemy linią ciągłą grubości 3px i kolorem
czerwonym i zaokrąglonymi rogami.
</div>
</body>
</html>

```

Pierwszy blok obramujemy linią ciągłą grubości 3px i kolorem zielonym.

Drugi blok obramujemy linią ciągłą grubości 3px i kolorem czerwonym i zaokrąglonymi rogami.

Czwarty blok obramujemy linią ciągłą grubości 3px i kolorem czerwonym i zaokrąglonymi rogami.

Rys. 10.15. Formatowanie obramowania w CSS

- **border-image** – służy do określenia cech obramowania obrazkowego. Za pomocą tej własności oraz grupy wartości można określić kilka różnych cech obramowania obrazkowego.

Za pomocą tej właściwości można określić:

- **source** – plik, za pomocą którego tworzy się obramowanie;
- **width** – szerokość obramowania;
- **outset** – określa rozmiar odległości obramowania od krawędzi danego obszaru;
- **repeat** – określa sposób powtarzania się niektórych elementów obramowania obrazkowego; może przyjmować następujące wartości: **stretch** (wartość domyślna), **repeat**, **round**, **space**;
- **slice** – określa, w jaki sposób dany obrazek ma zostać podzielony na części.

PRZYKŁAD 10.23

Poniższy kod umożliwi obramowanie bloku za pomocą obramowania obrazkowego:

```

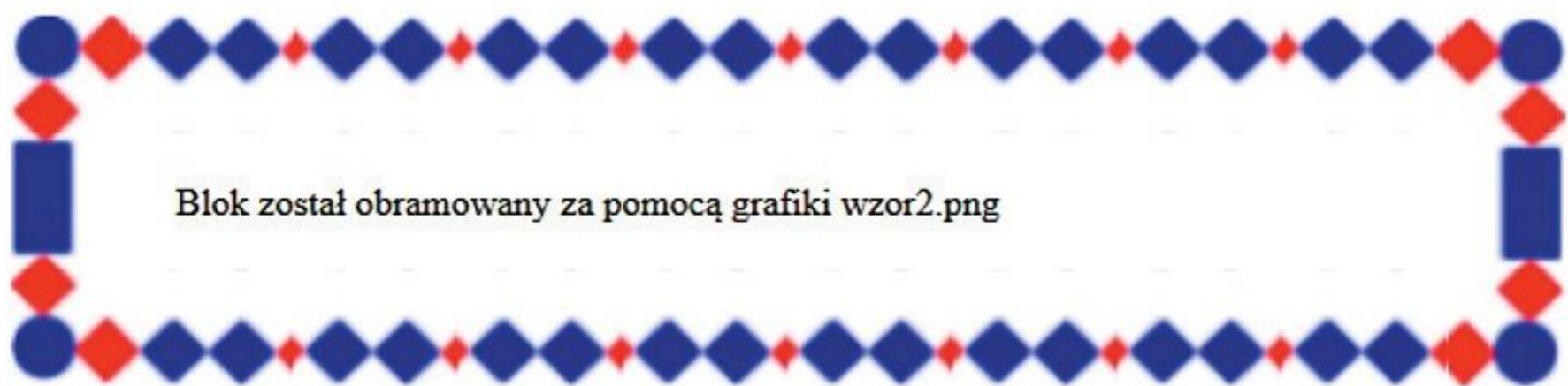
<!DOCTYPE html>
<html>
<meta charset="utf-8">
<style>
#blok1
{
margin:0 auto;

```



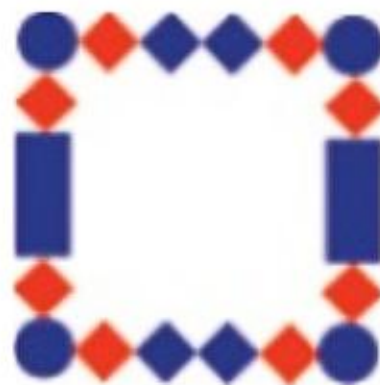
```
border-width: 60px;
border-style: solid;
margin-top: 20px;
border-image: url("wzor2.png") 27 / 50px / 10px round;
height: 30%;
width: 50%;
}
</style>
<body>
<div id="blok1">
Blok został obramowany za pomocą grafiki wzor2.png
</div>
</body>
</html>
```

Efektom będzie obramowanie przedstawione na rys. 10.16.



Rys. 10.16. Zastosowanie obramowania obrazkowego

Do utworzenia tego typu obramowania wykorzystano plik graficzny wzor2.png przedstawiony na rys. 10.17.



Rys. 10.17. Wzór do obramowania obrazkowego

! UWAGA

Za pomocą właściwości **border** można określić kilka różnych cech obramowania wybranego obszaru strony.

Tabele

Do właściwości formatujących tabele zalicza się:

- **border** – służy do obramowania poszczególnych elementów tabeli;
- **padding** – służy do określenia odległości tekstu od obramowania (tzw. margines wewnętrzny);
- **border-spacing** – służy do określenia szerokości między komórkami;
- **caption-side** – określa położenie nagłówka tabeli; właściwość ta może przyjmować następujące wartości:
 - **top** – nagłówek zostanie umieszczony nad górną krawędzią tabeli;
 - **bottom** – nagłówek zostanie umieszczony pod górną krawędzią tabeli;
- **border-collapse** – służy do połączenia dwóch sąsiadujących ze sobą obramowań;
- **background-color** – określa kolor tła poszczególnych elementów tabeli;
- **text-align** – wyrównanie tekstu w poziomie;
- **vertical-align** – wyrównanie tekstu w pionie; ta właściwość może przyjmować następujące wartości:
 - **middle** – tekst w komórce wyśrodkowany w pionie;
 - **bottom** – wyrównanie do dolnej krawędzi komórki;
 - **top** – wyrównanie do górnej krawędzi komórki;
- **width** – szerokość tabeli, wiersza lub komórki tabeli;
- **height** – wysokość tabeli, wiersza lub komórki tabeli.

PRZYKŁAD 10.25

Aby otrzymać tabelę przedstawioną na rys. 10.18, należy utworzyć plik html z poniższym kodem:

```
<html>
<head>
<title>Rzym i jego zabytki</title>
<meta charset="utf-8">
<style>
  img
  {
    width:200px;
    height:200px;
  }
  table
  {
    width:700px;
    height:300px;
    border: 8px #0066FF groove;
    caption-side:top;
    border-collapse: collapse;
```



```

}
td
{
text-align: center;
vertical-align: middle;
border: 1px grey solid;
color: #0066CC;
font-size: large;
}
.zdj
{
background-color: #FFFF99;
}
</style>
</head>
<body>
<table>
<caption>Zabytki Rzymu</caption>
<tr>
<td class="zdj"></td>
<td class="zdj"></td>
<td class="zdj"></td>
</tr>
<tr>
<td>Koloseum</td>
<td>Fontanna di Trevi</td>
<td>Bazylika Św. Piotra</td>
</tr>
</table>
</body>
</html>

```



Rys. 10.18. Zastosowanie formatowania tabel

Linki

Aby określić wygląd odsyłacza, trzeba najpierw określić jego stan, czyli ustalić, czy jest to link odwiedzony, czy np. aktywny. W tym celu należy zastosować tzw. **pseudoklasy**. W przypadków linków mogą to być:

- link podstawowy – pozwala określić atrybuty formatowania dla wszystkich odsyłaczy na stronie:

```
a: link  
{ właściwość: wartość;}
```

- link odwiedzony – pozwala określić atrybuty formatowania dla wszystkich odsyłaczy odwiedzonych przez użytkownika:

```
a: visited  
{ właściwość: wartość;}
```

- link, nad którym znajduje się kursor myszy:

```
a: hover  
{ właściwość: wartość;}
```

- link, na którym jest ustawiony fokus, np. za pomocą klawisza tabulacji:

```
a: focus  
{ właściwość: wartość;}
```

- link aktywny – pozwala określić atrybuty formatowania dla odsyłaczy, które zostały aktywowane przez użytkownika:

```
a: active  
{ właściwość: wartość;}
```

PRZYKŁAD 10.26

```
<html>  
<meta charset="utf-8">  
<style>  
a:link  
{  
font-size:30px;  
color:green;  
background-color: #dddddd;  
text-decoration: none;  
}  
a:visited
```



```
{
color:red;
}
a:hover
{
background-color: #FFFF33;
color: #660033;
}
a:focus
{
background-color: black;
color: white;
}
</style>
<body>
<a href="http://www.info.tekom.strefa.pl">
Przejdź do strony i pobierz zadanie
</a>
</body>
</html>
```

Powyższy kod spowoduje, że po załadowaniu strony link o treści: „Przejdź do strony i pobierz zadanie” przyjmie postać:

Przejdź do strony i pobierz zadanie

Po kliknięciu w link stanie się on odsyłaczem odwiedzionym i będzie miał następujący wygląd:

Przejdź do strony i pobierz zadanie

Najechnie na odsyłacz kursorem myszki spowoduje zmianę wyglądu linku na:

Przejdź do strony i pobierz zadanie

Natomiast kliknięcie w link klawiszem TAB spowoduje, że link przyjmie poniższą postać:

Przejdź do strony i pobierz zadanie

Kursory

Za pomocą właściwości `cursor` można określić typ kursora danego elementu. Własność ta może przyjmować następujące wartości:

- **auto** – przeglądarka internetowa określa typ kursora;
- **pointer** – kursor w postaci łapki.

PRZYKŁAD 10.27

```
<div style="cursor:pointer;">Przykład działania kursora typu  
<b>pointer</b></div>
```

Najechnie na tekst umieszczony w bloku spowoduje, że pojawi się kursor w kształcie białej łapki.

- **progress** – kursor sugerujący, że jest wykonywana dodatkowa czynność na stronie;
- **crosshair** – kursor w postaci krzyżyka;
- **move** – kursor przenoszenia;
- **help** – kursor ze znakiem zapytania;
- **text** – kursor umożliwiający zaznaczenie tekstu pionowo;
- **vertical-text** – kursor umożliwiający zaznaczanie tekstu poziomo.

Pozostałe wartości tej własności można znaleźć na stronie www.w3schools.com z dokładnym opisem i odpowiednim przykładem.

Przezroczystość

Za pomocą właściwości `opacity` można określić przezroczystość wybranego obszaru strony. Własność ta przyjmuje wartość będącą nieujemną liczbą z zakresu od 0 do 1.

PRZYKŁAD 10.28

```
<html>
<head>
<meta charset="utf-8">
<title></title>
<style>
#blokA
{
height:20%;
background-image: url("gerbery.jpg");
background-repeat:repeat;
opacity:0.7;
font-size:30px;
padding-top: 20px;
color:white;
}
```



```
</style>
```

```
</head>
```

```
<body>
```

```
<div id="blokA">
```

Gerbery to rośliny z rodziny astrowatych. Ta duża grupa skupia około 50 gatunków. Gerbery występują w południowo-wschodniej Afryce oraz na Madagaskarze. Można je również spotkać w tropikalnych rejonach Azji oraz na Tasmanii.

```
</div>
```

```
</body>
```

```
</html>
```

Kod z przykładu 10.28 utworzy blok, w którym zostanie określone tło obrazkowe, na którym będzie wyświetlony tekst czcionką wielkości 30 px i koloru białego. Właściwość **opacity** spowoduje, że procent przezroczystości wyniesie 70 procent, ponieważ została ustalona wartość 0,7.



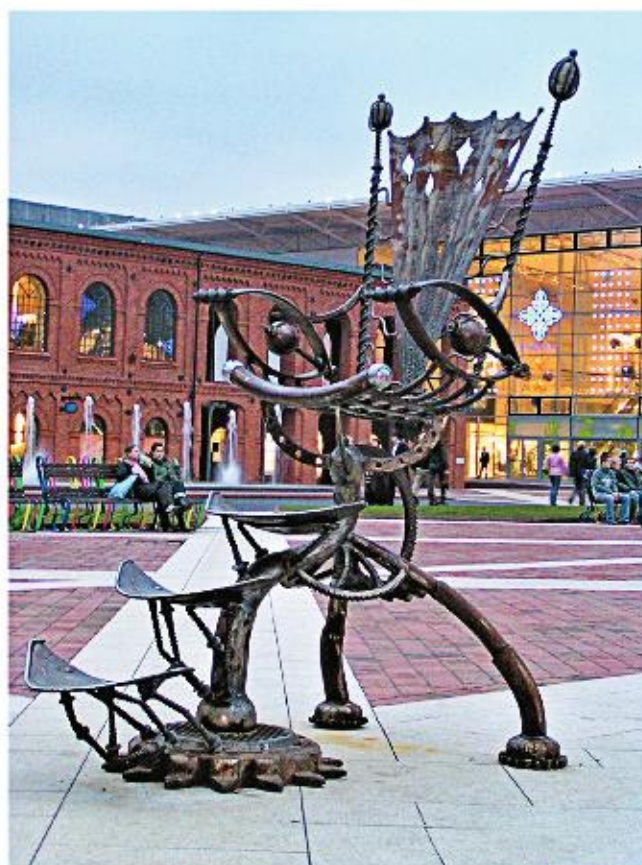
Rys. 10.19. Zastosowanie właściwości **opacity**

Za pomocą właściwości **filter** można zmienić kolorystykę wybranego obszaru strony. Ta własność może przyjmować następujące wartości:

- **none** – wartości domyślna;
- **grayscale()** – pozwala zmienić odcień szarości wybranego elementu graficznego;
- **sepia()** – pozwala zmienić procent sepii wybranego elementu graficznego;
- **saturate()** – pozwala zmienić procent nasycenia kolorów wybranego elementu;
- **hue-rotate()** – pozwala zmienić odcienie kolorów za pomocą koła barw;
- **invert()** – pozwala zmienić procent odwrócenia kolorów wybranego elementu graficznego;
- **brightness()** – pozwala zmienić procent jasności kolorów wybranego elementu graficznego;
- **contrast()** – pozwala zmienić procent zróżnicowania kolorów wybranego elementu graficznego;
- **blur()** – pozwala zmienić stopień rozmycia kolorów wybranego elementu graficznego;
- **opacity()** – pozwala zmienić procent przezroczystości kolorów wybranego elementu graficznego;
- **drop-shadow()** – pozwala określić efekt cienia dla kolorów.

PRZYKŁAD 10.29

Przykład ilustruje działanie kilku filtrów na pliku graficznym *lodz.jpg*.



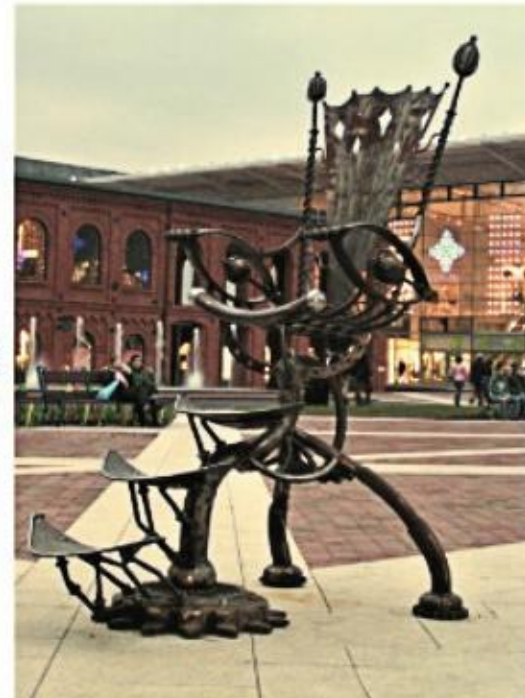
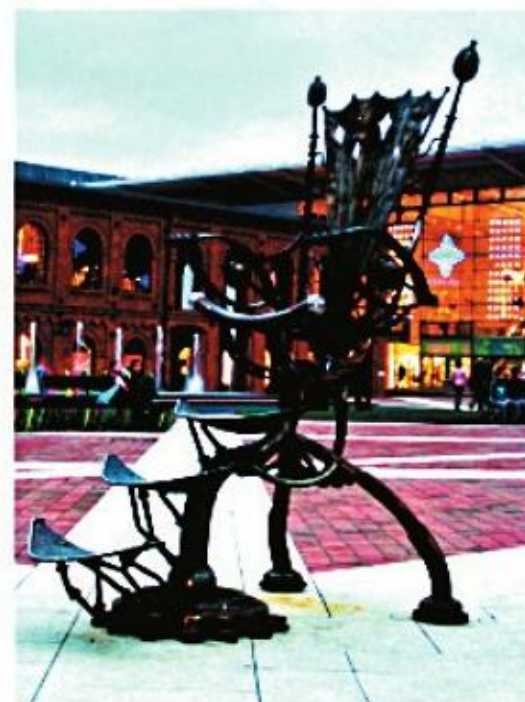
```
<html>
<head>
<meta charset="utf-8">
<style>
img
{
width:150px;
height:200px;
}
</style>
</head>
<body>
<table>
<tr>
<td></td>
<td>Filtr <i>grayscale</i></td>
<td></td>
<td>Filtr <i>sepia</i></td>
</tr>
<tr>
<td></td>
<td>Filtr <i>invert</i></td>
<td></td>
<td>Filtr <i>brightness</i></td>
</tr>
<tr>
```



```

<td></td>
<td>Filtr blur</td>
<td></td>
<td>Filtr contrast</td>
</tr>
</table>
</body>
</html>

```

Filtr *grayscale*Filtr *sepia*Filtr *invert*Filtr *brightness*Filtr *blur*Filtr *contrast*

Rys. 10.20. Zastosowanie filtrów

Pozostałe przydatne właściwości CSS

Podczas pracy można również spotkać właściwości ułatwiające wyświetlanie elementu, formatowanie wypływania treści kontenera bądź oblewanie tekstem elementu.

W poprzednich rozdziałach opisano znaczniki **<div>** i ****. Pierwszy z nich tworzy elementy blokowe, natomiast drugi stosuje się do definiowania elementów liniowych. Każdy blok zajmuje domyślnie 100% szerokości, a każdy kolejny jest umieszczony pod spodem. Elementy liniowe umieszcza się obok siebie. Natomiast nie można określić szerokości i wysokości elementów liniowych ani marginesów.

Za pomocą właściwości **display** można zmienić domyślny sposób zachowania elementów na stronie. Display przyjmuje następujące wartości:

- **none** – pozwala ukryć element na stronie;
- **inline** – pozwala na wyświetlenie elementu jako liniowego;
- **block** – pozwala na wyświetlenie elementu jako blokowego;
- **inline-block** – pozwala połączyć cechy elementu blokowego i liniowego, czyli można określić wartości marginesów i nie nastąpi po nim przejście do nowej linii.

PRZYKŁAD 10.30

```
<html>
<head>
<meta charset="utf-8">
<style>
.blok1
{
display: block;
width: 100px;
height: 100px;
margin: 5px;
background-color: green;
color:white;
}
.blok2
{
display: inline-block;|
width: 100px;
height: 100px;
margin: 5px;
background-color: red;
color:white;
}
</style>
</head>
<body>
<h4>Działanie display:block</h4>
<div class="blok1"></div>
<span class="blok1"></span>
<h4>Działanie display:inline-block</h4>
<div class="blok2"></div>
<span class="blok2"></span>
</body>
</html>
```

Działanie display:block



Działanie display:inline-block



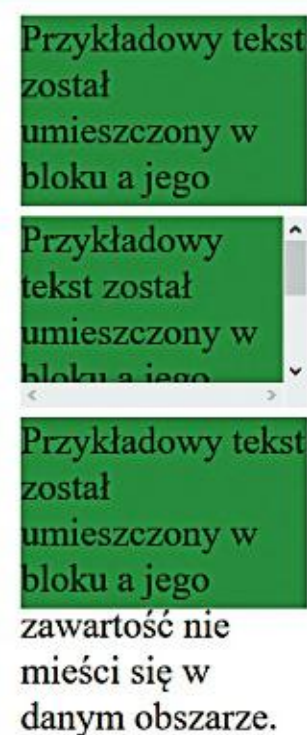
Rys. 10.21. Zastosowanie właściwości **display**

Kolejną niezwykle użyteczną właściwością jest **overflow**, która pozwala zarządzać treścią niemieszczącą się w danym obszarze strony. Przyjmuje ona następujące wartości:

- **visible** – wartość domyślna, zawartość niemieszcząca się w obszarze określonym zewnętrzną krawędzią prawego oraz dolnego marginesu wewnętrznego będzie widoczna;
- **hidden** – treść niemieszcząca się w danym obszarze zostanie ukryta;
- **scroll** – pozwala na dodanie do danego obszaru suwaków, umożliwiających przewijanie niemieszczącej się treści;
- **auto** – pozwala na dodanie do obszarów suwaków, gdy będzie to konieczne.

PRZYKŁAD 10.31

```
<html>
<head>
<meta charset="utf-8">
<style>
.blok1
{
  width: 150px;
  height: 100px;
  margin: 5px;
  background-color: green;
  font-size: 20px;
}
</style>
</head>
<body>
<div class="blok1" style="overflow:hidden;">Przykładowy tekst został umieszczony w
bloku a jego zawartość nie mieści się w danym obszarze. </div>
<div class="blok1" style="overflow:scroll;">Przykładowy tekst został umieszczony w
bloku a jego zawartość nie mieści się w danym obszarze. </div>
<div class="blok1" style="overflow:visible;">Przykładowy tekst został umieszczony w
bloku a jego zawartość nie mieści się w danym obszarze. </div>
</body>
</html>
```



Przykładowy tekst został umieszczony w bloku a jego zawartość nie mieści się w danym obszarze.

Przykładowy tekst został umieszczony w bloku a jego zawartość nie mieści się w danym obszarze.

Przykładowy tekst został umieszczony w bloku a jego zawartość nie mieści się w danym obszarze.

Rys. 10.22. Zastosowanie właściwości **overflow**

Właściwość **float** umożliwia zdefiniowanie sposobu otaczania elementów tekstem. Może przyjmować wartości:

- **left** – element zostanie ustawiony po lewej stronie względem obiektów sąsiadujących;
- **right** – element zostanie ustawiony po prawej stronie względem obiektów sąsiadujących;
- **none** – wartość domyślna, element nie sąsiaduje z żadnymi obiektami.

Tę właściwość stosuje się w przypadku, gdy kilka elementów blokowych zamierza się ułożyć obok siebie.

Właściwość **clear** określa sposób przylegania do elementów „pływających”. Może ona przyjmować następujące wartości:

- **none** – wartość domyślna;
- **left** – następny element będzie pod obiektem z lewej strony;
- **right** – następny element będzie pod obiektem z prawej strony;
- **both** – następny element będzie pod obiektem niezależnie od jego położenia.

PRZYKŁAD 10.32

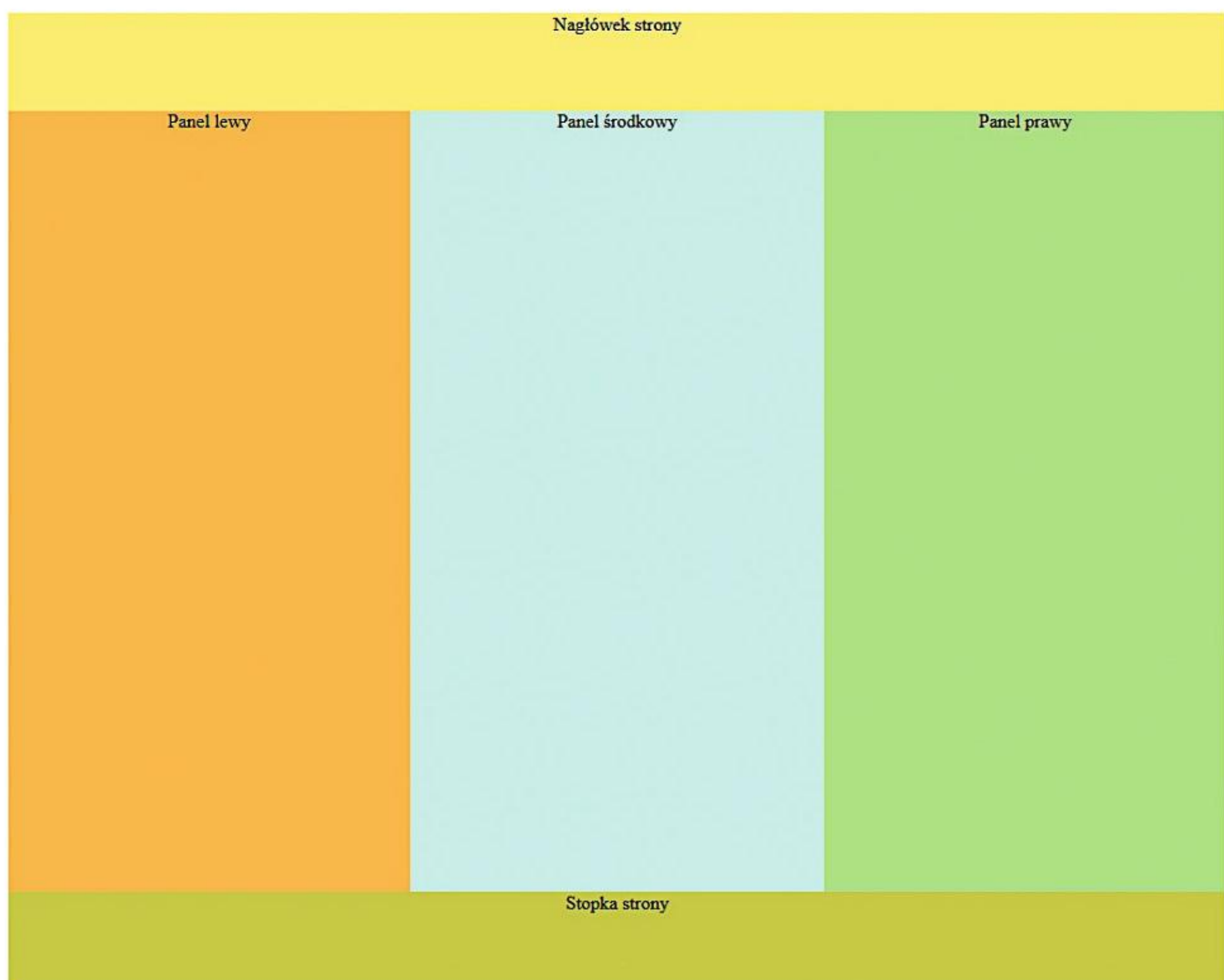
W przykładzie przedstawiono podział strony na pięć bloków: baner, panel lewy, panel środkowy, panel prawy i stopka. Aby umieścić trzy środkowe bloki obok siebie, należy zastosować polecenie **float:left**. Spowoduje ono, że panele lewy, środkowy i prawy będą ułożone jeden obok drugiego. Polecenie **clear:both** ma na celu usunięcie opływania bloków.

```
<html>
<meta charset="utf-8">
<style>
#baner
{
height:10%;
background-color:#FFFF99;
text-align:center;
}
#panel_lewy, #panel_srodkowy, #panel_prawy
{
height:80%;
float:left;
text-align:center;
}
#panel_lewy
{
background-color: #FFCC66;
width:33%;
}
#panel_srodkowy
{
background-color: #CCFFFF;
width:34%;
}
#panel_prawy
{
background-color: #CCFF99;
width:33%;
}
#stopka
{
height:10%;
text-align:center;
background-color:#CCCC66;
```



```
clear:both;
}
</style>
<body>
<div id="baner">Nagłówek strony</div>
<div id="panel_lewy">Panel lewy</div>
<div id="panel_srodkowy">Panel środkowy</div>
<div id="panel_prawy">Panel prawy</div>
<div id="stopka">Stopka strony</div>
</body>
</html>
```

Efektem powyższego kodu będzie strona następującej postaci:



Rys. 10.23. Zastosowanie właściwości **float** oraz **clear**

! UWAGA

Zestawienie najpopularniejszych właściwości i ich wartości specyfikacji CSS 3 zamieszczono w postaci załącznika na końcu podręcznika.

Jednostki miary oraz kolory w wartościach CSS

Kaskadowe arkusze stylów służą do określania cech wielu elementów. Stosuje się przy tym różne jednostki miar. Można je podzielić na jednostki względne i bezwzględne.

Do jednostek miar względnych zalicza się:

- **px** – piksele;
- **em** – proporcje elementu względem wielkości czcionki;
- **ex** – proporcja względem wysokości x danej czcionki;
- **%** – procenty.

PRZYKŁAD 10.33

```
p{  
  font-size:10%;  
  margin-left:20px;  
}
```

Do jednostek miar bezwzględnych zalicza się:

- **in** – cale;
- **pt** – punkty (jednostka wywodząca się z typografii; 72 pt = 1 in);
- **cm** – centymetry;
- **mm** – milimetry;
- **pc** – pica (jednostka wywodząca się z typografii; 1 pc = 12 pt).

PRZYKŁAD 10.34

```
p{  
  font-size:12pt;  
  margin-left:2in;  
}
```

Dodatkowo, w tworzeniu kaskadowych arkuszy stylów używa się jednostki kątowe oraz czasu. Do najczęściej stosowanych zalicza się:

- **ms** – milisekunda;
- **s** – sekunda,
- **deg** – stopnie;
- **rad** – radiany.

PRZYKŁAD 10.35

```
h5 {  
  pause-after:10s;  
}
```

W wartościach często podaje się nazwy kolorów za pomocą kilku dostępnych metod. Kodowanie kolorów w CSS można wykonać przez:

- podanie nazwy koloru –

```
h3 {color:red;}
```

- zapis za pomocą modelu RGB, z wykorzystaniem wartości liczbowych –

```
#stopka {color:rgb(255,0,0);}
```

- zapis za pomocą modelu RGB, z wykorzystaniem wartości procentowych –

```
p {color:rgb(100%,0%,0%);}
```

- zapis za pomocą modelu RGBA –

```
#menu {background-color: rgba(255, 0, 0, 0.3);}
```

- zapis za pomocą modelu HLS –

```
.menu {background-color: hsl(120, 100%, 50%);}
```

- zapis za pomocą modelu HSLA:

```
p.stopka {background-color: hsla(120, 100%, 25%, 0.3);}
```

- zapis korzystający z systemu szesnastkowego –

```
p {color:#FF0000;}
```

Dodatkowo, można określić stopień przezroczystości właściwością `opacity`, z wykorzystaniem wartości od 0.0 (pełna przezroczystość) do 1.0 (brak przezroczystości):

```
p {background-color:rgb(255,0,0);opacity:0.7;}
```

SPRAWDŹ SWOJĄ WIEDZĘ

1. Omów właściwości formatujące czcionkę.
2. Omów własności służące do obramowania elementów na stronie.
3. Jaka własność służy do określenia tła danego elementu strony?
4. Omów sposób formatowania linków na stronie.

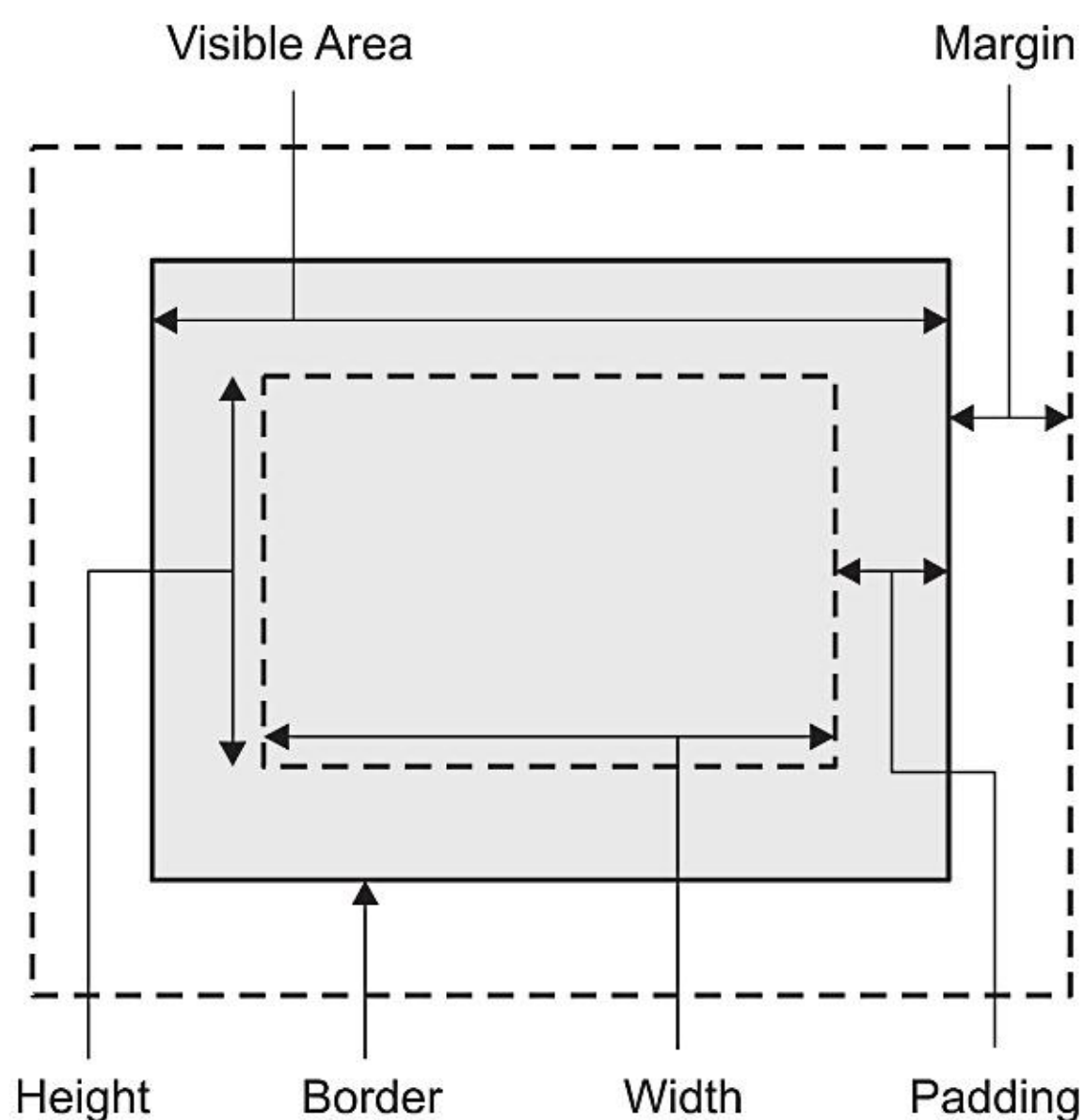
11

Model blokowy w CSS

ZAGADNIENIA

- Struktura układu blokowego
- Właściwości i wartości stosowane w modelu blokowym

Model blokowy w CSS opisuje bloki, które są tworzone dla elementów HTML. Model blokowy dokładnie definiuje ustawienie marginesów: zewnętrznego, wewnętrznego, obramowania i zawartość każdego elementu. Strukturę układu blokowego elementów HTML przedstawiono na rys. 11.1.



Rys. 11.1. Przykład modelu blokowego

Aby wytłumaczyć zasadę działania modelu blokowego w kaskadowych arkuszach stylów, posłużymy się przykładem formatowania prostego dokumentu hipertekstowego zawierającego jeden nagłówek drugiego stopnia i jeden blok `<div>`.


```

1  <!DOCTYPE html>
2  <html>
3
4  <body>
5
6  <h2>Przykład modelu blokowego</h2>
7
8  <div>Model blokowy CSS opisuje bloki, jakie tworzą się wokół
   elementów HTML zawierających treść. W myśl zasad określanych przez
   ten model, każdy element opcjonalnie może otrzymać dodatkowe
   parametry, opisujące jakie powinny być odległości formatowanego
   elementu względem innych elementów na stronie, jakie element powinien
   mieć obramowanie oraz jaka powinna być odległość pomiędzy zawartością
   elementu a jego obramowaniem.</div>
9
10 </body>
11 </html>

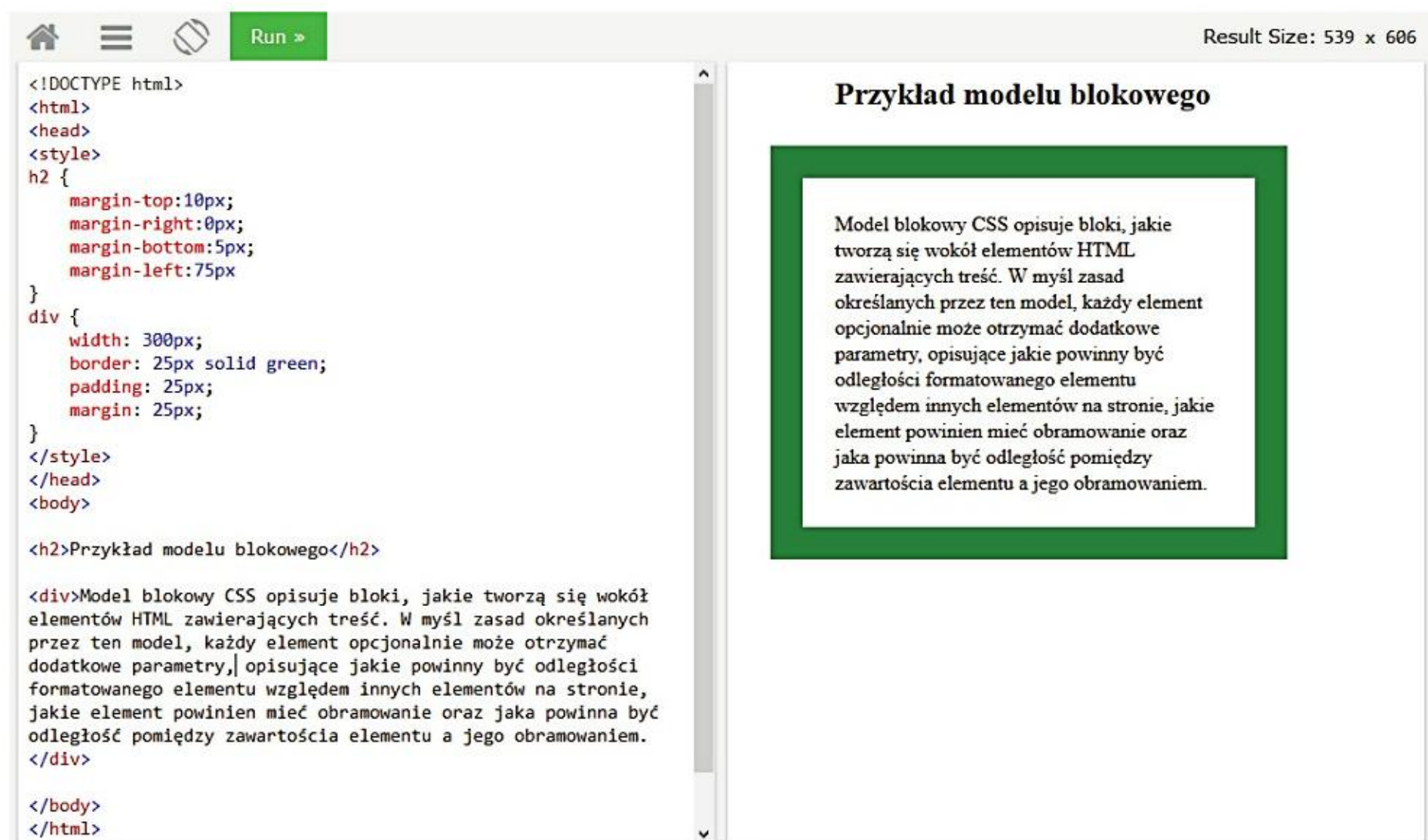
```

Rys. 11.2. Dokument HTML przed formatowaniem stylem CSS

Mamy tutaj dwa elementy HTML, które sformatujemy za pomocą modelu blokowego. Dla nagłówka zdefiniujemy wartości marginesów zewnętrznych **margin**, które określą położenie względem kontenera strony.

Dla bloku **<div>**, oprócz zdefiniowania marginesu zewnętrznego **margin** i wewnętrznego **padding**, określimy szerokość i kolor obramowania **border** oraz szerokość bloku **width**.

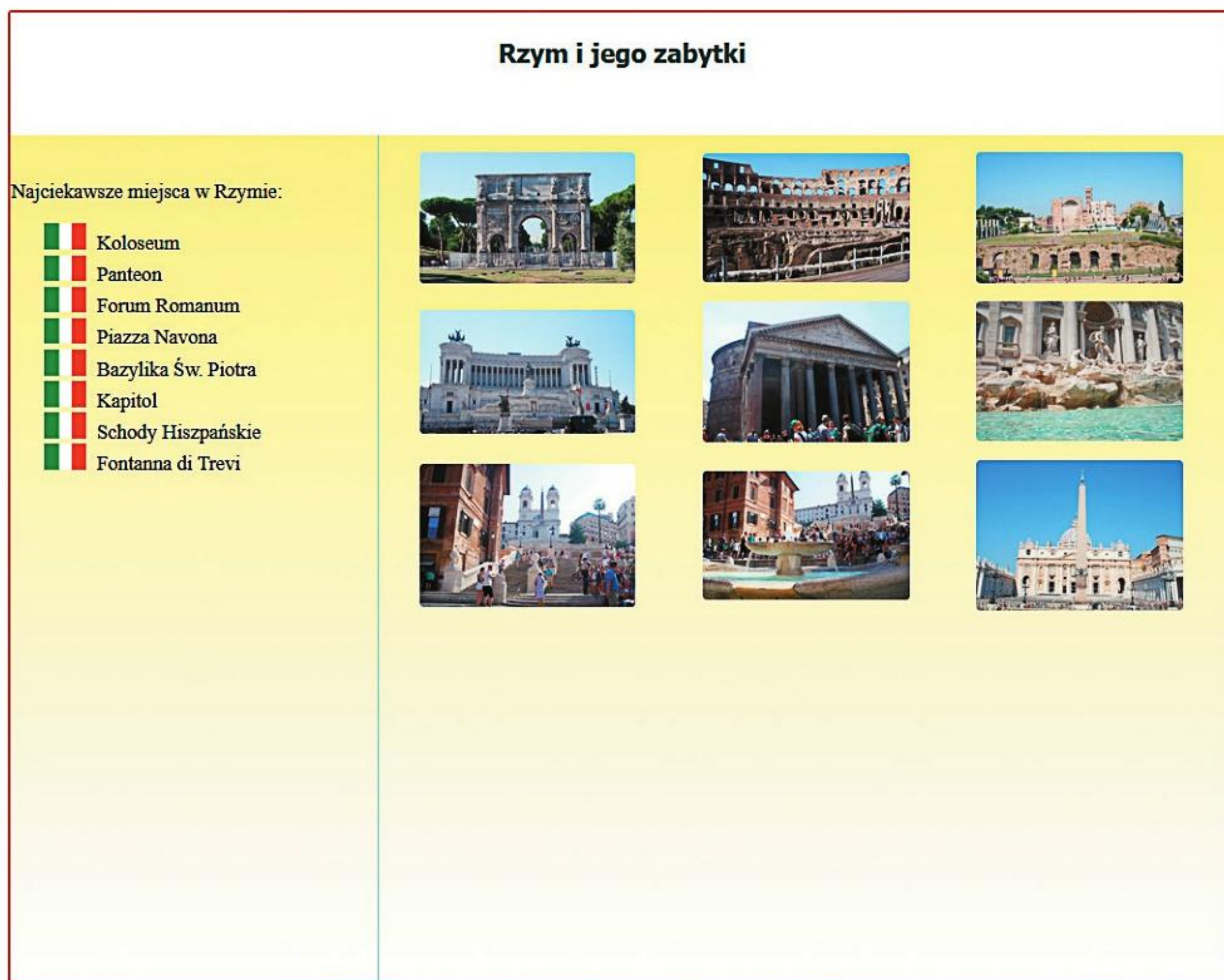
Po zapisaniu definicji stylu wewnętrznego w sekcji **<head>** otrzymamy efekt widoczny na rys. 11.3.



Rys. 11.3. Dokument po formatowaniu według modelu blokowego

PRZYKŁAD 11.1

Za pomocą modelu blokowego można stworzyć dowolny układ strony. Przykładem może być strona przedstawiona na rys. 11.4.



Rys. 11.4. Strona z podziałem na bloki

Stronę z rys. 11.4 utworzono na podstawie dwóch plików: rzym.html i rzym.css. W pierwszym z nich umieszczono treść strony, drugi jest plikiem zawierającym style css.

Kod źródłowy pliku rzym.html:

```
<html>
<head>
<title>Rzym i jego zabytki</title>
<meta charset="utf-8">
<link rel="stylesheet" type="text/css" href="rzym.css">
</head>
<body>
<div id="main">
```



```

<div id="baner">>
Rzym i jego zabytki
</div>
<div id="lewy">>
<p>Najciekawsze miejsca w Rzymie:</p>
<ul>
<li>Koloseum</li>
<li>Panteon</li>
<li>Forum Romanum</li>
<li>Piazza Navona</li>
<li>Bazylika Św. Piotra</li>
<li>Kapitol</li>
<li>Schody Hiszpańskie</li>
<li>Fontanna di Trevi
</li>
</ul>
</div>
<div id="prawy">>
<table>
<tr>
<td colspan="2"></td>
<td colspan="2"></td>
<td colspan="2"></td>
</tr>
<tr>
<td colspan="2"></td>
<td colspan="2"></td>
<td colspan="2"></td>
</tr>
<tr>
<td colspan="2"></td>
<td colspan="2"></td>
<td colspan="2"></td>
</tr>
</table>
</div>
</div>
</body>
</html>

```


Kod źródłowy pliku rzym.css:

```
#main
{
margin: 5% auto 0 auto;
width:80%;
height:90%;
border:4px double red;
border-radius:5px;
background-image: linear-gradient(#FFFF99, white);
}
#baner
{
font-family: Tahoma;
color:#003333;
padding-top: 2%;
height:10%;
font-weight: 900;
background-color: white;
text-align: center;
font-size: 30px;
}
#lewy
{
height:87%;
width:30%;
float:left;
padding-top: 2%;
}
#prawy
{
height:87%;
width:69%;
float:left;
border-left:1px #99cccc solid;
}
img
{
width:80%;
height:80%;
border-radius:5px;
}
```



```
table
{
border-spacing:15px 20px;
margin:auto;
text-align: center;
}

p
{
font-size: x-large;
}

li
{
list-style-image: url("flaga2.png");
list-style-position: inside;
font-size:x-large;
}
```

SPRAWDŹ SWOJĄ WIEDZĘ

1. Jaką funkcję pełni model blokowy w CSS?
2. Czym charakteryzuje się dokument HTML po sformatowaniu z użyciem stylu CSS?

12

Pozycjonowanie elementów w CSS

ZAGADNIENIA

- Rola pozycjonowania elementów
- Właściwości i wartości dotyczące pozycjonowania

Właściwości pozycjonowania pozwalają na precyzyjne określenie miejsca, w którym dany element ma być wyświetlany na stronie internetowej. Wszystkie elementy mogą być rozmieszczane na jeden z następujących sposobów:

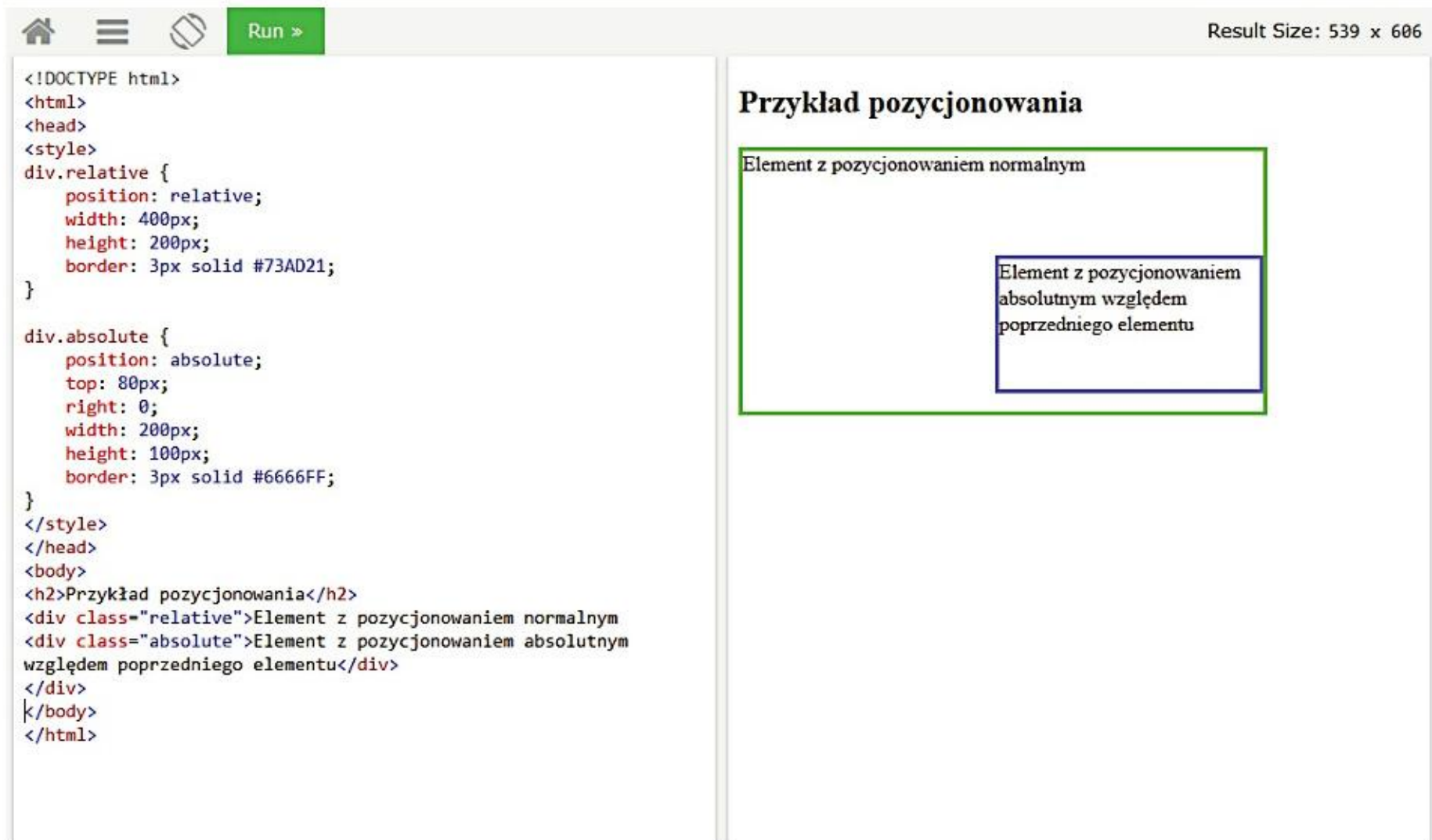
- **static** – w takim porządku, jak wynika to z układu w dokumencie hipertekstowym;
- **relative** – umieszczany względem swojego normalnego położenia;
- **absolute** – umieszczany w pozycji absolutnej, określanej względem pierwszego elementu przodka, mającego inną metodę pozycjonowania niż **static**;
- **fixed** – umieszczany względem okna przeglądarki.

Dodatkowo właściwości **top**, **right**, **bottom**, **left** pozwalają na określenie przesunięcia bloku elementu o wskazane wartości (nie działają dla wartości pozycji **static**). Zależnie od wybranej metody pozycjonowania oraz rodzaju wartości odnoszą się one do krawędzi elementu pozycjonowanego, elementu nadrzędnego lub krawędzi okna przeglądarki.

Za pomocą pozycjonowania **absolutnego** elementu określa się jego położenie względem granic okna przeglądarki lub innego elementu nadrzędnego, podając odległości w pikselach, jednostkach em bądź procentach.

Na potrzeby pozycjonowania **relatywnego** położenie elementu określa się względem granic naturalnego położenia pozycjonowanego elementu.

Korzystanie z pozycjonowania i przesunięć elementów względem układu normalnego może skutkować koniecznością określenia sposobu lub kolejności nakładania się jednego na drugi. Właściwością, która pozwoli na wymuszenie kolejności nakładania się elementów, jest **z-index**.

PRZYKŁAD 12.1**Rys. 12.1.** Przykład pozycjonowania elementów w CSS**PRZYKŁAD 12.2**

Tworzymy dokument hipertekstowy z wewnętrznym arkuszem stylów formatującym tekst i grafikę. Gotowy listing pokazano na rys. 12.2.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="utf-8">
5  <style>
6    p {
7      color:blue;
8    }
9    img {
10     position: absolute;
11     left: 0px;
12     top: 0px;
13     z-index: -1;
14   }
15 </style>
16 </head>
17 <body>
18
19 <h1>Przykład nagłówka z pozycjonowaniem grafiki</h1>
20 
21 <p>Zastosowanie właściwości z-index z wartością -1 spowoduje, że
   grafika osadzona zostanie za tekstem.</p>
22
23 </body>
24 </html>
  
```

Rys. 12.2. Dokument HTML z przykładem pozycjonowania

W stylach zastosowano właściwość **z-index** z wartością „-1”. Umożliwi to umieszczenie rysunku pod tekstem umieszczonym w nagłówku i akapicie. Efekt przedstawiono na rys. 12.3.



Przykład nagłówka z pozycjonowaniem grafiki

Zastosowanie właściwości z-index z wartością -1 spowoduje, że grafika osadzona zostanie za tekstem.

Rys. 12.3. Przykład zastosowania właściwości **z-index**



SPRAWDŹ SWOJĄ WIEDZĘ

1. Jaka jest różnica między pozycjonowaniem relacyjnym a absolutnym?
2. Do czego służy pozycjonowanie fixed?
3. Do czego służą właściwości top, right, bottom oraz left?

13

Praktyczne zastosowania CSS i HTML

ZAGADNIENIA

- Przykład formatowania formularza
- Przykład podziału strony na bloki

Znamy już podstawowe znaczniki języka HTML oraz zasady wykorzystania kaskadowych arkuszy stylów do formatowania dokumentów hipertekstowych. W tym rozdziale opiszemy kilka przykładów praktycznego wykorzystania tych umiejętności.

Przykład formatowania formularzy

Naszym zadaniem jest przygotowanie formularza, który będzie wysyłać na adres poczty elektronicznej wynik prostej ankiety dla uczniów. W tytule e-maila musi pojawić się tekst: „Temat ankiety”. Do wysłania e-maila należy zastosować najbezpieczniejszą z dostępnych metod. W ankiecie uczeń powinien podać imię i nazwisko oraz wybrać z dostępnej listy jeden z czterech dostępnych tematów kursów: HTML, CSS, CMS, SQL. Dodatkowo dwa pierwsze pola mają wymusić podanie danych. W tytule formularza należy wstawić tekst: „Ankieta dotycząca wyboru tematu kursu”, w formacie nagłówka pierwszego stopnia. Pod formularzem trzeba zdefiniować dwa przyciski: jeden do wysłania danych o nazwie „Submit,” a drugi do czyszczenia zawartości pól o nazwie „Reset”.

W dokumencie hipertekstowym piszemy kod spełniający warunki zadania. Zaczynamy od napisania struktury strony. W przykładzie pozostawiono pustą sekcję **<head>**, ale można ją uzupełnić według przykładów z poprzednich rozdziałów, np. deklaracją kodowania polskich znaków **<meta charset="utf-8">**. W sekcji **<body>** wstawiono nagłówek formularza zgodnie z wytycznymi i poniżej zdefiniowano poszczególne elementy formularza. Aby wymusić wysłanie e-maila z określonym tematem, korzystamy z atrybutu **action** w postaci:

```
<form action="mailto:wsip@wsip.pl?subject=Temat ankiety" method="post">
```

W kodzie stosujemy metodę POST, ponieważ jest bezpieczniejsza w porównaniu do drugiej metody GET ze względu na to, że parametrów transmisji nie widać w pasku przeglądarki.

Do pól mieszczących imię i nazwisko zastosowano typ tekstowy o określonym rozmiarze, natomiast do wyboru tematu kursu wykorzystano pole **select**, które służy do tworzenia rozwijalnych list z opcjami, spośród których użytkownik wybiera interesujące go pozycje. W opcje wpisano nazwy kursów: HTML, CSS, CMS, SQL.

Aby wymusić wypełnienie pól imieniem i nazwiskiem, posłużono się atrybutem **required**, który w sytuacji, gdy użytkownik nie wpisze danych do pola formularza, wyświetli odpowiedni komunikat.

Pod polami formularza, w oddzielnym bloku, wstawiono obok siebie dwa przyciski typu **reset** oraz **submit**.

Po zapisaniu dokumentu i otwarciu go w oknie przeglądarki otrzymano efekt przedstawiony na rys. 13.1.

The screenshot displays a web browser window with a form titled "Ankieta dotycząca wyboru tematu kursu". The form includes the following elements:

- Imię:** A text input field.
- Nazwisko:** A text input field.
- Tytuł kursu:** A dropdown menu with the following options: HTML, CSS, CMS, and SQL.
- Buttons:** "Reset" and "Submit" buttons.

The left pane of the browser shows the HTML code used to create the form:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<h1>Ankieta dotycząca wyboru tematu kursu</h1>
<form action="mailto:wsip@wsip.pl?subject=Temat ankiety" method="post">
<p><strong>Imię</strong>: <input type="text" name="Imie" size="30" maxlength="40"
required></p>
<p><strong>Nazwisko</strong>: <input type="text" name="Nazwisko" size="30"
maxlength="40" required></p>
<p><strong>Tytuł kursu</strong>: <select >
<option value ="HTML">HTML</option>
<option value ="CSS">CSS</option>
<option value ="CMS">CMS</option>
<option value ="SQL">SQL</option>
</select> </p>
<p><input type="reset" value="Reset">
<input type="submit" value="Submit"></p>
</form>
</body>
</html>
```

Rys. 13.1. Utworzony formularz w języku HTML

Formularz jest gotowy, choć wyraźnie brakuje mu estetycznego wyglądu oraz właściwego pozycjonowania. Aby nadać mu odpowiednie formatowanie, należy użyć kaskadowych arkuszy stylów. Dla czytelności zmian użyjemy stylów wewnętrznych oraz inline.

Styl wewnętrzny wstawiamy w sekcję **<head>**. Sformatuje on elementy: **h1** oraz dwa przyciski typu **reset** i **submit**. Dla elementu **h1** określamy kolor czcionki, rodzaj obramowania, efekt wyświetlania tekstu oraz jego wyśrodkowanie. Za pomocą grupowania selektorów formatujemy przyciski. Definiujemy dla nich kolejno: kolor tła, obramowanie, kolor tekstu, marginesy (tym samym wielkość przycisku), rodzaj kursora oraz zaokrąglenie obramowania.

Styl inline ma formatować wygląd pól formularza. Definiujemy w nim pozycjonowanie, kolor tła oraz kolor i styl wyświetlania tekstu.

Gotowy listing przedstawiono na rys. 13.2.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color:grey;
  border-style: ridge;
  text-shadow: 2px 2px #000;
  text-align: center;
}
input[type=submit], input[type=reset] {
  background-color: #4CAF50;
  border: 4px solid #ccc;
  color: white;
  padding: 16px 32px;
  margin: 4px 2px;
  cursor: pointer;
  border-radius: 8px;
}
</style>
</head>
<body>
<h1>Ankieta dotycząca wyboru tematu kursu</h1>
<form action="mailto:wsip@wsip.pl?subject=Temat ankiety" method="post">
<p><strong>Imię</strong>: <input type="text" name="Imie" size="30" maxlength="40" style="position: absolute; left: 150px;
background-color:#CCFF99; font-style: italic; color: gray" required></p>
<p><strong>Nazwisko</strong>: <input type="text" name="Nazwisko" size="30" maxlength="40" style="position: absolute;
left:150px; background-color:#CCFF99; font-style: italic; color: gray " required></p>
<p><strong>Tytuł kursu</strong>: <select style="position: absolute; left:150px; background-color:#CCFF99; font-style: italic;
color: gray ">
<option value = "HTML">HTML</option>
<option value = "CSS">CSS</option>
<option value = "CMS">CMS</option>
<option value = "SQL">SQL</option>
</select> </p>
<p><input type="reset" value="Reset">
<input type="submit" value="Submit"></p>
</form>
</body>
</html>
```

Rys. 13.2. Listing dokumentu hipertekstowego z formularzem

Po zapisaniu zmian i wyświetleniu formularza w przeglądarce uzyskamy efekt widoczny na rys. 13.3.

Ankieta dotycząca wyboru tematu kursu

Imię:

Nazwisko:

Tytuł kursu: HTML ▾

Rys. 13.3. Sformatowany formularz

Przykład podziału strony na bloki

W tym zadaniu mamy utworzyć stronę internetową szkolnej rady klasowej według określonych wytycznych. Cechy strony:

- zastosowany standard kodowania polskich znaków;
- zastosowany zewnętrzny arkusz stylów;
- podział strony na bloki: panel lewy, panel prawy, stopka – za pomocą znaczników sekcji;
- gotowy plik zapisany pod nazwą `rada_klasowa.html`.

Zawartość panelu lewego:

- nagłówek pierwszego stopnia o treści „Aktualności”;
- nagłówek trzeciego stopnia o treści „Co robimy?”;
- lista numerowana z elementami: Pomagamy w nauce, Informujemy o prawach ucznia, Prowadzimy finanse klasy;
- nagłówek trzeciego stopnia o treści: „Skład Rady”;
- tabela zgodna z rysunkiem 2, o wymiarach 3 wiersze na 2 kolumny, z czego w pierwszej kolumnie wszystkie wiersze są scalone w jedną komórkę. Treść komórki scalonej: Jan Abacki Przewodniczący, przed słowem „Przewodniczący” występuje łamanie linii. Treść kolumny 2 kolejno: Anna Babacka, Anna Cabacka, Krzysztof Nowak.

Zawartość panelu prawego:

- nagłówek pierwszego stopnia o treści „Nasza Klasa”;
- osadzony rysunek `klasa.jpg` w oddzielnym akapicie.

W stopce strony umieszczony w akapicie napis o treści: „Autor strony: Jan Abacki”.

Styl CSS witryny internetowej powinien zawierać formatowanie dla:

- panelu lewego: szerokość 40%, wysokość 600 px, kolor tła #FF6633;
- panelu prawego: szerokość 60%, wysokość 600 px, kolor tła #FF9933;
- stopki: kolor tła #993300, biały kolor czcionki, wysokość czcionki 150%, wyrównanie tekstu do prawej strony;
- tabeli i komórki: marginesy zewnętrzne na 20 px, wewnętrzne na 30 px, obramowanie na 2 px, linia ciągła brązowego koloru, obramowanie połączone;
- znacznika `h1`: wyrównanie tekstu do środka;
- grafiki: szerokość 450 px, wysokość 350 px.

Niewymienione właściwości obiektów przybierają wartości domyślne. Zewnętrzny styl CSS zostaje zapisany pod nazwą `rada_klasowa.css` i połączony z plikiem `rada_klasowa.html`.

Pierwszą czynnością będzie utworzenie pliku `rada_klasowa.html`. Kodujemy w nim strukturę strony zawierającą kodowanie polskich znaków, połączenie z zewnętrznym plikiem arkusza stylów oraz podział strony na bloki za pomocą znaczników `<div>`. Kod gotowego pliku pokazano na rys. 13.4.

Kolejnym krokiem stanie się utworzenie pliku `rada_klasowa.css`, zawierającego styl formatowania poszczególnych elementów strony według wytycznych. Efekt pracy pokazano na rys. 13.5.


```

1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4   <meta charset="utf-8"/>
5   <title>Rada klasowa klasy 2TA</title>
6   <link rel="stylesheet" type="text/css" href="rada_klasowa.css">
7 </head>
8 <body>
9   <div id="lewy"><h1>Aktualności</h1>
10    <h3>Co robimy?</h3>
11    <ol>
12      <li>Pomagamy w nauce</li>
13      <li>Informujemy o prawach ucznia</li>
14      <li>Prowadzimy finanse klasy</li>
15    </ol>
16    <h3>Skład Rady:</h3>
17    <table callspecing="0">
18      <tr>
19        <td rowspan="4">Jan Abacki <br>Przewodniczący</td>
20        <td>Anna Babacka</td>
21      </tr>
22      <tr>
23        <td>Anna Cabacka</td>
24      </tr>
25      <tr>
26      </tr>
27      <tr>
28        <td>Krzysztof Nowak</td>
29      </tr>
30    </table>
31  </div>
32  <div id="prawy"><h1>Nasza Klasa</h1>
33    <p></p>
34  </div>
35  <div id="stopka">
36    <p>Autor strony: Jan Abacki</p>
37  </div>
38 </body>
39 </html>

```

Rys. 13.4. Listing pliku rada_klasowa.html

```

1 #lewy
2 {
3   background-color:#ff6633;
4   width:40%;
5   height:600px;
6   float:left;
7 }
8 #prawy
9 {
10  background-color:#ff9933;
11  width:60%;
12  height:600px;
13  float:left;
14 }
15 #stopka
16 {
17  background-color:#993300;
18  color:white;
19  font-size:150%;
20  text-align:right;
21 }
22 td
23 {
24  padding:20px;
25  margin:30px;
26  border: 2px solid brown;
27 }
28 img
29 {
30  height:350px;
31  width:450px;
32 }
33 h1
34 {
35  text-align:center;
36 }

```

Rys. 13.5. Listing pliku rada_klasowa.css

Po wczytaniu pliku `rada_klasowa.html` do przeglądarki internetowej jest widoczny efekt końcowy, pokazujący współpracę między dokumentem hipertekstowym a połączonym z nim zewnętrznym kaskadowym arkuszem stylów (rys. 13.6).



Rys. 13.6. Przykład strony internetowej z podziałem na bloki

SPRAWDŹ SWOJĄ WIEDZĘ

1. Na rys. 13.2 przedstawiono kod dokumentu hipertekstowego tworzącego formularz sformatowany za pomocą CSS. Przeanalizuj ten kod i zastanów się, czy można by było go uprościć po przeniesieniu niektórych elementów formatowania inline do stylu wewnętrznego, tworząc wspólny selektor.
Utwórz nową wersję tego dokumentu. Uwzględnij własne poprawki. Omów z nauczycielem otrzymany efekt.
2. Utwórz szablon strony internetowej w pliku `index.html` oraz zewnętrzny arkusz stylów w pliku `przyklad.css`. Stronę internetową utwórz według następujących wytycznych:
 - zastosowany właściwy standard kodowania polskich znaków;
 - podane informacje o autorze strony;
 - tytuł strony widoczny na karcie przeglądarki: „Przykład strony internetowej”;
 - arkusz stylów w pliku o nazwie `przyklad.css`, właściwie połączony z kodem strony;
 - podział strony na bloki: baner, panele lewy i prawy, stopka – zrealizowany za pomocą znaczników sekcji;
 - zawartość banera: nagłówek pierwszego stopnia o treści: „Mój pierwszy szablon strony internetowej”.

Zewnętrzny arkusz stylów utwórz według następujących wytycznych:

- baner: kolor RGB tła (66, 22, 42), biały kolor czcionki, rozmiar czcionki 140%, wyrównanie tekstu do środka, wysokość 100 px;
 - panel lewy: kolor RGB tła (142, 115, 110), szerokość 30%, wysokość 600 px, biały kolor czcionki;
 - panel prawy: kolor RGB tła (230, 230, 20), szerokość 70%, wysokość 600 px, rozmiar czcionki 120%;
 - stopka: kolor RGB tła (142, 115, 110), biały kolor czcionki, wyrównanie tekstu do prawej strony, wysokość 100 px;
 - niewymienione właściwości obiektów przybierają wartości domyślne.
3. Wykonaj menu do interfejsu aplikacji odtwarzającej i nagrywającej obraz wideo zgodnie ze wzorem. Wykorzystaj do tego celu wyłącznie język HTML oraz wewnętrzny kaskadowy arkusz stylów CSS. Efekt swojej pracy zapisz w pliku o nazwie menu.html.



4. Przeanalizuj kod strony internetowej z osadzonym wewnętrznym arkuszem stylów, który pokazano na rysunku. Znajdź i popraw błędy składni kodu.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta name="utf-8">
5    <meta http-equiv="Content-Language" content="pl">
6    <meta name="keywords" content="">
7  <style>
8    <!--nieodwiedzony link-->
9    a:link {
10      color: #FEF;
11    }
12
13    <!--odwiedzony link-->
14    a:visited {
15      color: bardzojasnozielony;
16    }
17
18    /* najechany kursorem link */
19    a:hover {
20      color: #hotpink;
21    }
22
23    /* wybrany link */
24    a:active {
25      color: blue;
26    }
27  </style>
28 </head>
29 <body>
30 <video width=500 height=400 controls>
31   
32 </video>
33 <p><b><a href="index.php">To jest link do mojej strony</a></b></p>
34 </body>
35 </html>

```


5. Uzupełnij poprawiony kod w taki sposób, aby odsyłacz do strony **index.php** można było uruchomić przyciskiem, którego wzór pokazano na rysunku. Formatowanie przycisku wykonaj w oparciu na kaskadowych arkuszach stylów.

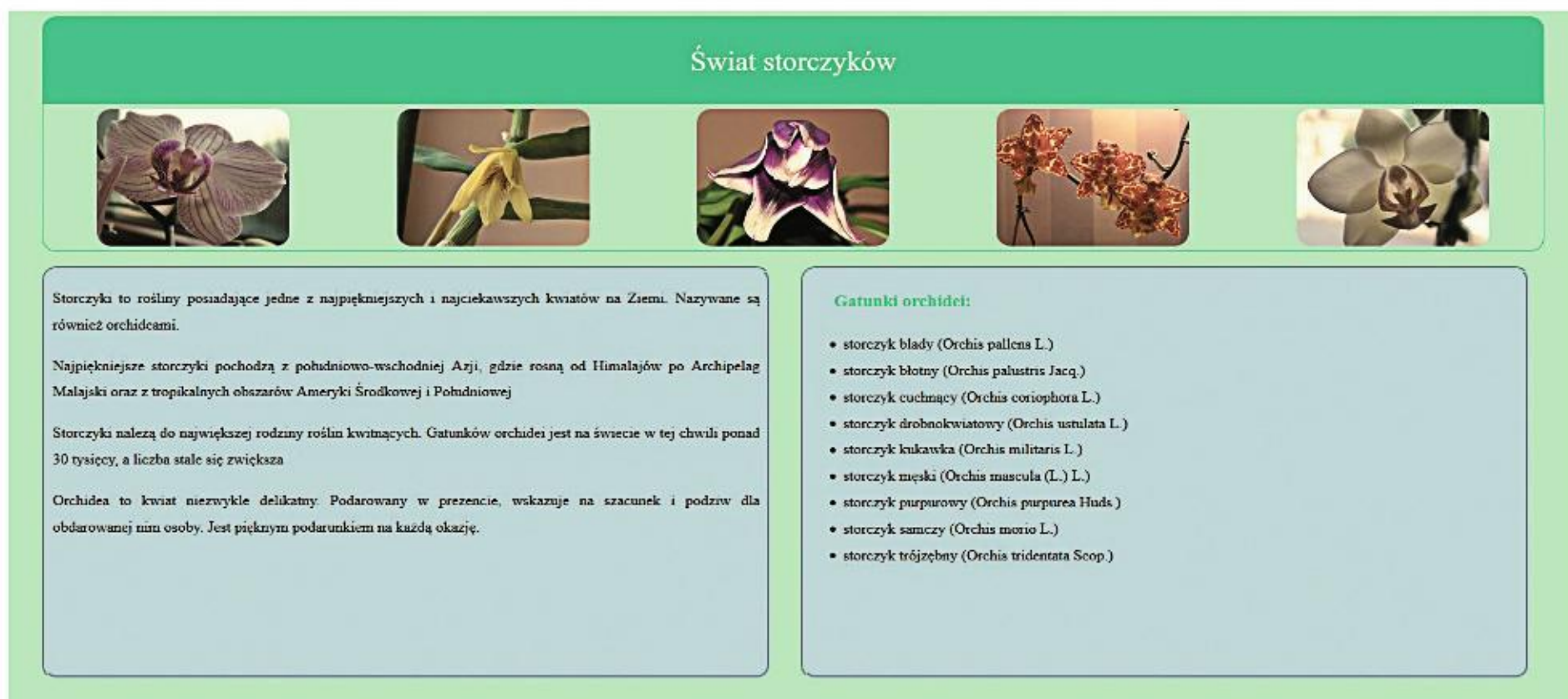
Moja strona

6. Utwórz dokument hipertekstowy opisujący Góry Stołowe. Dodaj do niego rysunek ilustrujący piękno tego pasma górskiego. Za pomocą zewnętrznego kaskadowego arkusza stylów sformatuj tekst i obraz według następujących wytycznych:

- tytuł strony „Góry Stołowe” w nagłówku drugiego stopnia;
- tytuł strony w kolorze zielonym oraz wyśrodkowany;
- obraz o wymiarach 400 px na 400 px;
- wyrównanie obrazu do prawej strony.

Gotowy plik hipertekstowy zapisz pod nazwą **index.html**, a plik z zewnętrznym arkuszem stylów zapisz pod nazwą **style.css**. Zaprezentuj nauczycielowi efekt pracy w przeglądarce internetowej.

7. Wykonaj witrynę internetową przedstawioną na rysunku.



Cechy CSS witryny internetowej

Plik **style.css** zawiera formatowanie dla:

- sekcji głównej: kolor tła #CCFFCC;
- głównego kontenera, w którym znajdują się pozostałe elementy strony: szerokość 90%, wysokość 90%, marginesy zewnętrzne ustawione na auto;
- banera: kolor tła #66CC99, kolor czcionki biały, rozmiar czcionki 30 px, wyrównanie tekstu do środka, wysokość banera 8%, górny margines wewnętrzny 2%, zaokrąglenie rogów górnego lewego i prawego na 15 px;

- blok pod banerem: wysokość 20%, rozmiar czcionki 20 px, wyrównanie tekstu do środka, obramowanie lewe, prawe i dolne – kolor #66CC99, linia ciągła grubości 1 px;
- paneli lewego oraz prawego: kolor tła #818181, szerokość 48%, wysokość 55%, wyrównanie tekstu do środka, kolor czcionki biały, górny margines zewnętrzny 1%;
- dodatkowo w prawym panelu: odstępy między wierszami 1,8 em,
- paragrafów: wyjustowanie tekstu, marginesy wewnętrzne lewy i prawy 1%, odstępy między wierszami 1,8 em;
- nagłówków trzeciego stopnia: kolor czcionki #66CC99, wcięcie w tekście 4%;
- obrazków: szerokość 200 px, wysokość 150, zaokrąglenie rogów na 15 px, zastosowany filtr sepia na 60%;
- tabeli: wysokość i szerokość 100%, wyrównanie tekstu do środka.

Cechy witryny:

- nazwa pliku: **index.html**;
- zastosowany standard kodowania polskich znaków;
- tytuł strony widoczny na karcie przeglądarki: „Świat storczyków”;
- arkusz stylów w pliku o nazwie **style.css** właściwie połączony z kodem strony;
- podział strony na bloki: główny kontener, baner, blok pod banerem, panel lewy, panel prawy – zrealizowany za pomocą znaczników sekcji zgodnie z rysunkiem;
- zawartość banera: tekst „Świat storczyków”;
- zawartość bloku pod banerem: wstaw tabelę z dowolnymi zdjęciami storczyków;
- zawartość panelu lewego: wstaw poniższy tekst i podziel go na trzy akapity:
Storczyki to rośliny mające jedne z najpiękniejszych i najciekawszych kwiatów na Ziemi. Nazywane są również orchideami. Najpiękniejsze storczyki pochodzą z południowo-wschodniej Azji, gdzie rosną od Himalajów po Archipelag Malajski, oraz z tropikalnych obszarów Ameryki Środkowej i Ameryki Południowej. Storczyki należą do największej rodziny roślin kwitnących. Gatunków orchidei jest na świecie w tej chwili ponad 30 tysięcy, a ich liczba stale się zwiększa. Orchidea to kwiat niezwykle delikatny. Podarowany w prezencie, wskazuje na szacunek i podziw dla obdarowanej nim osoby. Jest pięknym podarunkiem na każdą okazję.
- zawartość panelu prawego: wstaw nagłówek trzeciego stopnia o treści „Gatunki orchidei”. Poniżej wykaz nieuporządkowany, w którym wstaw następujące elementy:
 - storczyk błądy (*Orchis pallens* L.);
 - storczyk błotny (*Orchis palustris* Jacq.);
 - storczyk cuchnący (*Orchis coriophora* L.);
 - storczyk drobnokwiatowy (*Orchis ustulata* L.);
 - storczyk kukawka (*Orchis militaris* L.);
 - storczyk męski (*Orchis mascula* (L.) L.);
 - storczyk purpurowy (*Orchis purpurea* Huds.);
 - storczyk samczy (*Orchis morio* L.);
 - storczyk trójzębny (*Orchis tridentata* Scop.).

8. Wykonaj witrynę internetową przedstawioną na rysunku.



Cechy CSS witryny internetowej

Plik **style1.css** zawiera formatowanie dla:

- sekcji głównej: kolor tła #FFCC66;
- głównego kontenera, w którym znajdują się pozostałe elementy strony: szerokość 90%, wysokość 90%, marginesy zewnętrzne ustawione na auto;
- blok pionowy: szerokość 25%, wyrównanie tekstu do środka, margines zewnętrzny lewy 5%, margines zewnętrzny prawy 2%;
- blok zawierający grafikę: margines zewnętrzny górny 5%, wyrównanie tekstu do środka, obramowanie podwójne, grubości 2 px koloru #CC6633;
- blok zawierający opis danego miasta: wysokość 30%, kolor tła #FFFFCC, margines zewnętrzny górny 1%, margines wewnętrzny górny 2%;
- paragrafów: wyrównanie tekstu, kolor czcionki #330000, rozmiar czcionki 14 px, odstępy między wierszami 1,5 em, margines zewnętrzny 4%;
- nagłówków drugiego stopnia: kolor czcionki #CC3333, czcionka Verdana;
- obrazków: szerokość 280 px, wysokość 130.

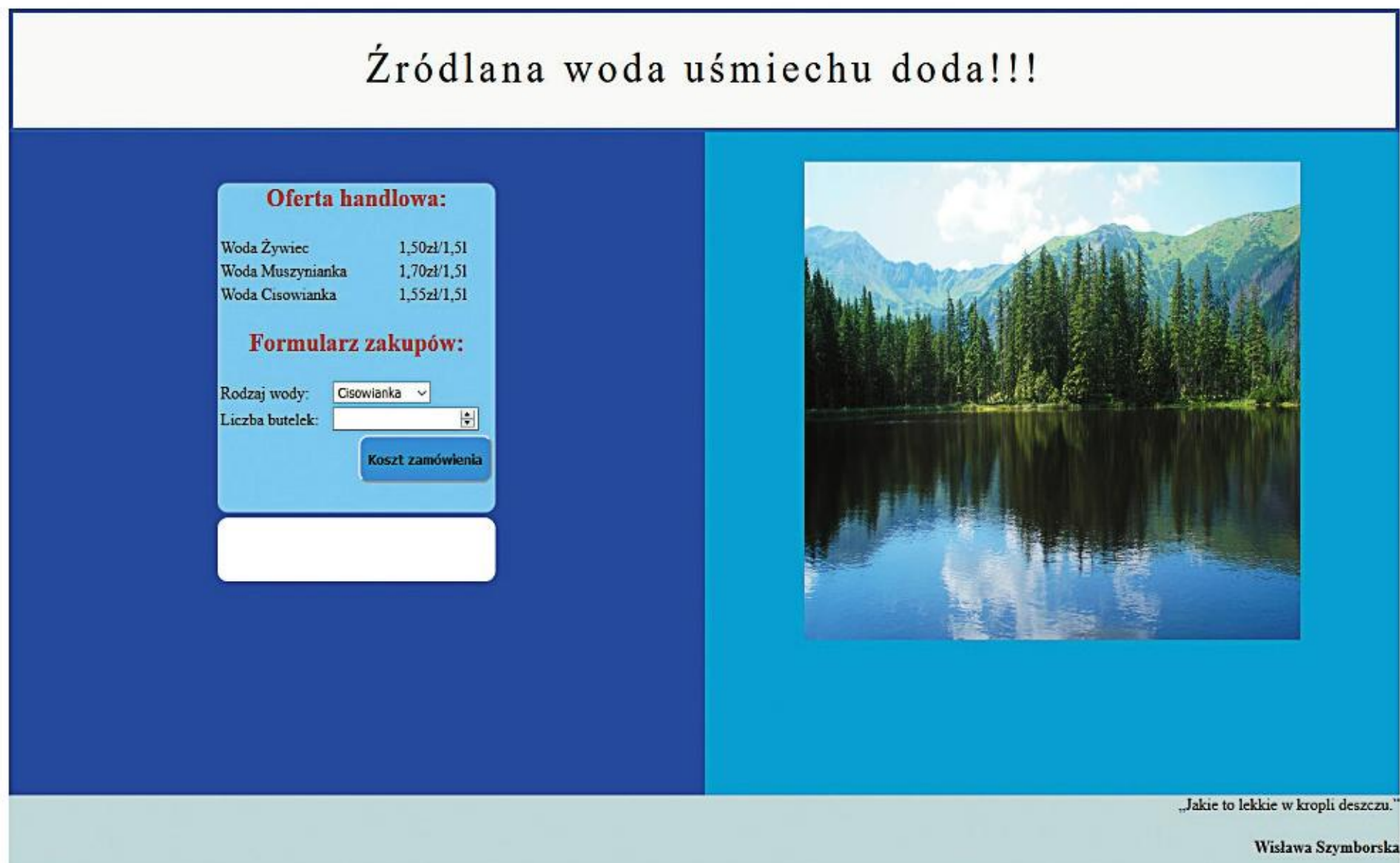
Cechy witryny:

- nazwa pliku: **miasta.html**;
- zastosowany standard kodowania polskich znaków;
- tytuł strony widoczny na karcie przeglądarki: „Zwiedzamy Świat”;
- arkusz stylów w pliku o nazwie **style1.css** prawidłowo połączony z kodem strony;

- podział strony na bloki: główny kontener, trzy bloki pionowe (w każdym z nich wstaw naprzemiennie blok z grafiką i blok opisujący dane miasto, zrealizowany za pomocą znaczników sekcji zgodnie z rysunkiem);
- zawartość bloków pionowych: wstaw naprzemiennie blok ze zdjęciem danego miasta (pobierz z internetu dostępne zdjęcia lub wykorzystaj własne) i blok z opisem danego miasta. Opis ma zawierać dwa zdania umieszczone w akapicie. Przed opisem w nagłówku drugiego stopnia wstaw nazwę miasta.

Do wykonania ćwiczenia możesz wybrać sześć dowolnych miast.

9. Wykonaj witrynę internetową przedstawioną na rysunku.



Cechy CSS witryny internetowej

Plik **style2.css** zawiera formatowanie dla:

- głównego kontenera, w którym znajdują się pozostałe elementy strony: szerokość 80%, wysokość 80%, marginesy zewnętrzne ustawione na auto;
- banera: kolor tła #f9f9f9, rozmiar czcionki 40 px, wyrównanie tekstu do środka, wysokość banera 10%, górny margines wewnętrzny 2%, odstępy między literami 5 px, obramowanie koloru #3366cc linią typu groove grubości 4 px;
- panelu lewego: kolor tła #003399, szerokość 50%, wysokość 74%, górny margines wewnętrzny 2;
- panelu prawego: kolor tła #3399CC, szerokość 50%, wysokość 74%, górny margines wewnętrzny 2%, wyrównanie tekstu do środka;
- stopki: kolor tła #cfd8dc, wysokość 9%, wyrównanie tekstu do prawej strony;
- blok z formularzem: marginesy wewnętrzny i zewnętrzny auto, wysokość 310 px, szerokość 280 px, kolor tła #99CCFF, zaokrąglenie rogów 10 px;

- blok pod formularzem: marginesy wewnętrzny i zewnętrzny auto, szerokość 280 px, wysokość 60 px, białe tło, zaokrąglenie rogów 10 px, wyrównanie tekstu do środka;
- nagłówek drugiego stopnia: wyrównanie tekstu do środka, czcionka koloru #990000;
- obrazków: szerokość 500 px, wysokość 450 px;
- tabeli: szerokość 100%, wyrównanie tekstu do lewej;
- przycisku: kolor tła #3399ff, wysokość 45 px, czcionka pogrubiona, zaokrąglenie rogów 10 px.

Cechy witryny:

- nazwa pliku: **woda.html**;
- zastosowany standard kodowania polskich znaków;
- tytuł strony widoczny na karcie przeglądarki: „Hurtownia wody źródlanej”;
- arkusz stylów w pliku o nazwie **style2.css** właściwie połączony z kodem strony;
- podział strony na bloki: główny kontener, baner, panel lewy, panel prawy (w nim blok z formularzem i blok pod formularzem) oraz stopka – zrealizowany za pomocą znaczników sekcji zgodnie z rysunkiem;
- zawartość banera: tekst „Źródłana woda uśmiechu doda!!!”;
- zawartość panelu lewego:
 - blok formularza: nagłówek drugiego stopnia o treści „Oferta handlowa:”. Pod tekstem umieść tabelę złożoną z trzech wierszy i dwóch kolumn:

woda Żywiec	1,50 zł/1,5 l
woda Muszynianka	1,70 zł/1,5 l
woda Cisowianka	1,55 zł/1,5 l

 Pod tabelą nagłówek drugiego stopnia o treści „Formularz zakupów:”. Pod spodem formularz umieszczony w tabeli zawierający listę z trzema nazwami wód źródłanych, pole numeryczne oraz przycisk „Koszt zamówienia”;
 - blok pod formularzem pozostaw pusty;
- zawartość panelu prawego: dowolna grafika przedstawiająca krajobraz górski;
- zawartość stopki: cytat „Jakie to lekkie w kropli deszczu”; poniżej w nagłówku czwartego stopnia „Wisława Szymborska”.

V. Systemy zarządzania treścią – CMS

- Publikacja stron i aplikacji internetowych
- Rodzaje systemów CMS
- Wybór i instalacja CMS
- Personalizacja systemu CMS
- Zarządzanie treścią w systemie CMS

14

Publikacja stron i aplikacji internetowych

ZAGADNIENIA

- Serwery WWW
- Usługi hostingowe
- Budowa adresu domenowego
- Kopiowanie plików strony na serwer

Jednym z ostatnich etapów pracy nad stroną internetową jest jej publikacja. Aby strona internetowa była publicznie dostępna, musi zostać opublikowana na serwerze WWW oraz mieć unikalny publiczny adres domenowy będący alfanumeryczną konwersją adresu IP. Adres domenowy wpisuje się w pasek adresu przeglądarki internetowej w końcowej części adresu URL.



Rys. 14.1. Budowa adresu domenowego

Serwer WWW udostępnia swoje zasoby dyskowe do przechowywania plików składających się na stronę internetową oraz obsługuje żądania protokołu komunikacyjnego HTTP, który pozwala na przesyłanie dokumentów hipertekstowych w sieci Internet. Dodatkowo serwer WWW oferuje wiele usług pozwalających na stabilną i bezpieczną dostępność strony internetowej w sieci oraz oprogramowanie backendowe, ułatwiające zarządzanie witryny. Do najpopularniejszych serwerów zalicza się darmowy **Apache** oraz **IIS** firmy Microsoft. Oprócz serwera WWW, obsługa stron i aplikacji wymaga instalacji dodatkowych pakietów umożliwiających obsługę skryptów PHP i baz danych, np. SQL. Na rynku jest dostępnych kilka zintegrowanych pakietów serwerowych zawierających wszystkie te usługi. W następnych rozdziałach zostanie opisany jeden z nich – XAMPP.

Stronę internetową można publikować na serwerze własnym lub skorzystać z usług jednego z dostawców usług hostingowych (ang. *provider*) obecnych na rynku. Jeżeli

zdecydujemy się na własny serwer, musimy pamiętać, że do tego celu trzeba wydzielić osobny komputer pracujący przez 24 godz. na dobę oraz dysponować zewnętrznym (publicznym) adresem IP „widocznym” w sieci Internet.

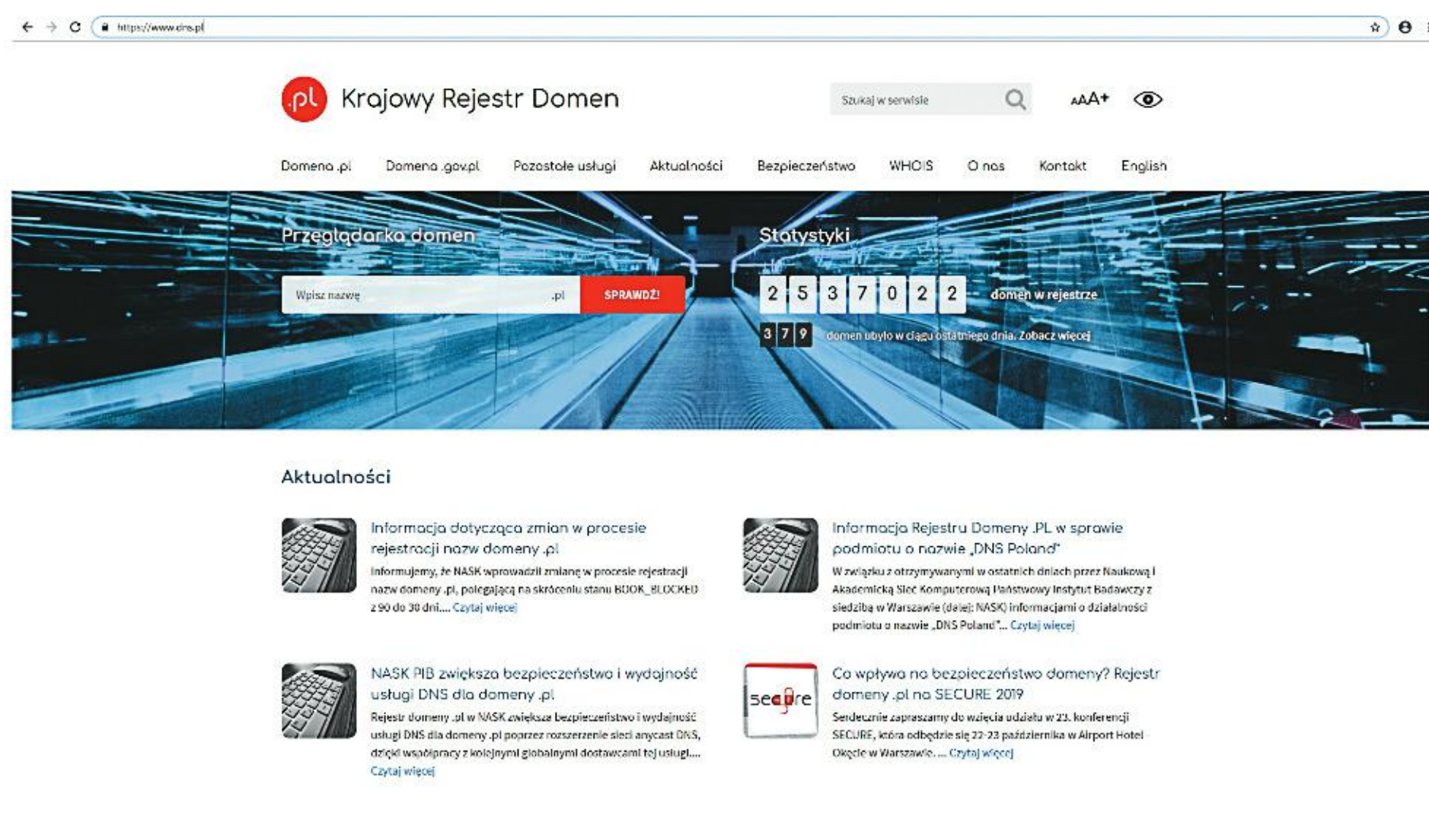
Usługa hostingu jest udostępniania przez dostawcę usług internetowych zasobów serwerów oraz technologii pozwalających na publikację stron i aplikacji internetowych. Za dodatkową opłatą dostawcy oferują wsparcie techniczne, usługi zabezpieczające (certyfikat SSL, backup strony) oraz FTP.

Zewnętrzny serwer WWW w ramach usługi hostingowej jest wygodniejszym i bezpieczniejszym rozwiązaniem, ale z reguły wiąże się z opłatami. Często jednak w jednym miejscu otrzymujemy zarówno gotowy do pracy serwer WWW, jak i adres domenowy widoczny w sieci Internet.

Jeżeli naszym celem jest posiadanie domeny pozwalającej na dobre pozycjonowanie, możemy samodzielnie ją zdefiniować i zarejestrować w Krajowym Rejestrze Domen prowadzonym przez NASK. Rejestr domen jest dostępny pod adresem <https://www.dns.pl/>.

Na stronie można sprawdzić dostępność domen najwyższego i drugiego poziomu oraz zarejestrować i opłacić wybraną domenę. Jeżeli zarejestrujemy domenę w NASK bądź u innego operatora domen lokalnych, możemy ją delegować do naszego dostawcy usług hostingowych.

Następnym krokiem jest połączenie naszej strony w serwerze WWW z naszym adresem domenowym i wgranie plików tworzących stronę do folderu utworzonego na serwerze WWW.



Rys. 14.2. Strona Krajowego Rejestru Domen

Do kopiowania plików na serwer zdalny można wykorzystać narzędzia dostępne z poziomu panelu administratora usługi hostingowej lub własnego klienta usługi **FTP** (ang. *File Transfer Protocol*).

Na potrzeby serwera lokalnego (naszego komputera PC podniesionego do roli serwera WWW) wystarczy skopiowanie plików do odpowiedniego folderu. W przypadku własnego serwera Apache można kopiować pliki strony do folderu `/var/www/html`. Jeżeli korzysta się z pakietu serwerowego XAMPP, pliki kopiuje się do folderu `C:\xampp\htdocs\`.

W przypadku serwerów zdalnych (dostępnych po wykupieniu usługi hostingowej) provider powinien poinformować, w którym miejscu znajduje się folder do kopiowania plików strony, oraz podać parametry logowania do usługi FTP. Najczęściej w takiej sytuacji serwery WWW mają foldery przypisane do użytkowników:

domains/nazwadomeny/public_html

Należy również pamiętać o tym, że domyślnie plik startowy powinien mieć nazwę **index** z rozszerzeniem `.html` lub `.php`.

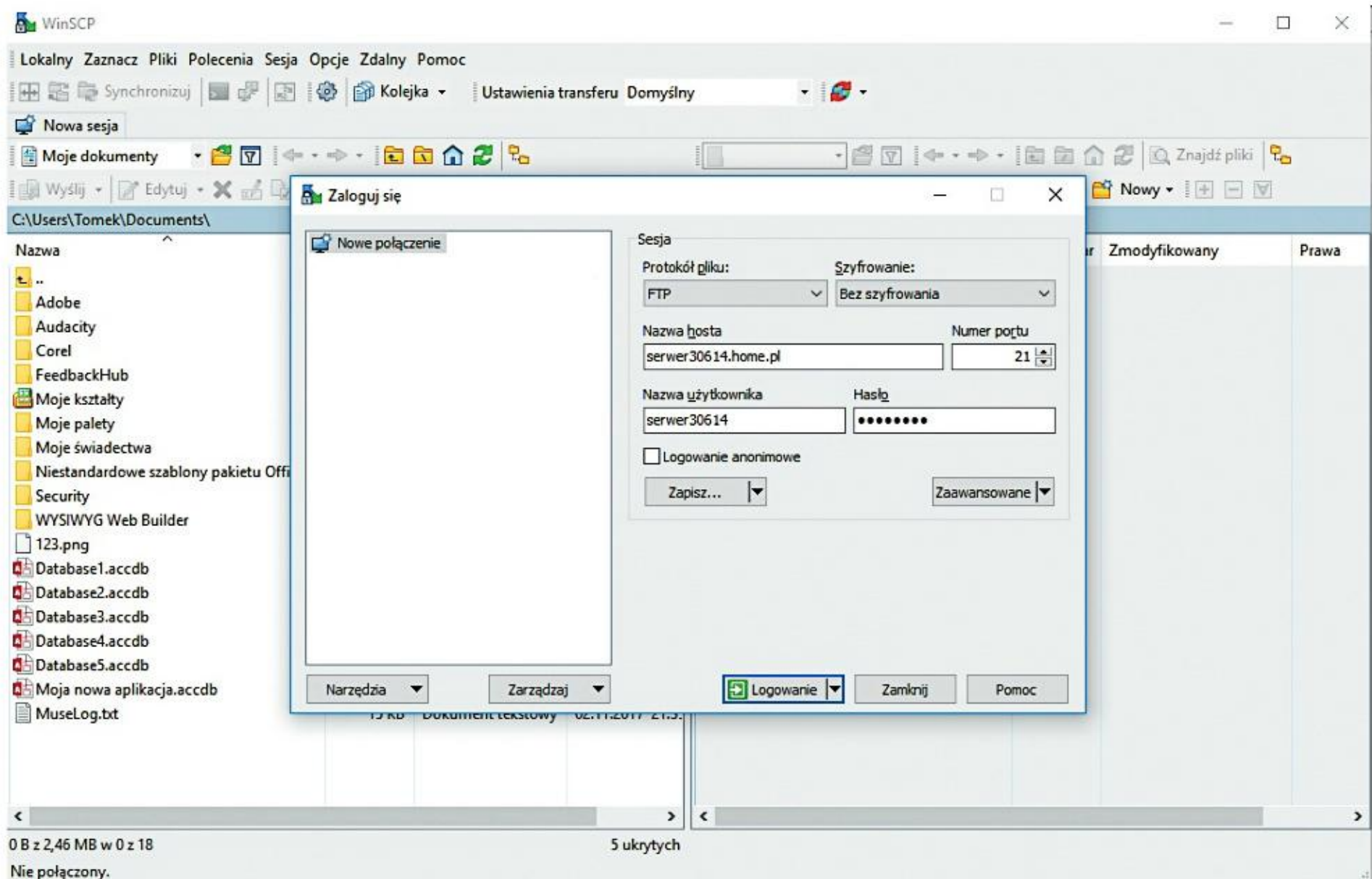
Aby przesłać pliki do folderów na zdalnych serwerach, należy zainstalować klienta FTP. Najczęściej stosuje się programy:

- Total Commander (<http://totalcmd.pl/>),
- FileZilla (<https://filezilla-project.org/>),
- WinSCP (<https://winscp.net>).

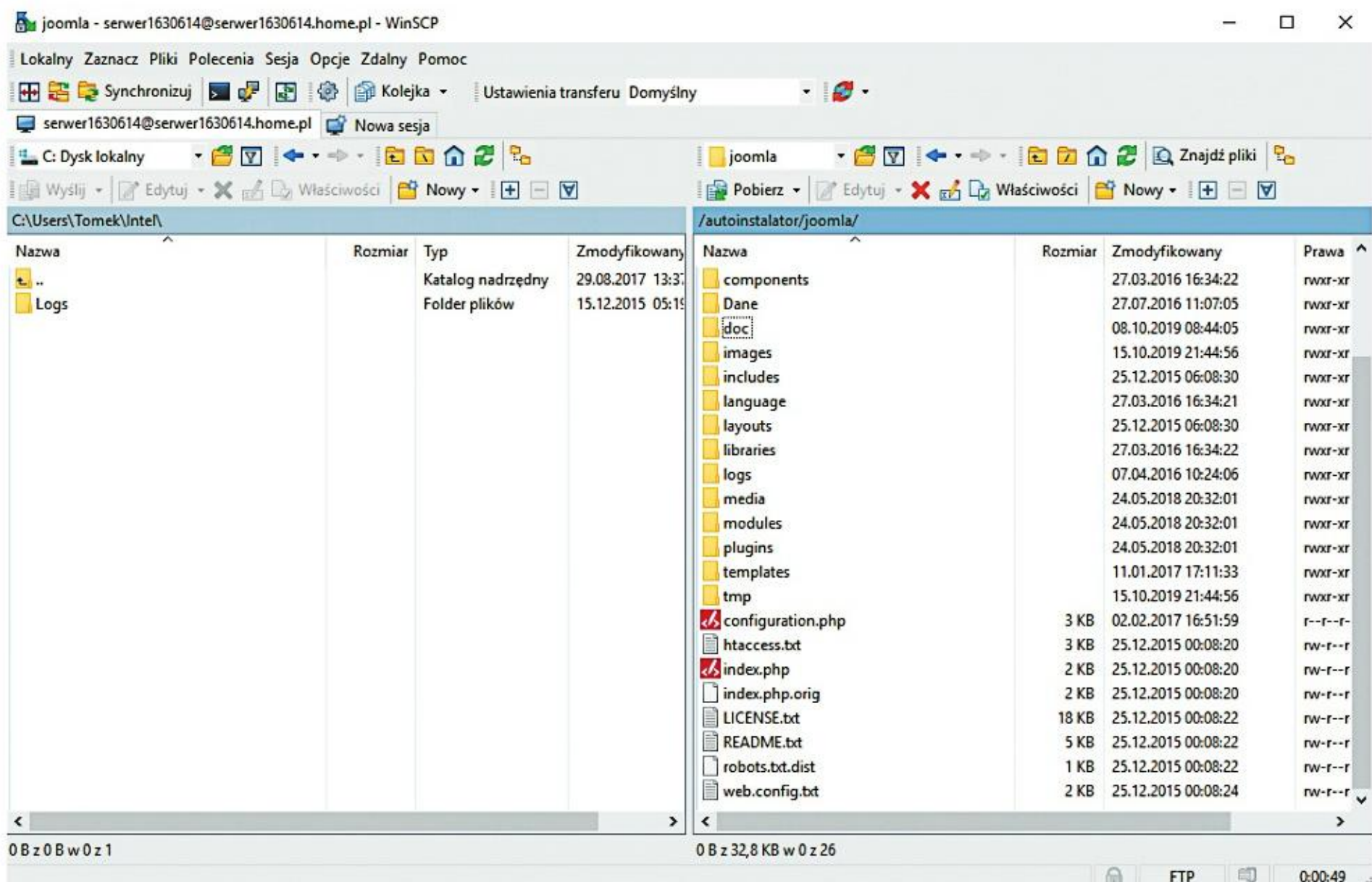
Do ustalenia połączenia należy skonfigurować naszego klienta FTP – ustawić połączenie przez podanie następujących informacji:

- nazwa serwera FTP (np. `serwer30614.home.pl`),
- nazwa użytkownika (np. `serwer30614`),
- hasło użytkownika,
- numer portu,
- opcje szyfrowania (opcjonalnie).

Po wykonaniu konfiguracji klienta FTP można połączyć się z naszym serwerem WWW, aby wgrać pliki tworzące stronę internetową. Po zalogowaniu mamy dostępne dwa okna. W jednym oknie są widoczne drzewo katalogowe i pliki naszego lokalnego komputera. W drugim oknie możemy zobaczyć zawartość naszego katalogu udostępnionego dzięki usłudze hostingowej. Teraz można kopiować dowolne pliki między oknami. Po zaznaczeniu pliku wystarczy uruchomić jego przesyłanie za pomocą poleceń menu podręcznego albo klawiszy funkcyjnych (np. `[F5]`). W trakcie operacji kopiowania plików (zwłaszcza na serwery linuxowe) zaleca się nieużywanie wielkich liter, znaków diakrytycznych oraz znaków specjalnych.



Rys. 14.3. Przykład konfiguracji klienta FTP



Rys. 14.4. Przykład połączenia z serwerem WWW za pomocą klienta FTP

Po skopiowaniu plików należy sprawdzić poprawność działania strony – wpisać w przeglądarce internetowej adres domenowy.

W ten sam sposób można korzystać z gotowych rozwiązań aplikacji internetowych zwanych CMS. Są one oferowane również przez dostawców hostingowych jako rozwiązania dedykowane i dobrze współpracujące z zasobami serwera.

SPRAWDŹ SVOJE UMIEJĘTNOŚCI

1. Pobierz i zainstaluj wybraną aplikację klienta FTP. Skonfiguruj nowe połączenie według danych dostarczonych przez nauczyciela. Wykonaj połączenie z serwerem FTP.
2. Zastanów się, jaki adres byłby najlepszy dla twojej strony internetowej. Sprawdź w katalogu domen NASK, czy taka nazwa domeny jest dostępna oraz jaka jest cena za jej rejestrację.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Co to jest usługa hostingowa?
2. Gdzie można zarejestrować własną domenę dla strony internetowej?
3. Jaka jest różnica między adresem sieciowym prywatnym a publicznym?
4. Do czego służy klient FTP?

15

Rodzaje systemów zarządzania treścią

ZAGADNIENIA

- Definicja CMS
- Rodzaje systemów zarządzania treścią
- Wybór CMS do określonych zadań

Co to jest CMS?

System zarządzania treścią **CMS** (ang. *Content Management System*) jest najczęściej stosowanym rozwiązaniem podczas tworzenia mniej lub bardziej zaawansowanych witryn internetowych. Są to przeważnie gotowe rozwiązania, czyli grupa aplikacji internetowych do tworzenia, zarządzania i personalizacji stron internetowych. Podstawowym celem oprogramowania CMS jest oddzielenie zawartości informacyjnej strony internetowej od sposobu jej prezentacji. W praktyce wygląda to tak, że po wprowadzeniu nowych treści za pomocą artykułu przez uprawnionego użytkownika zapisuje się ją w bazie danych. Następnie system zarządzania treścią generuje dynamicznie strony internetowe na podstawie treści pochodzącej z bazy danych oraz zadeklarowanych szablonów. Pozwala to na bardziej elastyczne, czytelne i wygodniejsze zarządzanie stroną internetową niż w przypadku zastosowania statycznych stron internetowych.

Systemy zarządzania treścią buduje się na podstawie bazy danych, serwerów WWW i języków skryptowych wykonywanych po stronie serwera (ang. *server-side*), np. PHP. Do obsługi systemu, np. aktualizacji strony, może wymagać użycia oprogramowania klienckiego korzystającego z takich protokołów, jak FTP czy SFTP, natomiast do edycji artykułów – z wbudowanych edytorów WYSIWYG.

Istotnym elementem budowy systemu CMS są **templatki** (ang. *template*), czyli gotowe szablony strony, które definiują prezentację treści. Wykorzystanie templatek sprawia, że zmiana koncepcji graficznej całego serwisu sprowadza się do przygotowania i zamiany szablonu. W skład systemu CMS wchodzi również gotowe, intuicyjne panele zarządzania stroną. Dzięki temu proces publikacji strony w internecie i jej obsługa są wygodniejsze i prostsze. Użytkownik nie musi ingerować bezpośrednio w kod strony, a tym samym nie wymaga się od niego znajomości wszystkich zastosowanych w CMS technologii. Dodatkowo, treści na stronie można prezentować jednocześnie w różnych formatach, np. PDF.

Najczęściej systemy CMS są pisane w językach: HTML, ASP, JSP, PHP i Python.

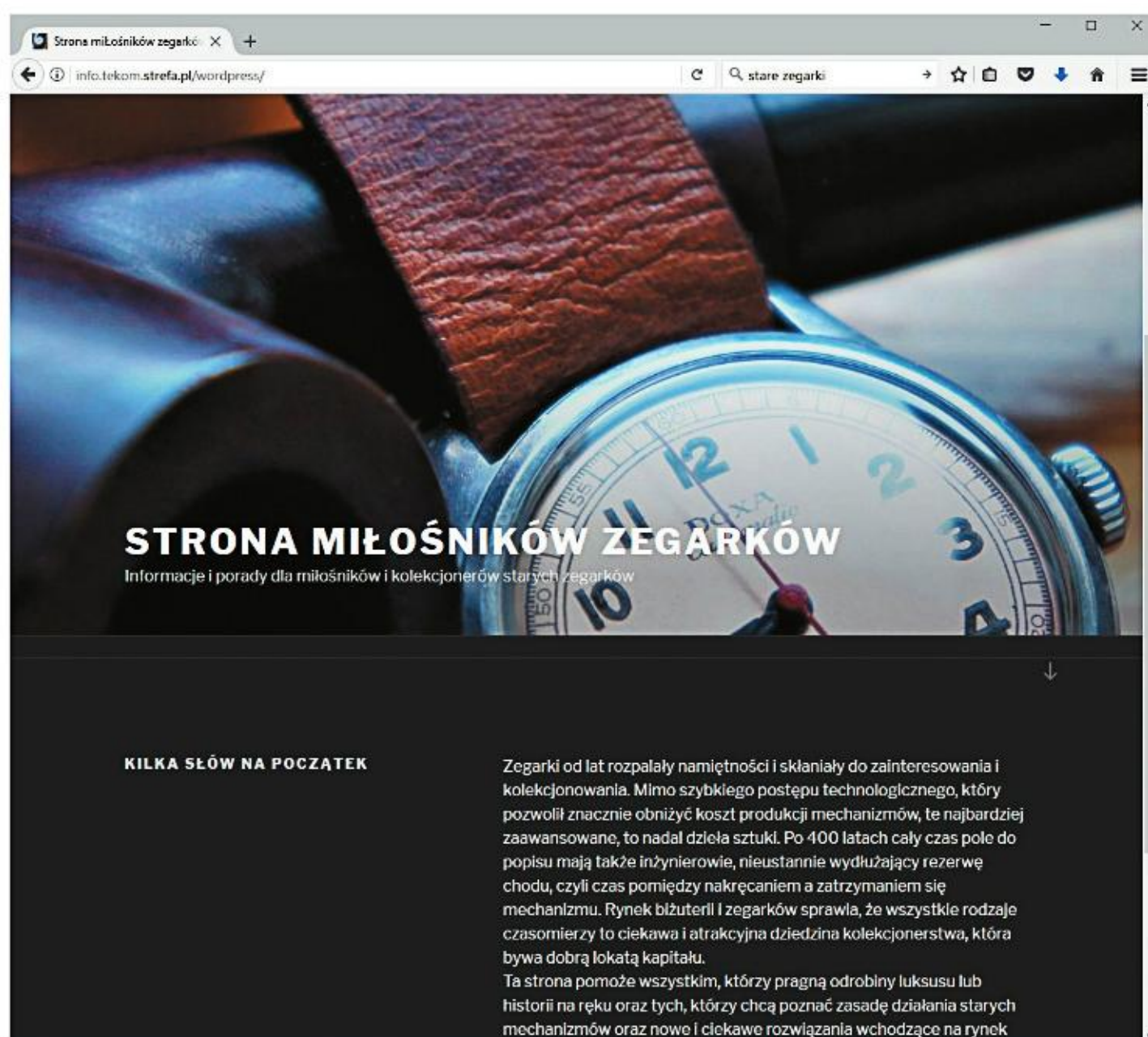
Rodzaje systemów CMS

Tworzenie systemów zarządzania treścią jest często spotykanym produktem w projektach internetowych, dlatego można spotkać wiele rodzajów tego oprogramowania. Ze względu na to, kto i w jakim celu tworzy CMS oraz jakie są jego warunki rozpowszechniania i licencjonowania, rozróżnia się:

- **systemy komercyjne** – płacone i tworzone pod zamówienie konkretnego klienta; często nazywane systemami o zamkniętym kodzie źródłowym; charakteryzują się zwiększonym zakresem bezpieczeństwa, indywidualnym szablonem (templatką) oraz okrojoną liczbą dodatków (ang. *widgets*) i wtyczek (ang. *plugins*);
- **open source** – bezpłatne systemy z otwartym kodem źródłowym oferujące bardzo dużą liczbę dodatków, aplikacji i wtyczek;
- **systemy własne** – autorskie CMS-y do prywatnego stosowania.

Ze względu na budowę systemu CMS rozróżnia się:

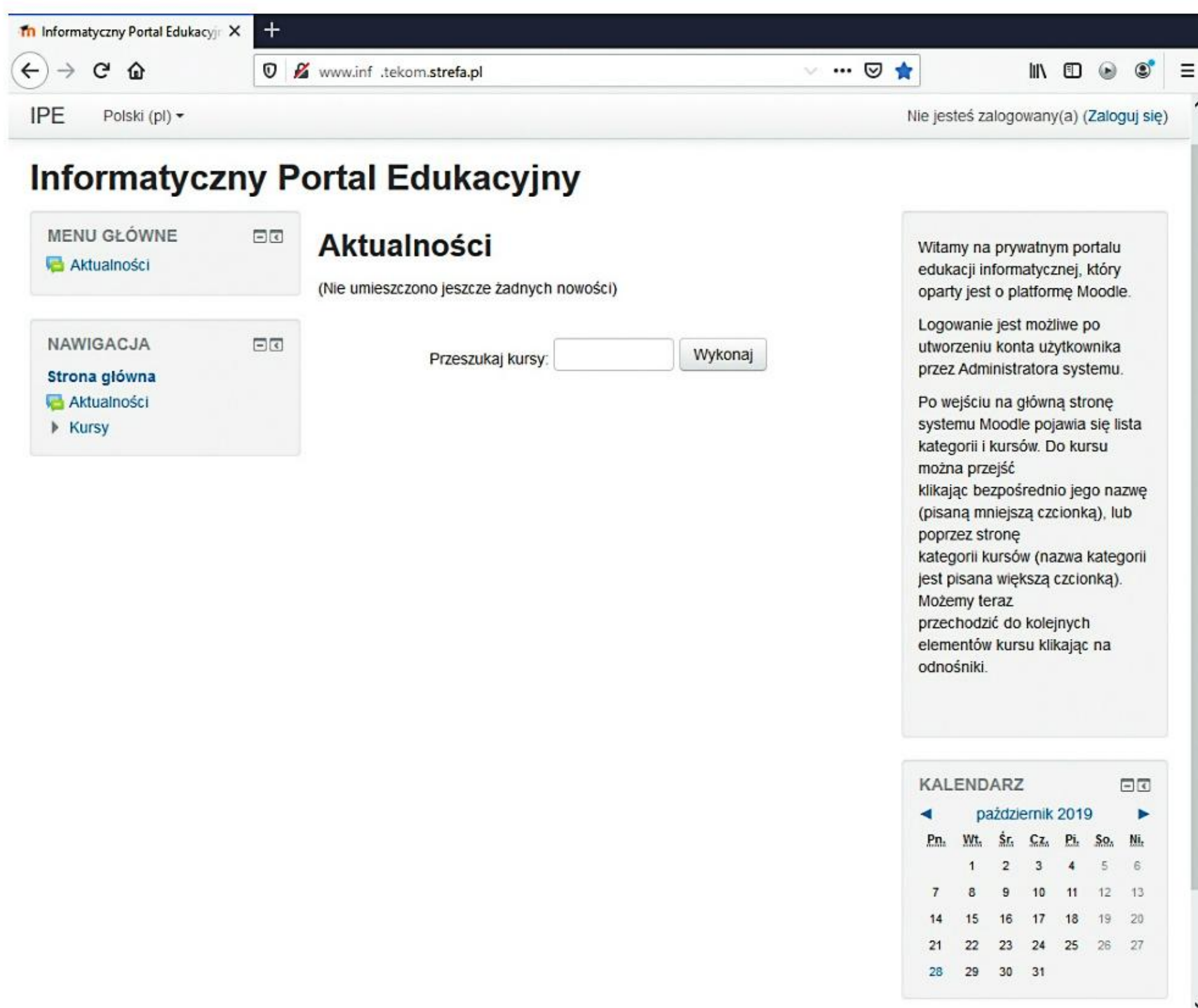
- **content management frameworks** – narzędzie do budowy CMS-ów, czyli zbiór aplikacji potrzebnych do projektowania i programowania systemu CMS;
- **page-based systems** – systemy o transparentnych konsolach; umożliwiają edycję w ciele strony, nie wymagają konsoli do zarządzania treścią;
- **module-based systems** – systemy CMS oparte na modułach; najbardziej popularny rodzaj CMS;
- **content object systems** – systemy koncentrujące się wokół tzw. obiektów, czyli małych części informacji, które można reprodukować w wielu miejscach strony WWW.



Rys. 15.1. Przykład CMS WordPress

Ze względu na zastosowanie systemów zarządzania treścią rozróżnia się:

- systemy do tworzenia systemów zarządzania treścią (np. Aplin);
- systemy do zarządzania przepływem dokumentów, wiedzy i zasobów;
- systemy do zarządzania treścią informacyjnej witryny internetowej lub blogu (np. Joomla! bądź WordPress);
- systemy do prezentacji i zarządzania zdjęciami i innymi zasobami multimedialnymi (np. Gallery);
- systemy do tworzenia i zarządzania sklepami internetowymi (np. Magento);
- systemy do organizacji nauczania na odległość (np. Moodle);
- systemy do zarządzania projektami i ich realizacją (np. Redmine).



Rys. 15.2. Przykład CMS Moodle

Wybór systemu CMS

Znaczenie zarówno opisu rodzajów systemów zarządzania treścią, jak i ich funkcjonalności sprawia, że wybór właściwego systemu do konkretnego zastosowania jest kluczowy.

Poza funkcjonalnością systemu ważne jest, aby sprawdzić wymagania danej wersji CMS dotyczące serwera WWW i obsługiwanych aplikacji (np. wersji serwera bazy danych MySQL bądź języka skryptowego PHP).

W przypadku instalacji systemu CMS na serwerze zdalnym warto upewnić się, czy dysponuje się odpowiednią ilością miejsca na dysku i możliwością utworzenia nowej bazy danych.

Najczęściej systemy zarządzania treścią wykorzystuje się do tworzenia informacyjnych stron internetowych i ich obsługi. Do najpopularniejszych zalicza się CMS Joomla! i WordPress. Ich popularność opiera się głównie na: stabilności pracy, bezpieczeństwie, bezpłatnym dostępie oraz ogromnej liczbie dodatków i rozszerzeń, pozwalających na tworzenie różnorodnych witryn do użytku prywatnego oraz firmowego.



SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Sprawdź na stronie <http://www.wordpress-polska.pl>, jakie wymagania dotyczące serwera WWW, SQL-a oraz PHP należy spełnić, aby zainstalować najnowszą wersję CMS WordPress.



SPRAWDŹ SWOJĄ WIEDZĘ

1. Jak nazywają się systemy CMS o transparentnych konsolach?
2. Jakie zastosowanie ma CMS o nazwie Moodle?
3. Wyszukaj w zasobach sieci Internet dwa rozwiązania CMS do tworzenia sklepu internetowego. Porównaj interfejs i funkcjonalność tych rozwiązań.
4. Opisz, czym różnią się CMS-y komercyjne od rozwiązań open source.
5. Co to są templatki i jaką rolę odgrywają w konfiguracji strony internetowej?

16

Instalacja CMS w usłudze hostingowej

ZAGADNIENIA

- Przygotowanie do instalacji CMS
- Rodzaje metod instalacji CMS
- Instalacja ręczna CMS Joomla!
- Instalacja CMS WordPress za pomocą autoinstalatora

Po wyborze systemu CMS należy przygotować się do jego instalacji. Dostępne są różne możliwości publikacji własnej strony internetowej opartej na CMS. Do każdej z nich trzeba się przygotować inaczej. Poniżej opisano metody instalacji systemu CMS z wykorzystaniem:

- zewnętrznej usługi hostingowej (serwera w sieci globalnej);
- lokalnego pakietu serwerowego.

Każde rozwiązanie ma zalety i wady. Hosting lokalny wymaga od administratora większej wiedzy, dedykowania komputera na usługi serwerowe, który praktycznie będzie pracował całą dobę, oraz obsługi i serwisowania w czasie pracy. Hosting zewnętrzny sprowadza się tylko do zarządzania samą stroną. Kwestia sprzętu, serwisowania i zapewnienia bezpieczeństwa leży po stronie usługodawcy, a jakość i zakres usługi praktycznie określają cena i szczegóły umowy.

Wybór usługi hostingowej

Aby strona funkcjonowała, musi istnieć w internecie, dlatego trzeba mieć publiczny adres IP i domenę, pod którą będzie widoczna witryna. Jednym ze sposobów rozwiązania tego problemu jest wykupienie usługi hostingowej. Należy pamiętać, że dobry pakiet hostingowy jest tak samo ważny, jak sama strona czy aplikacja. Na rynku działa wiele firm oferujących różny zakres usług. Różnią się one liczbą baz danych, pojemności konta czy limitu transferu. Mogą również oferować w cenie usługi dostęp do domeny, co oszczędza czas i pracę związaną z wyborem, opłaceniem i transferem domeny z NASK, czyli instytutu krajowego rejestru nazw internetowych.

The screenshot displays the home.pl website's hosting selection interface. At the top, there are navigation links for Domeny, Hosting, Sklepy, Office 365 i Poczta, Aplikacje, and Reklama. A shopping cart icon shows a total of 0,00 zł. Below the navigation, four hosting packages are presented in a grid:

- Hosting Start + homeSSL Start:** Idealny na początek. Price: 22,90 zł netto / rok (Promocja Rabat - 95%). Features: 70 GB storage, unlimited transfer, 25 databases, CMS installer, SSH access, email, and SSL certificate.
- Hosting Biznes + homeSSL Start:** Optymalne rozwiązanie. Price: 9,82 zł netto / rok (Promocja Rabat - 98%). Features: 100 GB storage, unlimited transfer, 50 databases, CMS installer, SSH access, email, and SSL certificate.
- Hosting Profesjonalny + Ochrona WWW Biznes:** Kilka witryn w jednym miejscu. Price: 96,00 zł netto (Promocja Rabat - 88%). Features: 500 GB storage, unlimited transfer, unlimited databases, CMS installer, SSH access, email, SSL certificate, and WWW protection.
- Hosting Premium + Ochrona WWW Biznes:** Zaawansowany projekt i duży ruch. Price: 120,00 zł netto (Promocja Rabat - 90%). Features: 2 TB storage, unlimited transfer, unlimited databases, CMS installer, SSH access, email, SSL certificate, and WWW protection.

Each package has a 'WYBIERAM' (I choose) button at the bottom.

Rys. 16.1. Parametry przykładowych pakietów hostingowych

Jak przedstawiono na rys. 16.1, dostawcy usług hostingowych oferują różne parametry pakietów. Do najważniejszych cech, na które należy zwrócić uwagę w trakcie wyboru hostingu, należą:

- **pojemność serwera** – w zależności od wielkości i funkcji strony należy wybrać pojemność, która zapewni prawidłowe działanie strony i pozwoli umieścić wszystkie ważne dla niej pliki, grafiki oraz inne elementy;
- **przepustowość** łącza internetowego (transfer) – podczas projektowania strony lub aplikacji działającej w internecie należy brać pod uwagę jej docelową popularność; jeżeli transfer danych będzie ograniczony, to dostęp do strony w okresach jej największej popularności będzie dla potencjalnych zainteresowanych także ograniczony;
- **ochrona strony** – np. przed atakami DDoS, których celem jest zablokowanie dostępu do strony; w tym aspekcie zawiera się również tworzenie kopii bezpieczeństwa oraz certyfikacja SSL;
- **dostępność wielu baz danych** – ten czynnik umożliwia budowanie wielu stron internetowych na subdomenach dostępnych pod domeną główną; każdy CMS wymaga bowiem utworzenia dedykowanej bazy danych;
- **dostępność wsparcia** – w razie problemu z funkcjonowaniem strony klient może skontaktować się ze swoim dostawcą, by przywrócić stronę do działania;

- **autoinstalator** – bardzo pomocne narzędzie dla osób niemających dużego doświadczenia z instalowaniem systemów CMS za pomocą klienta FTP; w takich rozwiązaniach okazuje się dostępnych kilka najpopularniejszych CMS-ów w pełni kompatybilnych z daną usługą hostingową; instalacja wybranego pakietu odbywa się bardzo szybko i bezproblemowo;
- **klient FTP** – częstą praktyką dostawców usług hostingowych jest oferowanie klienta FTP online; nie trzeba wtedy instalować tej aplikacji na dysku, a wszystkie operacje kopiowania plików wykonuje się w przeglądarce internetowej;
- **cena** – jeden z głównych parametrów podczas wyboru pakietu hostingowego; zasada jest prosta – niska cena oznacza mniej usług i gorsze parametry; należy pamiętać również o tym, że niska cena może dotyczyć tylko pierwszego roku usługi, w następnych latach opłaty często rosną;
- **rodzaj hostingu (współdzielony bądź dedykowany)** – jeżeli wybierze się hosting współdzielony, to trzeba pamiętać, że zasoby serwera, na którym znajduje się usługa, są dzielone na grupę klientów; serwery dedykowane pozwalają na samodzielność w konfiguracji wielu parametrów oraz ich maksymalne wykorzystanie na potrzeby tylko jednego klienta;
- **jakość zasobów serwerów hostingu** – duże znaczenie ma rodzaj zastosowanego sprzętu do obsługi serwerów usługodawcy hostingu; przykładowo: zastosowanie wydajnych dysków twardych SSD zamiast wolniejszych HDD może istotnie wpływać na jakość działania CMS-a.

Dla naszej strony opartej na systemie zarządzania treścią wybraliśmy pakiet hostingowy Hosting Biznes z certyfikatem SSL.

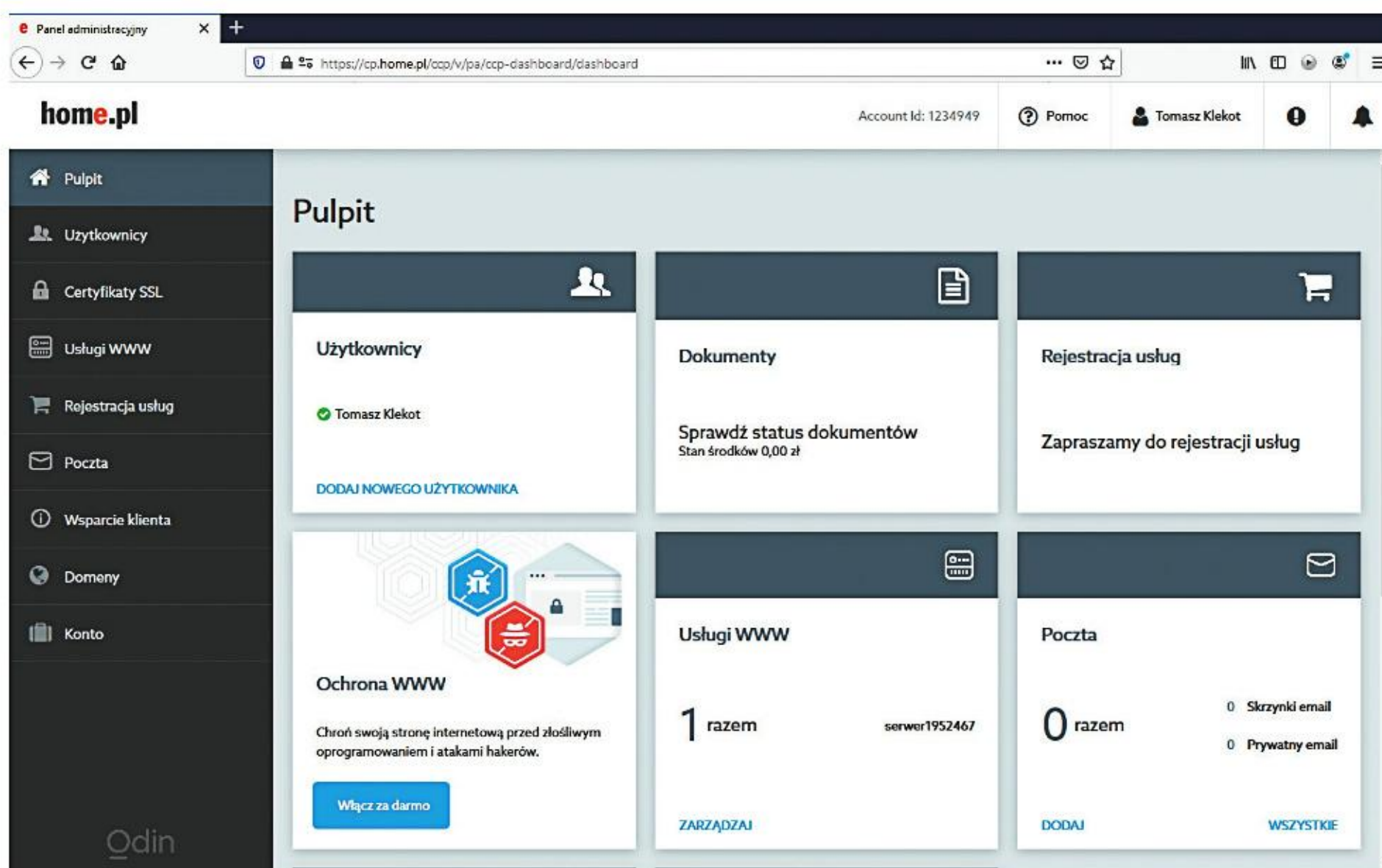
Na naszym serwerze hostingowym, po zarejestrowaniu usługi, przystępujemy do przygotowania ręcznej instalacji za pomocą klienta FTP i przeglądarki internetowej.

UWAGA

Jeżeli mamy domenę kupioną i zarejestrowaną w NASK, należy delegować ją na serwery DNS naszego obecnego usługodawcy. W tym celu trzeba zadeklarować w konfiguracji domeny adresy DNS operatora, do którego delegujemy domenę, a następnie przypisujemy domenę do wykupionej usługi. Zazwyczaj do jednej usługi hostingowej można przypisać wiele domen.

Konfiguracja usługi hostingowej

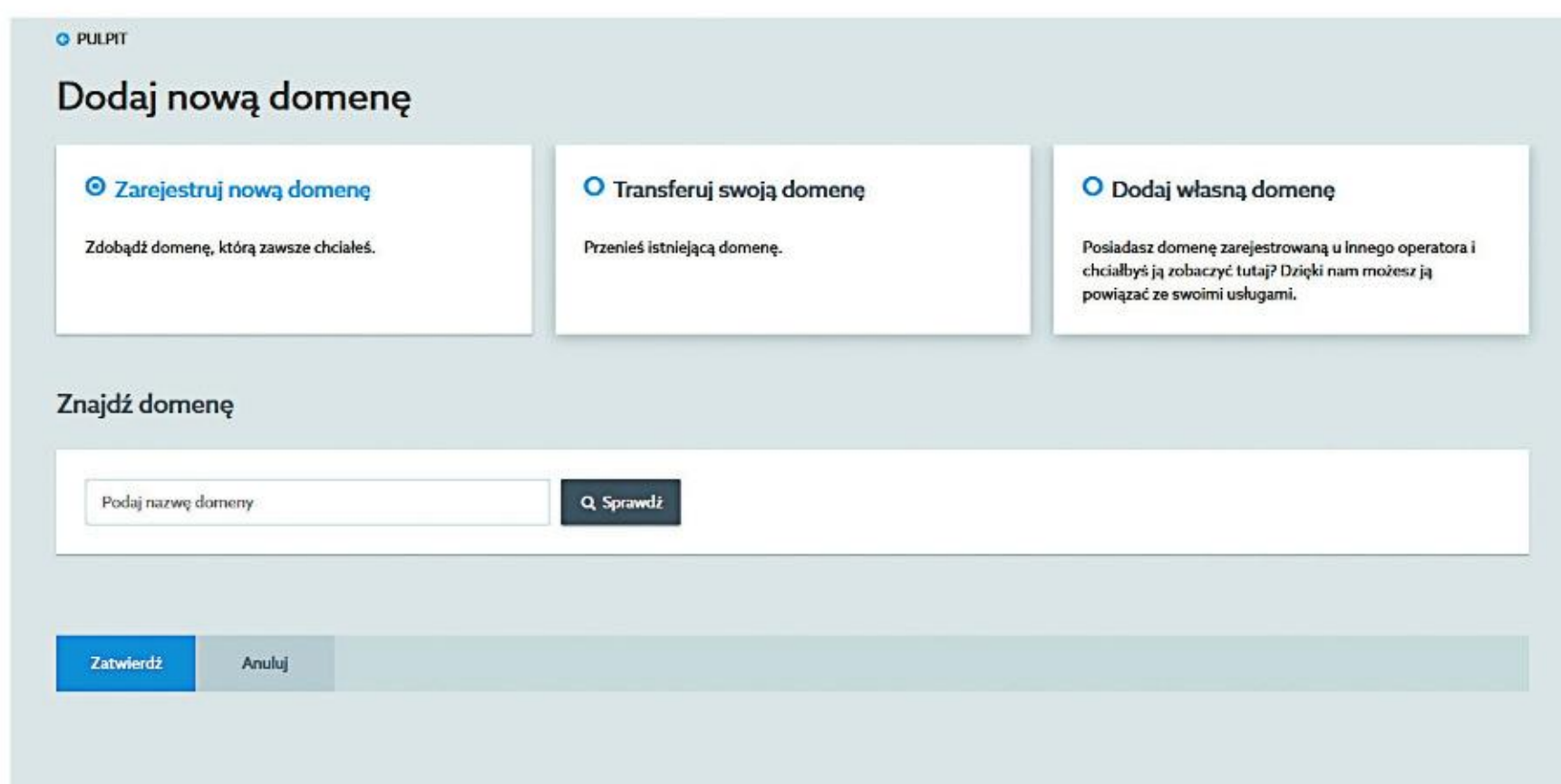
Pierwszym krokiem w trakcie konfiguracji jest zalogowanie się do panelu administracyjnego usługi u naszego dostawcy hostingu. Na skrzynkę mailową otrzymamy login i hasło niezbędne do logowania.



Rys. 16.2. Okno panelu administracyjnego usługi hostingowej

W panelu administracyjnym mamy wszelkie narzędzia dotyczące konfiguracji wszystkich usług, takich jak poczta elektroniczna, domeny, strony, a także możliwość zarządzania opłatami i ewentualnymi użytkownikami współzarządzającymi usługą.

Następnym krokiem jest zarejestrowanie domeny, do której będzie przypięta nasza strona internetowa utworzona w CMS. Najpierw wyszukujemy wolne domeny i wybieramy taką, która będzie najlepiej identyfikować treści strony. Możemy w tym miejscu, co jest widoczne na rys. 16.3, wykonać również transfer domeny z innej usługi hostingowej lub wpisać domenę zarejestrowaną samodzielnie w NASK.



Rys. 16.3. Okno wyszukiwania domen w rejestrze NASK

Dla naszej strony wyszukaliśmy domenę inf03.pl. Była dostępna, dlatego wykonaliśmy opłatę i automatyczną rejestrację w naszej usłudze hostingowej. Po przeprowadzeniu tej operacji otrzymamy potwierdzenie aktywacji domeny (rys. 16.4).

Domeny					
Dodaj nową domenę					
Domena	Hosting DNS	Status rejestracji	Wygasa	Włączone dla usług	Opcje
inf03.pl	Włączone	Aktywna	29 paź 2020		Opcje

Rys. 16.4. Lista aktywnych domen naszej usługi hostingowej

Po rejestracji domeny musimy przypisać ją do naszego serwera w usłudze hostingowej. Możliwość przeprowadzenia tej czynności znajdziemy w panelu administracyjnym w menu **Usługi WWW**. Przy danej nazwie serwera wybieramy opcję **Zarządzaj**.

home.pl

- Pulpit
- Użytkownicy
- Certyfikaty SSL
- Usługi WWW**
- Rejestracja usług
- Poczta
- Wsparcie klienta
- Domeny
- Konto

Usługi WWW

SERWER1952467
Id subskrypcji: 5857810

Hosting Biznes Apache wygasa za 365 dni

Dostępne: 90GB

Konta e-mail

Bazy danych

Domeny brak

FTP brak

OPCJE...

Przypisz domenę

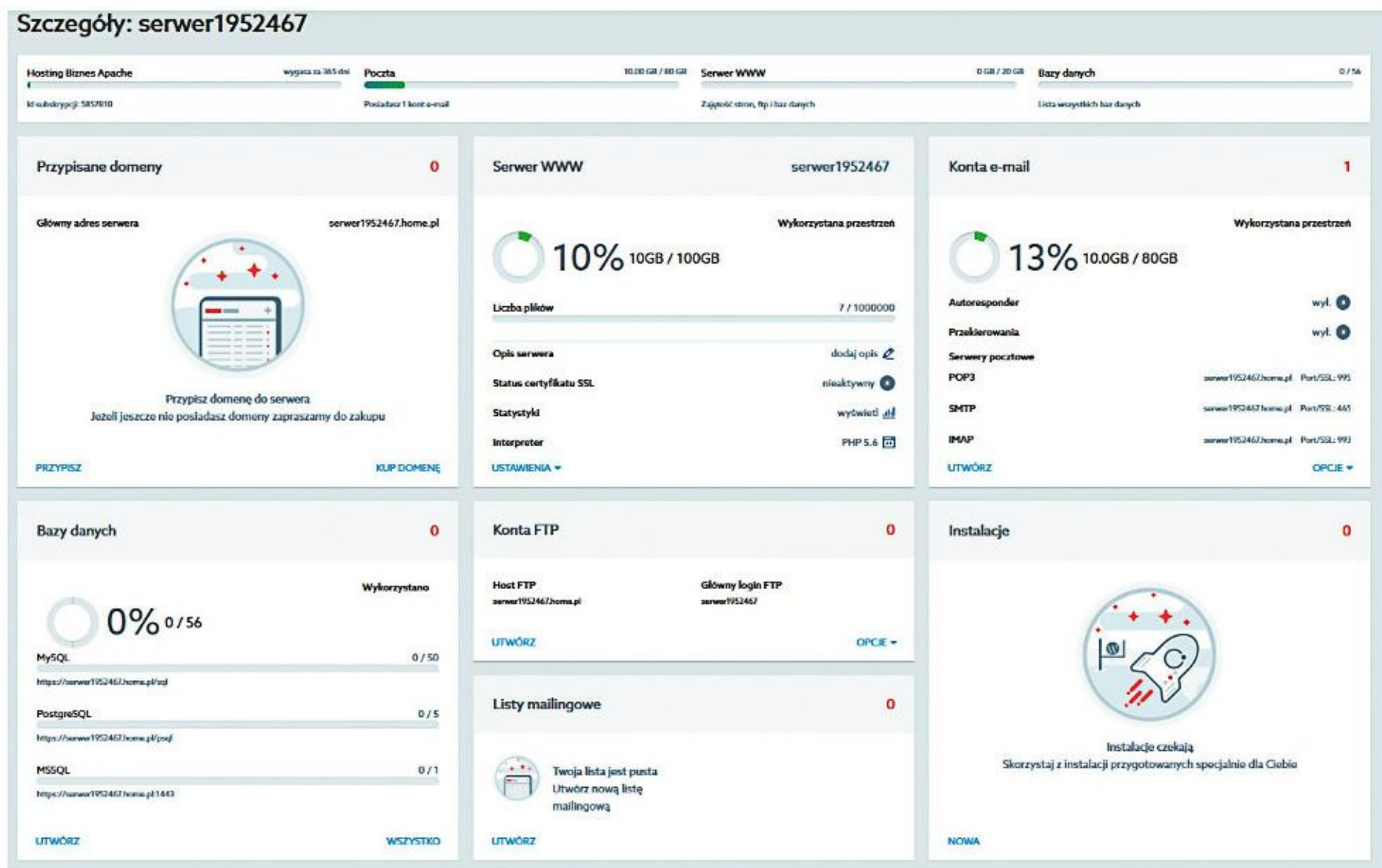
Utwórz skrzynkę e-mail

Nowa baza danych

Zarządzaj

Rys. 16.5. Uruchamianie panelu administracyjnego serwera

W otwartym oknie szczegółów serwera naszej usługi hostingowej mamy kompleksowy przegląd statusu wszystkich usług i parametrów jego pracy. W oknie **Przypisane domeny** dodajemy zarejestrowaną domenę. Dzięki temu po instalacji systemu CMS pod tym adresem domenowym wyświetli się nasza strona internetowa.



Rys. 16.6. Okno panelu administracyjnego serwera

Kolejnym etapem jest utworzenie nowej bazy danych, z której będzie korzystać nasz CMS. W tym celu wybieramy opcję **Bazy danych** w oknie szczegółów serwera i w oknie utwórz wprowadzamy nazwę nowej bazy danych oraz hasło. Nazwa użytkownika będzie w tym wypadku taka sama jak nazwa bazy.

Nowa baza danych

Konfiguracja

Typ bazy danych: MySQL

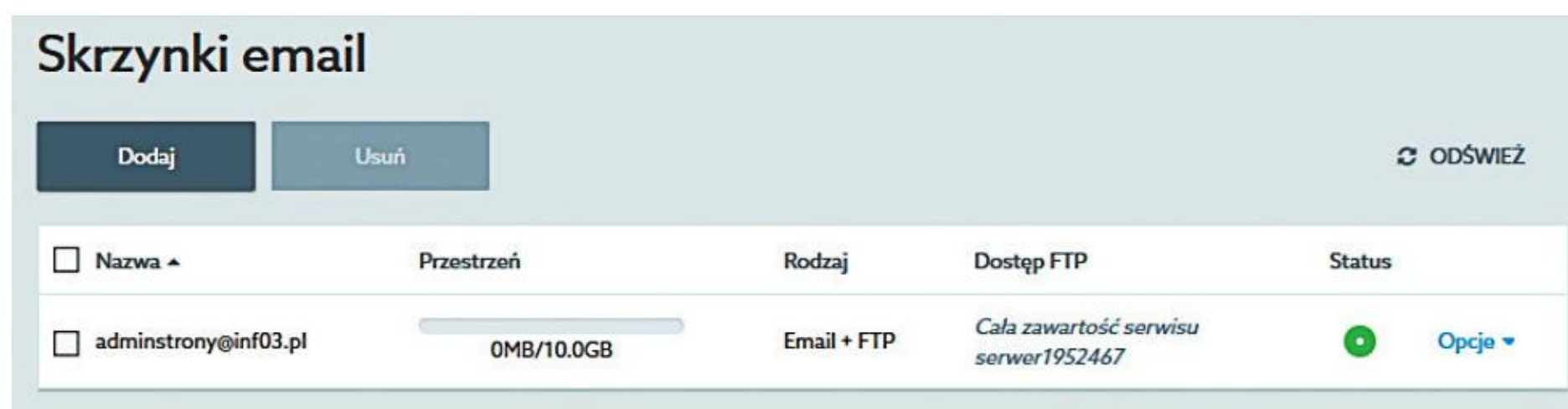
Nazwa bazy: 31693816_

Hasło bazy danych: → 👁

Opcjonalny komentarz:

Rys. 16.7. Tworzenie nowej bazy danych

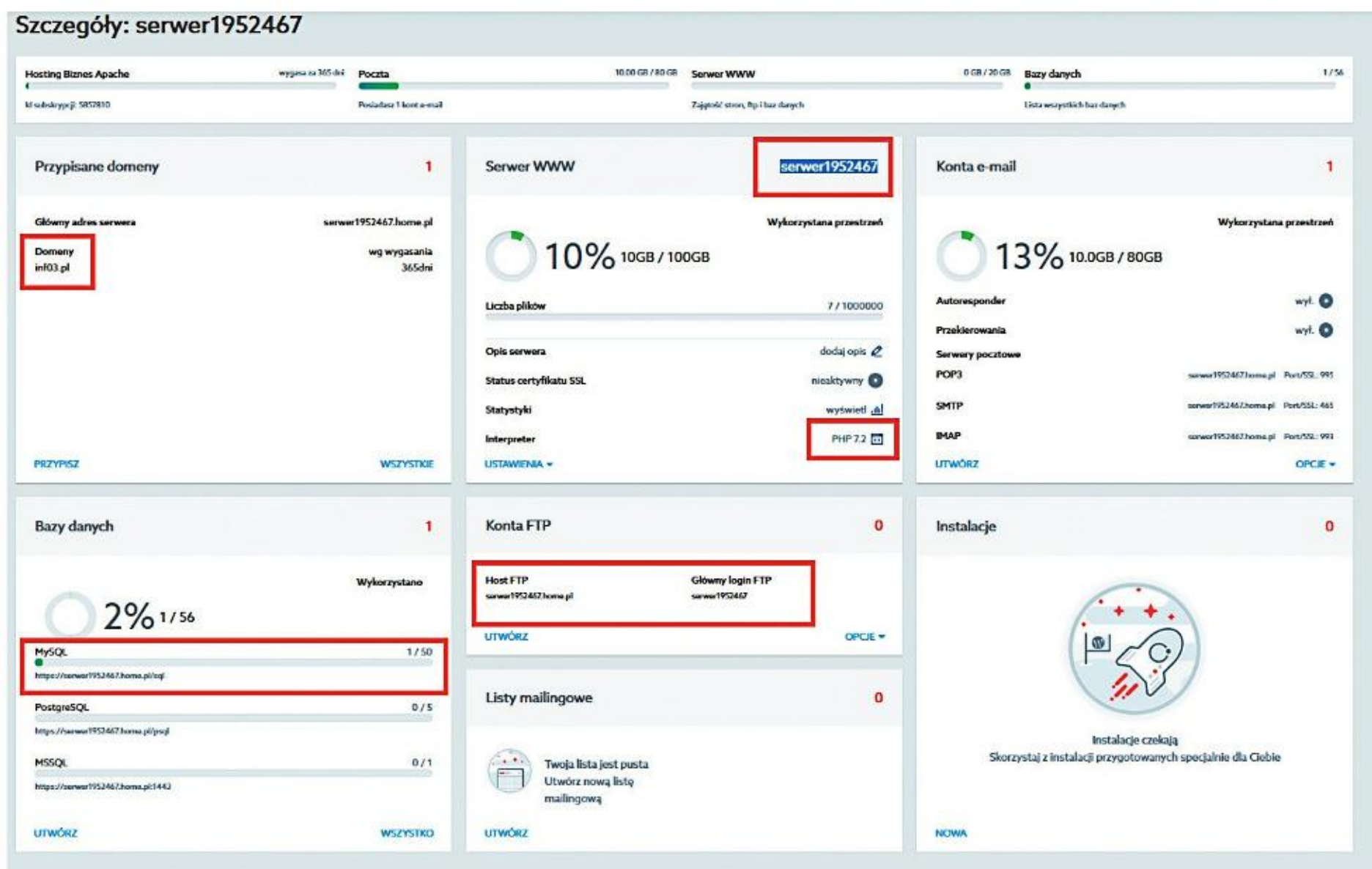
Ostatnią czynnością jest utworzenie konta pocztowego, które będzie nam potrzebne do konfiguracji kontaktu z administratorem w serwisie CMS oraz obsługi klienta FTP.



Rys. 16.8. Okno tworzenia skrzynki pocztowej

Po zakończeniu konfiguracji usługi hostingowej warto spisać wszystkie istotne parametry potrzebne do instalacji pakietu CMS. Na rys. 16.9 zaznaczyliśmy je czerwonymi prostokątami. W naszym przypadku będą to następujące dane (oprócz haseł):

- nazwa domeny: inf03.pl,
- nazwa serwera: serwer1952467,
- host FTP: serwer1952467.home.pl,
- użytkownik FTP: adminstrony@inf03.pl (adres pocztowy dla domeny),
- interpreter PHP: PHP 7.2,
- nazwa bazy danych SQL: 31693816_inf,
- rodzaj Bazy danych: MySQL,
- użytkownik bazy danych: 31693816_inf,
- konto e-mail: adminstrony@inf03.pl,
- wersja MySQL: 5.7 (5.7.18).



Rys. 16.9. Podsumowanie konfiguracji serwera

Instalacja ręczna na serwerze zdalnym CMS Joomla!

Jedną z możliwości instalacji CMS jest instalacja ręczna. Do tej operacji użyjemy **CMS Joomla!**, jednego z najpopularniejszych tego typu pakietów. Zaczynamy od wyboru wersji pakietu oraz jego pobrania. Wchodzimy na stronę <https://www.joomla.pl> i szukamy informacji o wymaganiach dotyczących tej dystrybucji CMS. Na rys. 16.10 przedstawiono zestawienie parametrów minimalnych i zalecanych dla wersji 3.XX.

Oprogramowanie	Zalecane	Minimum
PHP (wylączone Magic Quotes GPC)	5.3.10 +	5.3.10 +
Obsługiwane bazy danych:		
MySQL ^[1] (wymagana obsługa InnoDB)	5.1 +	5.1 +
MSSQL	10.50.1600.1 +	10.50.1600.1 +
PostgreSQL	8.3.18 +	8.3.18 +
Obsługiwane serwery:		
Apache(z mod_mysql, mod_xml, and mod_zlib) ^{[2][3]}	2.x+	2.x+
Nginx	1.1	1.0
Microsoft IIS	7	7

Rys. 16.10. Wymagania dla CMS Joomla!

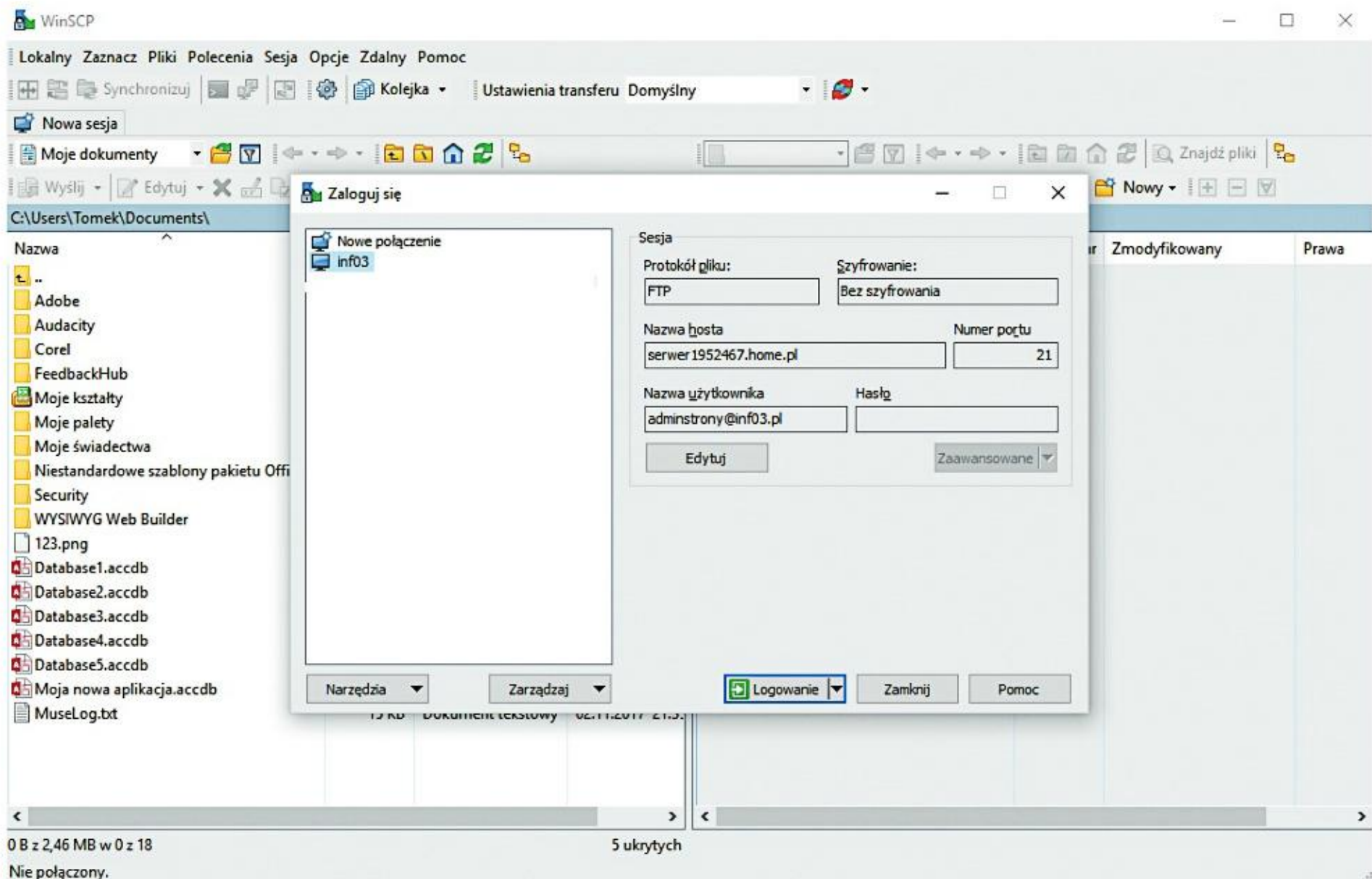
Po wybraniu najlepszej dla naszej usługi hostingowej wersji systemu CMS Joomla! przystępujemy do pobrania pliku skompresowanego ZIP. W naszym przypadku będzie to plik o nazwie **Joomla_3.9.6-Stable-Full_Package**.

UWAGA

Jeżeli system CMS wymaga innego interpretera PHP niż ten, który jest wybrany w naszej usłudze hostingowej, należy sprawdzić w panelu administracyjnym serwera, czy istnieje możliwość wyboru innej wersji tego oprogramowania, oraz przeprowadzić zmianę.

W celu skopiowania plików systemu CMS Joomla! na serwer naszej usługi hostingowej musimy skonfigurować klienta FTP. Na rys. 16.11 przedstawiono przykład konfiguracji, w której wykorzystano parametry ustalone w poprzednich krokach.

Po zatwierdzeniu ustawień klienta FTP wybieramy przycisk **Logowanie** i łączymy się z serwerem (podajemy hasło). Połączony z serwerem klient FTP wyświetli nam w lewym oknie domyślny katalog naszego komputera, natomiast w prawym – zawartość głównego katalogu naszego zdalnego serwera. Teraz musimy skopiować na serwer pliki systemu CMS Joomla! pobrane ze strony projektu. Jeżeli na naszym serwerze będzie istniała jedna strona lub aplikacja internetowa, możemy skopiować pliki do katalogu **/public_html**.



Rys. 16.11. Konfiguracja klienta FTP w celu skopiowania plików instalacyjnych

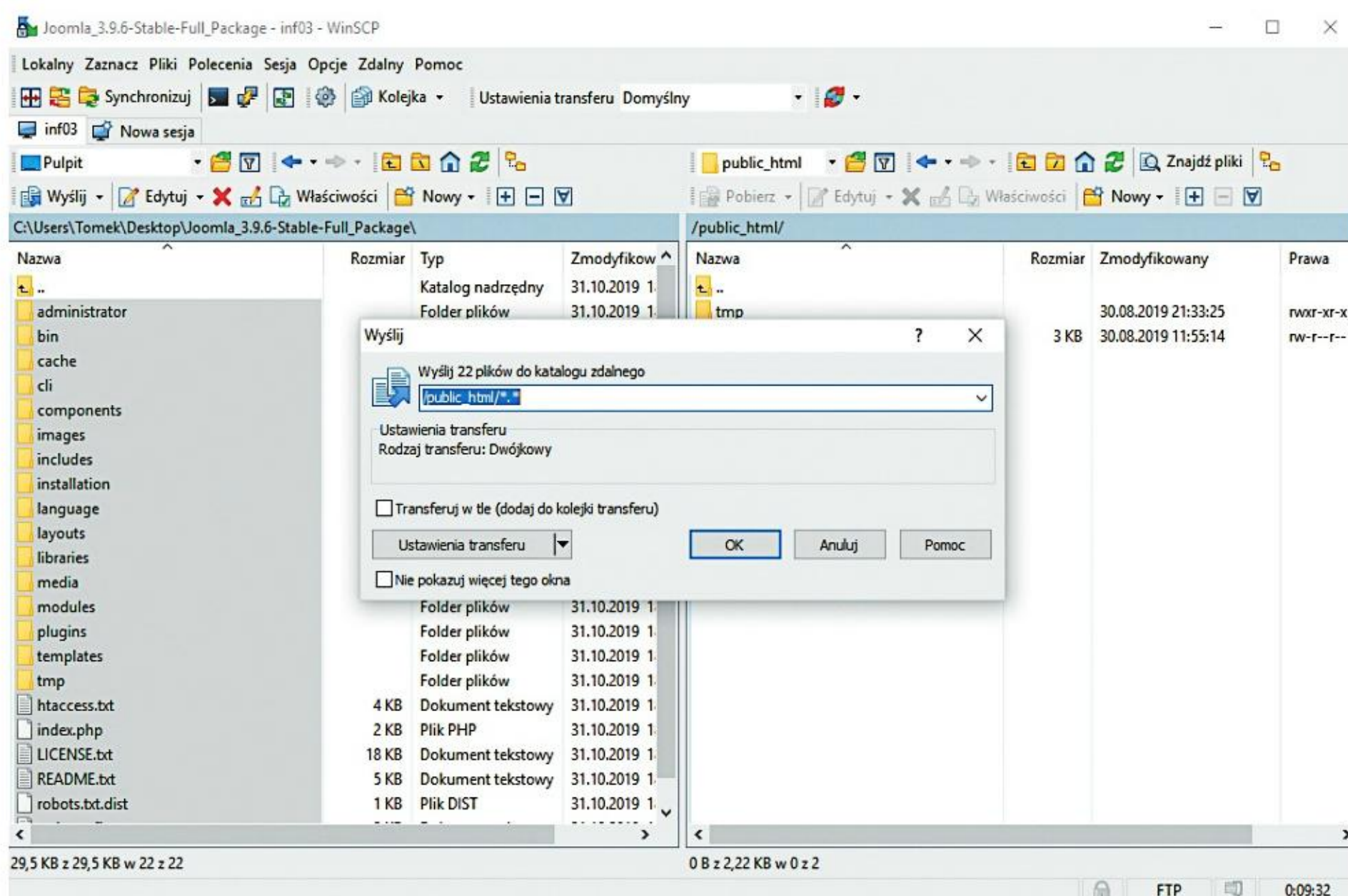
Jeżeli natomiast będziemy na serwerze obsługiwać wiele stron z oddzielnymi bazami danych, to musimy utworzyć dedykowany katalog w **/public_html** i dopiero do niego kopiować pliki. W takiej sytuacji trzeba pamiętać, że do obsługi każdej strony należy utworzyć oddzielną bazę danych oraz subdomenę i skojarzyć ją z utworzonym katalogiem.

Aby przesłać pliki, rozpakowujemy folder ZIP w dowolnym miejscu na naszym komputerze. W kliencie FTP, w lewym oknie wchodzimy do rozpakowanego katalogu i zaznaczamy wszystkie pliki. W prawym oknie wchodzimy do katalogu docelowego.

! UWAGA

Można przesłać na serwer cały spakowany pakiet, a następnie go rozpakować. Można również najpierw rozpakować pakiet na swoim komputerze w pustym katalogu, a następnie przesłać zawartość tego katalogu do katalogu witryny **/public_html** na serwerze.

Po zaznaczeniu plików uruchamiamy polecenie Wyślij z menu uruchomionego prawym klawiszem myszy lub klawiszem funkcyjnym [F5]. Przebieg tego procesu przedstawiono na rys. 16.12.



Rys. 16.12. Kopiowanie plików na serwer

Po zakończeniu kopiowania pakiet CMS Joomla! jest gotowy do instalacji.

Możemy zastosować trzy metody instalacji systemu CMS:

- instalacja z edycją pliku config.php;
- instalacja bez edytowania żadnych plików – wszelkie dane do połączenia z bazą podaje użytkownik podczas instalacji;
- instalacja za pomocą autoinstalatora, bez jakiegokolwiek ingerencji użytkownika.

W razie wykorzystania pierwszej metody wszystkie parametry instalacji (nazwa bazy danych, adres serwera SQL, nazwa użytkownika bazy i hasło dostępu do bazy) muszą zostać wpisane do pliku config.php. Edycje pliku możemy wykonać za pomocą dowolnego edytora plików tekstowych.

Po wybraniu drugiej metody należy uruchomić instalację przez wpisanie adresu strony w przeglądarce. Zależnie od tego, czy pliki CMS Joomla! skopiowano do głównego katalogu domeny, czy też do dedykowanego podkatalogu, adres powinien wyglądać następująco:

`http://www.nasza-domena.pl/installation/install.php`

lub

`http://www.nasza-domena.pl/nazwa katalogu/installation/install.php`

Na rys. 16.13 przedstawiono okno rozpoczynające prawidłowe uruchomienie instalatora.

The screenshot shows the Joomla! installation wizard in a web browser. The browser's address bar displays 'www.inf03.pl/installation/index.php'. The Joomla! logo is at the top, followed by the text 'Joomla! jest wolnym oprogramowaniem wydany na licencji GNU Powszechna Licencja Publiczna.' Below this, there are three steps: '1 Konfiguracja witryny' (selected), '2 Baza danych', and '3 Podsumowanie'. A language selector shows 'Polski (PL)' and a 'Dalej' button is present.

Podstawowa konfiguracja

Nazwa witryny * Wpisz nazwę swojej witryny Joomla!.

Opis witryny Opisz zwięźle treść witryny. Opis jest używany przez wyszukiwarki. Optymalnie: nie więcej niż 20 słów.

Szczegóły konta super użytkownika

E-mail administratora * Wpisz swój adres e-mail. Będzie to adres super użytkownika witryny.

Nazwa administratora * Ustal nazwę konta super użytkownika.

Hasło administratora * Ustal hasło konta super użytkownika i powtórz je w polu poniżej.

Powtórz hasło administratora *

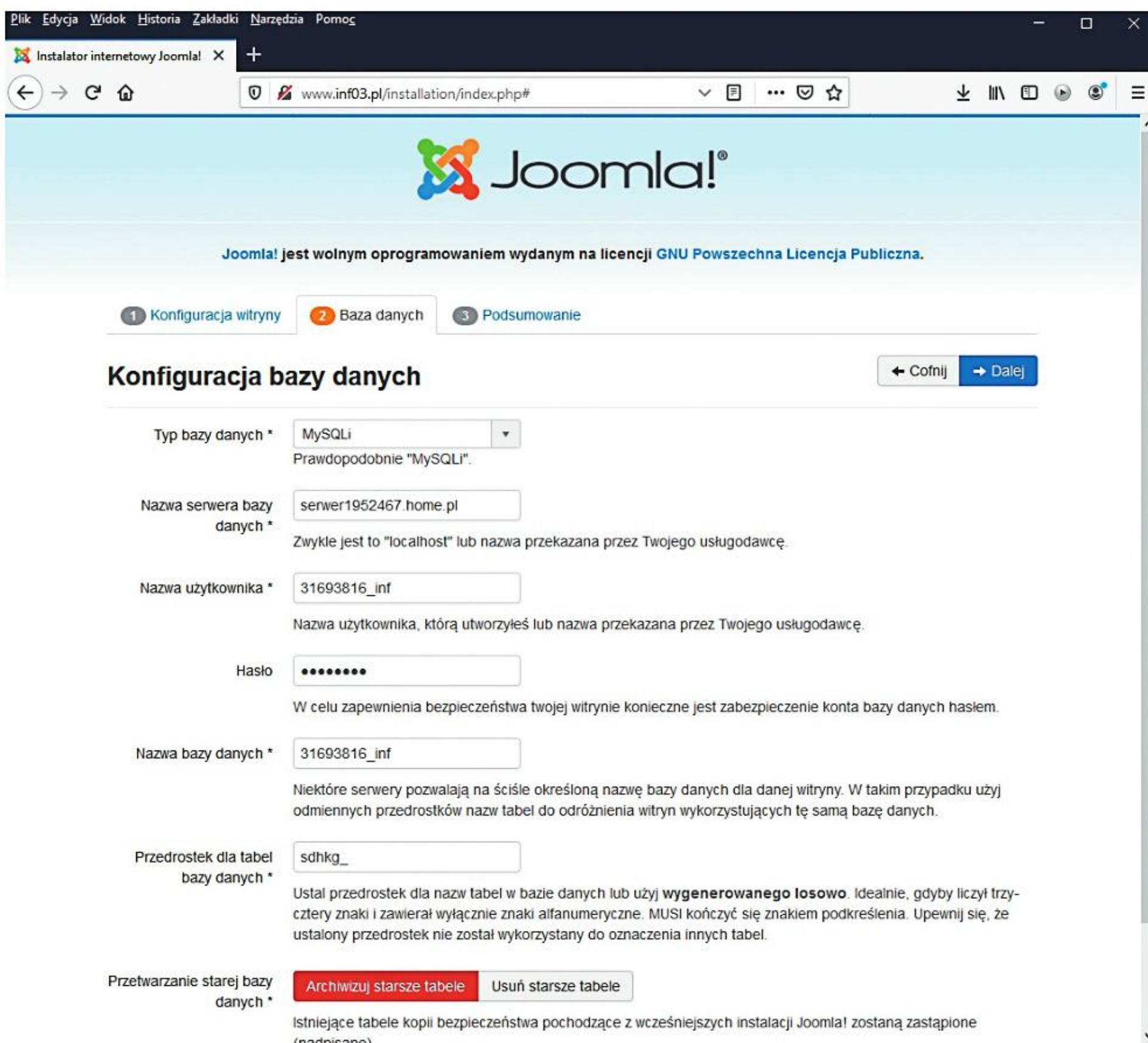
Witryna wyłączona Wyłącz witrynę na czas, dopóki nie zostanie zainstalowana i skonfigurowana. Witrynę można włączyć po zakończeniu instalacji na stronie Konfiguracja globalna.

Dalej

Rys. 16.13. Okno instalatora Joomla!

Proces instalacji składa się z trzech kroków. W pierwszym podajemy podstawowe parametry strony, takie jak: nazwa witryny, adres e-mail administratora oraz jego nazwę i hasło. Na rys. 16.13 pokazano poszczególne okna wypełnione przykładowymi danymi. Po ich uzupełnieniu i naciśnięciu klawisza **Dalej** przechodzimy do następnego kroku.

W drugim etapie instalacji należy podać parametry dotyczące konfiguracji bazy danych. Na rys. 16.14 przedstawiono przykładową konfigurację bazy według parametrów naszej usługi hostingowej.



The screenshot shows the Joomla! installation interface in a web browser. The browser's address bar displays `www.inf03.pl/installation/index.php#`. The Joomla! logo is prominently displayed at the top, followed by the text: "Joomla! jest wolnym oprogramowaniem wydany na licencji GNU Powszechna Licencja Publiczna." Below this, a progress bar indicates three steps: 1. Konfiguracja witryny, 2. Baza danych (active), and 3. Podsumowanie. The main heading is "Konfiguracja bazy danych", with "Cofnij" and "Dalej" buttons on the right. The configuration fields include: "Typ bazy danych" (MySQLi), "Nazwa serwera bazy danych" (serwer1952467.home.pl), "Nazwa użytkownika" (31693816_inf), "Hasło" (masked with dots), "Nazwa bazy danych" (31693816_inf), and "Przedrostek dla tabel bazy danych" (sdhkg_). Each field has a descriptive note. At the bottom, there are buttons for "Archiwizuj starsze tabele" and "Usuń starsze tabele", with a note about replacing existing backup tables.

1 Konfiguracja witryny 2 Baza danych 3 Podsumowanie

Konfiguracja bazy danych

← Cofnij → Dalej

Typ bazy danych * MySQLi
Prawdopodobnie "MySQLi".

Nazwa serwera bazy danych * serwer1952467.home.pl
Zwykle jest to "localhost" lub nazwa przekazana przez Twojego usługodawcę.

Nazwa użytkownika * 31693816_inf
Nazwa użytkownika, którą utworzyłeś lub nazwa przekazana przez Twojego usługodawcę.

Hasło
W celu zapewnienia bezpieczeństwa twojej witrynie konieczne jest zabezpieczenie konta bazy danych hasłem.

Nazwa bazy danych * 31693816_inf
Niektóre serwery pozwalają na ściśle określoną nazwę bazy danych dla danej witryny. W takim przypadku użyj odmiennych przedrostków nazw tabel do odróżnienia witryn wykorzystujących tę samą bazę danych.

Przedrostek dla tabel bazy danych * sdhkg_
Ustal przedrostek dla nazw tabel w bazie danych lub użyj **wygenerowanego losowo**. Idealnie, gdyby liczył trzy-cztery znaki i zawierał wyłącznie znaki alfanumeryczne. MUSI kończyć się znakiem podkreślenia. Upewnij się, że ustalony przedrostek nie został wykorzystany do oznaczenia innych tabel.

Przetwarzanie starej bazy danych * Archiwizuj starsze tabele Usuń starsze tabele
Istniejące tabele kopii bezpieczeństwa pochodzące z wcześniejszych instalacji Joomla! zostaną zastąpione (nadpisane).

Rys. 16.14. Konfiguracja bazy danych w instalatorze Joomla!

Po wyborze opcji **Dalej** przechodzimy do podsumowania instalacji. Tutaj możemy również zdecydować, czy instalację przeprowadzimy na przykładowych danych, czy też pozostawimy ją tzw. pustą. Jeżeli wszystkie parametry podano poprawnie, możemy przejść do instalacji pakietu. Na rys. 16.15 przedstawiono podsumowanie naszej instalacji.

Podsumowanie

Wyślij konfigurację

Po instalacji wyślij ustawienia konfiguracyjne dla `adminstrony@inf03.pl` na e-maila.

Podstawowa konfiguracja

Nazwa witryny	INF03_techinik informatyk
Witryna wyłączona	Nie
E-mail administratora	adminstrony@inf03.pl
Nazwa administratora	admininf03

Konfiguracja bazy danych

Typ bazy danych	mysql
Nazwa serwera bazy danych	serwer1952467.home.pl
Nazwa użytkownika	31693816_inf
Nazwa bazy danych	31693816_inf
Przedrostek dla tabel bazy danych	sdhkg_
Przetwarzanie starej bazy danych	Archiwizuj starsze tabele

Test przed instalacją

Jeżeli któryś z warunków nie został spełniony (został oznaczony następująco **Nie**), podejmij stosowne kroki, by to zmienić. Nie można zainstalować Joomla!, dopóki nie zostaną spełnione poniższe wymagania.

Wersja PHP >= 5.3.10	Tak
Magiczne apostrofy GPC wyłączone	Tak
Rejestracja zmiennych globalnych	Tak
Obsługa kompresji Zlib	Tak
Obsługa XML	Tak
Obsługiwane bazy danych: (pgsql, pdo, postgresql, sqlite, mysql, pdomysql)	Tak
Język wielobajtowy - neutralny	Tak
Nadpisywanie funkcji MB - wyłączone	Tak
Obsługa parsera INI	Tak
Obsługa JSON	Tak
configuration.php zapisywalny?	Nie

Ustawienia zalecane:

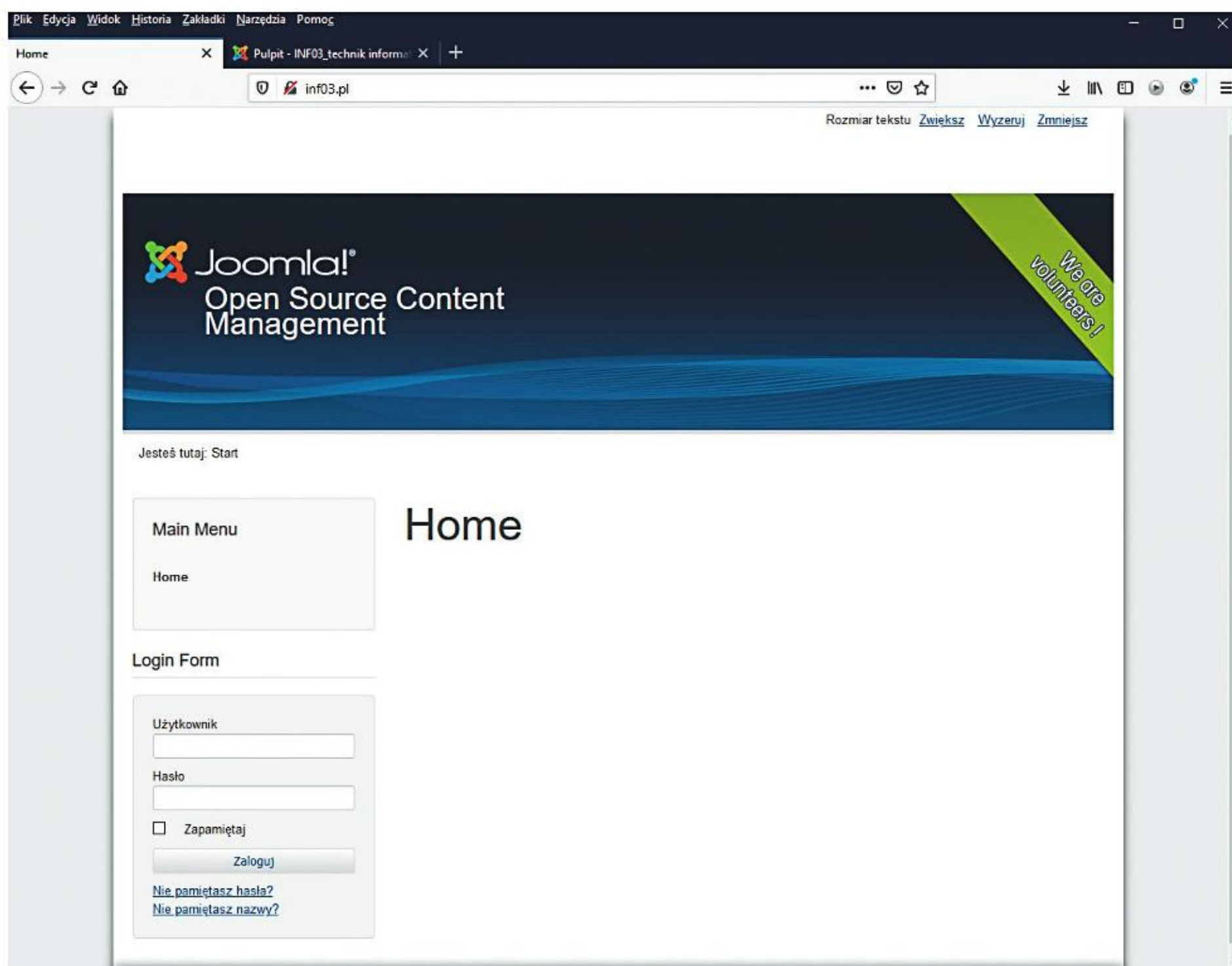
Poniższe zalecane ustawienia PHP zapewnią pełną zgodność ze środowiskiem Joomla!.

Joomla! będzie jednak działać, nawet gdy twoje ustawienia niezupełnie odpowiadają tym zalecanym.

Wytyczna	Zalecane	Bieżące
Tryb bezpieczny	Wyłączone	Wyłączone
Wyświetlanie błędów	Wyłączone	Włączone
Przesyłanie plików	Włączone	Włączone
Magiczne apostrofy Runtime	Wyłączone	Wyłączone
Buforowanie wyjścia	Wyłączone	Wyłączone
Automatyczny start sesji	Wyłączone	Wyłączone
Rdzenna obsługa ZIP	Włączone	Włączone

Rys. 16.15. Podsumowanie konfiguracji instalacji

Po zakończeniu konfiguracji procesu instalacji naciskamy klawisz **Instaluj**. Na serwerze zostaną zainstalowane poszczególne komponenty i utworzone pliki strony. Po zakończeniu tego procesu CMS będzie gotowy do pracy. Na rys. 16.16 przedstawiono wygląd domyślnej strony pakietu Joomla!



Rys. 16.16. Pierwsze uruchomienie strony pakietu Joomla!

Po zakończeniu instalacji pakietu Joomla! należy pamiętać o usunięciu katalogu instalacyjnego pakietu po stronie serwera. Teraz możemy przystąpić do konfiguracji witryny oraz zarządzania jej treścią.

Instalacja systemu CMS WordPress za pomocą autoinstalatora

Instalację za pomocą autoinstalatora przeprowadzimy za pomocą usługi hostingu Strefa.pl. Zainstalujemy tym razem CMS WordPress. Wszystkie poprzednie kroki są bardzo podobne. Wykupujemy hosting, logujemy się do panelu administratora i dodajemy domenę. Autoinstalator znajduje się najczęściej w karcie narzędzi witryn WWW. Na rys. 16.17 pokazano przykład okna narzędzi witryn WWW w usłudze hostingu Strefa.pl.

Narzędzia witryn WWW


Menedżer plików
 Zarządzaj plikami i katalogami.


Dostęp FTP
 W tej sekcji możesz zarządzać dostępem FTP przez zmianę hasła na nowe.


Konfiguracja strony WWW
 Sekcja konfiguracyjna Serwisu WWW zawiera ustawienia konfiguracyjne twojej strony WWW.


Autoinstalator aplikacji
 Instaluj nowe aplikacje i zarządzaj już zainstalowanymi.


Katalogi chronione
 Możesz dodać nowe katalogi chronione oraz zarządzać i usuwać istniejące. Zarządzanie katalogiem chronionym pociąga za sobą zmianę jego nazwy i nagłówka, włączenie/wyłączenie wspomaganie SSL oraz zarządzanie użytkownikami chronionego katalogu.


Statystyki WWW Webalizera
 Sekcja ta zawiera narzędzia przeglądania twoich statystyk WWW w innej przeglądarce i odblokowywanie/blokowanie obsługi statystyk WWW.


Menedżer logu
 Uzyskaj dostęp do nieprzetworzonych logów swojej strony WWW. Pobierz je na swój komputer lokalny.


Zarządzanie cronem
 Zaplanuj wykonanie komend hosta na twoim serwerze WWW.

Ogólny

ID

Nazwa <http://www.info.tekom.strefa.pl>

W synchronizacji

USUŃ

Usługi hostingu

1-3 z 3

USŁUGA HOSTINGU	STAN	SUBSKRYPCJA	POŁOŻENIE	ŁĄCZE
Hosting DNS	❗ Nie w synchronizacji	Strefa.pl Hosting Srebrny		Przełączaj szczegóły
Hosting WWW	✅ W synchronizacji	Strefa.pl Hosting Srebrny	/info.tekom.strefa.pl	Przełączaj szczegóły
Hosting poczty	✅ W synchronizacji	Strefa.pl Hosting Srebrny		Przełączaj szczegóły

Rys. 16.17. Wybór autoinstalatora systemów CMS

Po wyborze autoinstalatora aplikacji określamy, jaką aplikację chcemy zainstalować.

Zainstaluj aplikację

Select application to install

1-5 z 5

APLIKACJA	CATEGORY	SUMMARY
 Drupal	Content management	Open Source Content System zarządzania i silnik blogów
 gallery	E-commerce	Gallery is a powerful photo gallery.
 Joomla	Content management	Joomla! to zaawansowany system open source do zarządzania treścią...
 Piwik	Analytics	Open source web analytics
 WordPress	Blog	WordPress jest nowoczesną, semantyczną, osobistą platformą publikacyjną, koncentrującą się na estetyce, zgodności z internetowymi standardami i użyteczności.

1-5 z 5

ANULUJ

Rys. 16.18. Okno wyboru aplikacji do instalacji

Autoinstalator sam sprawdzi, czy parametry techniczne są spełnione, a w dalszej części procedury instalacyjnej poprosi o dane dotyczące nazwy katalogu do instalacji, domeny, z którą serwis ma być powiązany, oraz nazwy i hasła superadministradora pakietu CMS.

Installing WordPress

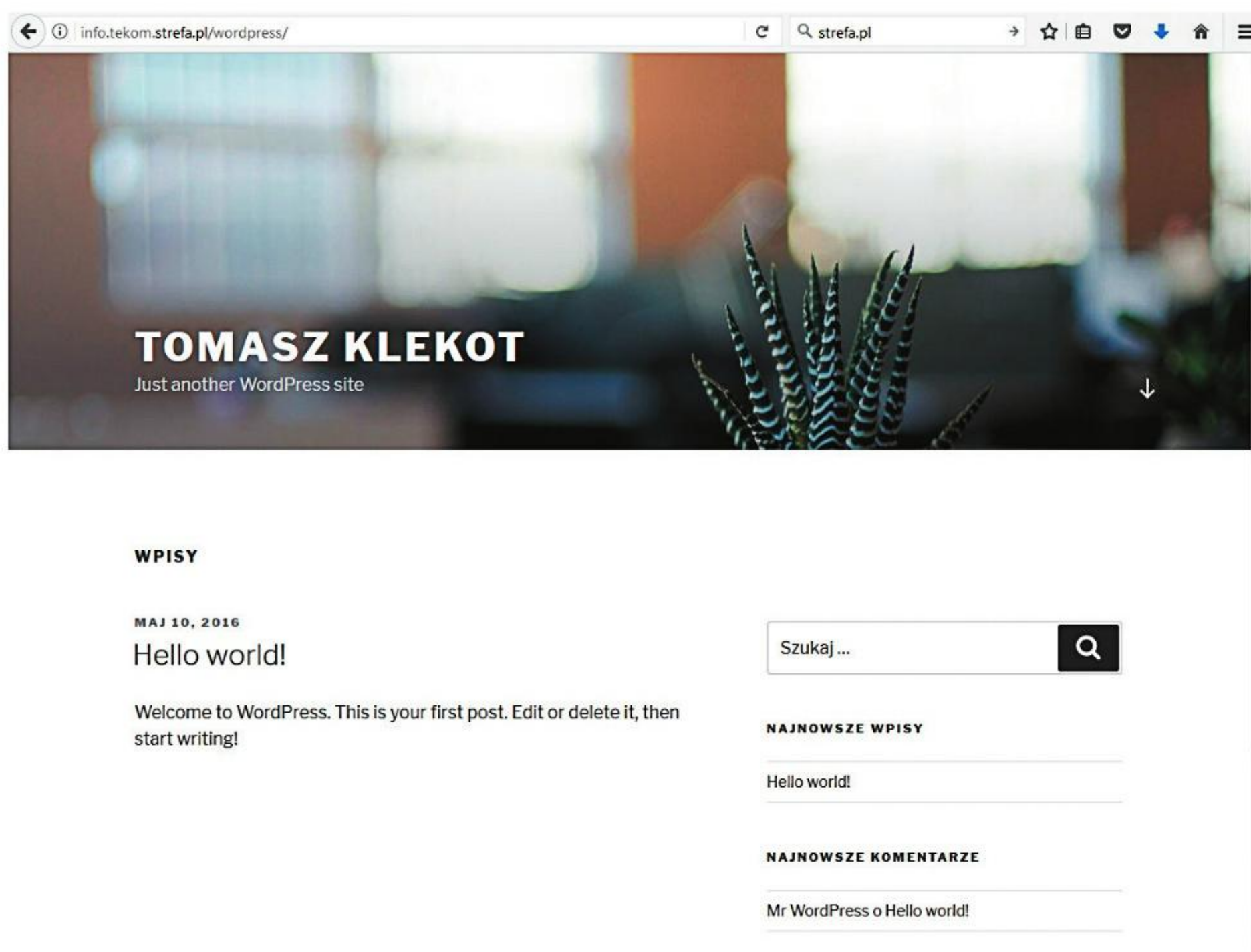
Wymaganie 1

WYMAGANIA		WYMAGANA WARTOŚĆ
db		Ok
php		Ok
extension	mysqli	Ok
safe-mode	false	Ok
version min	5.2.4	Ok

[<< POWRÓT](#)
[NASTĘPNY >>](#)
[ANULUJ](#)

Rys. 16.19. Weryfikacja wymagań instalacji systemu CMS

Po zatwierdzeniu wszystkich danych rozpocznie się proces instalacji. Instalacja nie wymaga kopiowania plików i jest przeprowadzana na serwerze WWW – dlatego trwa kilkanaście sekund. Po jej zakończeniu wpisujemy adres naszej domeny, z którą powiązaliśmy serwis. W oknie przeglądarki pojawi się strona główna naszego serwisu internetowego w domyślnej konfiguracji.



Rys. 16.20. Domyślne okno systemu CMS po instalacji



SPRAWDŹ SVOJE UMIEJĘTNOŚCI

1. Wyszukaj oferty usług hostingowych. Porównaj ich parametry oraz warunki korzystania. Utwórz plik w edytorze tekstu i przedstaw w nim wyniki porównania. Wyniki umieść w tabeli. Gotowy plik przedstaw do oceny nauczycielowi.
2. Skorzystaj z autoinstalatora i zainstaluj dowolny CMS na serwerze usługi hostingowej. Proces instalacji przeprowadź pod nadzorem nauczyciela. Zapisz parametry instalacji (login i hasło administratora). Dane logowania do usługi hostingowej otrzymasz od nauczyciela lub skorzystaj z darmowych ofert hostingowych.
3. Zainstaluj CMS Joomla! na zdalnym serwerze (skorzystaj z klienta FTP). Proces instalacji przeprowadź pod nadzorem nauczyciela. Zapisz parametry instalacji (login i hasło administratora). Dane do wykonania instalacji (nazwa i hasło bazy danych oraz inne) otrzymasz od nauczyciela.



SPRAWDŹ SWOJĄ WIEDZĘ

1. Opisz trzy metody instalacji systemu CMS.
2. Jakie usługi i technologie webowe musi mieć serwer, aby zainstalować dowolny CMS?
3. Jakie parametry są istotne podczas wyboru usługi hostingowej?
4. Jaką rolę w procesie instalacji systemu CMS na zdalnym serwerze odgrywa klient FTP?
5. Dlaczego po instalacji CMS Joomla! należy usunąć katalog instalacyjny?

17

Instalacja CMS na lokalnym serwerze

ZAGADNIENIA

- Rodzaje lokalnych pakietów serwerowych
- Opis programów XAMPP i JAMP
- Elementy składowe pakietu XAMPP

Dowolny plik aplikacji internetowej zawierający skrypt stworzony w PHP oraz elementy z zapytaniami SQL znajdują się na serwerze WWW. Lokalna przeglądarka internetowa zwraca się w odpowiednim momencie do serwera o wykonanie danego skryptu. Interpreter PHP wykonuje skrypt i wysyła efekt działania skryptu, czasami wraz z danymi pobranymi z bazy danych MySQL, na lokalny komputer użytkownika (wyświetla go w oknie przeglądarki). Sposób takiej współpracy nazywa się strukturą klient-serwer i wymaga odpowiedniej konfiguracji usług serwerowych.

Aby sprawnie projektować i tworzyć aplikacje internetowe opierające swoją funkcjonalność na skryptach PHP i bazie danych SQL, należy najpierw przygotować środowisko serwerowe do uruchomienia, testowania i zarządzania tymi aplikacjami.

Dobrym rozwiązaniem jest dysponowanie stałym pakietem hostingowym, który opisano w poprzednich rozdziałach. Jest to jednak drogie rozwiązanie i stwarza problem podczas każdorazowego przesyłania do testowania plików przez FTP. Poza tym nie sprawdza się w sytuacji, kiedy chcemy przetestować przykładowe projekty lub własne aplikacje, zanim je opublikujemy. Wtedy najlepszym rozwiązaniem jest instalacja serwerowych pakietów, które pozwalają na instalację i konfigurację lokalnego serwera WWW oraz dodatkowych usług. Jest to możliwe dzięki wykorzystaniu wewnętrznego adresu IP każdego komputera – 127.0.0.1. Odpowiadająca mu nazwa domenowa to **localhost**.

W sieci istnieje kilka sprawdzonych rozwiązań, do których zalicza się:

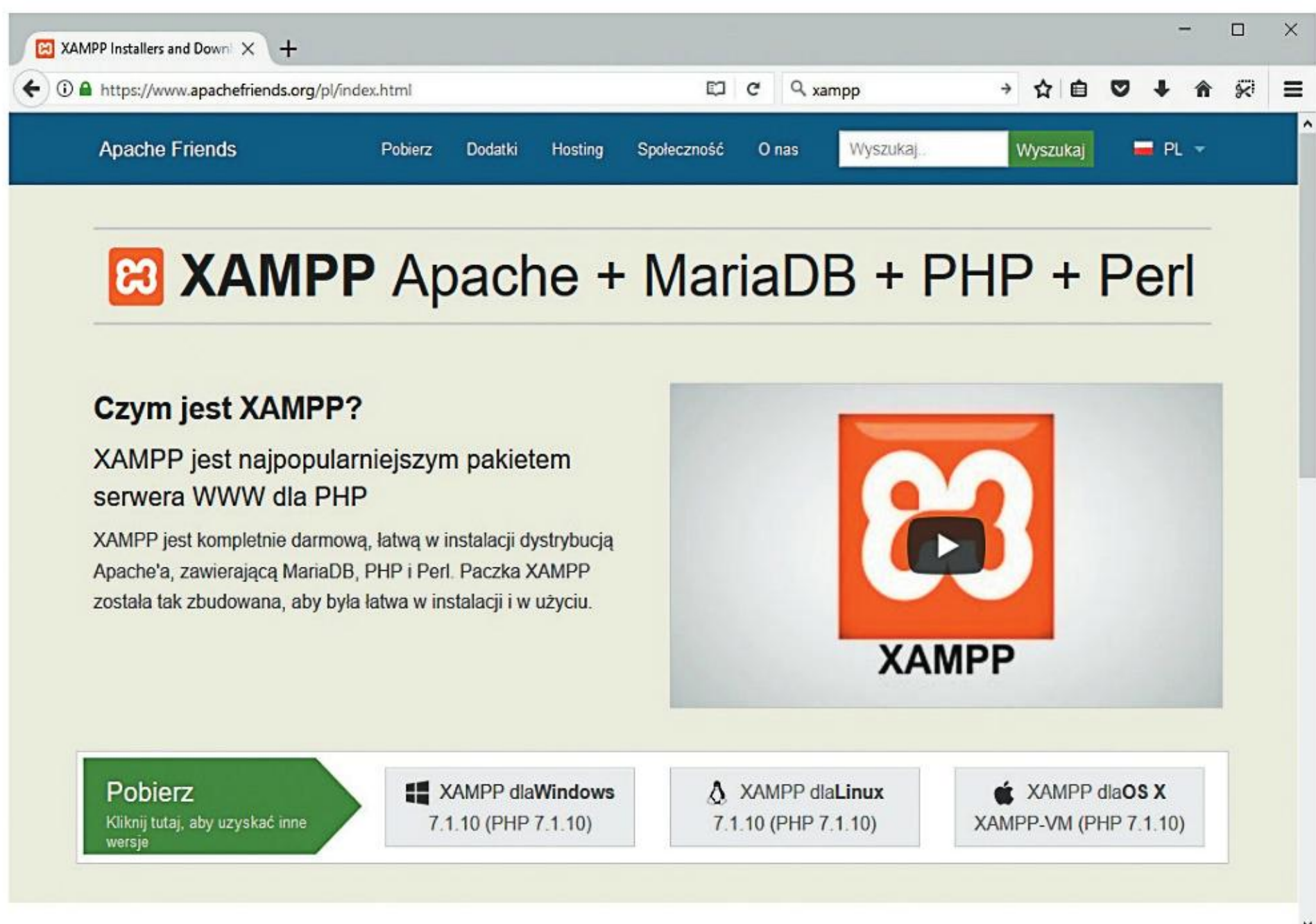
- pakiet XAMPP,
- testową platformę JAMP,
- pakiet WebServ,
- Krasnal.

Można również samodzielnie zainstalować na sieciowym systemie operacyjnym określone usługi i funkcje oraz odpowiednio je skonfigurować. W naszym przykładzie opiszemy jednak zastosowanie gotowych rozwiązań, które można zainstalować na każdym lokalnym systemie operacyjnym.

Jednym z gotowych rozwiązań dostępnych nieodpłatnie i powszechnie wykorzystywanych jest pakiet serwerowy XAMPP. XAMPP jest kompilacją wolnego oprogramowania, które można za darmo kopiować zgodnie z warunkami GNU General Public License. W skład pakietu wchodzi:

- serwer WWW Apache,
- relacyjna baza danych MySQL (lub Maria DB),
- interpreter PHP,
- interpreter Perl,
- FileZilla FTP Server,
- phpMyAdmin,
- OpenSSL.

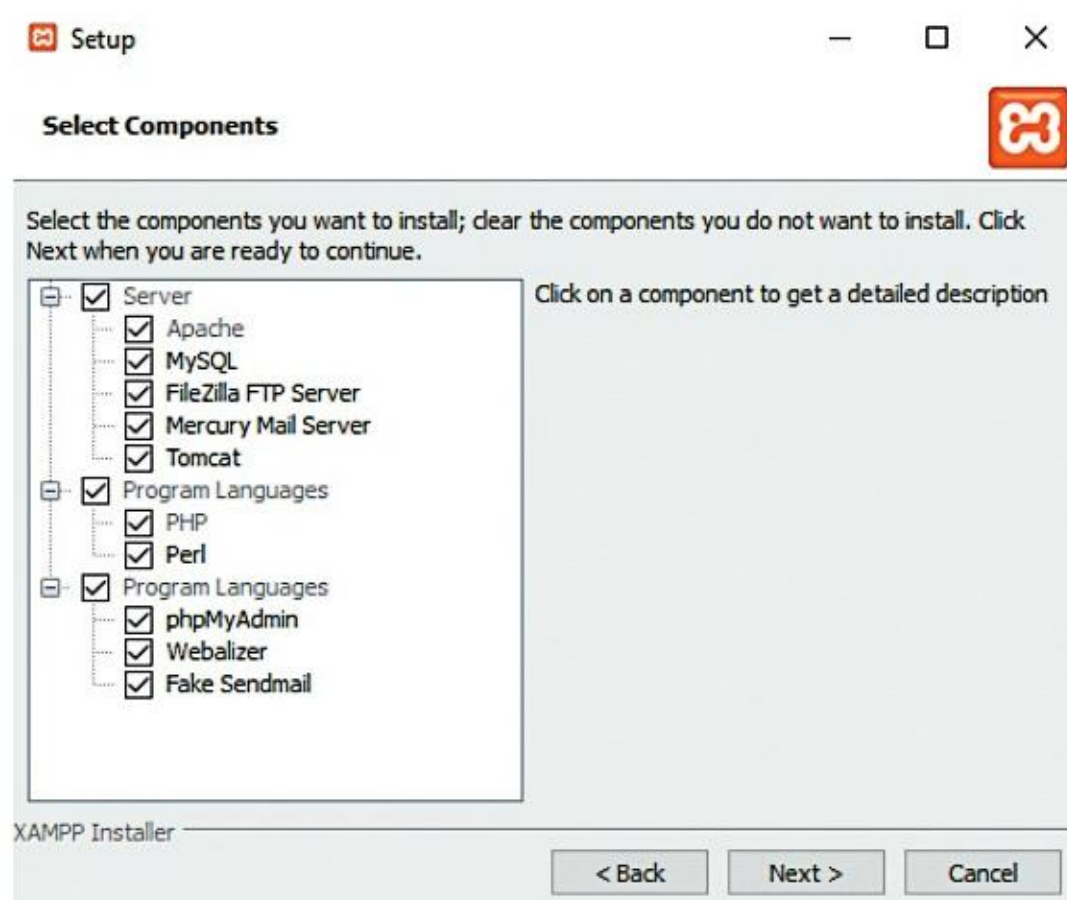
Ten zestaw pozwala na postawienie zarówno lokalnego, jak i publicznego serwera stron internetowych z obsługą skryptów i współpracującą z bazą danych. Bez tych narzędzi nie byłoby możliwe prawidłowe działanie naszych aplikacji internetowych. XAMPP jest pakietem wieloplatformowym, z którym możemy pracować w systemach operacyjnych MS Windows, Linux oraz Mac OS. Pliki instalacyjne są dostępne na stronie internetowej pod adresem <https://www.apachefriends.org/pl/index.html>.



Rys. 17.1. Strona internetowa społeczności XAMPP

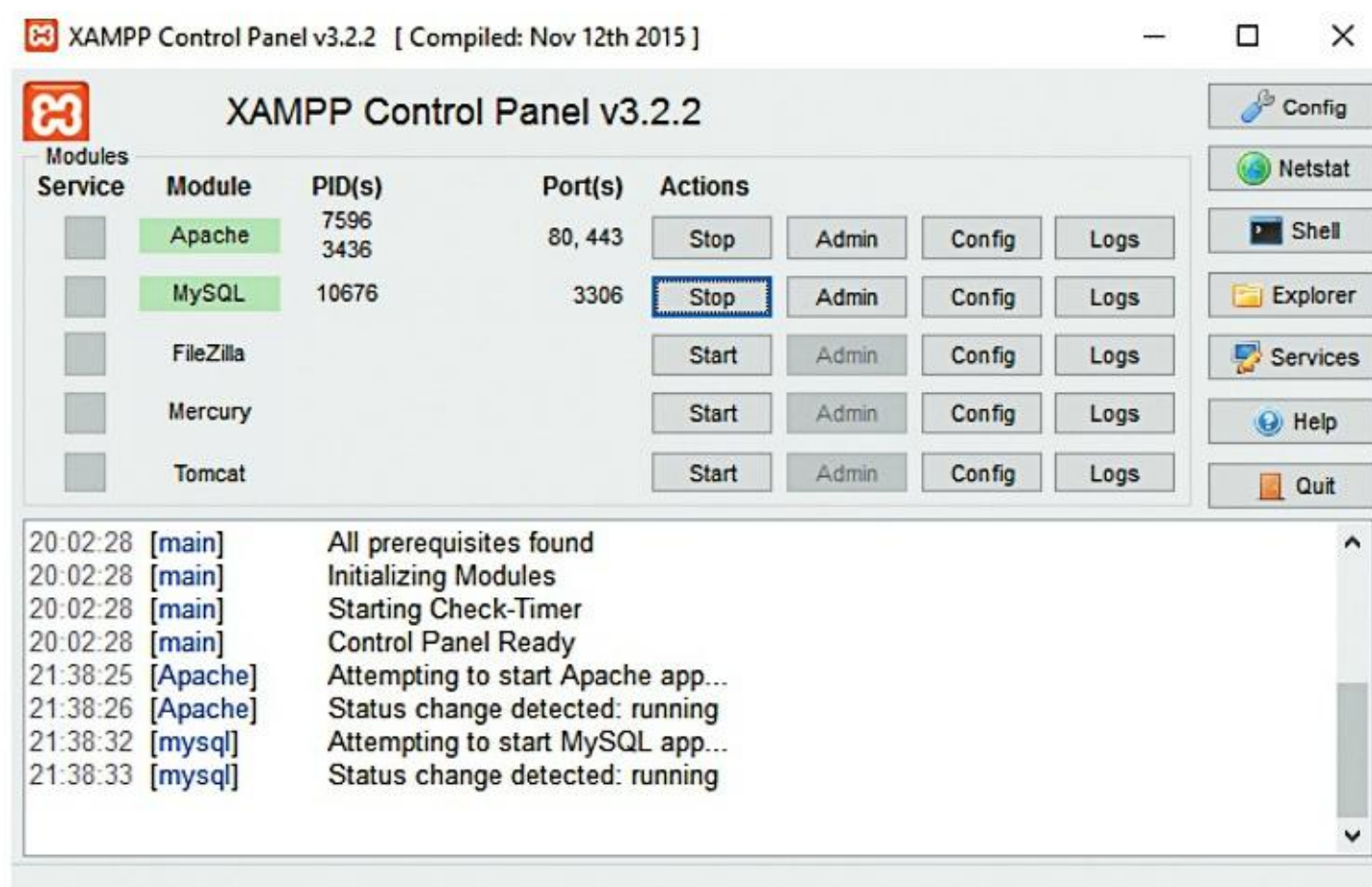
Instalacja pakietu XAMPP

Po pobraniu pliku instalatora można przystąpić do instalacji pakietu. Należy pamiętać, że instalacja powinna być prowadzona z konta administratora oraz że czasowo należy wyłączyć program antywirusowy. Podczas instalacji pakietu trzeba podać folder docelowy instalacji, a także wybrać komponenty programu. Podczas instalacji w systemie MS Windows dodatkowo należy zaakceptować zmiany w regułach zapory firewall dotyczące serwera Apache i bazy danych MySQL.



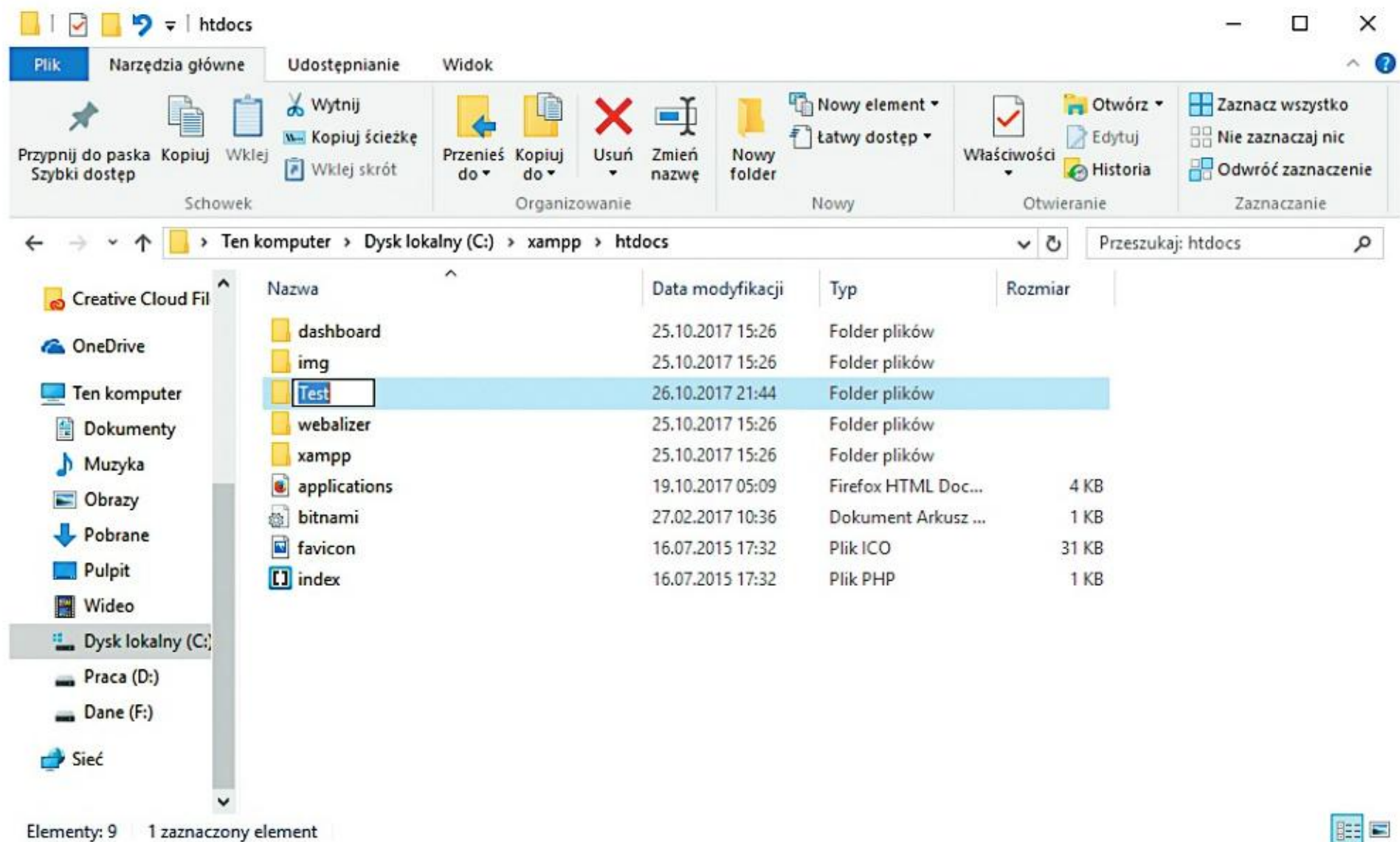
Rys. 17.2. Okno instalacji pakietu XAMPP

Po zakończeniu instalacji pakietu uruchomi się panel kontrolny. W nim można kontrolować pracę poszczególnych komponentów bądź przeprowadzić ich konfigurację. Do poprawnego działania aplikacji internetowej należy uruchomić moduły Apache i MySQL.



Rys. 17.3. Panel kontrolny XAMPP

Aby sprawdzić poprawność działania serwera i interpretera PHP, należy przeprowadzić test skryptu PHP. W tym celu w folderze `\xampp\htdocs` tworzymy podfolder **Test**.

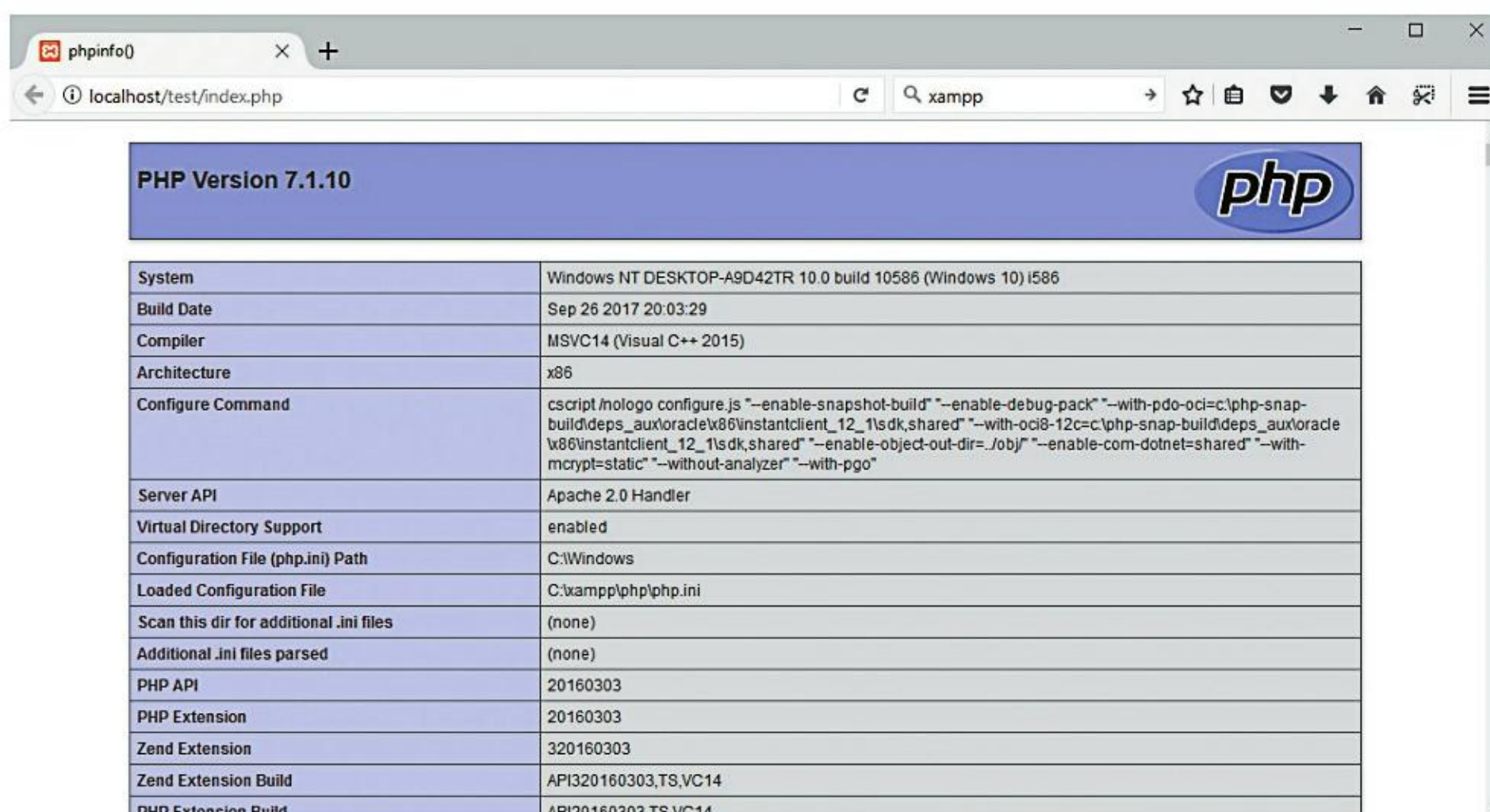


Rys. 17.4. Tworzenie folderu **Test**

Następnie w edytorze tekstowym tworzymy plik o nazwie `index.php` zawierający prosty skrypt testujący dostępność interpretera PHP:

```
<?php
    phpinfo ();
?>
```

Utworzony plik kopiujemy do założonego wcześniej folderu. Następnie uruchamiamy przeglądarkę internetową i wpisujemy w pasek adresu: **`http://localhost/test/index.php`**. Po zatwierdzeniu operacji w oknie przeglądarki powinna pojawić się strona z wygenerowaną informacją przez serwer o wersji i konfiguracji interpretera PHP.



Rys. 17.5. Okno testu PHP

Teraz jesteśmy pewni, że serwer jest prawidłowo zainstalowany i gotowy do pracy.

Funkcje pakietu XAMPP

Oprócz głównej roli pakietu XAMPP, jaką jest realizacja usług serwera WWW, oferuje on również wiele funkcji pozwalających na zarządzanie i konfigurację pakietu, administrowanie bazą danych czy rozbudowywanie naszego serwera o dodatkowe funkcje.

Terminal

Z poziomu panelu kontrolnego pakietu XAMPP dla Windows uruchamiamy okno konsoli znakowej za pomocą opcji Shell. Możemy wykorzystać polecenia systemowe Windows.

W systemie Linux można również zarządzać pakietem XAMPP z poziomu konsoli.

PRZYKŁAD 17.1

Do podstawowych poleceń, które możemy wykorzystać w konsoli, należą komendy uruchamiające usługi serwerowe Apache i bazy danych MySQL:

```
sudo/opt/lampp/lampp start - uruchomienie pakietu XAMPP;
sudo/opt/lampp/lampp startapache - uruchomienie samego serwera Apache;
sudo/opt/lampp/lampp startapache - uruchomienie samej bazy MySQL;
sudo/opt/lampp/lampp stop - wyłączenie pakietu XAMPP.
```

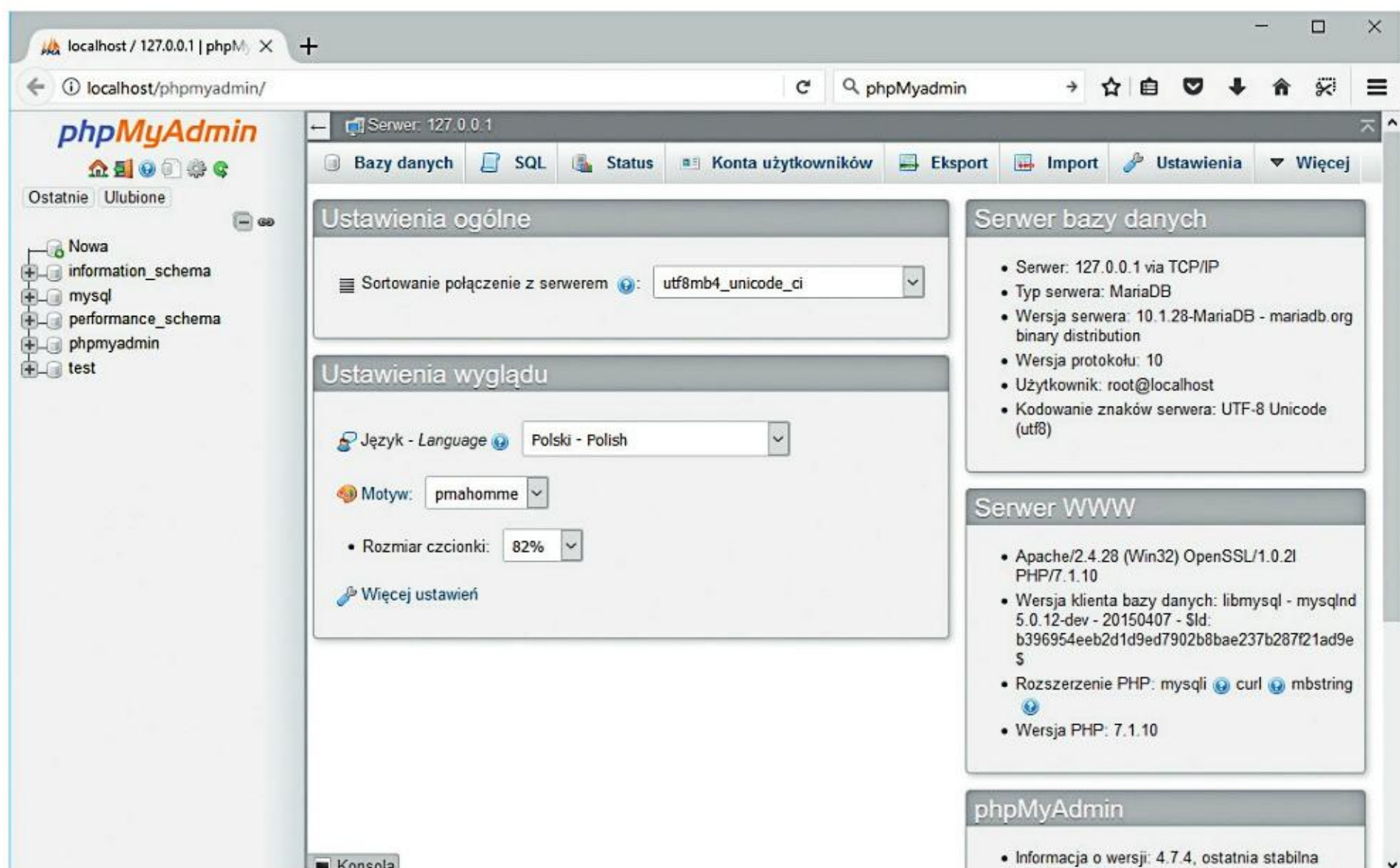

phpMyAdmin

Narzędzie phpMyAdmin pozwala zarządzać bazą danych MySQL, która funkcjonuje w ramach pakietu XAMPP. Ten program zapewnia realizację wszystkich podstawowych funkcji: tworzenie i usuwanie baz danych, tabel, umieszczanie rekordów oraz administrowanie bazami.

Często używaną opcją jest importowanie struktury bazy danych z zewnętrznego pliku sql. Aby wykonać tę czynność, trzeba skorzystać z zakładki **Import**.

Z poziomu phpMyAdmin można również realizować zapytania SQL do bazy danych. Otrzymujemy wówczas rezultat w postaci wyświetlonych rekordów spełniających warunki zapytania (zakładka **SQL**).

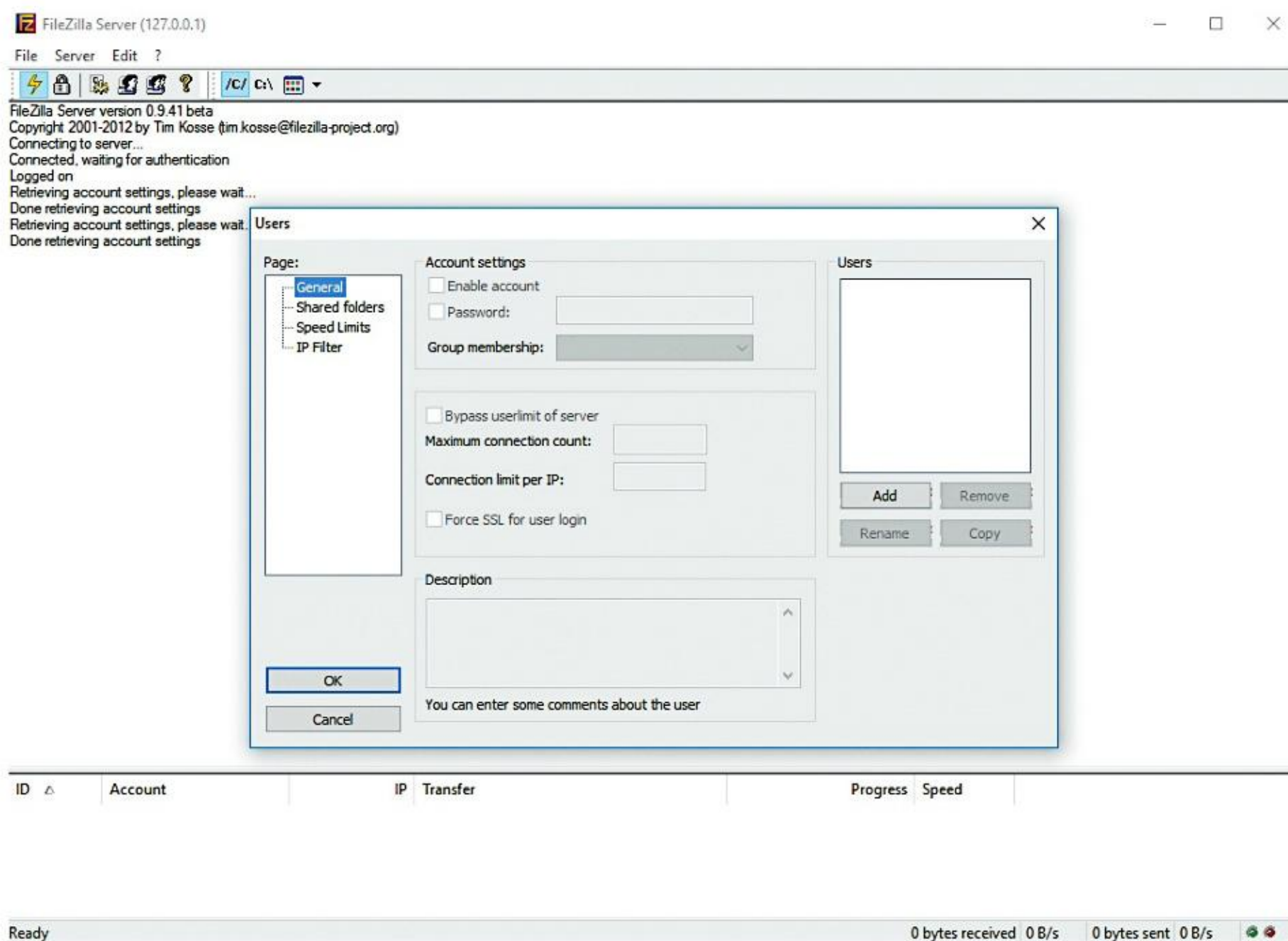
Dodatkowo program wyposażono w narzędzia do wykonywania kopii zapasowej baz danych MySQL (zakładka **Eksport**) oraz obsługi kont użytkowników (zakładka **Konta użytkowników**).



Rys. 17.6. Okno phpMyAdmin

Narzędzia administracyjne

Dzięki wybraniu przycisku Admin w każdej uruchomionej usłudze możemy konfigurować ją z poziomu administratora. Do tego celu służą również przyciski Config, pozwalające uruchomić do edycji podstawowe pliki konfiguracyjne. Przyciski Log umożliwiają kontrolę ruchu w danej usłudze.



Rys. 17.7. Konfiguracja serwera FTP

Instalacja CMS na lokalnym pakiecie XAMPP

Po zainstalowaniu i uruchomieniu pakietu serwerowego XAMPP należy utworzyć nowy folder o nazwie Joomla w katalogu **htdocs** serwera XAMPP. Domyślnie jest on zlokalizowany w ścieżce **C:\xampp\htdocs**. Następnie do utworzonego folderu trzeba wypakować skompresowany plik instalacyjny wybranej wersji Joomla!.

Przed rozpoczęciem instalacji musimy utworzyć pustą bazę danych (np. o nazwie Joomla) za pomocą opisanego wcześniej phpMyAdmin. Będzie to domyślna baza danych dla naszej instalacji CMS.

Teraz możemy już przystąpić do instalacji CMS. W pasku adresu wpisujemy **http://localhost/Joomla**, co uruchomi znany nam już proces instalacji. W pierwszym kroku wpisujemy parametry witryny oraz dane administratora. W drugim kroku uzupełniamy informacje dotyczące bazy danych według następujących wytycznych:

- typ bazy danych: MySQLi;
- nazwa serwera bazy danych: localhost;
- nazwa użytkownika: root;
- hasło bazy danych: zostawiamy puste;
- nazwa bazy danych: Joomla (taką, jaką utworzyliśmy w phpMyAdmin);
- przedrostek dla tabel bazy danych: domyślnie.

Następnie wybieramy przycisk **Instalacja**. Jeżeli wszystko wpisaliśmy poprawnie, pojawi się okno kończące instalację. Tak jak podczas instalacji na serwerze zdalnym, musimy usunąć folder instalacyjny. Nasze adresy do obsługi CMS będą następujące:

- adres URL witryny: **`http://localhost/Joomla`**;
- URL do administracji: **`http://localhost/Joomla/administrator`**.

UWAGA

Nazwy folderu i bazy danych oraz dane administratora CMS mogą być dowolne i zależą od ustaleń twórcy strony.

SPRAWDŹ SVOJE UMIEJĘTNOŚCI

1. Pobierz pakiet instalacyjny XAMPP właściwy dla twojego systemu operacyjnego i zainstaluj go na lokalnym komputerze. Przeprowadź test poprawności działania interpretera PHP.
2. Zainstaluj dowolny system CMS (zgodny z funkcją, którą ma realizować twoja strona) na lokalnym pakiecie serwerowym XAMPP (lub innym pakiecie serwerowym). Proces instalacji przeprowadź pod nadzorem nauczyciela. Udokumentuj proces instalacji zrzutami ekranu i opisem. Wykonaną dokumentację instalacji zapisz w pliku i przedstaw do oceny nauczycielowi. Zapisz dane administratora CMS niezbędne do pierwszego logowania na zaplecze witryny.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Sprawdź na stronie projektu Joomla!, jakie dodatki są oferowane w ramach pakietu Bitnami XAMPP.
2. Wyszukaj informacje na temat innych pakietów serwerowych odgrywających podobną rolę co aplikacja XAMPP. Wyniki wyszukiwania zapisz w pliku tekstowym lub w zeszycie.
3. Z jakich słów powstał akronim XAMPP?
4. Do jakiego folderu należy wypakować pliki systemu CMS, aby zainstalować go za pomocą pakietu XAMPP? Odpowiedź uzasadnij.

18

Personalizacja i zarządzanie systemem CMS Joomla!

ZAGADNIENIA

- Logowanie do panelu administratora
- Personalizacja witryny
- Edycja treści w CMS Joomla!

Po instalacji systemu CMS Joomla! uruchamia się witryna oparta na domyślnych ustawieniach i z domyślną templatką (szablonem).

Każda struktura CMS składa się z dwóch części nazywanych **front end (witryna)** i **back end (zaplecze)**. Front end to widoczna dla użytkowników witryna internetowa, back end to warstwa administracyjna witryny, która jest dostępna dla zalogowanych do panelu administracyjnego użytkowników.

Aby zmienić wygląd, konfigurować ustawienia i edytować treści, trzeba zalogować się do panelu administratora. W systemie CMS w pasku adresu przeglądarki internetowej należy wpisać adres według następującego szablonu:

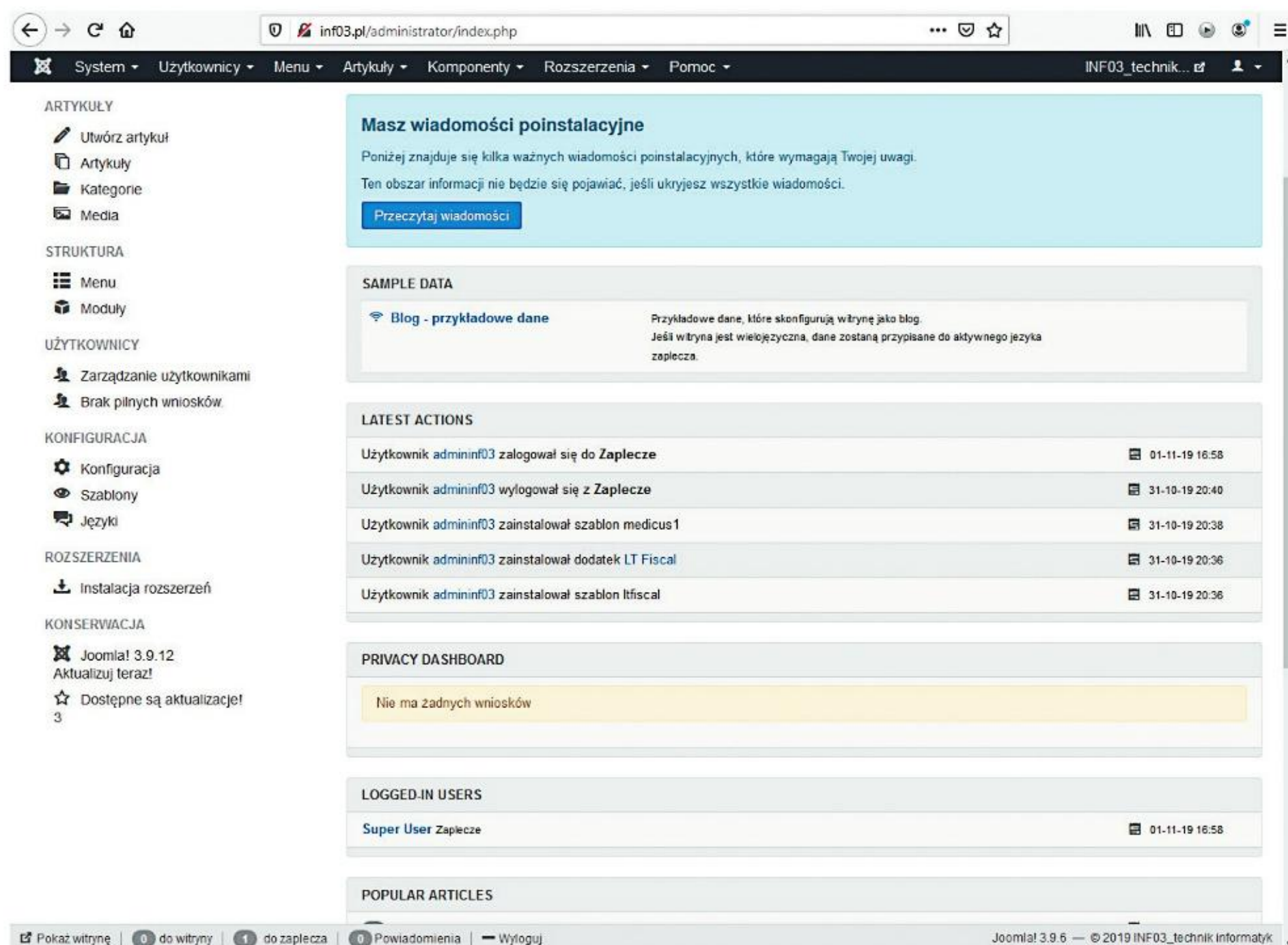
`http://www.nasza-domena.pl/administrator`

Po zatwierdzeniu adresu pojawi się okno logowania do panelu administratora. Tutaj trzeba wpisać nazwę użytkownika i hasło, ustalone podczas instalacji systemu, oraz wybrać język domyślny zaplecza.



Rys. 18.1. Logowanie do panelu administratora CMS Joomla!

Po zalogowaniu otwiera się okno kokpitu użytkownika o uprawnieniach administratora. Uprawnienia administratorów mogą się różnić. Superadministrator ma najwyższe uprawnienia dla całego zaplecza strony. Można również utworzyć profile administratorów – redaktorów z uprawnieniami do tworzenia artykułów, wstawiania grafik itp. Ułatwia to podział prac po stronie zaplecza witryny.



Rys. 18.2. Pulpit zaplecza systemu CMS Joomla!

Do najważniejszych czynności, które powinniśmy wykonać po instalacji systemu CMS jako superadministrator, należą:

- instalacja aktualizacji i pakietu językowego;
- dodanie użytkowników-redaktorów, którzy będą przygotowywać treści na stronę;
- personalizacja wyglądu witryny, czyli edycja lub zmiana szablonu (templatki), oraz edycja jej ustawień;
- dodanie kluczowych elementów graficznych, takich jak ikona strony, logo;
- ustalenie właściwego szablonu strony, menu i innych elementów;
- sprawdzenie aktualizacji i ewentualne wykonanie kopii zapasowej serwisu.

Nie będziemy opisywali wszystkich czynności dotyczących zarządzania systemem CMS, ponieważ każda witryna jest inna i wymaga indywidualnej personalizacji oraz konfiguracji. Skupimy się tylko na najważniejszych, przykładowych czynnościach, takich jak: instalacja pakietu językowego, instalacja i edycja wyglądu szablonu strony, tworzenie

menu oraz redakcja artykułów na stronę. Każda grupa programistów tworzących dany system zarządzania treścią oferuje na swoich stronach dokumentację, porady i samouczki, które pozwalają dokładnie poznać każdego CMS-a. Podczas bardziej zaawansowanych prac na zapleczu witryny niezbędne będzie korzystanie z dokumentacji lub forum.

Konfiguracja wstępna witryny

Po zalogowaniu się na zaplecze witryny widzimy okno pulpitu z informacjami dotyczącymi wszystkich najistotniejszych parametrów działania strony (rys. 18.2). Po lewej stronie okna umieszczono menu, z którego możemy wybrać skróty do konfiguracji poszczególnych elementów witryny. Ten sam efekt otrzymamy, jeśli skorzystamy z górnego menu.

Istotne parametry całego systemu CMS konfigurujemy w menu **Konfiguracja globalna**. W zakładkach: **Witryna**, **System**, **Serwer**, **Filtrowanie** tekstu oraz **Uprawnienia** dokonuje się zmian istotnych dla działania CMS Joomla!.

W zakładce **Witryna** możemy m.in.:

- zmienić nazwę witryny;
- włączyć lub wyłączyć publikację witryny;
- ustawić domyślny edytor WYSIWIG;
- określić domyślny poziom zasięgu (np. dostęp do treści dla każdego bądź dla zalogowanych użytkowników);
- dodać słowa kluczowe dla robotów wyszukiwarek;
- zoptymalizować treści dla wyszukiwarek;
- ustawić opcje ciasteczek.

W zakładce **System** możemy skonfigurować elementy dotyczące diagnostyki, pamięci podręcznej oraz sesji. Bez uzasadnionej konieczności nie powinno się zmieniać ustawień domyślnych. Zakładka **Serwer** umożliwia dokonywanie zmian danych wprowadzonych podczas instalacji systemu CMS. Możemy m.in. zmienić parametry bazy danych czy poczty elektronicznej. **Ustawienia filtrowania tekstu** odpowiadają za bezpieczeństwo witryny. Najlepszym rozwiązaniem jest pozostawienie ustawień domyślnych.

Bardzo ważne są ustawienia dostępne w zakładce **Uprawnienia**. Mamy tutaj możliwość ustalania uprawnień do określonych działań dla poszczególnych grup użytkowników. Dzięki temu jesteśmy w stanie kontrolować pracę licznej grupy użytkowników systemu zarówno po stronie witryny, jak i zaplecza. Na rys. 18.3 przedstawiono hierarchiczną strukturę praw dostępu.

Ustawienia uprawnień (praw dostępu)

Zarządzanie uprawnieniami grup użytkowników. Dodatkowe informacje o uprawnieniach poniżej.

Public	Działanie	Wybierz nowe ustawienia	Wyliczone ustawienia
– Guest	Logowanie do witryny	Dziedziczone ▼	Dozwolone (Dziedziczone)
– Manager	Logowanie do zaplecza	Dziedziczone ▼	Dozwolone (Dziedziczone)
... – Administrator	Dostęp do wyłączonej witryny	Dziedziczone ▼	Dozwolone (Dziedziczone)
– Registered	Pełna kontrola	Dziedziczone ▼	Niedozwolone (Dziedziczone)
... – Author	Konfigurowanie, ale bez ACL	Dziedziczone ▼	Niedozwolone (Dziedziczone)
... – Editor	Dostęp do interfejsu	Dozwolone ▼	Dozwolone
... – Publisher	Tworzenie	Dziedziczone ▼	Dozwolone (Dziedziczone)
– Super Users	Usuwanie	Dziedziczone ▼	Dozwolone (Dziedziczone)
	Dostosowanie	Dziedziczone ▼	Dozwolone (Dziedziczone)
	Dostosowanie stanu	Dziedziczone ▼	Dozwolone (Dziedziczone)
	Dostosowanie własnych	Dziedziczone ▼	Dozwolone (Dziedziczone)
	Dostosowanie pól dodatkowych	Dziedziczone ▼	Niedozwolone (Dziedziczone)

Rys. 18.3. Ustawienia dostępu w CMS Joomla!

W ustawieniach praw dostępu w CMS Joomla! mamy dostępne następujące ustawienia:

- **dziedziczone** – będą użyte uprawnienia z konfiguracji globalnej lub z pozycji nadrzędnej, jeśli taka istnieje;
- **zabronione** – bez względu na ustawienia w konfiguracji globalnej lub pozycji nadrzędnej dostęp zawsze będzie zabroniony; to ustawienie wpływa na pozycje podrzędne;
- **dozwolone** – umożliwi operacje na tym elemencie, jeśli ustawienie nie zostanie nadpisane ustawieniami w konfiguracji globalnej; w razie konfliktu z konfiguracją globalną bądź ustawieniami grupy nadrzędnej te ustawienia nie będą obowiązywać, a konflikt zostanie oznaczony jako Niedozwolone (Blokada);
- **nieustawione** – jest używane jedynie w odniesieniu do grupy Wszyscy w konfiguracji globalnej; grupa Wszyscy jest nadrzędna wobec wszystkich innych grup. Uprawnienia nieustawione są równoważne ustawieniu zabronione, ale mogą zostać zmienione w grupie potomnej, komponentcie, kategorii oraz konkretnej pozycji.

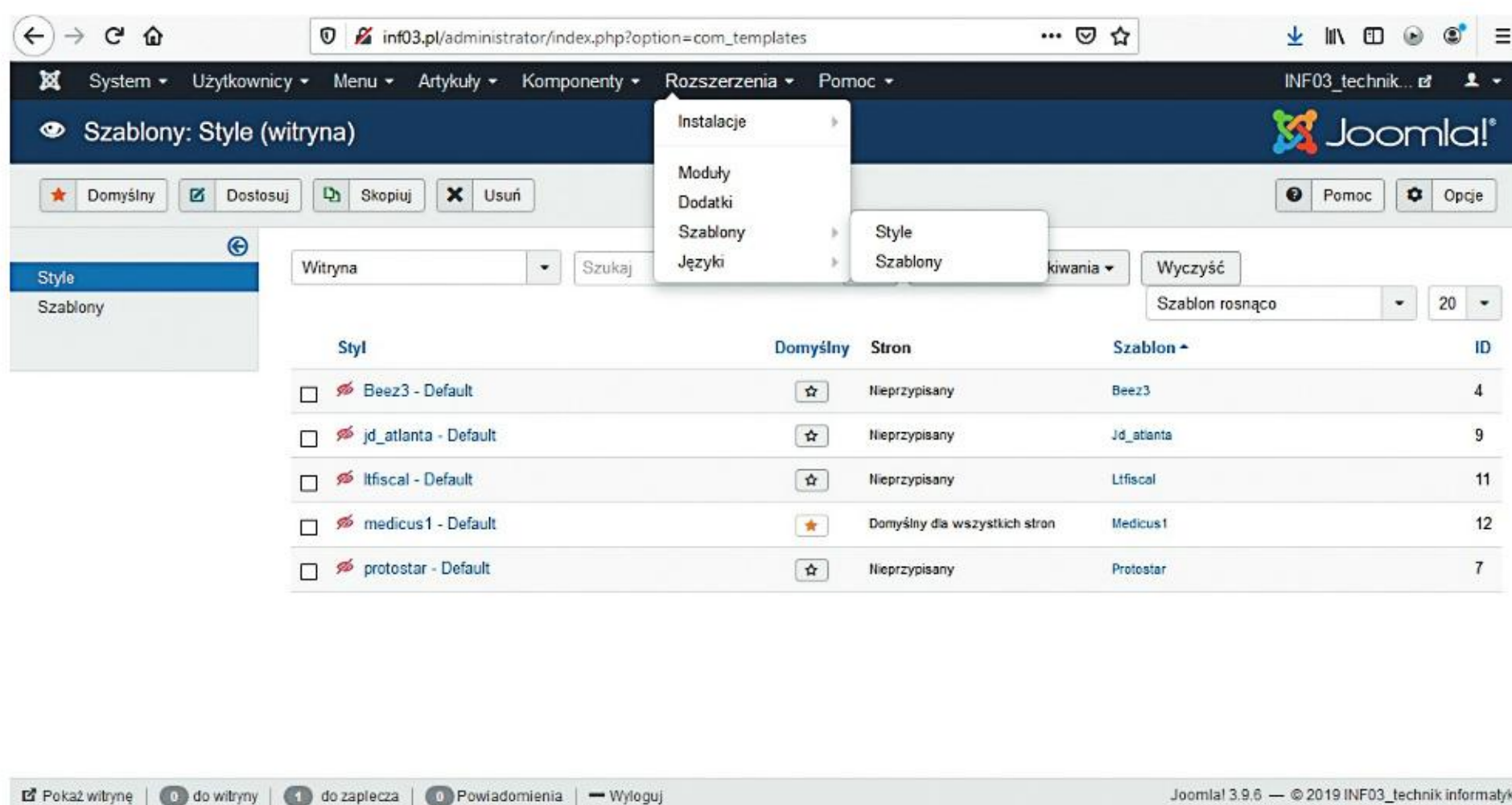
Instalacja szablonu

Za indywidualny wygląd każdej witryny zarządzanej przez pakiet CMS odpowiada szablon. Szablony w CMS Joomla! to rozszerzenia składające się z pakietu plików określających wzorce rozmieszczenia i formatowania treści na stronach witryny lub zaplecza. Szablony decydują o kompozycji stron, kształtach, rozmiarach, kolorystyce stałych elementów witryny oraz wszystkich istotnych cechach typograficznych. Szablony można tworzyć samodzielnie albo skorzystać z gotowych rozwiązań, które mogą być oferowane bezpłatnie oraz za określona opłatą. Z CMS Joomla! są instalowane dwa szablony standardowe witryny (Prostar i Beez3) oraz dwa szablony zaplecza (Hathor i Isis). W jednej witrynie można stosować wiele szablonów, np. w każdym dziale inny szablon. Jeden szablon można również wykorzystać w wielu witrynach.

Każdy szablon powinien składać się z co najmniej trzech elementów:

- pliku **index.php**, w którym jest zawarty wzorzec dokumentu HTML, definiującego strukturę treści witryny;
- pliku arkusza stylów css, określającego układ i reguły formatowania treści strony;
- skryptu konfiguracyjnego **templateDetails.xml**, w którym znajdują się dane szablonu oraz kod opcji konfiguracyjnych.

W sieci Internet znajduje się wiele stron oferujących szablony różnego typu. Aby wykorzystać na swojej stronie gotowy szablon, należy się upewnić, czy można go użyć bezpłatnie, nie naruszając praw autorskich, oraz pobrać skompresowany pakiet na swój komputer.



Rys. 18.4. Szablony w CMS

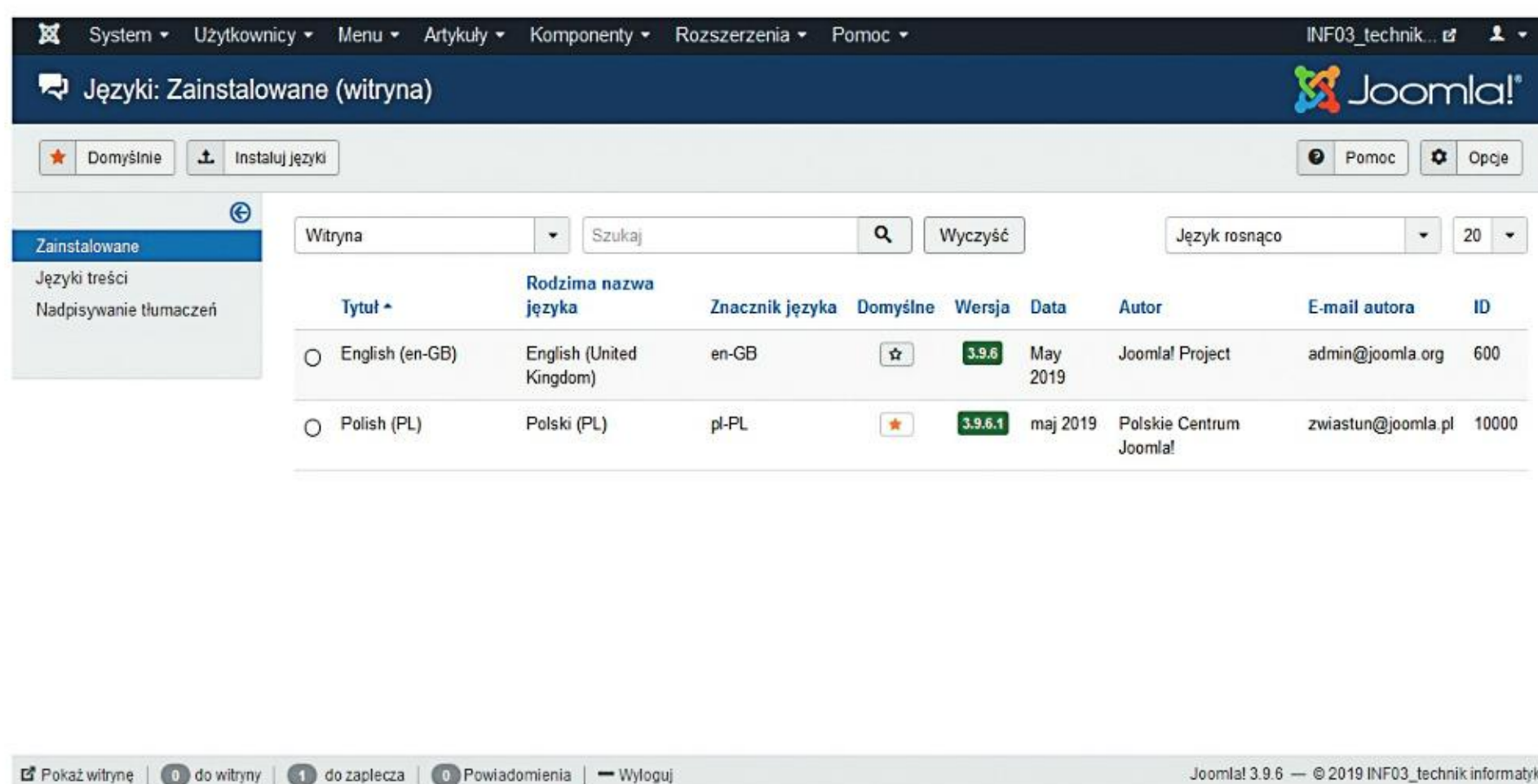
Logujemy się na zaplecze naszej witryny oraz uruchamiamy z menu **Rozszerzenia – Instalacje – Instalowanie**. Następnie zaznaczamy zakładkę **Instaluj z pakietu** i przeciągamy pobrany wcześniej plik na wskazany obszar zaplecza. W tym momencie rozpocznie się proces instalacji szablonu. Jeżeli wszystko odbędzie się bez

problemu, zainstalowany szablon będzie widoczny w oknie szablonów (menu **Rozszerzenia – Szablony**). Na rys. 18.4 przedstawiono okno zainstalowanych szablonów w naszym CMS Joomla!.

Instalacja pakietu językowego

Jednymi z ważniejszych cech wyróżniających CMS Joomla! są rozbudowana kontrola języka i zdolność do tworzenia wielojęzycznych stron internetowych. Pakiet językowy do każdej wersji Joomla! jest dostępny na stronie <https://www.joomla.pl/pobierz-joomla>. Aby go zainstalować, logujemy się na zaplecze naszej witryny oraz uruchamiamy z menu **Rozszerzenia – Instalacje – Instalowanie**. Następnie zaznaczamy zakładkę **Instaluj z pakietu** i przeciągamy pobrany wcześniej plik na wskazany obszar zaplecza. W tym momencie rozpocznie się proces instalacji pakietu. Jeżeli wszystko przebiegnie sprawnie, pakiet językowy zostanie zainstalowany. Następnie wchodzimy jeszcze raz do menu **Rozszerzenia – Instalacje – Instalowanie języka**. Wyświetlą się dostępne pakiety językowe. Wyszukujemy ostatnio dodany pakiet języka polskiego i wybieramy przycisk instalacji.

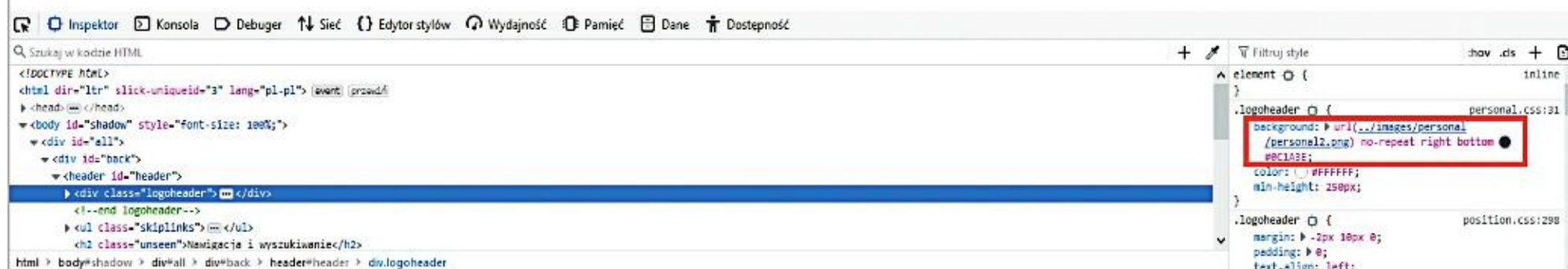
Po zainstalowaniu nowego języka można go ustawić jako domyślny na stronie internetowej bądź jako domyślny w zapleczu witryny, a także stworzyć dodatkową konfigurację dla wielu języków (gdy opracowuje się wielojęzyczną stronę internetową).



Rys. 18.5. Okno wyboru domyślnego języka dla zaplecza i witryny

Zmiana logo witryny

Każdy gotowy szablon zawiera domyślne elementy graficzne. Wśród nich jest również logo strony. Na rys. 16.16 pokazano wygląd domyślny szablonu Joomla!. Za pomocą np. Inspektora (narzędzie dla twórców witryn przeglądarki Firefox) lub analizy pliku **index.php** ustalamy nazwę i położenie pliku logo. W naszym szablonie nazywa się on **personal2.png** i znajduje się w katalogu **./images/personal** (rys. 18.6).



Rys. 18.6. Analiza szablonu za pomocą aplikacji Inspektor

Najprostszym sposobem zmiany logo strony jest podmiana pliku graficznego. Przygotowujemy logo i wysyłamy je do odpowiedniego folderu za pomocą klienta FTP (zastępujemy nim plik domyślny).

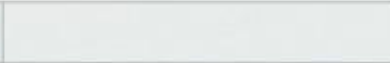
Innym sposobem jest skonfigurowanie wzbogaceń szablonu z poziomu zaplecza witryny: wchodzimy do konfiguracji domyślnego szablonu (rys. 18.7) z menu **Rozszerzenia – Szablony**.

Nazwa stylu * Beez3 - Default

Szczegóły Wzbogacenie Przypisz do pozycji menu

Mała obejmia (%) 53

Duża obejmia (%) 72

Logo   Wybierz X

Tytuł witryny

Opis witryny

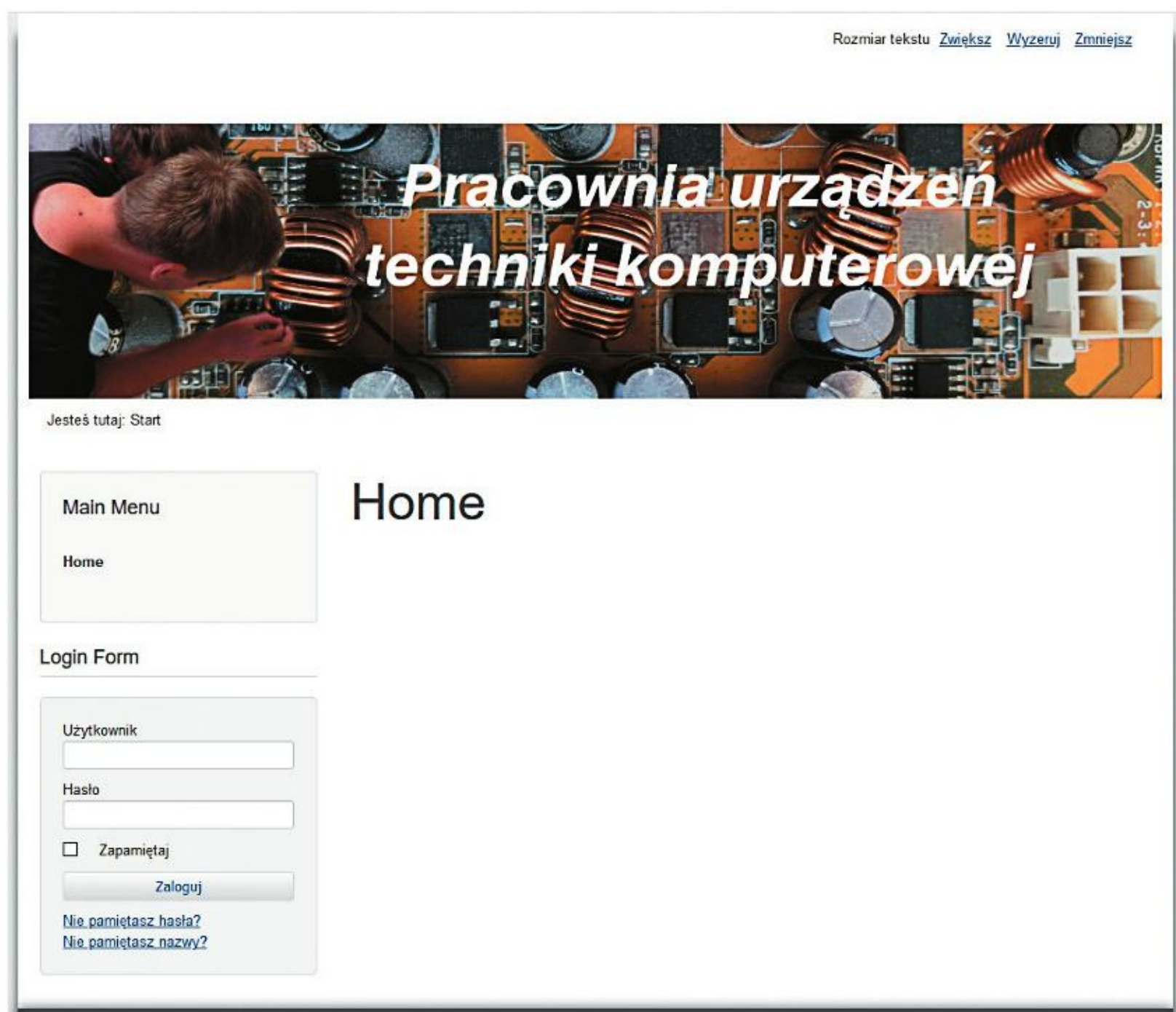
Położenie nawigacji przed treścią

Komponenty wymagające Bootstrap

Kolor szablonu Osobisty

Rys. 18.7. Konfiguracja szablonu za pomocą menu Wzbogacenie

Efektem naszych działań powinna być strona z nowym logo, które będzie nawiązywało do treści zawartych na stronie. Na rys. 18.8 pokazano efekt zamiany logo domyślnego dla Joomla! szablonu Beez3.



Rys. 18.8. Zmienione logo szablonu CMS Joomla!

Tworzenie menu

Każda strona internetowa, tworzona zgodnie ze standardami, musi mieć prosty, przejrzysty i spójny system nawigacji zapewniający łatwy i szybki dostęp do wszystkich treści i funkcji strony. Tę funkcję spełniają elementy strony zwane **menu**. Tworzenie menu w systemie Joomla! stanowi bardzo ważny etap w konfiguracji powstającej witryny. CMS Joomla! ma wbudowany system zarządzania menu. Jest on skonstruowany wokół dwóch podstawowych typów obiektów: komponentów i modułów. Do tworzenia i zarządzania menu po stronie zaplecza CMS służą:

- **projektant menu** – komponent zaplecza administracyjnego, który udostępnia narzędzia do tworzenia i konfiguracji menu;
- **projektant pozycji menu** – pulpit kontrolny z narzędziami do tworzenia i zarządzania pozycjami menu, który jest dodawany po utworzeniu każdego menu;
- **menedżer modułów** – komponent zaplecza administracyjnego, który służy do tworzenia i konfiguracji modułów udostępniających menu na stronie.

! UWAGA

Należy pamiętać o kluczowej zasadzie: w każdej witrynie musi istnieć co najmniej jedno menu główne, w którym znajduje się co najmniej jedna domyślna pozycja.

Aby dodać do witryny menu, należy zrobić trzy rzeczy:

- utworzyć menu za pomocą **Projektanta menu**;
- utworzyć i skonfigurować moduł wyświetlający menu za pomocą **Menedżera modułów**;
- dodać do menu pozycje menu za pomocą **Projektanta pozycji menu** oraz wybrać rodzaj wyświetlania treści (np. artykuł).

Aby utworzyć nowe menu, logujemy się do zaplecza witryny Joomla! oraz wybieramy pozycję **Menu – Projektant menu – Utwórz menu**. Uzupełniamy szczegóły dotyczące tworzonego menu:

- **Tytuł** – sygnalizujący zawartość menu; jest on widoczny tylko na zapleczu witryny;
- **Typ menu** – unikalna jednowyrazowa nazwa; nazwa typu menu jest identyfikatorem odróżniającym menu od innych;
- **Opis** – dodatkowy opis przeznaczenia menu; opis nie jest nigdzie wyświetlany.

Na rys. 18.9 przedstawiono przykład tworzenia menu **Materiały z zajęć**.

Rys. 18.9. Okno tworzenia menu

Aby skorzystać w witrynie z nowego menu, trzeba jeszcze utworzyć moduł wyświetlający to menu: wybieramy **Rozszerzenia – Moduły – Utwórz**. W oknie dialogowym wyszukiujemy rodzaj modułu – **Menu**. Następnie uzupełniamy szczegóły modułu:

- **Tytuł** – wymagany, choć nie zawsze wyświetlany w witrynie;
- **Wybierz menu** – wybieramy nazwę menu, do którego przyporządkowujemy moduł;
- **Pozycja** – określa, w którym miejscu strony będzie wyświetlane menu; pozycje są zdefiniowane w szablonie witryny (to, jak podejrzeć rozmieszczenie pozycji modułów w szablonie, opiszemy w następnym podrozdziale);
- **Pokaż pozycje podmenu** – zezwalamy, jeżeli będzie to menu rozwijane.

Opcje **Rozpocznij od poziomu** i **Najniższy poziom** wykorzystuje się do tworzenia menu lokalnych (wielopoziomowych).

Na rys. 18.10 pokazano przykładowe utworzenie modułu **Menu_gorne**.

The screenshot shows the Joomla! administrator interface for creating a new module. The top navigation bar includes links for System, Użytkownicy, Menu, Artykuły, Komponenty, Rozszerzenia, and Pomoc. The main header displays 'Moduły: Menu' and the Joomla! logo. Below the header, there are buttons for 'Zapisz', 'Zapisz i zamknij', 'Zapisz i nowy', and 'Anuluj'. The form itself is titled 'Menu' and has a sub-header 'Witryna'. The description reads 'Moduł wyświetla menu na stronach witryny'. The form is divided into two main sections. The left section contains fields for 'Wybierz menu' (set to 'Materiały z zajęć'), 'Pozycja bazowa' (set to 'Widoczny'), 'Rozpocznij od poziomu' (set to '1'), 'Najniższy poziom' (set to 'Wszystkie'), and 'Pokaż pozycje podmenu' (set to 'Tak'). The right section contains fields for 'Pokaż tytuł' (set to 'Pokaż'), 'Pozycja' (set to 'Góra [position-1]'), 'Stan' (set to 'Opublikowano'), 'Rozpocznij publikację' (empty), 'Zakończ publikację' (empty), 'Dostęp' (set to 'Public'), 'Porządek' (set to 'Brak'), 'Język' (set to 'Wszystkie'), and 'Notatka' (empty). At the bottom, there is a status bar showing 'Pokaż witrynę', '0 do witryny', '1 do zaplecza', '0 Powiadomienia', and a 'Wyloguj' button. The Joomla! version '3.9.6' and copyright '© 2019 Joomla!' are also displayed.

Rys. 18.10. Okno tworzenia modułu

Następnym etapem jest utworzenie pozycji w naszym menu. Aby to zrobić, wchodzimy do **Menu – Pozycje** menu oraz naciskamy przycisk **Utwórz**. Po wpisaniu nazwy pozycji menu wybieramy **Typ pozycji menu**. Każdej tworzonej pozycji menu można nadać określony typ. Mamy do dyspozycji m.in.:

- Artykuły,
- Doniesienia,
- Tagi,
- Wpinacz,
- Wyszukiwarkę,
- Łączy systemowe (linki).

Jeżeli pozycja odnosi się do artykułu lub listy artykułów, należy najpierw utworzyć artykuły.

The screenshot shows the Joomla! administration interface for creating a new menu item. The top navigation bar includes links for System, Users, Menu, Articles, Components, Extensions, and Help. The main heading is 'Pozycje menu: Utwórz'. Below the heading are buttons for 'Zapisz', 'Zapisz i zamknij', 'Zapisz i nowy', and 'Anuluj'. The form fields include:

- Tytuł menu ***: Informacje
- Alias**: Generuj z tytułu
- Typ pozycji menu ***: Pojedynczy artykuł (Wybierz)
- Wybierz artykuł ***: Informacja (Dostosowanie, Wyczyść)
- Łącze**: index.php?option=com_content&view=article
- Otwórz w**: W głównym oknie przeglądarki
- Styl szablonu**: - Użyj domyślnych -
- Pokaż w menu ***: Materiały z zajęć
- Pozycja macierzysta**: Główna pozycja
- Kolejność**: Nowa kolejność będzie ustalona po zapisaniu.
- Stan**: Opublikowano
- Strona domyślna**: Tak (Nie)
- Dostęp**: Public
- Język**: Wszystkie
- Uwaga**: (empty text area)

The bottom status bar shows 'Pokaż witrynę', '0 do witryny', '1 do zaplecza', '0 Powiadomienia', and 'Wyloguj'. The Joomla! version is 3.9.6, and the copyright is © 2019 Joomla!.

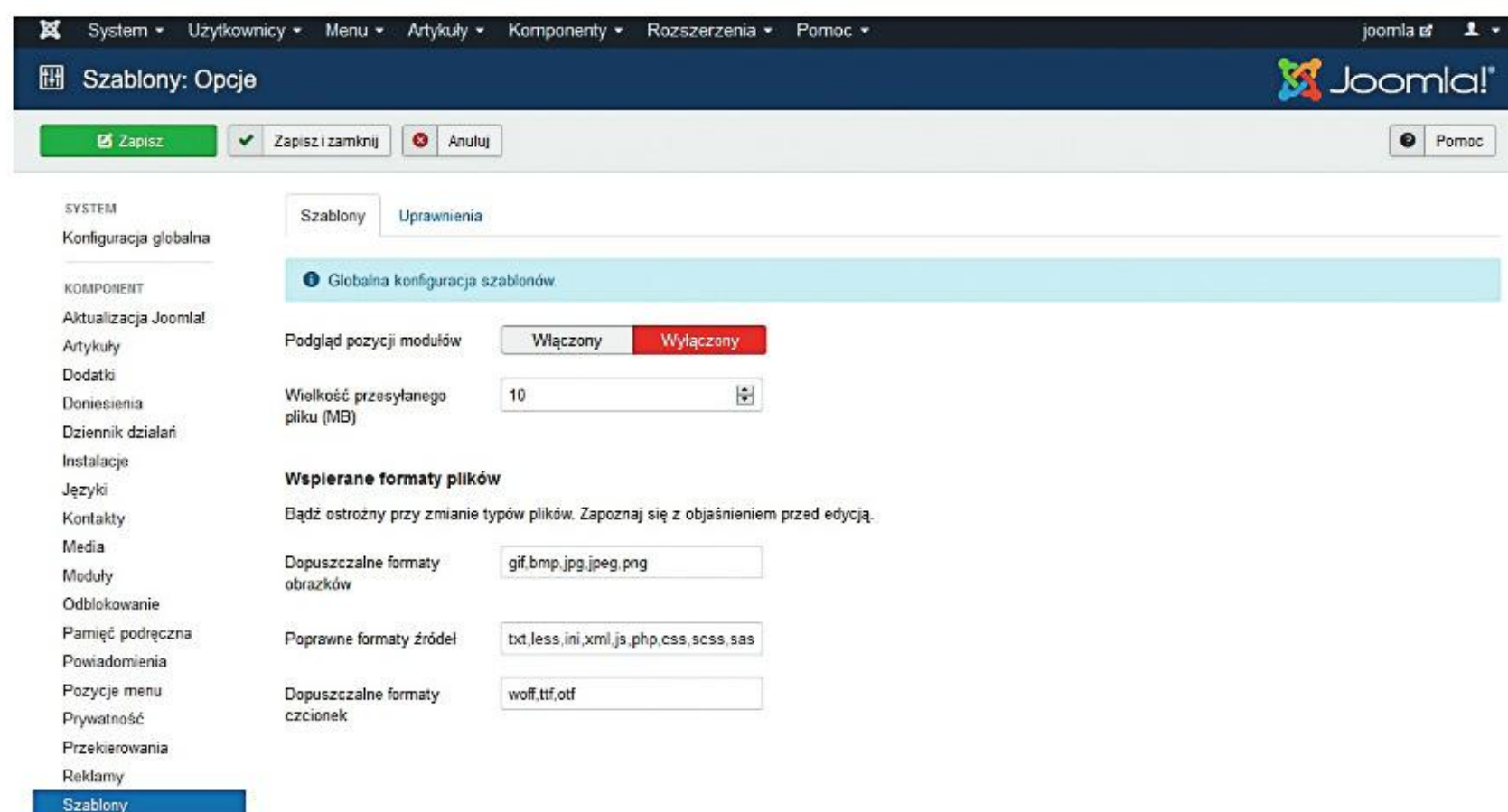
Rys. 18.11. Okno tworzenia pozycji menu

Po zatwierdzeniu pozycji menu powinno być ono widoczne w menu witryny.

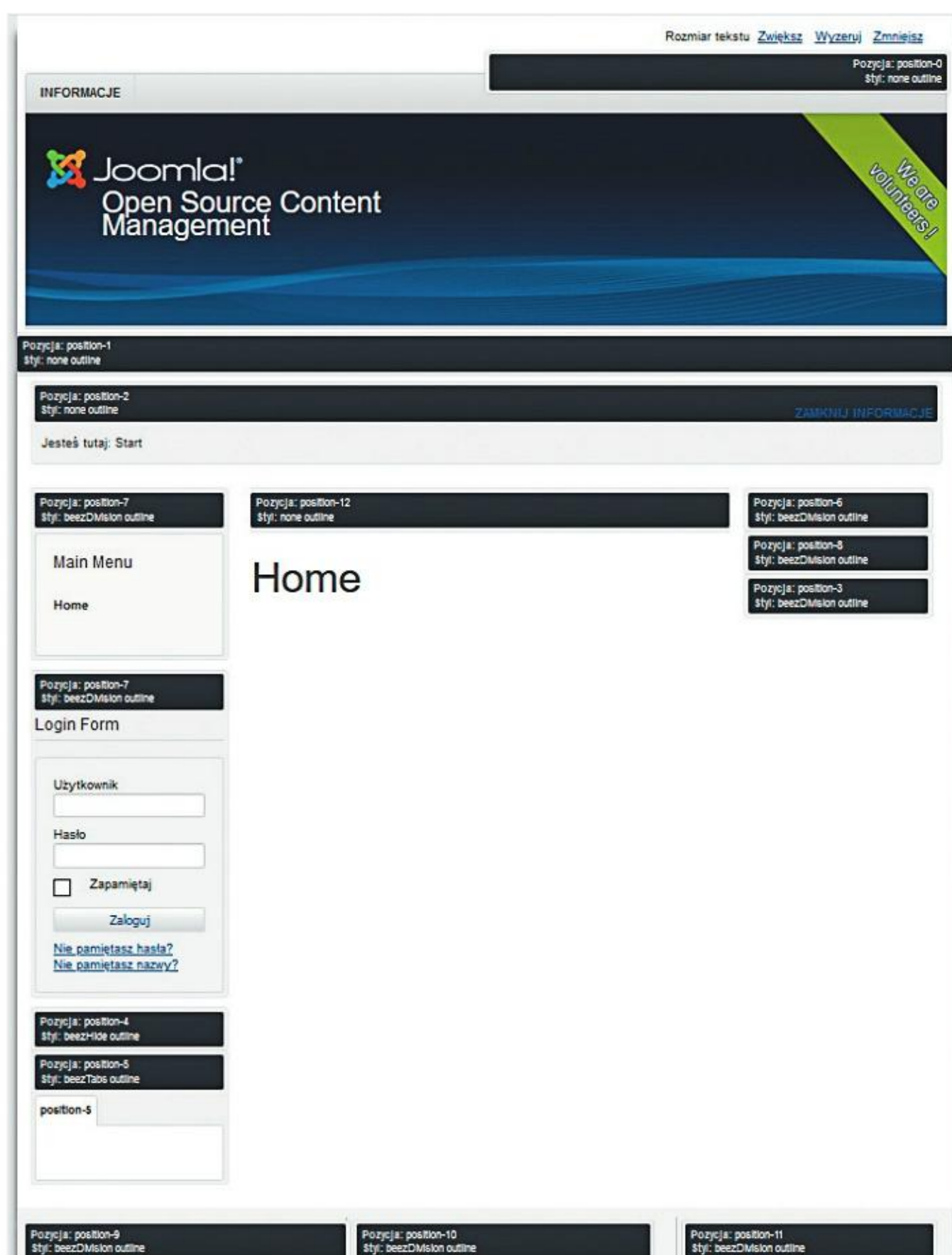
Pozycje w szablonie

Podczas konfigurowania modułów w CMS Joomla! należy wskazać, w którym miejscu strony ma się on wyświetlać. Na moduły przeznaczono w szablonie obszary zwane **pozycjami modułów**. Domyślnie CMS Joomla! ma wyłączony podgląd pozycji w szablonie. Aby zobaczyć, jak nazywają się poszczególne pozycje oraz w jakim miejscu szablonu się znajdują, należy zalogować się do zaplecza witryny, a następnie uruchomić **Rozszerzenia – Szablony**. W oknie szablonów wybieramy przycisk **Opcje** i włączamy **Podgląd pozycji modułów**.

Po zapisaniu wyboru i powrocie do okna Szablony wybieramy kartę **Szablony**. Przy naszym domyślnym szablonie strony naciskamy opcję **Podgląd**. W oknie przeglądarki wyświetli się nasza witryna z zaznaczonymi poszczególnymi pozycjami (rys. 18.13). Ze względu na bezpieczeństwo witryny należy wyłączyć możliwość podglądu modułów, gdy już nie będzie nam potrzebna.



Rys. 18.12. Włączenie podglądu pozycji szablonu



Rys. 18.13. Pozycje szablonu Beez3

Redagowanie artykułów

Zarządzanie treścią w systemie CMS sprowadza się głównie do dodawania i zarządzania artykułami. Artykuł może tworzyć stronę startową, listę aktualności albo wyświetlać się w postaci pojedynczego dokumentu bądź linku. Aby prawidłowo zarządzać systemem CMS, trzeba stworzyć obraz struktury fizycznej witryny w postaci przyporządkowania artykułów do elementów menu.

Podczas tworzenia menu i jego elementów w postaci odnośników określa się, jaki rodzaj artykułu będą otwierać. Kluczowe dla przyporządkowania artykułu do elementów strony są nadanie nazwy oraz dołączenie go do określonej kategorii.

Przed przystąpieniem do edycji artykułów należy utworzyć ich określone kategorie (np. aktualności, archiwum). Wykonuje się to po zalogowaniu na zaplecze strony oraz po wyborze menu **Artykuły – Kategorie – Utwórz kategorię**.

Po utworzeniu kategorii można przystąpić do edycji artykułów. Po wyborze opcji **Artykuły – Artykuły – Utwórz artykuł**, po otwarciu okna artykułów, wpisuje się jego tytuł, alias i kategorię. Sama edycja wpisu jest intuicyjna i prosta, ponieważ sprowadza się do obsługi wewnętrznego edytora WYSIWYG. Można tutaj edytować tekst bądź wstawić obiekty graficzne.

Gotowy artykuł zapisuje się i sprawdza, czy jest widoczny w określonej lokalizacji. Jeżeli artykuł ma być bezpośrednio powiązany z odsyłaczem elementu menu, należy go wybrać w oknie edycji danego elementu menu jako związany.

System Użytkownicy Menu Artykuły Komponenty Rozszerzenia Pomoc Joomla!

Artykuły: Utwórz artykuł

Zapisz Zapisz i zamknij Zapisz i nowy Anuluj Pomoc

Tytuł * Cele kształcenia Alias Generuj z tytułu

Treść Obrazki i łącza Opcje Opcje publikacji Opcje konfiguracji Uprawnienia

Edycja Wstaw Widok Format Tabela Narzędzia

Cele kształcenia w zawodzie technik informatyk.

Celem kształcenia zawodowego jest przygotowanie uczących się do życia w warunkach współczesnego świata, wykonywania pracy zawodowej i aktywnego funkcjonowania na zmieniającym się rynku pracy.

Absolwent szkoły prowadzącej kształcenie w zawodzie technik informatyk powinien być przygotowany do wykonywania zadań zawodowych:

W zakresie kwalifikacji **INF.02. Administracja i eksploatacja systemów komputerowych, urządzeń peryferyjnych i lokalnych sieci komputerowych:**

- 1) przygotowania do pracy systemu komputerowego i urządzeń peryferyjnych;
- 2) administrowania systemami operacyjnymi;
- 3) serwisowania i naprawiania urządzeń techniki komputerowej;
- 4) przygotowania i eksploatacji lokalnej sieci komputerowej;

Wzrost jakości i efektywności pracy, a także bezpieczeństwa i niezawodności systemu komputerowego i urządzeń peryferyjnych.

Stan: Opublikowano

Kategoria *: Uncategorised

Wyróżnione: Tak Nie

Dostęp: Public

Język: Wszystkie

Tagi: Wpisz lub wybierz opcję

Notatka:

Pokaż witrynę 0 do witryny 1 do zaplecza 0 Powiadomienia Wyloguj Joomla! 3.9.6 — © 2019 Joomla!

Rys. 18.14. Okno edycji wpisu

Po wykonaniu personalizacji witryny oraz uzupełnieniu treści można sprawdzić ostateczny wygląd naszego CMS od strony witryny (trzeba wpisać adres domenowy).



Rys. 18.15. Wygląd gotowej witryny internetowej

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Pobierz i zainstaluj najnowszy pakiet językowy dla twojego systemu CMS Joomla!.
2. Wyszukaj i zainstaluj szablon, który najlepiej będzie pasował do treści prezentowanej przez twoją witrynę.
3. Sprawdź rozmieszczenie pozycji twojego szablonu CMS.
4. Utwórz dowolne menu zawierające cztery pozycje odnoszące się do określonej kategorii artykułów, pojedynczego artykułu oraz linku do wybranej strony internetowej.
5. Utwórz kilka artykułów jako typ pozycji menu utworzonych w punkcie 4.
6. Utwórz artykuł, który będzie głównym artykułem witryny podłączonym do domyślnej pozycji głównego menu.



SPRAWDŹ SWOJĄ WIEDZĘ

1. Przeprowadź analizę dostępnych systemów CMS do tworzenia witryn internetowych. Przygotuj zestawienie zalet i wad dla trzech wyszukanych systemów zarządzania treścią. Określ, który – twoim zdaniem – jest najlepszy i uzasadnij dlaczego.
2. Hasło administratora witryny CMS powinno być silne. Opisz zasady tworzenia silnych haseł.
3. Jaką rolę odgrywają pozycje w szablonie witryny?
4. Opisz typy pozycji menu dostępne w CMS Joomla! 3.X.

VI. Projektowanie grafiki komputerowej

- Pojęcie grafiki komputerowej
- Grafika rastrowa
- Grafika wektorowa
- Edytory graficzne
- Projektowanie i tworzenie grafiki komputerowej

19

Obraz cyfrowy

ZAGADNIENIA

- Powstawanie obrazu cyfrowego
- Podstawowe parametry opisujące grafikę komputerową
- Grafika rastrowa
- Grafika wektorowa
- Rodzaje formatów plików graficznych

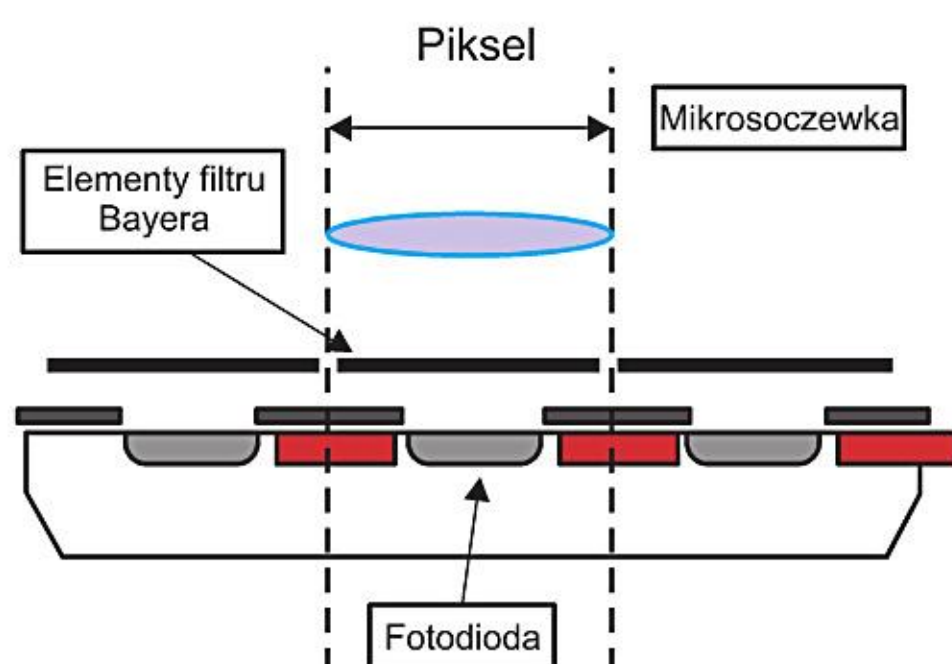
Pojęcie obrazu, znane człowiekowi od niemal 30 000 lat, najczęściej kojarzy się z malarstwem. Obraz definiuje się jako dwuwymiarową płaską formę plastyczną, na której za pomocą farb, po zastosowaniu różnych technik malarskich i graficznych, wykonano zapis pewnych treści związanych z rzeczywistością lub fantazją.

Wraz z pojawieniem się technologii cyfrowej pojawiła się możliwość zapisywania również w taki sposób obrazu.

Obraz cyfrowy (ang. *digital image*) to dwuwymiarowy obraz analogowy zapisany w postaci binarnej, możliwy do przechowywania w komórkach pamięci.

Obraz cyfrowy może powstać przez wykonanie zdjęcia aparatem z matrycą światłoczułą, zeskanowanie obrazu analogowego lub utworzenie mapy bitowej za pomocą aplikacji do tworzenia i edycji grafiki.

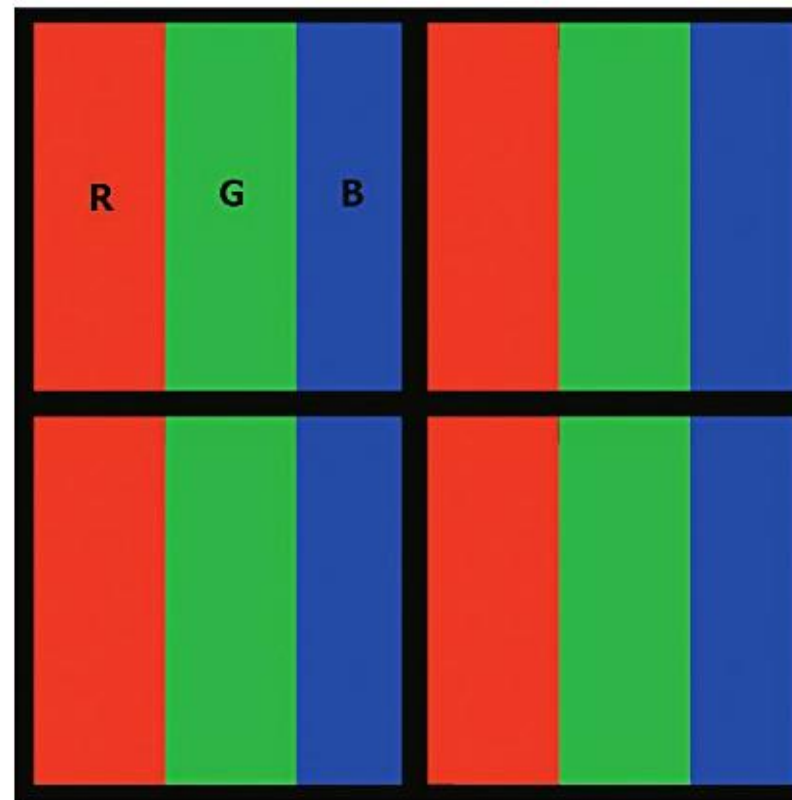
Pierwsze próby skanowania fotografii analogowej sięgają 1957 r., ale dopiero po wynalezieniu matrycy światłoczułej w 1969 r. można mówić o początku ery cyfrowego obrazu. Pierwszy egzemplarz matrycy **CCD** (ang. *Charge Coupled Device*) miał zaledwie osiem elementów światłoczułych, umieszczonych w jednym rzędzie. Jeden element światłoczuły rejestruje, a następnie pozwala odczytać sygnał elektryczny dla jednego piksela.



Rys. 19.1. Przekrój matrycy CMOS

Dodatkowe zastosowanie filtra barwnego RGB umożliwia rejestrację natężenia określonej szerokości spektrum światła w danym punkcie matrycy.

Zasada wyświetlania obrazu na monitorach komputerowych zależy od ich budowy i zastosowanej technologii, ale zazwyczaj jeden piksel wyświetlający element obrazu składa się z trzech punktów odpowiadających za podstawowe składowe barwy RGB.



Rys. 19.2. Piksele w monitorze LCD

Obraz cyfrowy opisuje się takimi pojęciami, jak:

- **piksel** (ang. *pixel*) – najmniejszy element dwuwymiarowego obrazu cyfrowego, który ma stałą barwę i określone położenie;
- **punkt** (ang. *dot*) – najmniejszy element obrazu, który nie ma już wewnętrznej struktury;
- **linia** (ang. *line*) – poziomy zbiór punktów lub pikseli;
- **rozdzielczość obrazu** (ang. *image resolution*) – opisuje liczbę punktów obrazu przypadających na jednostkę długości.

Rozdzielczość obrazu najczęściej wyraża się w:

- **dpi** (*dot per inch*) – punktach na cal;
- **ppi** (*pixel per inch*) – pikselach na cal;
- **lpi** (*line per inch*) – liniach na cal;
- **rozmiarze obrazu** (ang. *image size*) – liczbie punktów na szerokość i wysokość obrazu (np. 800 × 600 dpi);
- **formacie obrazu** (ang. *image aspect ratio*) – wymiarach wyświetlanego obrazu wyrażonych ilorazem jego długości do wysokości (np. 4 : 3).

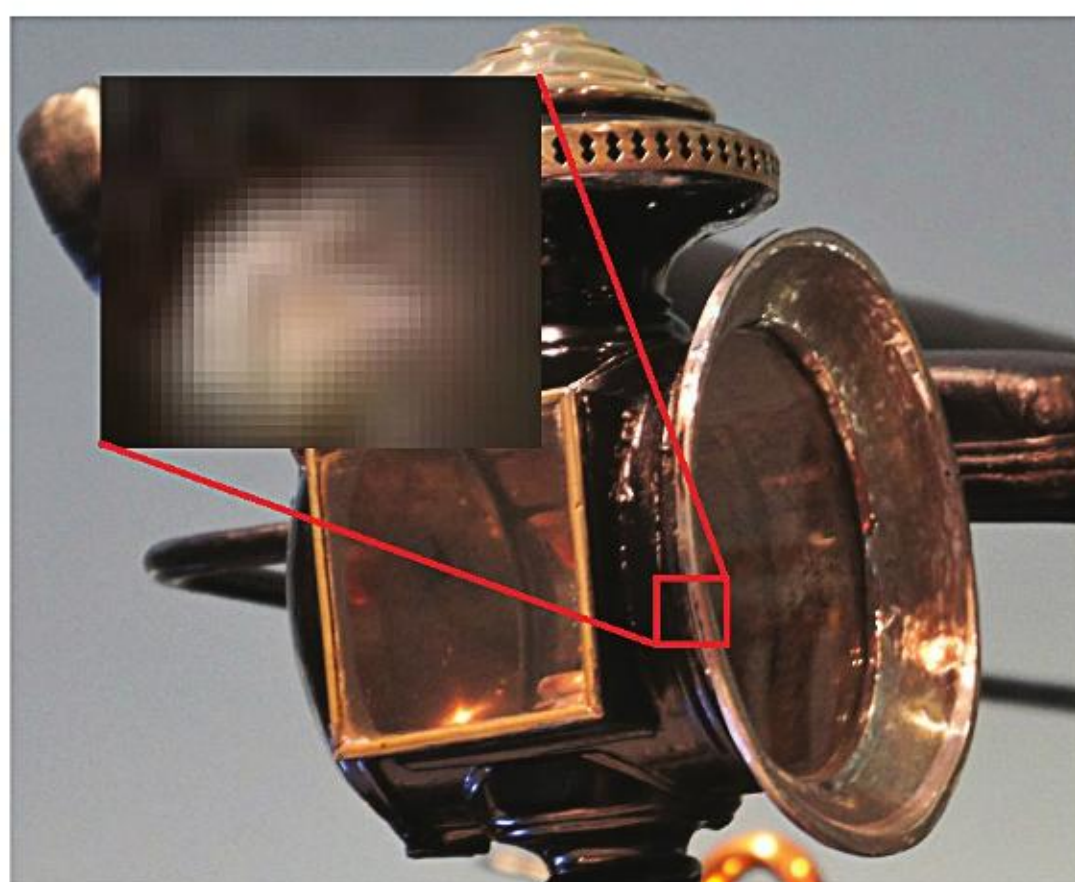
Grafika komputerowa

Możliwość zapisu obrazu w formie cyfrowej umożliwiło powstanie całej dziedziny informatyki zajmującej się wykorzystaniem technik komputerowych do celów wizualizacji artystycznej. Jest to tak szerokie zagadnienie, że funkcjonuje na rynku jako oddzielny zawód, do którego wykonywania, oprócz umiejętności informatycznych i matematycznych, potrzeba również zmysłu plastycznego i wyobraźni.

Grafika rastrowa

Matrycę cyfrową buduje się z wielu milionów pikseli. Umożliwia to zapis i przetwarzanie obrazów wysokiej jakości. Obraz zapisany za pomocą matrycy tworzy **bitmapę** (ang. *bitmap*), czyli tablicę o określonej liczbie pikseli opatrzonych atrybutami barwy. Obrazy cyfrowe opisane bitmapą nazywa się **grafiką rastrową** (ang. *raster image*).

Jakość grafiki rastrowej określa się przez całkowitą liczbę pikseli (**rozdzielczość obrazu**) oraz ilości informacji o barwie zawartej w każdym pikselu. Im większe są liczba pikseli w danym rozmiarze obrazu oraz liczba bitów opisujących barwę, tym obraz ma lepszą jakość. Zbyt mała liczba pikseli powoduje powstanie efektu **pikselizacji**, czyli uwidocznienia się poszczególnych wiązek pikseli, co prowadzi do deformacji obrazu.



Rys. 19.3. Efekt pikselizacji

Rozdzielczość często podaje się jako liczbę pikseli na jednostkę długości i opisuje jednostką **ppi** (ang. *pixels per inch*). Przykładowo: rozdzielczość dobrej jakości drukarki domowej osiąga 200 ppi. Oznacza to, że na wydruku o wielkości 1 cal na 1 cal znajduje się 40 000 pikseli.

Liczba bitów pamięci przeznaczonej do zapisania informacji o barwie nazywa się **głębią barwy**. Jej jednostką jest **bpp** (ang. *bits per pixel*). Barwa opisana pojedynczym bitem tworzy obraz monochromatyczny, czyli jednokolorowy.

Współczesna grafika rastrowa stosuje głębie 24-, 32- oraz 48-bitową. Wierną reprezentację obrazu rzeczywistego daje już zastosowanie kodowania 24-bitowego. W modelu RGB w zapisie 24-bitowym otrzymuje się paletę 16,7 mln kolorów. Tę liczbę uzyskuje się z 256 odcieni z każdej składowej modelu RGB ($28 \cdot 28 \cdot 28 = 224$).

Obrazy rastrowe można również tworzyć za pomocą aplikacji do edycji rastrowej. Grafika powstaje wówczas przez określenie wymiarów rysunku i nadanie określonych parametrów pikselom za pomocą narzędzi dostępnych w programie oraz zapisaniu efektu pracy w pliku o formacie rastrowym.

Do najczęściej używanych edytorów grafiki rastrowej należą: Adobe Photoshop, GIMP i MS Paint.

Podczas zapisywania obrazu cyfrowego, zarówno tworzonego przez kamerę czy aparat, jak i przez edycję w aplikacji, bardzo duże znaczenie ma wybór formatu pliku. Format może decydować o jakości, ilości pamięci potrzebnej do zapisu pliku lub o możliwości jego dalszej obróbki. W grafice rastrowej istotnym elementem w operacji zapisu jest **kompresja obrazu** (ang. *data compression*). Kompresja umożliwia zmniejszenie rozmiaru pamięci zajmowanej przez plik. Jeżeli operacja kompresji pozwoli na powrót do oryginału, to nazywa się ją **kompresją bezstratną**. Jeżeli natomiast kompresja spowoduje utratę pewnej ilości danych, zwie się ją **kompresją stratną**.

Do najpopularniejszych formatów plików grafiki rastrowej z kompresją stratną zalicza się:

- **TIFF** – standard plików graficznych przeznaczonych do wydruku lub publikacji o dużej głębi kolorów, kompresji stratnej i bezstratnej oraz umożliwiający przechowywanie kanału alfa;
- **JPEG** – format powszechnie stosowany w fotografii cyfrowej, mający dobrą jakość obrazu i niewielkie rozmiary pliku;
- **JPEG 2000** – nowsza wersja formatu JPEG oferująca lepszą kompresję;
- **DjVu** – format stworzony do przechowywania zeskanowanych dokumentów w formie elektronicznej.

Do najpopularniejszych formatów plików grafiki rastrowej z kompresją bezstratną zalicza się:

- **BMP** – domyślny format systemu Windows, charakteryzuje się dużą wielkością pliku;
- **GIF** – format powszechnie stosowany na stronach internetowych do zapisu prostej grafiki lub animacji z 8-bitową głębią koloru;
- **PNG** – format stworzony do zapisu grafiki na potrzeby stron internetowych.

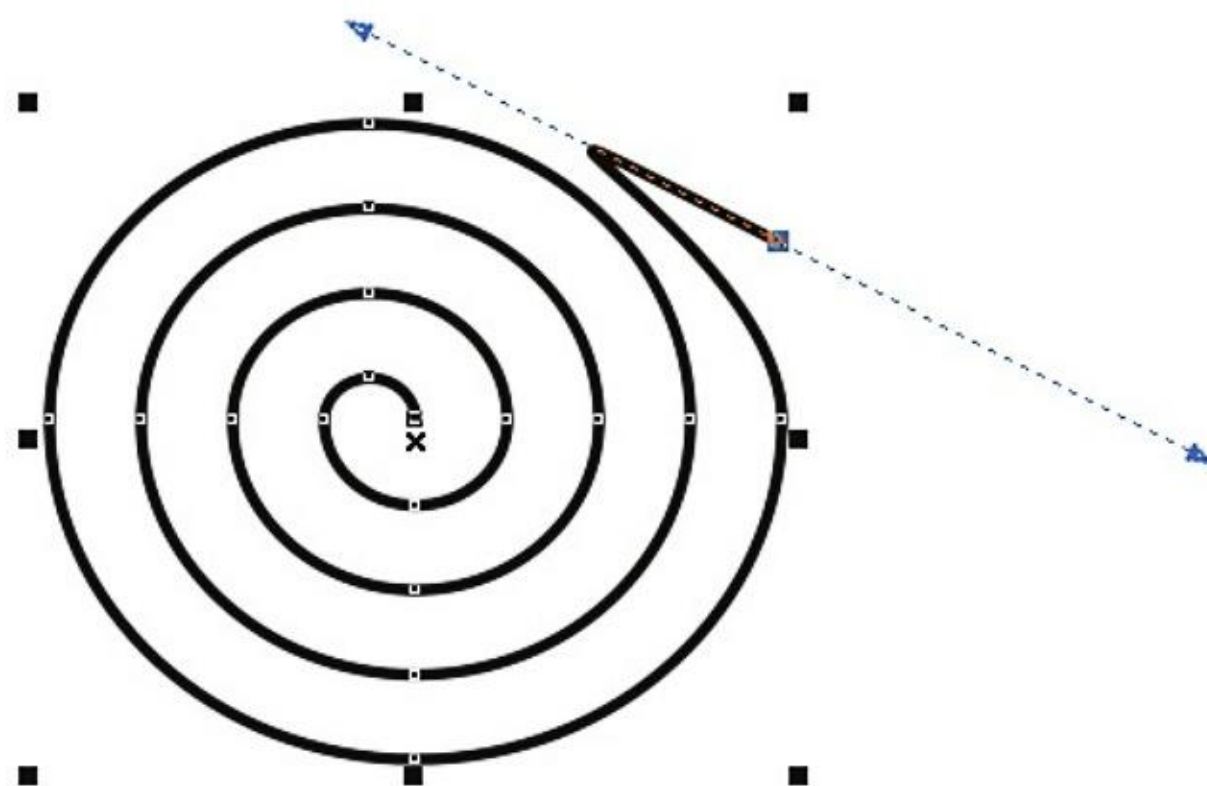
Inne formaty dotyczące grafiki rastrowej:

- **XCF** – wewnętrzny format popularnego programu GIMP pozbawiony kompresji;
- **RAW** – format zawierający wszystkie dane obrazu niezbędne do dalszej obróbki;
- **PSD** – mapa bitowa programu Adobe Photoshop, może przechowywać wiele warstw.

Grafika wektorowa

Grafika wektorowa (ang. *vector graphics*) to obraz cyfrowy zapisany w postaci obiektów lub figur geometrycznych opisanych wzorami matematycznymi. Dzięki temu zmiany parametrów obrazu nie powodują strat w jakości czy proporcjach. Ten rodzaj grafiki jest przydatny w tworzeniu obrazów z niewielką liczbą szczegółów oraz takich, które nie są dokładnym odzwierciedleniem rzeczywistości. Główne zastosowanie grafiki wektorowej znajdziemy w reklamie, projektowaniu, grach komputerowych lub efektach specjalnych w filmach.

Obrazy wektorowe powstają w edytorach graficznych. Do edycji figur geometrycznych służą **krzywe Béziera** (ang. *Bézier curves*), które są opisane punktami mającymi współrzędne wyrażone funkcjami liczb rzeczywistych.



Rys. 19.4. Grafika wektorowa

Do najczęściej używanych edytorów grafiki wektorowej należą: CorelDraw, Adobe Illustrator, Inkscape.

Do najpopularniejszych formatów plików grafiki wektorowej zalicza się:

- **AI** – format programu Adobe Illustrator;
- **SVG** – format do publikacji dwuwymiarowej grafiki wektorowej statycznej i animowanej na stronach internetowych;
- **SWF** – format wektorowej grafiki programu Adobe Flash;
- **CDR** – format macierzysty programu CorelDraw.

Podczas obróbki grafiki komputerowej może zachodzić potrzeba konwersji obrazu opisanego w formie wektorowej na obraz rastrowy w celu wyświetlenia na monitorze ekranowym, wydrukowania lub zapamiętania w pliku w formacie rastrowym. Ten proces nazywa się **rasteryzacją**.

Grafika 3D

Grafika 3D zajmuje się głównie wizualizacją obiektów trójwymiarowych. W tym typie grafiki obiekty są umieszczone w przestrzeni trójwymiarowej. Celem programu komputerowego jest przede wszystkim przedstawienie trójwymiarowego świata na dwuwymiarowym obrazie. W grafice trójwymiarowej obiekty są przedstawiane za pomocą **wokseli** lub **opisu matematycznego**.

Woksel to pojedynczy, najmniejszy element składający się na przestrzeń w grafice trójwymiarowej. W grafice 3D stanowi on odpowiednik piksela występującego w grafice dwuwymiarowej.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Oceń prawdziwość poniższych zdań.

W grafice rastrowej	Prawda	Fałsz
najmniejszym elementem dwuwymiarowego obrazu cyfrowego jest woksel.		
format tiff charakteryzuje się tylko bezstratną kompresją obrazu.		
rozdzielczość jest często podawana jako liczba pikseli na jednostkę długości.		

W grafice wektorowej	Prawda	Fałsz
do edycji figur geometrycznych służą krzywe Béziera.		
każdy edytor ma swoje dedykowane rozszerzenie plików.		
zmiany parametrów obrazu nie powodują strat w jakości.		

- Opisz, w jaki sposób powstaje obraz cyfrowy.
- Wyjaśnij zasadę działania matrycy CDD. Wyszukaj w zasobach sieci Internet urządzenia (skanery, aparaty fotograficzne, kamery cyfrowe), które są wyposażone w matryce CDD.
- Wymień różnice między grafiką rastrową a wektorową.
- Sporządź tabelę, w której zawrzesz rodzaje grafiki komputerowej oraz ich zastosowanie z przykładami.
- Wyjaśnij, na czym polega różnica między kompresją stratną a bezstratną.
- Sporządź zestawienie dostępnych na rynku programów do obróbki grafiki rastrowej i wektorowej.

20

Edytor grafiki rastrowej GIMP

ZAGADNIENIA

- Zasady pracy w edytorze GIMP
- Przykłady zastosowań praktycznych

Do edycji grafiki cyfrowej można używać wielu aplikacji o różnym poziomie możliwości tworzenia i przetwarzania obrazów. W tym rozdziale opisano kilka z tych, które oferują duże możliwości, a są dostępne w ramach licencji **GPL** (ang. *General Public License*), czyli nieodpłatnie z otwartym źródłem.

Edytor grafiki rastrowej GIMP

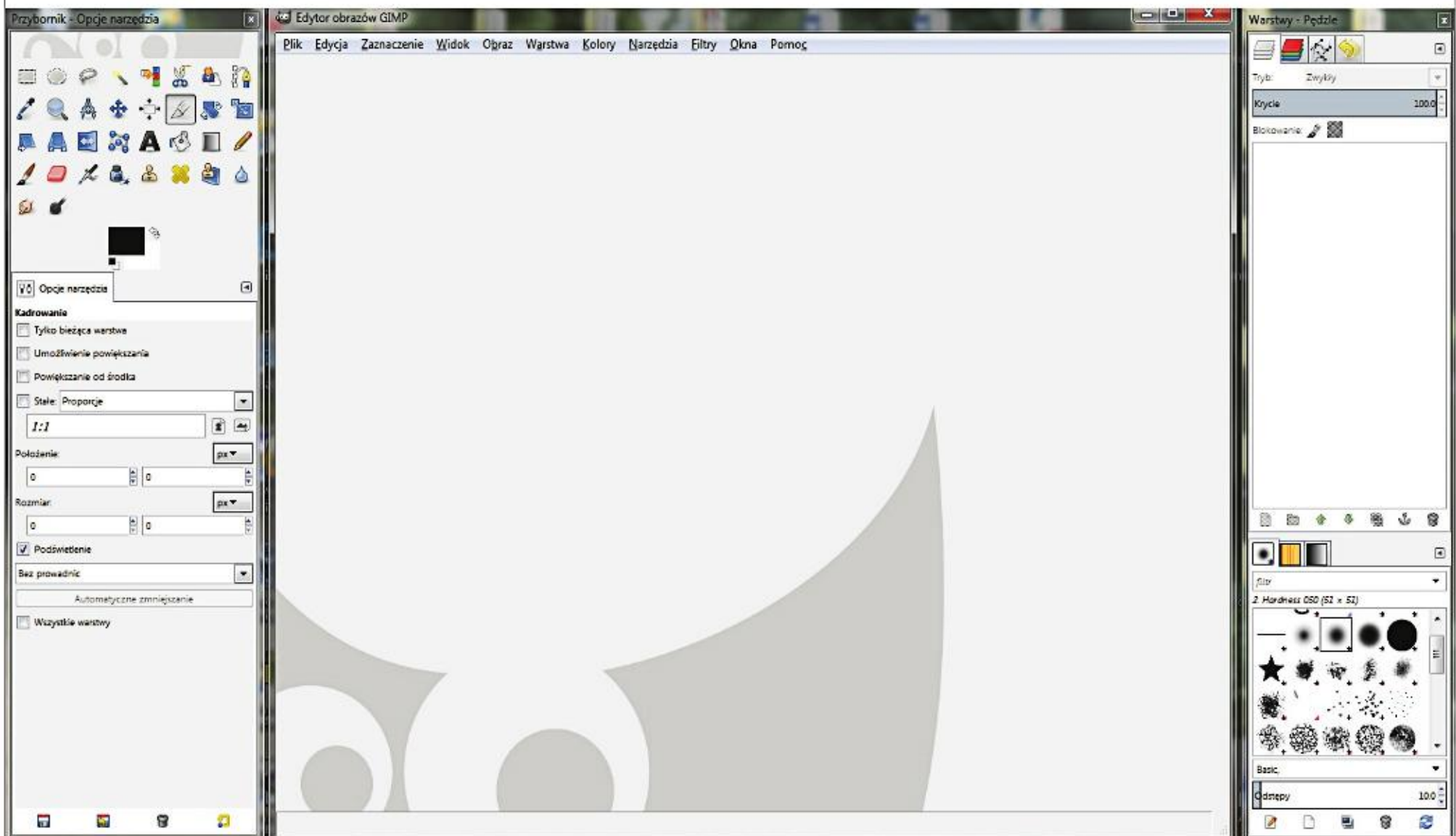
GIMP (ang. *GNU Image Manipulation Program*) to bezpłatny program do edycji grafiki rastrowej, który w ostatnich latach zdobył dużą popularność i stanowi poważną konkurencję dla Adobe Photoshop Elements – najtańszej wersji legendarnej aplikacji.

Technik informatyk lub technik programista podczas pracy nad witrynami i aplikacjami internetowymi edytują prostą grafikę lub przetwarzają już gotowe pliki (np. zdjęcia), które dopasowuje do założeń projektowych albo technicznych. Na tym etapie nie powinny być realizowane zadania przeznaczone dla wykwalifikowanego grafika komputerowego. Dlatego taki edytor graficzny jak GIMP idealnie nadaje się do tego typu zadań. Bardziej zaawansowane projekty graficzne lub artystyczne wizualizacje powinny być tworzone przez profesjonalne agencje plastyczne.

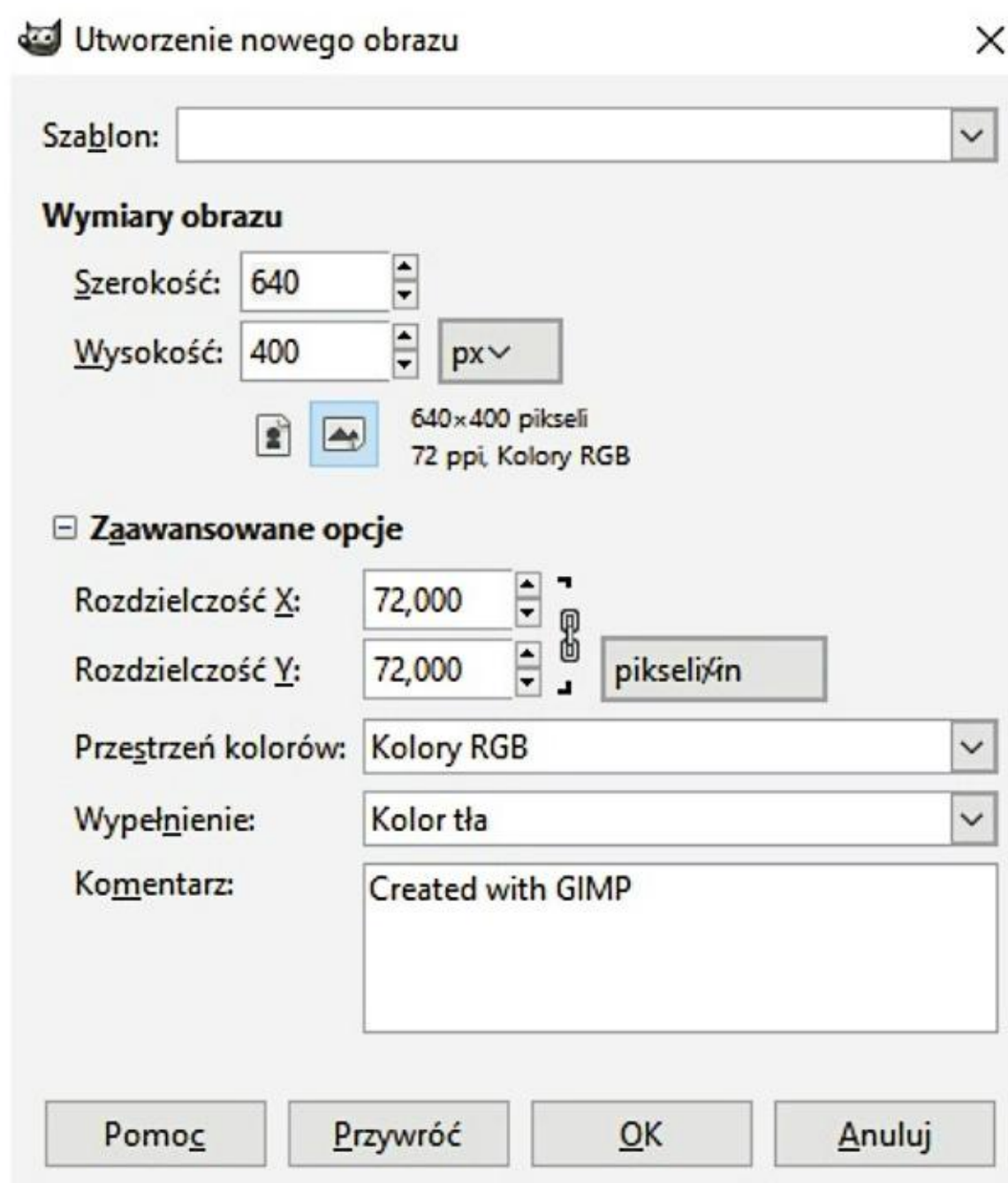
Program GIMP jest dostępny na systemy: Windows, Linux oraz Mac OS. Pliki instalacyjne można pobrać bezpośrednio ze strony projektu: <https://www.gimp.org>.

Instalacja w systemach z rodziny Windows jest intuicyjna i nie powinna przysporzyć problemów. Po pierwszym uruchomieniu mamy do dyspozycji główne okno programu oraz okna przyborników.

Nowy obraz można utworzyć przez zakładkę **Plik** umieszczoną w menu poleceń okna głównego. Należy określić jego wymiary. Domyślnie są ustawione wymiary 640 px na 400 px z rozdzielczością 72 ppi i modelem barw RGB. Po wyborze opcji zaawansowanych można wybrać rodzaj wypełnienia, dodać komentarz lub zmienić domyślne ustawienia.

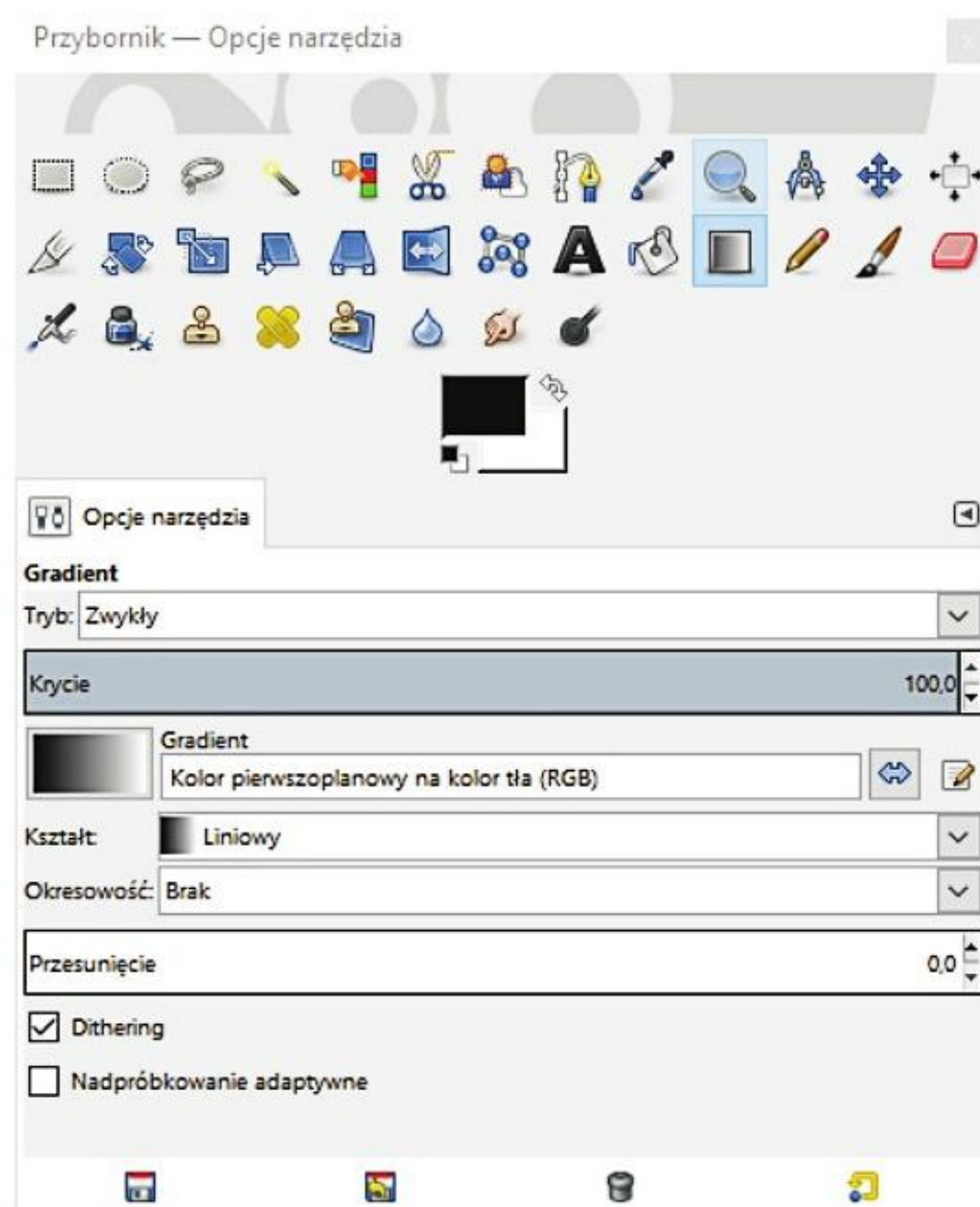


Rys. 20.1. Okno główne programu GIMP



Rys. 20.2. Okno tworzenia nowego obrazu w edytorze GIMP

Po utworzeniu nowego obrazu główne okno robocze zyskuje prowadnice – pionową oraz poziomą. W tym momencie jest gotowe do edycji. Edycję umożliwiają narzędzia umieszczone w przyborniku. Każde podwójne naciśnięcie lewego klawisza myszy na wybranym narzędziu w przyborniku skutkuje otwarciem okna z opcjami narzędzia.



Rys. 20.3. Okno przybornika w edytorze GIMP

Program GIMP oferuje wiele narzędzi rysowniczych, domyślnie zgrupowanych w przyborniku. Zalicza się do nich:

- ołówek (skrót N),
- pędzel (skrót P),
- aerograf (skrót A),
- stalówkę (skrót K),
- wypełnienie (skrót Shift + B),
- gradient (skrót L),
- gumkę (skrót Shift + E),
- klonowanie (skrót C),
- klon perspektywy, który klonuje przekształcony obraz,
- łątkę (skrót H),
- rozmywanie / wyostrażanie, które rozmywa lub wyostraża (skrót Shift + U),
- rozsmarowywanie (skrót S),
- rozjaśnianie / wypalenie, rozjaśniające lub przyciemniające element obrazu (skrót Shift + D).

Obok przybornika znajduje się okno **Warstwy – Pędzle**. Umieszczono tutaj zakładki warstw, kanałów, ścieżek oraz historii działań. Warstwy to jedno z najważniejszych narzędzi w edytorach graficznych. Można z nich korzystać w pracy zarówno z obrazem rastrowym, jak i wektorowym.

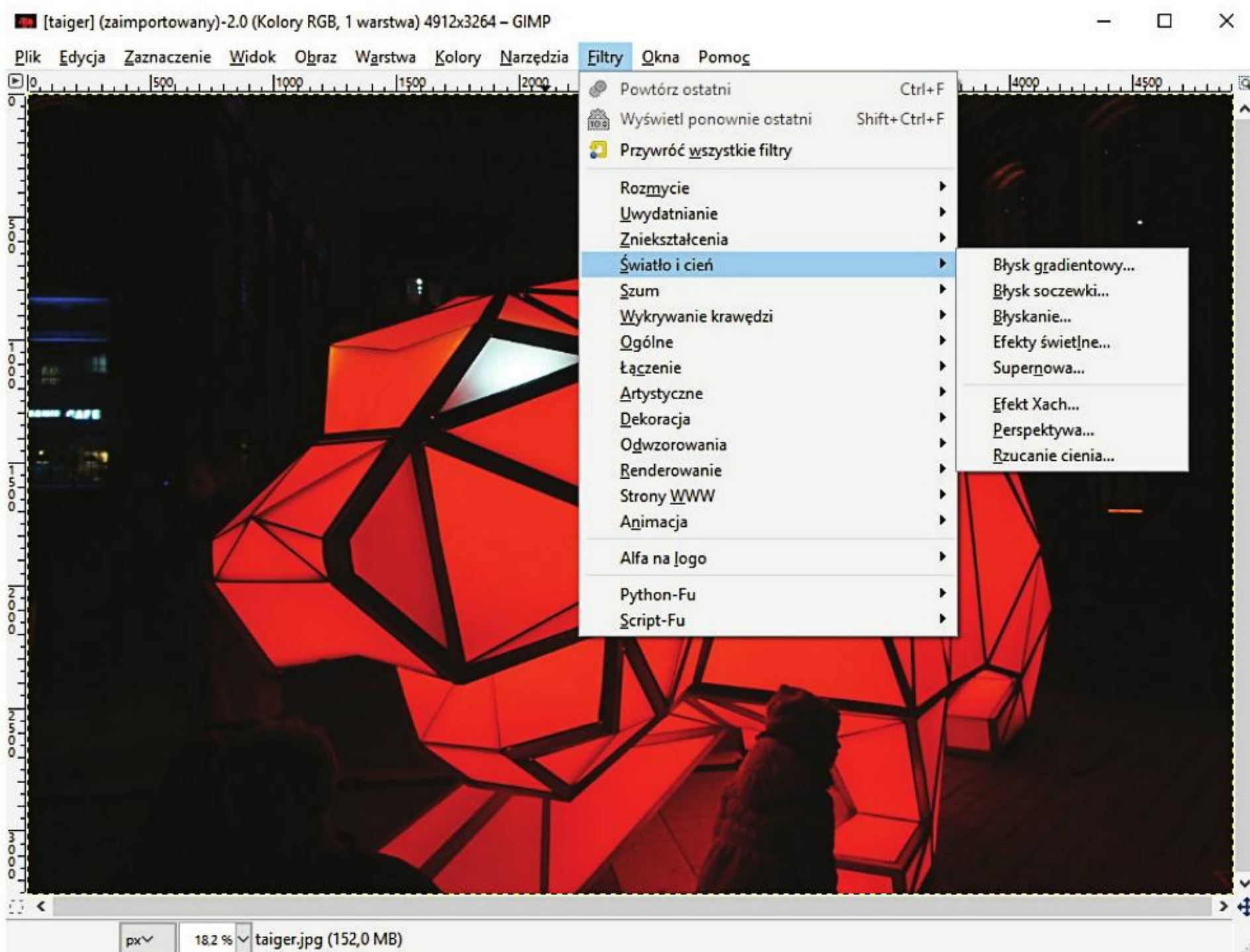
Warstwa to niezależny poziom obrazu, który ma unikalną nazwę i własności. Poszczególne warstwy leżą jedna na drugiej (tworzą stos). Wymiary każdej z warstw mogą być dowolne: warstwa jest większa lub mniejsza od głównego obrazu. Dzięki wykorzystaniu warstw można modyfikować poszczególne elementy stosu, przy czym nie naruszy się całego obrazu.



Rys. 20.4. Okno warstw

Filtry

Edytor GIMP ma grupę filtrów i efektów do zmiany parametrów obrazów oryginalnych. Na rys. 20.5 przedstawiono moment wyboru filtru z listy dostępnych w programie GIMP.



Rys. 20.5. Przykładowe filtry w programie GIMP

Import i eksport plików graficznych w GIMP

Oprócz tworzenia obrazów graficznych z rozszerzeniem xcf, możemy pracować na obrazach cyfrowych w różnych formatach. Wtedy należy skorzystać z menu **Otwórz** lub przeciągnąć plik graficzny na obszar roboczy. GIMP obsługuje większość najpopularniejszych formatów. Po wczytaniu plik otworzy się w oryginalnym rozmiarze. Okno z wczytanym plikiem również zmienia swój wygląd i pojawiają się na nim dodatkowe informacje:

- liczba warstw i rozmiar obrazu,
- jednostka miary,
- kursor zmiany położenia,
- skala powiększenia,
- informacje na pasku stanu.

Po wykonaniu edycji obrazów cyfrowych należy zapisać efekty pracy. Po wybraniu menu **Plik – Zapisz jako** możemy zapisać plik tylko w formacie xcf. Jeżeli chcemy zapisać edytowany plik graficzny w innym formacie, musimy wybrać opcję **Plik – Wyeksportuj jako...**

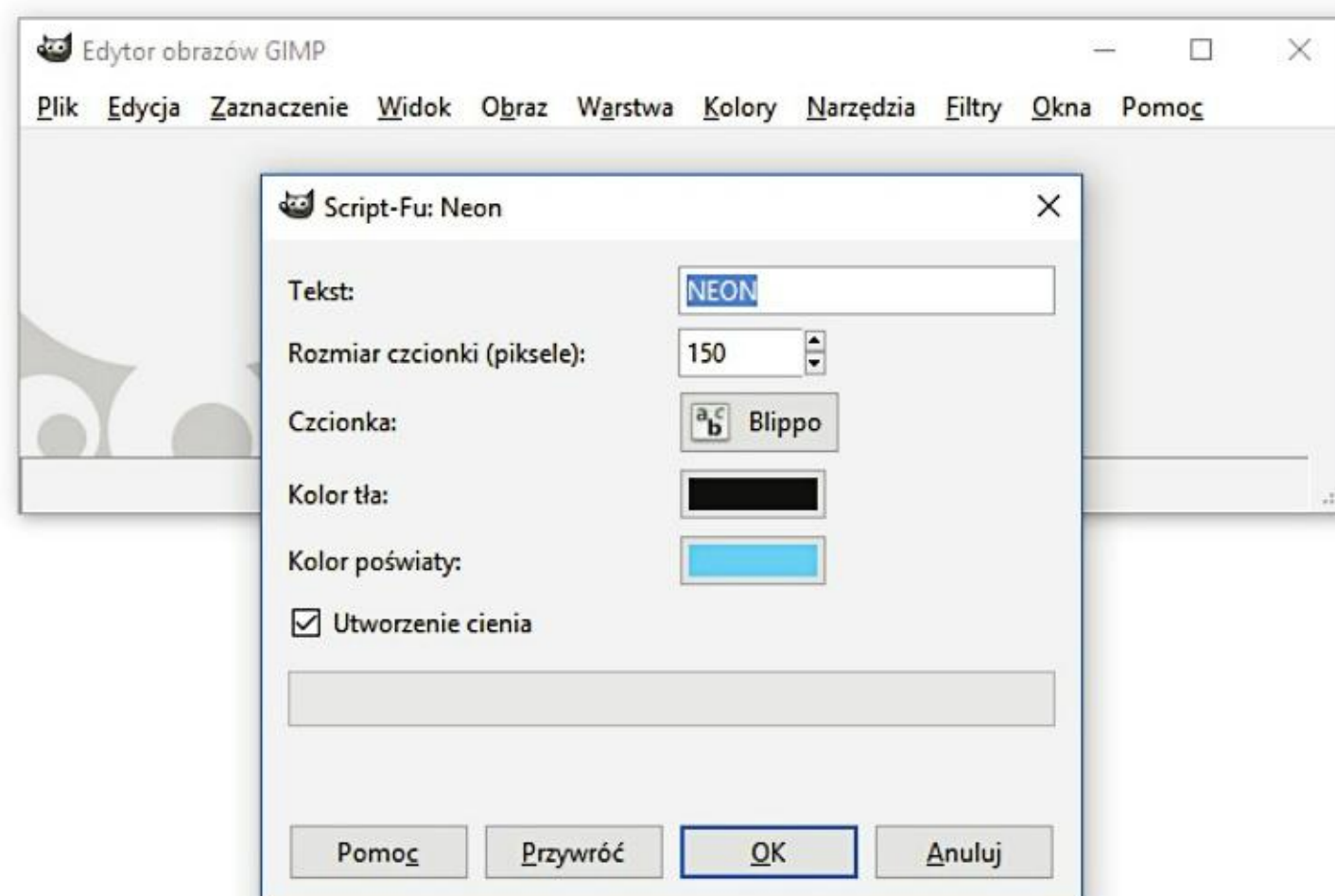


Rys. 20.6. Otwarty rysunek w programie GIMP oraz okno opcji Wyeksportuj jako

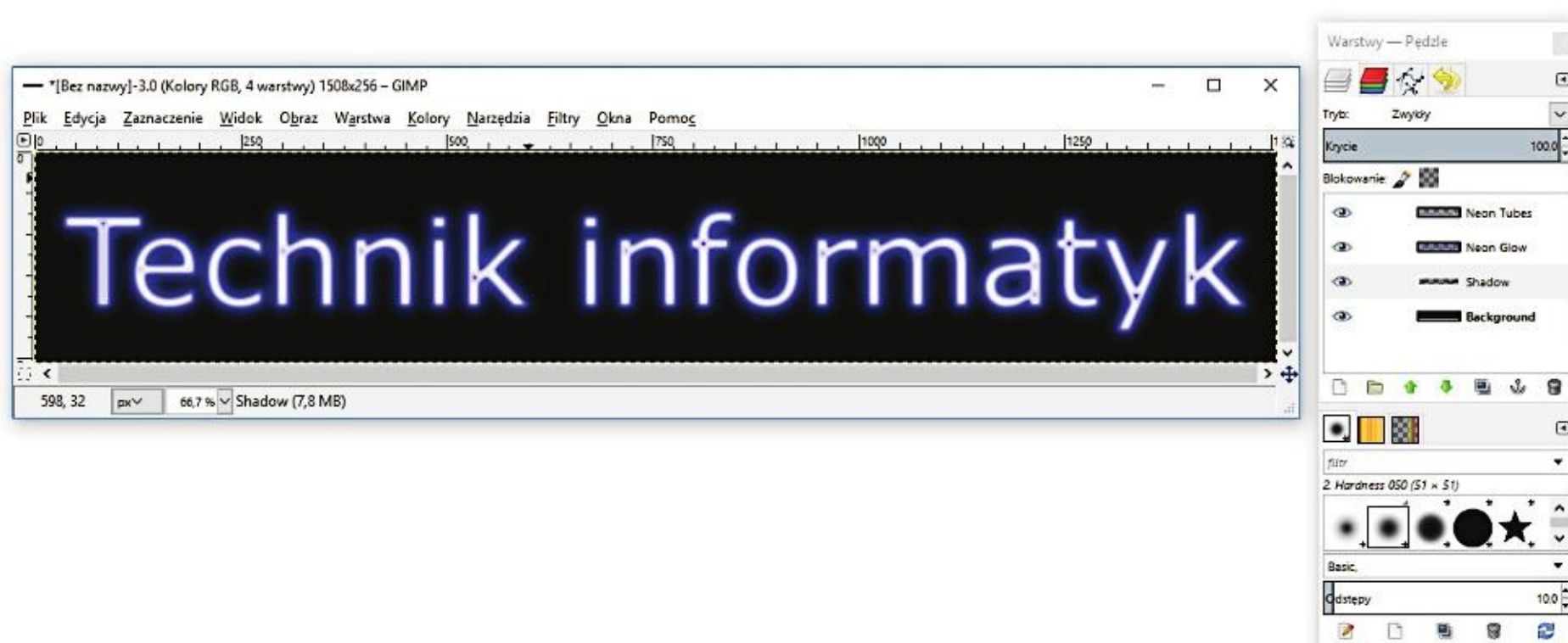
PRZYKŁAD 20.1

Za pomocą edytora GIMP utworzymy proste logo na potrzeby strony internetowej. Po uruchomieniu programu postępujemy według poniższych kroków.

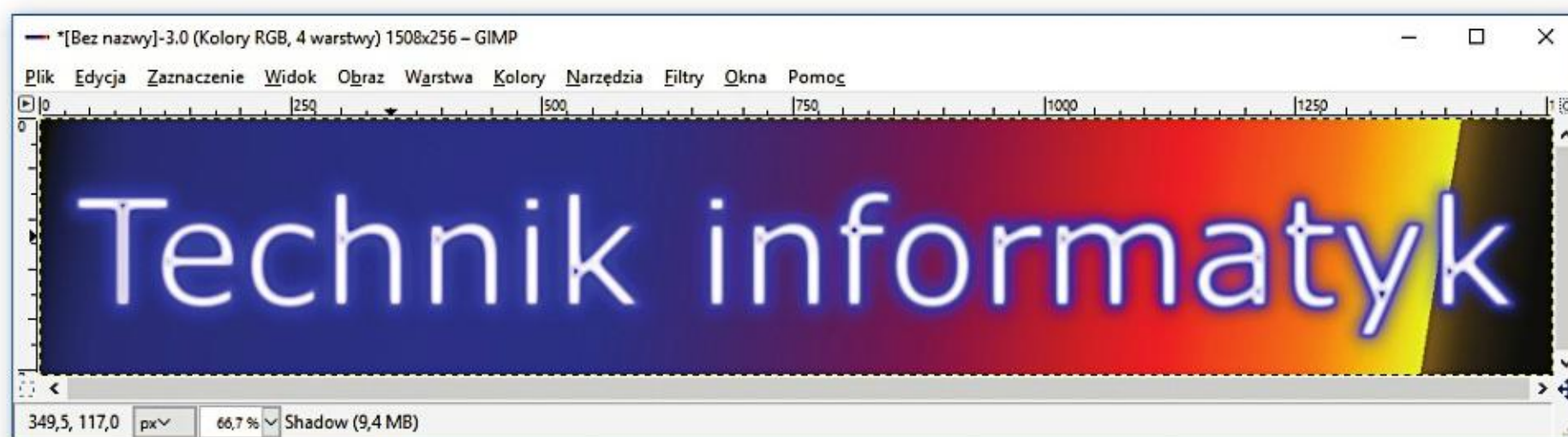
Krok 1. Z menu Plik wybieramy opcję Utwórz – Logo. Następnie z dostępnej listy wybieramy typ logo – Neon. W oknie dialogowym wybieramy kolor tła, kolor poświaty, tekst oraz rozmiar czcionki.



Krok 2. Po zatwierdzeniu zmian program wygeneruje gotowe logo. W oknie warstw widzimy warstwy, które posłużyły do utworzenia logo.



Krok 3. Zaznaczamy warstwę Background. Z przybornika wybieramy narzędzie Gradient. Z listy dostępnych rodzajów wskazujemy wzór Skyline oraz kształt liniowy. Cursor gradientu umieszczamy w górnym lewym rogu okna logo. Naciskamy lewy klawisz myszy i przeciągamy cursor w dolny prawy róg okna. Gotowy efekt widzimy na rysunku.

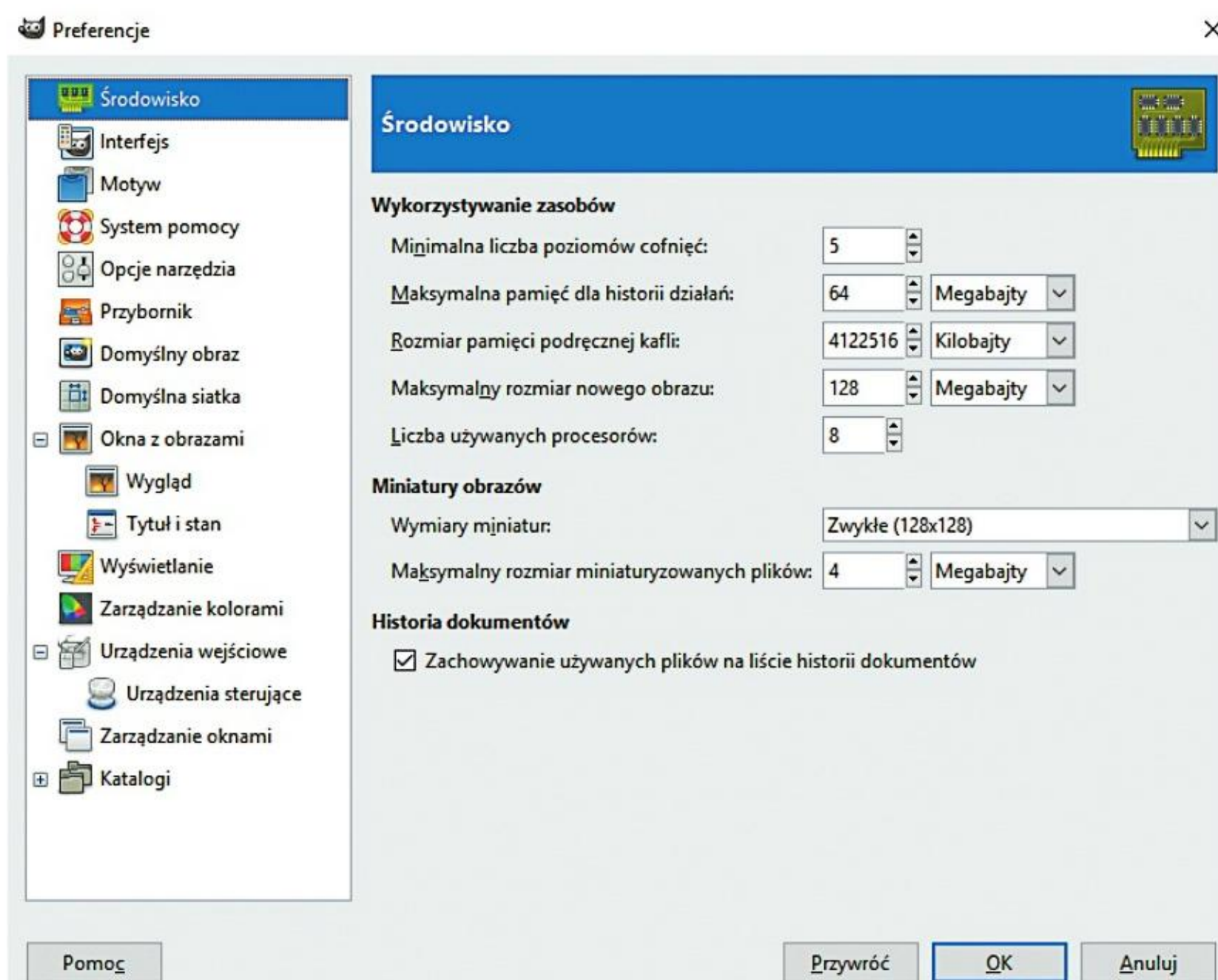


Krok 4. Po zakończonej edycji należy połączyć warstwy oraz wyeksportować obraz do formatu wykorzystywanego na stronach internetowych. Aby połączyć warstwy, korzystamy z opcji Obraz – Spłaszcz obraz. Wszystkie warstwy połączą się wtedy w jedną i obraz będzie gotowy do zapisu w dowolnym z dostępnych formatów. Wybieramy opcję **Plik >> Wyeksportuj jako...**, następnie podajemy nazwę i rozszerzenie oraz miejsce docelowe na dysku. Gotowe logo możemy osadzać w dowolnym dokumencie hipertekstowym.

Podobny efekt można osiągnąć dzięki tworzeniu poszczególnych warstw ręcznie i dzięki korzystaniu z narzędzi rysowniczych do edytowania poszczególnych elementów obrazu.

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Pobierz i zainstaluj program GIMP. Po przeczytaniu dokumentacji programu zapoznaj się – z pomocą nauczyciela – z jego interfejsem i możliwościami. Wybierz menu Edycja – Preferencje i postaraj się dopasować opcje programu oraz narzędzi do zadań realizowanych na zajęciach (np. domyślny rozmiar obrazu).



2. Skorzystaj z opcji Plik – Utwórz – Przyciski i zaprojektuj przycisk menu, który będzie można wykorzystać do wysłania e-maila na dowolnej stronie internetowej. Podczas eksportu najlepiej wybierz format png z przezroczystym tłem. Efekt pracy zapisz pod nazwą przycisk.
3. Zimportuj dowolne zdjęcie do programu GIMP. Zastosuj cztery wybrane filtry do jego zmiany. Efekt zmian skonsultuj z nauczycielem. Gotowy obraz wyeksportuj do formatów: png, bmp oraz jpg. Opisz różnice w parametrach tych plików.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Co oznacza akronim GIMP?
2. Jakie edytory grafiki rastrowej są alternatywą dla programu GIMP?
3. Jaki format plików graficznych jest dedykowanym formatem programu GIMP?
4. Jaki domyślny rozmiar pliku ma każdy nowy obraz w programie GIMP?

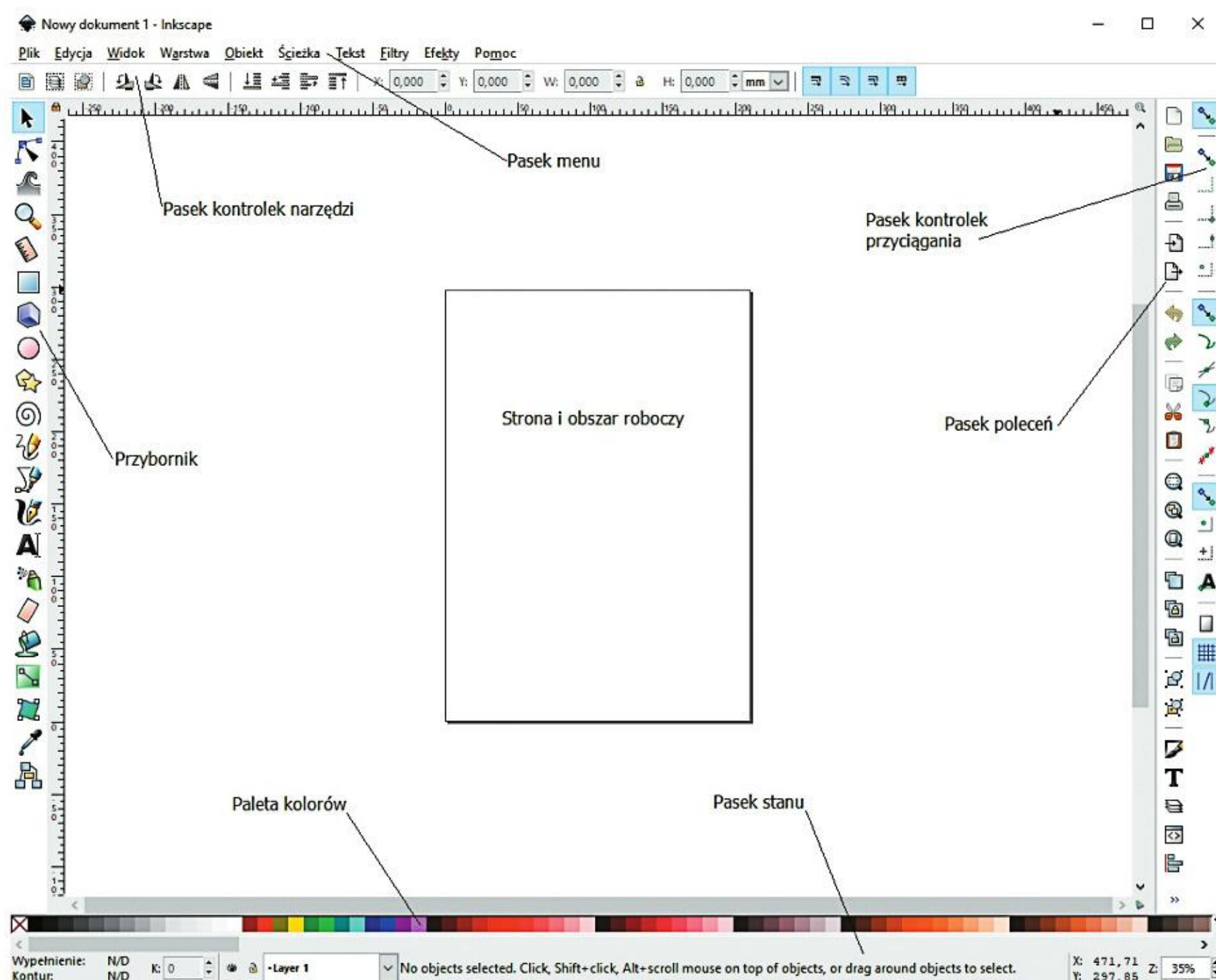
21

Edytor grafiki wektorowej Inkscape

ZAGADNIENIA

- Interfejs obsługi programu Inkscape
- Zasady tworzenia figur geometrycznych
- Edycja ścieżek
- Dobór i edycja tekstu

Do najbardziej popularnych darmowych edytorów grafiki wektorowej należy Inkscape. Korzysta on z otwartego standardu W3C SVG (ang. *Scalable Vector Graphics*), jako swojego rodzimego formatu. Jest darmowym oprogramowaniem typu open source. Inkscape tworzy grupa autorów zaangażowanych w projekt. Pliki instalacyjne na różne platformy sprzętowe są dostępne do pobrania na stronie <https://inkscape.org>.



Rys. 21.1. Okno główne programu Inkscape

Obrazy utworzone w programie Inkscape są zapisywane w formacie SVG, który jest obsługiwany przez wszystkie główne przeglądarki internetowe (bez dodatkowych wtyczek) oraz wspomagany przez język HTML 5. Warto przypomnieć, że obrazy SVG można tworzyć bezpośrednio z poziomu skryptu osadzonego w kodzie HTML. Dodatkowo Inkscape umożliwia eksportowanie obrazów do formatu PNG, obsługującego przezroczystość, oraz PDF, co może być istotnym argumentem do stosowania właśnie tej darmowej aplikacji.

Pionowy pasek znajdujący się po lewej stronie głównego okna to przybornik. Zawiera on narzędzia do rysowania i edycji obiektów. W górnej części okna, poniżej menu głównego, znajduje się pasek poleceń, na którym umieszczono przyciski realizujące główne polecenia programu. Poniżej tego paska znajduje się pasek kontrolek narzędziowych z elementami kontrolnymi przypisanymi do aktualnie wybranego narzędzia. Na pasku stanu, położonym na dole okna, są wyświetlone informacje dotyczące tworzonego obrazu.

Wszystkie narzędzia w edytorze grafiki wektorowej nazywa się narzędziami kształtu. Na każdym utworzonym lub zaznaczonym kształcie są widoczne **uchwyty** w postaci małych kwadratów. Po wybraniu kursorem myszy dowolnego uchwytu można edytować kształt.

W obszarze roboczym można zaznaczać jeden wybrany obiekt lub grupę obiektów. Poza tym cechy obrazu wektorowego pozwalają na łączenie różnych obiektów w grupę. Grupa podczas przemieszczania bądź przekształcania zachowuje się tak jak pojedynczy obiekt. Połączone obiekty można w dowolny sposób rozdzielać, edytować i ponownie łączyć. W programie Inkscape są dostępne zaawansowane działania na obszarach ścieżek (tzw. kształtowanie): suma (spawanie), różnica (przycięcie), część wspólna. Tak jak w aplikacjach do edycji grafiki rastrowej, obiekty można wypełniać kolorem lub edytować barwę konturu.

W edycji obrazów wektorowych korzysta się również z warstw. Pozwala to oddzielić poszczególne etapy pracy. Przydatnymi funkcjami są także powielanie obiektów oraz zmiana kolejności ich położenia. Istotną cechą aplikacji jest możliwość importowania obrazów rastrowych. Program Inkscape to potężne narzędzie do edycji grafiki wektorowej. W następnych rozdziałach opiszemy jego podstawowe możliwości.

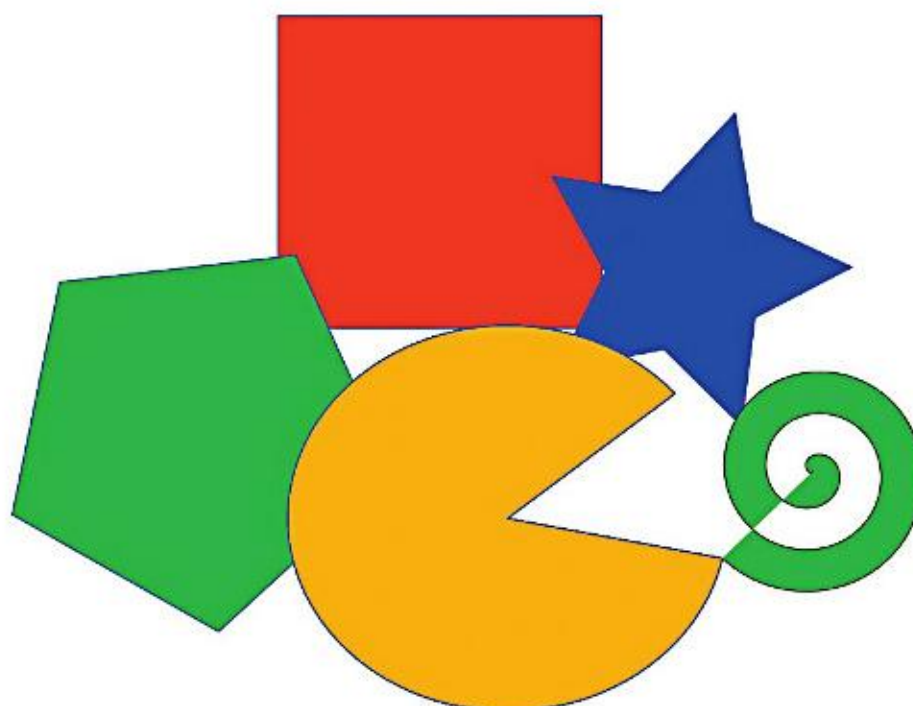
Rysowanie figur w programie Inkscape

Do rysowania figur geometrycznych wykorzystuje się przybornik. Znajdują się w nim ikony umożliwiające rysowanie prostych i złożonych figur 2D i 3D. Oprócz wskazania określonej figury kursorem myszki, edytor oferuje możliwość wyboru za pomocą klawiszy funkcyjnych, co może znacznie usprawnić pracę. W tabeli 21.1 przedstawiono dostępne opcje przeznaczone do rysowania figur.

Tabela 21.1. Narzędzia do tworzenia figur geometrycznych

prostokąty i kwadraty	[F4]	
obiekty 3D	[Shift] + [F4]	
okręgi, elipsy i łuki	[F5]	
gwiazdy i wielokąty	[*]	
spirale	[F9]	

Dodatkowo program umożliwia rysowanie linii prostych – [Shift] + [F6] – i ręczne kreślenie linii krzywych – [F6].

**Rys. 21.2.** Przykładowe figury rysowane w programie Inkscape

Najczęściej używanym narzędziem jest **Wskaźnik**. Jest to najwyżej położony przycisk na przyborniku. Wskaźnik można również wybrać, naciskając klawisz [F1] lub klawisz [Space]. Po wyborze wskaźnika wokół obiektu ukaże się osiem uchwytów w kształcie strzałek. Za ich pomocą można:

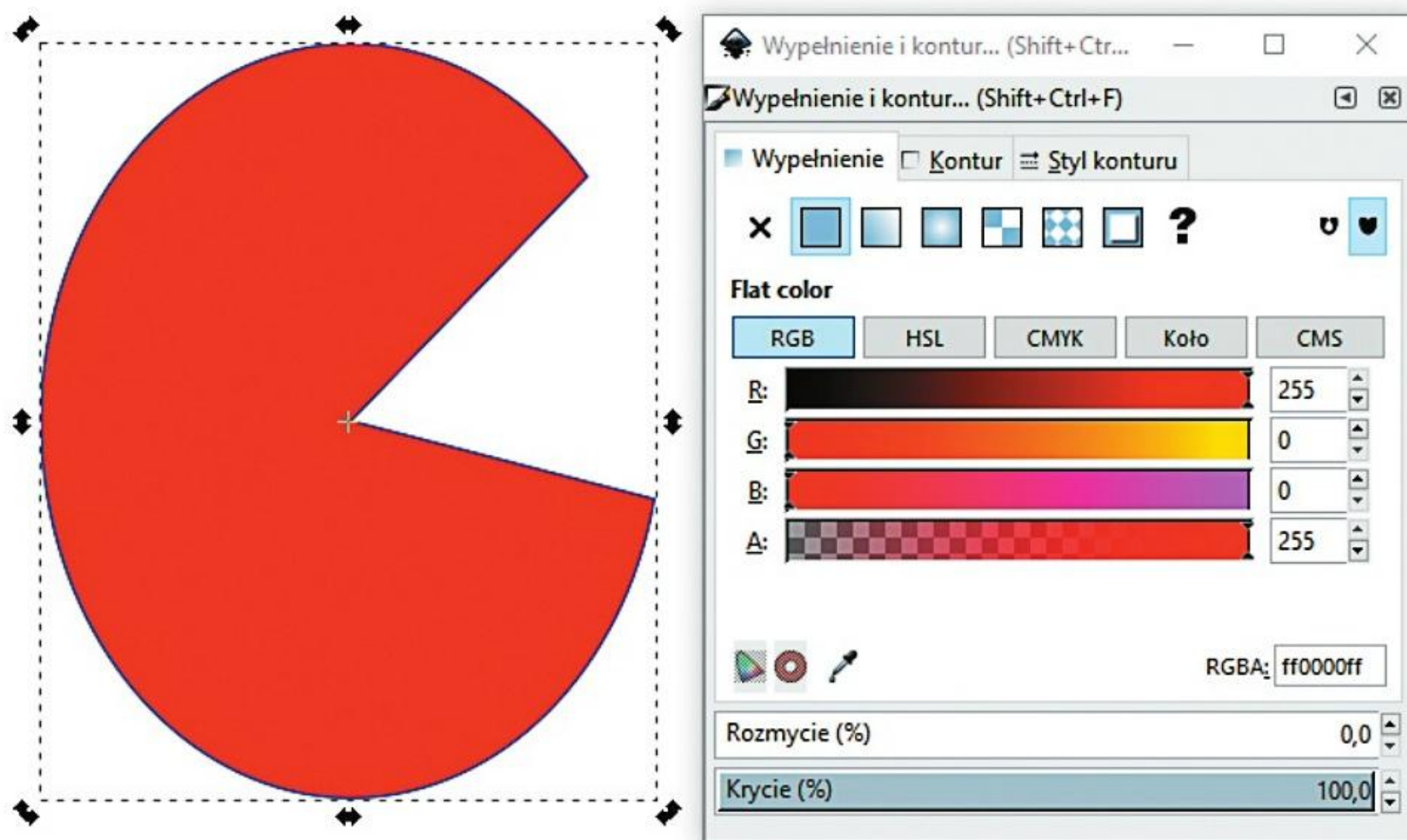
- przesuwając obiekt przez przeciąganie go;
- skalować obiekt przez przesuwanie jednego z uchwytów.

Po ponownym kliknięciu na zaznaczonym obiekcie uchwyty zmieniają wygląd i umożliwiają:

- obrócenie obiektu przez przesuwanie uchwytów narożnych;
- pochylenie obiektu za pomocą przeciągania nienarożnych uchwytów.

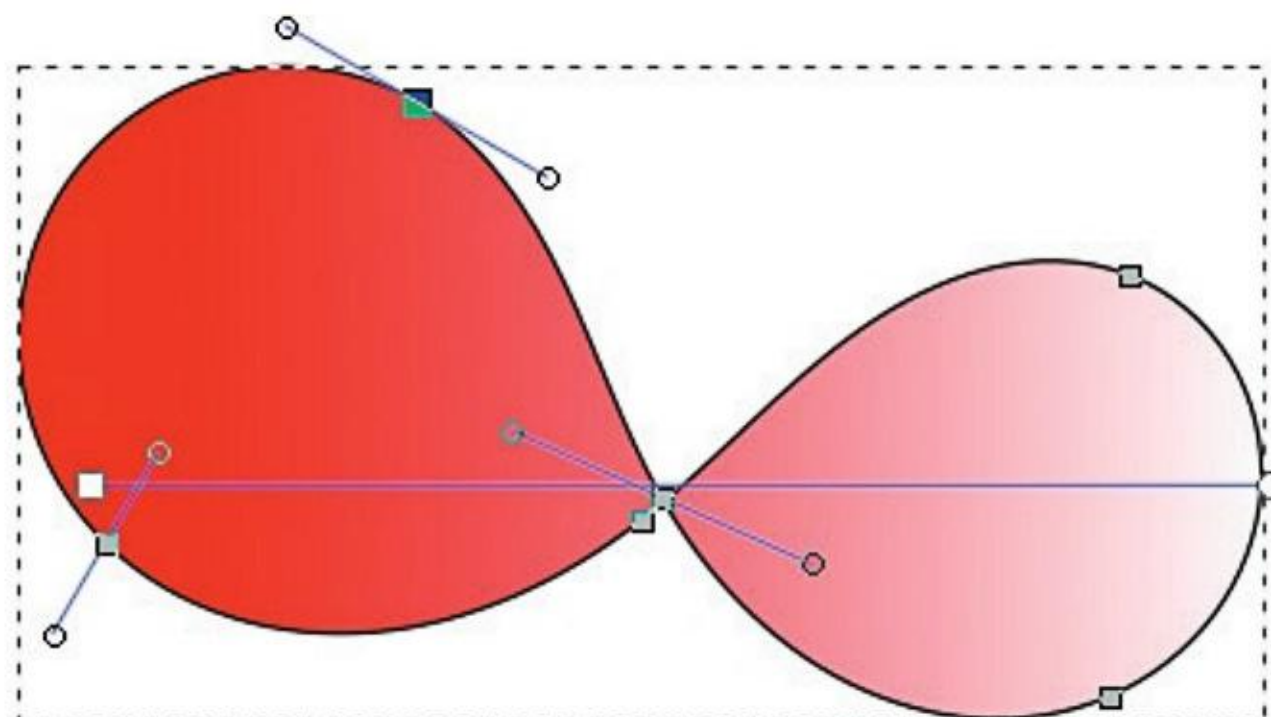
Po wyborze wskaźnika można używać numerycznych pól wprowadzania na **Pasku kontrolek narzędzi**, żeby ustawić dokładne wartości współrzędnych (X i Y) i wymiarów (S i W) selekcji.

Zmiana koloru wypełnienia, konturu, stylu konturu oraz innych właściwości jest dostępna po zaznaczeniu obiektu wskaźnikiem oraz uruchomieniu menu **Wypełnienie i kontur** ([Shift] + [Ctrl] + [F]). Kolor można wybrać w jednym z dostępnych modeli barw lub wskazać na palecie kolorów umieszczonej w dolnej części obszaru roboczego. Dodatkowo, oprócz jednolitego koloru, można zastosować gradient liniowy, gradient radialny, gradient mesh oraz desień.



Rys. 21.3. Wypełnienie i kontur

Największą zaletą grafiki wektorowej jest możliwość tworzenia dowolnego kształtu lub zmiany kształtu narysowanego obiektu: wszystko jedno, czy jest nim figura geometryczna, czy tekst. Do edycji własnych form graficznych służą **ścieżki** oraz **krzywe Béziera**. Edycja kształtu krzywej jest możliwa dzięki prowadnicom i węzłom.

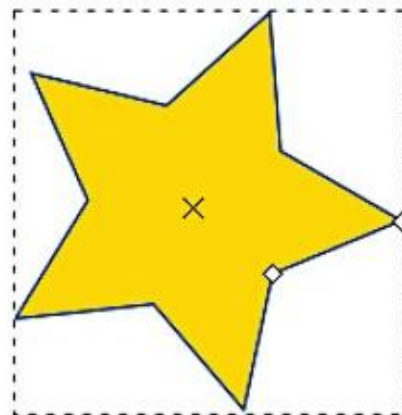


Rys. 21.4. Edycja obiektów za pomocą ścieżek

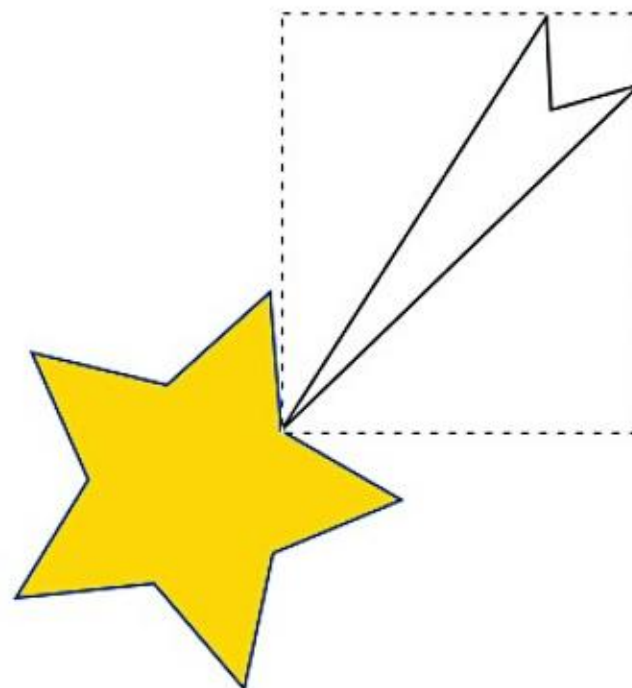
PRZYKŁAD 21.1

Spróbujemy narysować kometę. Wykorzystamy opisane powyżej możliwości. Tworzenie obrazu wykonujemy w poniższych krokach.

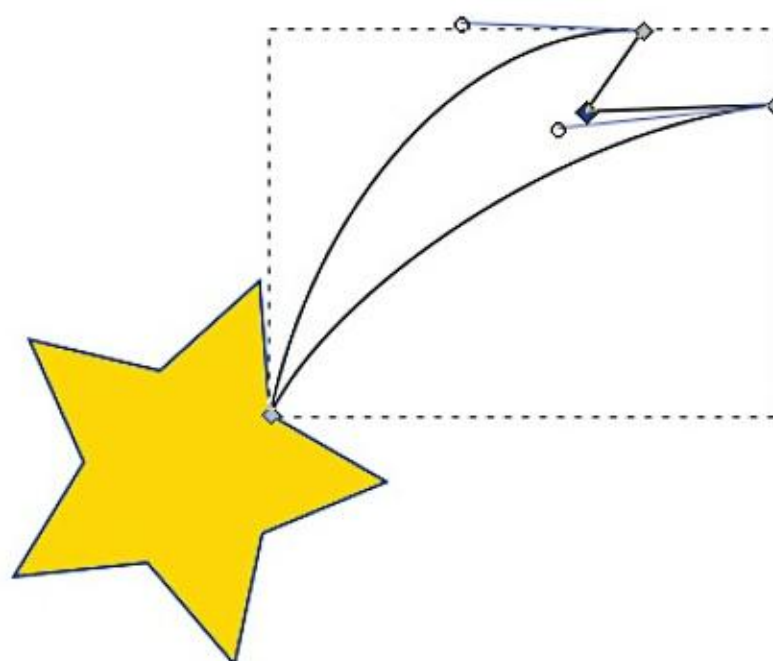
Krok 1. Wybieramy narzędzie do rysowania gwiazd i wielokątów. Rysujemy gwiazdę na stronie w obszarze roboczym. Zaznaczamy ją wskaźnikiem oraz wybieramy kolor żółty jako wypełnienie.



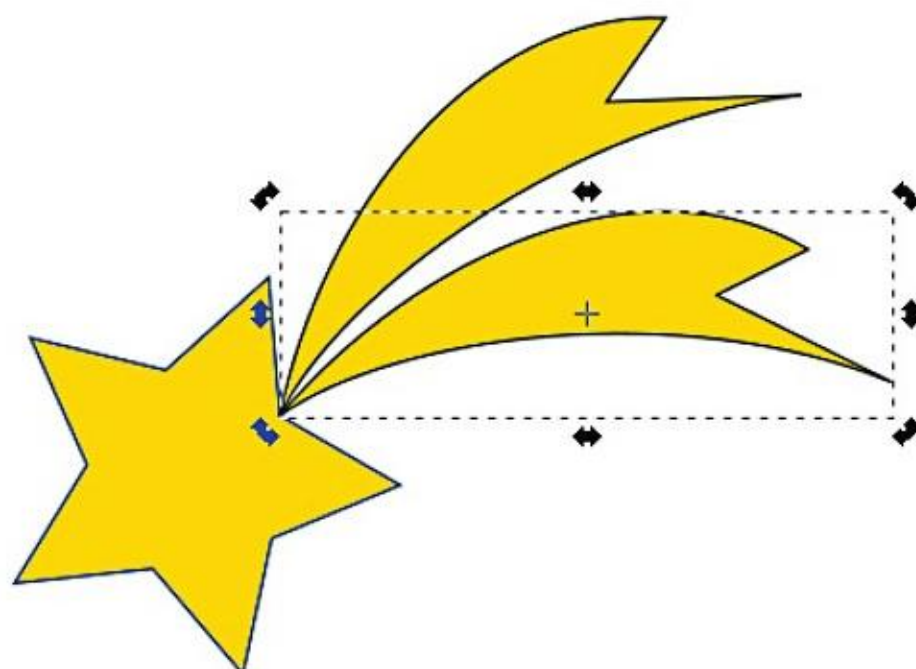
Krok 2. Za pomocą narzędzia do rysowania krzywych Béziera i linii prostych rysujemy szablon warkocza komety.



Krok 3. Następnym etapem będzie modelowanie warkocza za pomocą narzędzia do edycji węzłów. Mamy wówczas możliwość nadania właściwego kształtu i położenia warkocza.

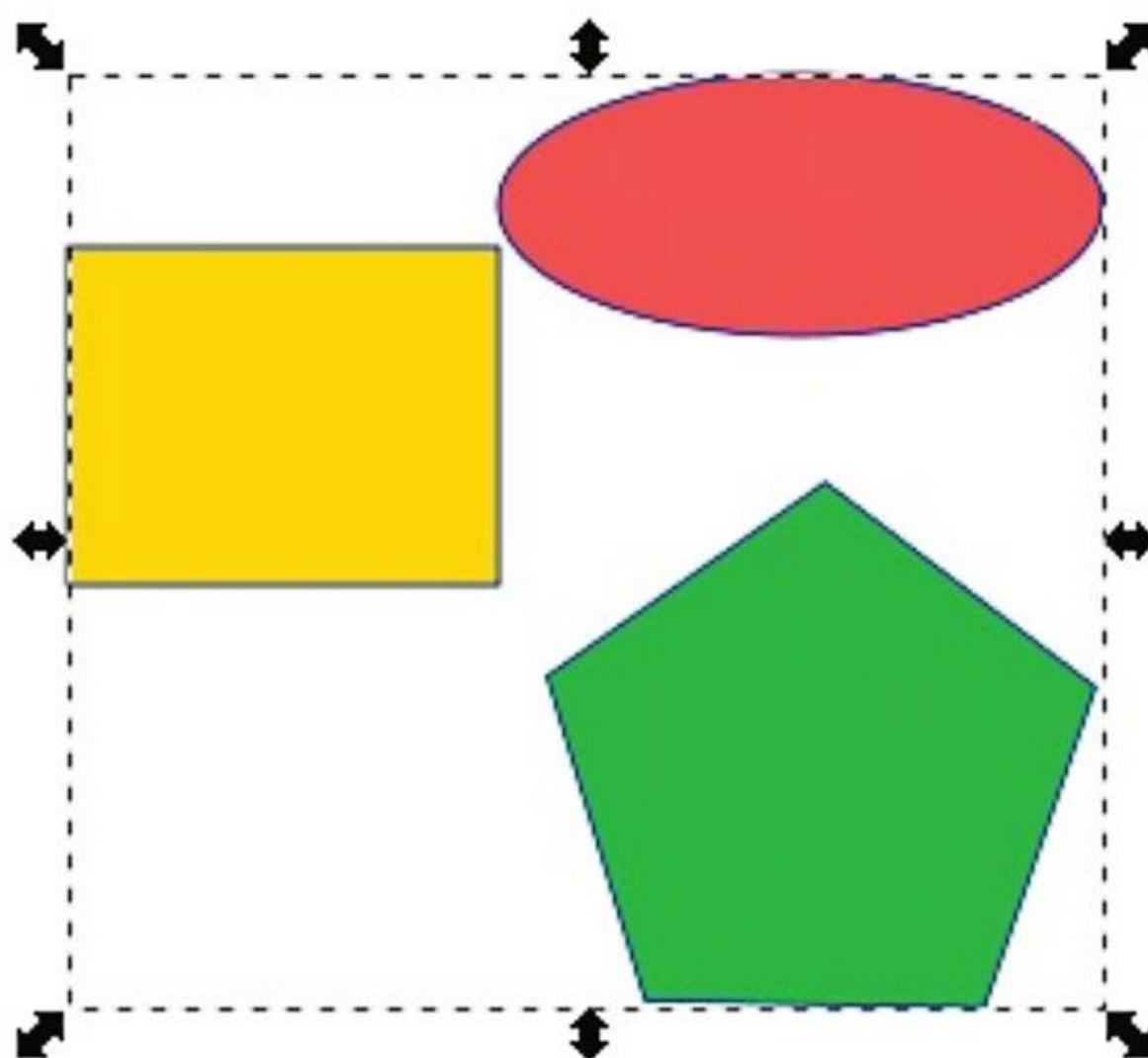


Krok 4. Po operacji modelowania warkocza zaznaczamy obiekt wskaźnikiem i wybieramy kolor wypełnienia, taki sam jak gwiazdy. Następnie, za pomocą menu Edycja – Powiel, wykonujemy ustaloną liczbę kopii warkocza. Po operacji powielenia, korzystając z opcji wskaźnika, obracamy i przesuwamy kopie w odpowiednie miejsce, aby uzyskać końcowy efekt (widoczny na rysunku).



Grupowanie obiektów

W edytorze Inkscape możemy wykonywać modyfikacje oraz inne działania na obiektach łączonych w grupy. Grupa zachowuje się wtedy jak pojedynczy obiekt. Żeby utworzyć grupę, należy wybrać jeden albo więcej obiektów i nacisnąć [Ctrl] + [G]. Aby rozgrupować jedną lub więcej grup, trzeba wcisnąć [Shift] + [Ctrl] + [G].



Rys. 21.5. Grupowanie obiektów

Pozostałe operacje

Jedną z przydatnych operacji jest **powielanie** obiektu ([Ctrl] + [D]). Kopia obiektu jest umieszczana dokładnie powyżej oryginału i zaznaczana. Podobną funkcjonalnością charakteryzuje się narzędzie do **klonowania** ([Alt] + [D]). Opcja klonuj udostępnia jednak więcej możliwości, np. klonowanie wielokrotne tworzące układ klonów według ustawień użytkownika.

Kolejną przydatną funkcją jest możliwość układania obiektów w stos. Pozwala to na ułożenie obiektów jeden na drugim według wymagań tworzonego projektu. Aby przenieść dany obiekt, należy go zaznaczyć oraz z menu **Obiekt** wybrać jedną z dostępnych opcji:

- przenieś w górę,
- przenieś w dół,
- przenieś na wierzch,
- przenieś na spód.

Polecenia w menu **Ścieżka** pozwalają m.in. na wykonywanie operacji logicznych na obiektach. Jeżeli dwa obiekty zachodzą na siebie w stosie, możemy po zaznaczeniu ich wskaźnikiem wykonać następujące operacje:

- suma,
- różnica,
- część wspólna,
- wykluczenie,
- podział,
- rozcięcie ścieżki.



Rys. 21.6. Przykład podstawowych operacji logicznych na obiektach

Tekst

Za pomocą programu Inkscape można tworzyć proste i złożone obiekty tekstowe, takie jak nagłówki, banery, logotypy.

Aby dodać do projektu obiekt tekstowy, wystarczy uruchomić narzędzie **Tworzenie i modyfikowanie obiektów tekstowych** ([F8]) i wybrać miejsce na obszarze roboczym do edycji. Aby zmienić rodzinę czcionek, styl, rozmiar i wyrównanie, należy otworzyć okno dialogowe **Tekst i czcionka**. W tym oknie znajduje się zakładka **Tekst**, w której można edytować zaznaczony obiekt tekstowy. Bywa to wygodniejsze niż edycja bezpośrednio na obszarze roboczym, chociażby dlatego, że w tej karcie działa sprawdzanie pisowni. Tekst, jak każdy obiekt w Inkscape, można modyfikować. Oprócz zastosowania efektów dostępnych w programie, można przekształcić tekst w ścieżkę ([Shift] + [Ctrl] + [C]) i modyfikować każdą literę indywidualnie. Przykład takiej operacji przedstawiono na rys. 21.7.



Rys. 21.7. Przykład modyfikacji tekstu (wyżej tekst wyjściowy, niżej – zmodyfikowany)

Autorzy stron internetowych wykorzystują edytor grafiki wektorowej do tworzenia banerów, przycisków, ikon lub innych elementów graficznej konstrukcji strony.

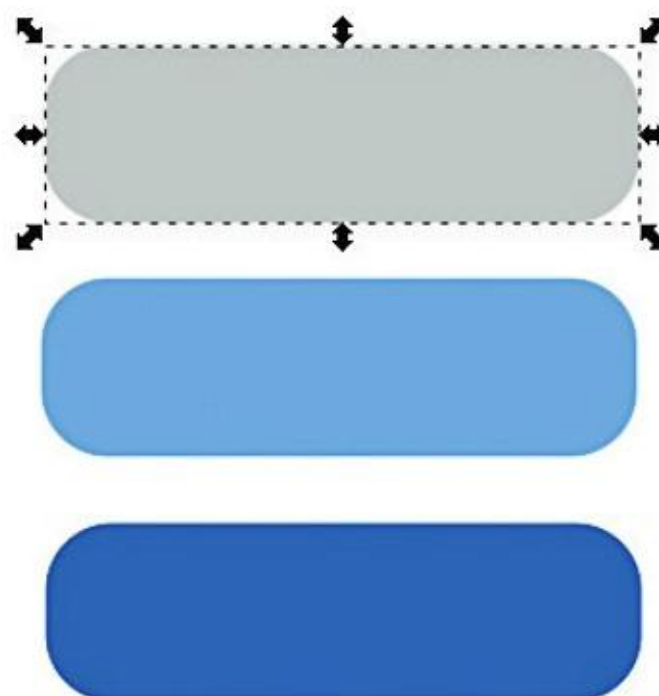
PRZYKŁAD 21.2

Najczęściej przygotowywanym elementem na potrzeby tworzenia stron i aplikacji internetowych są przyciski. Naszym zadaniem będzie wykonanie przycisku o nazwie OBLICZ za pomocą edytora Inkscape. Efekt naszej pracy musimy zapisać w pliku z rozszerzeniem png. Tworzenie przycisku wykonujemy w pięciu krokach.

Krok 1. Uruchamiamy program Inkscape. Rysujemy prostokąt z jasnoszarym kolorem wypełnienia i krawędzi (RGBA:ccccccff) oraz delikatnie zaokrąglonymi narożnikami krawędzi.



Krok 2. Wykonujemy dwie kopie naszego prostokąta. W jednym z nich zmieniamy kolor wypełnienia i krawędzi na jasny odcień niebieskiego (RGBA:80b3ffff), w drugim – na niebieski (RGBA: 0066ffff).



Krok 3. Następnie przeciągamy niebieski prostokąt na szary, który będzie imitował cień przycisku. Nie zawsze uda nam się umieścić idealnie jeden prostokąt na drugim, dlatego musimy skorygować położenie właściwościami X i Y na górnym pasku kontrolek narzędzi. W ten sam sposób postępujemy z prostokątem jasnoniebieskim. Efekt jest widoczny na rysunku poniżej.



W ten sposób mamy ustalony kształt przycisku, jego kolor oraz efekt grubości i cieniowania.

Krok 4. Za pomocą narzędzia do wstawiania tekstu edytujemy tekst OBLICZ. Dobieramy czcionkę Rockwell Extra Bold, wielkość oraz kolor.



Krok 5. Przycisk jest praktycznie skończony, ale na stronie potrzebujemy mieć najczęściej trzy wersje przycisku: przed wybraniem, aktywny (po najechaniu kursorem myszki) oraz wciśnięty. Dlatego musimy zrobić dwie kopie naszego przycisku i lekko je zmodyfikować. W aktywnym zmienimy tylko kolor wypełnienia pierwszego prostokąta na jaśniejszy odcień niebieskiego. W przycisku wciśniętym zmniejszymy odstęp między pierwszym a drugim przyciskiem, aby stworzyć efekt wciśnięcia. Końcowy efekt pracy jest widoczny na rysunku poniżej. Na koniec eksportujemy każdy przycisk do formatu png.

**OBLICZ****OBLICZ****OBLICZ**

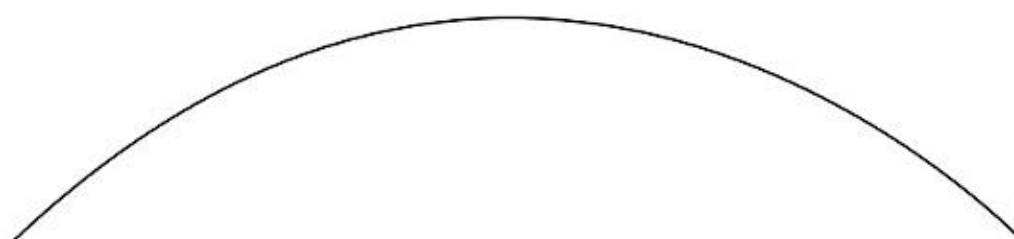
Bardzo przydatną funkcją w tworzeniu obrazów wektorowych jest **opisywanie tekstu na krzywej**. Ta operacja pozwala na wykonanie ciekawych logotypów, ikon oraz innych elementów strony.

PRZYKŁAD 21.3

Aby ukształtować tekst względem krzywej, należy postępować według poniższych kroków.

Krok 1. Rysujemy linię (np. za pomocą pióra) i odpowiednio ją modelujemy (edytujemy węzły). Powyżej edytujemy tekst i jednocześnie nadajemy mu odpowiednie styl i kolor.

TECHNIK INFORMATYK



Krok 2. Kolejnym etapem pracy będzie zaznaczenie napisu oraz ścieżki i wybranie **Tekst – Wstaw na ścieżkę**. Następnie możemy usunąć kolor konturu w oknie **Wypełnienie i kontur**. Efekt końcowy przedstawiono na rysunku.

TECHNIK INFORMATYK

SPRAWDŹ SVOJE UMIEJĘTNOŚCI

1. Skorzystaj z narzędzi oraz funkcji edytora Inkscape i narysuj flagę Unii Europejskiej.
2. Wykonaj projekt ikony (Plik – New from Template – Icon) na stronę internetową XVI Liceum Ogólnokształcącego. Projekt ma być zgodny ze wzorem przedstawionym na rysunku. Zapisz projekt jako ikona.png.



3. Wykonaj w programie Inkscape dwa przyciski nawigacyjne z narysowanymi strzałkami w lewo i w prawo. Każdy przycisk powinien mieć trzy wersje (przed wybraniem, aktywny oraz wciśnięty). Każdy przycisk wykonaj w takim samym zestawie barw. Gotowe przyciski zapisz pod nazwami prawo.png i lewo.png.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Jakie zastosowanie mogą mieć na stronach internetowych obrazy wykonane w programie Inkscape?
2. Wyjaśnij, co umożliwia przekształcenie obiektu w ścieżkę.
3. Jakiego skrótu należy użyć, aby wybrać narzędzie do rysowania gwiazd i wielokątów?

22

Edycja gotowych obrazów cyfrowych

ZAGADNIENIA

- Poprawianie jakości obrazów cyfrowych
- Zasady obróbki obrazów
- Wykonywanie zrzutów z ekranu

Oprócz tworzenia obrazów cyfrowych, w edytorach graficznych często wykonuje się obróbkę plików graficznych pochodzących ze skanera, kamery, aparatu cyfrowego lub zrzutu ekranowego. Sprowadza się to do następujących operacji:

- poprawy parametrów obrazu, takich jak naświetlenie, kolor, kontrast,
- kadrowanie,
- skalowanie,
- przekształcenie (obróć, odbicie),
- przekształcanie geometryczne.

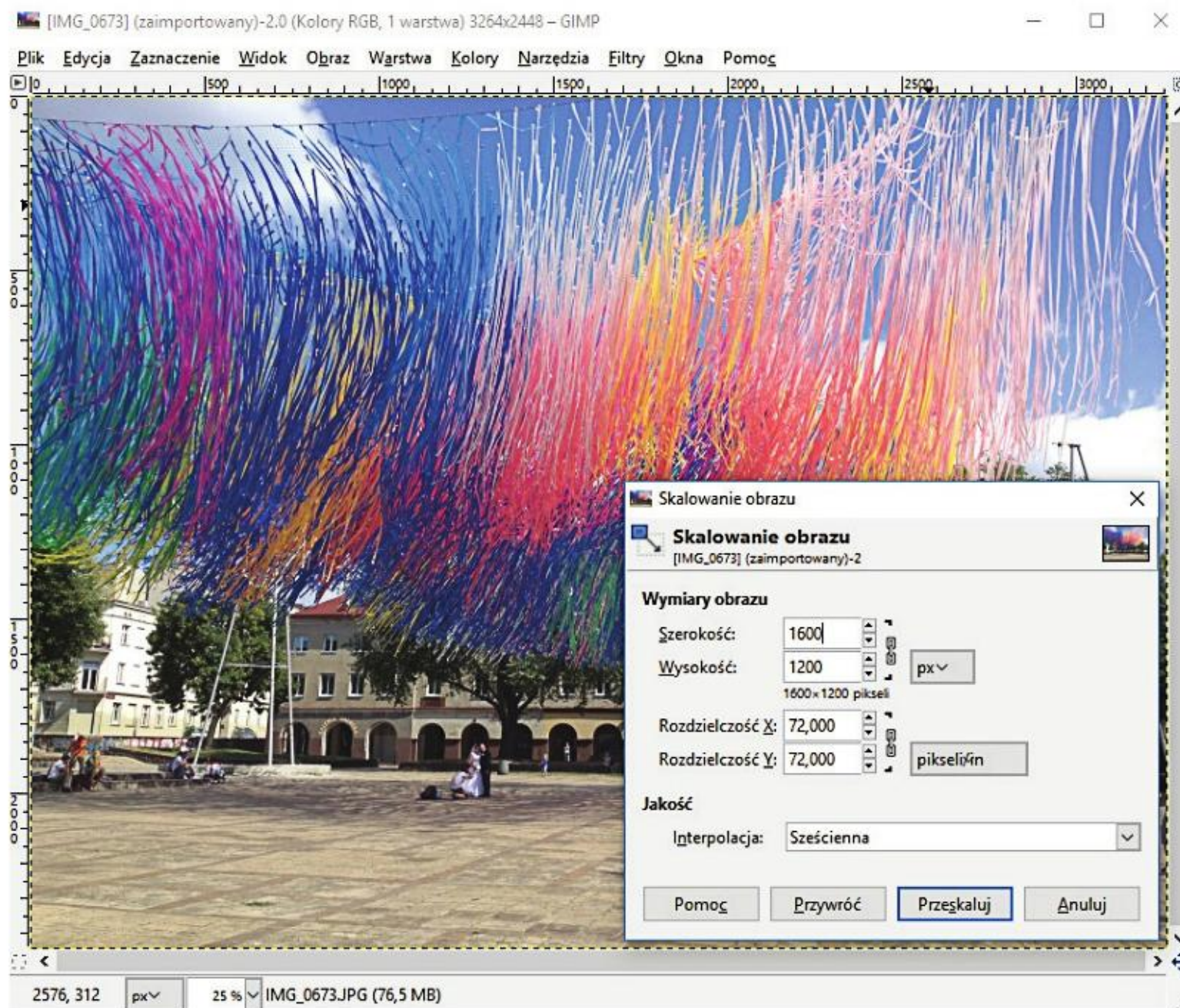
Dodatkowo wykonuje się konwersję między formatami zapisu pliku lub proste techniki poprawy zdjęć typu:

- redukcja czerwonych oczu,
- retusz,
- fotomontaż.

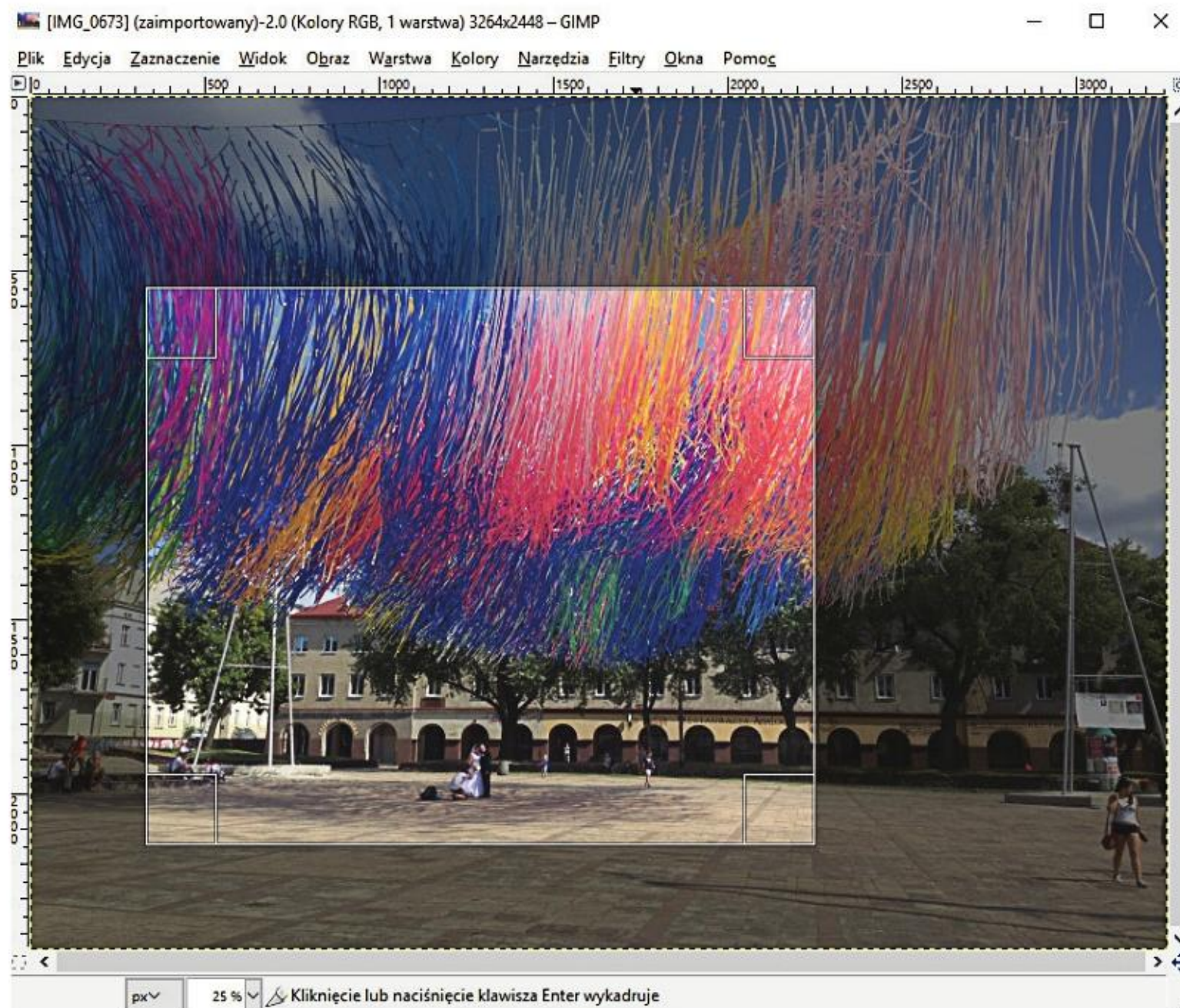
Często wykonywaną czynnością jest również tworzenie i obróbka zrzutów z ekranu.

Skalowanie to proporcjonalna zmiana rozmiarów obrazu względem oryginału. Skalowanie w programie GIMP można wykonać za pomocą opcji **Obraz >> Skaluj obraz**. Ta funkcja pozwala na dopasowanie rozmiaru obrazu do parametrów strony internetowej oraz zmianę rozdzielczości.

Kadrowanie to przycięcie oryginalnego obrazu w celu uzyskania optymalnego kadru do wykorzystania jako oddzielny plik graficzny.

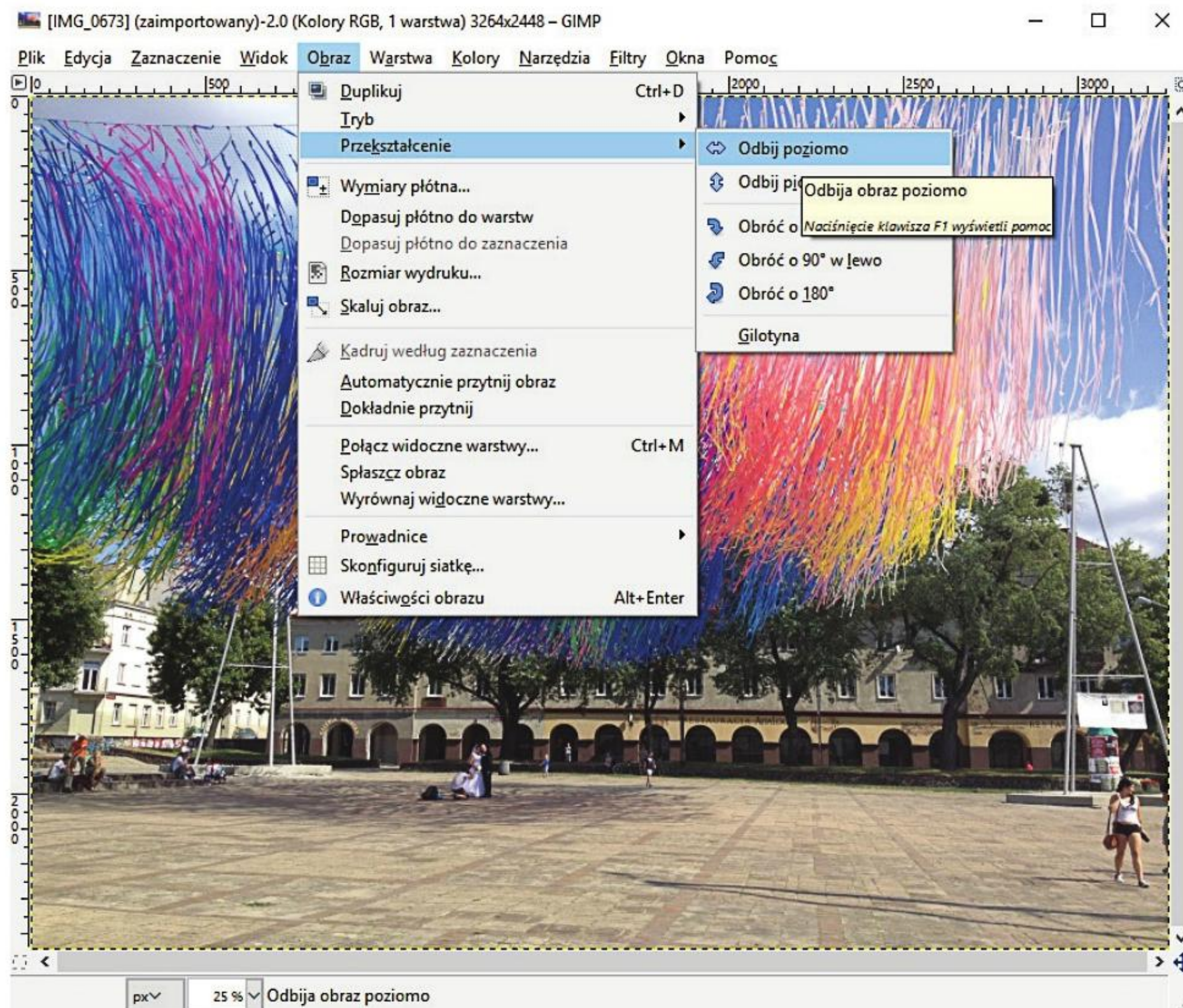


Rys. 22.1. Operacja skalowania obrazu



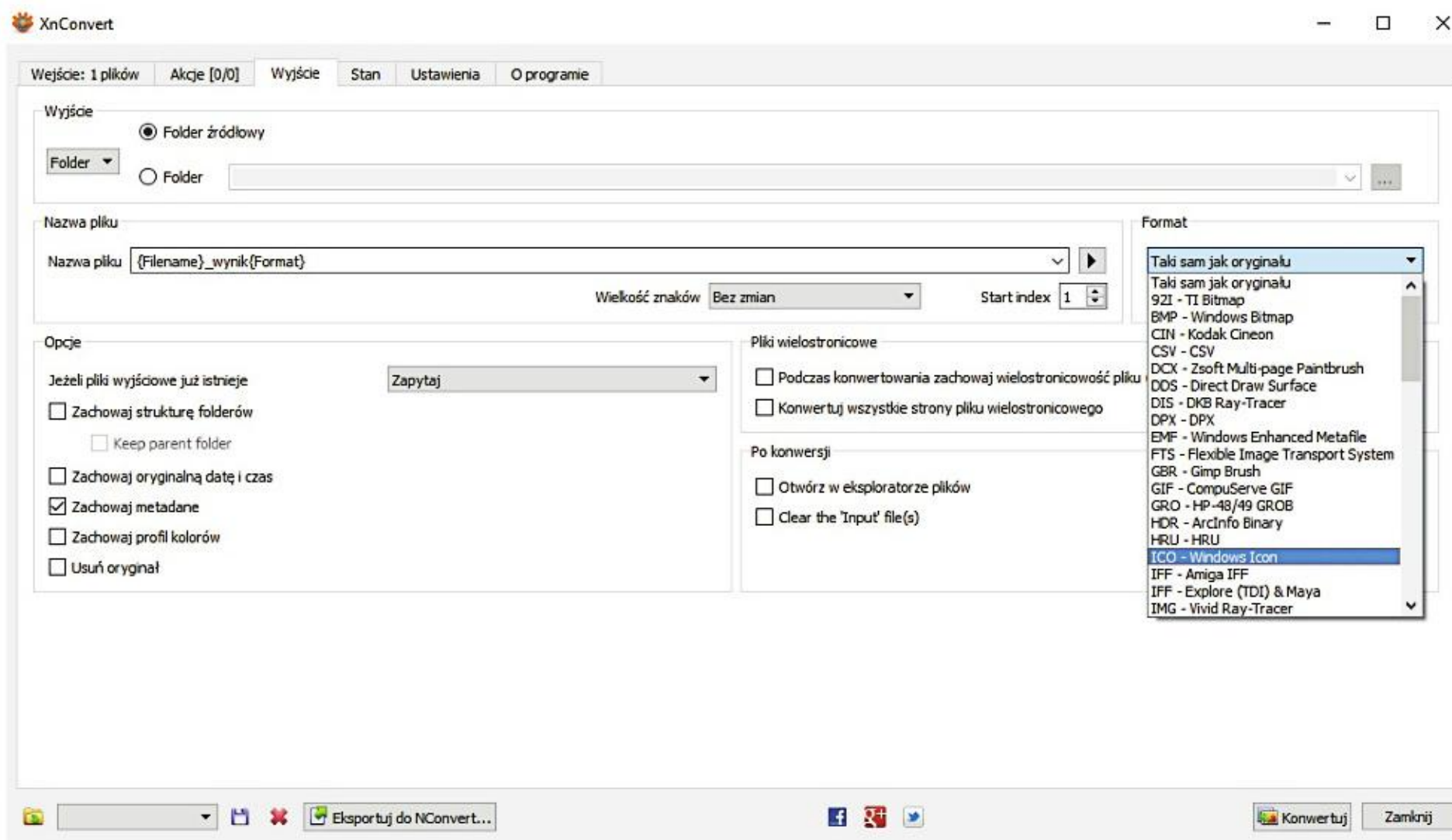
Rys. 22.2. Operacja kadrowania

Przekształcenie pozwala na wykonanie lustrzanego odbicia lub obrotu o określony kąt. Na rys. 22.3 przedstawiono sposób wykonania takiej operacji.



Rys. 22.3. Operacja przekształcania obrazu

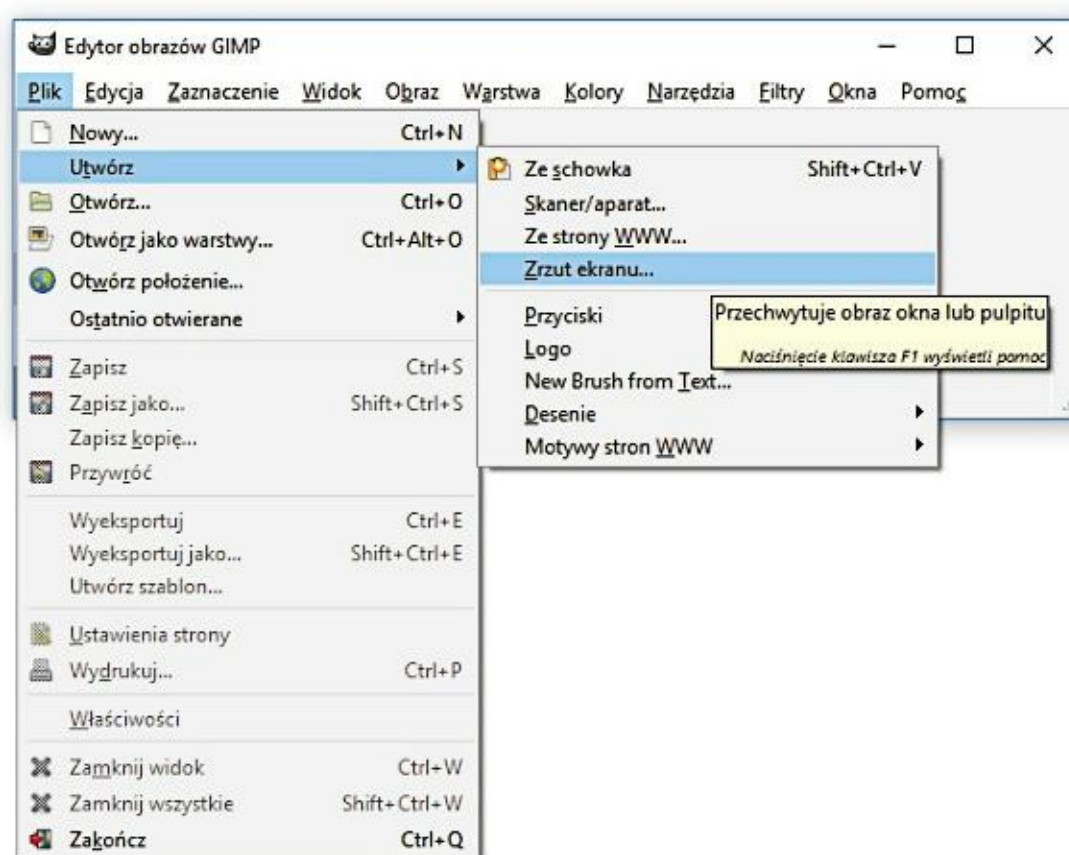
Konwersja formatów polega na zapisaniu (wyeksportowaniu) tego samego pliku w innym formacie obsługiwanym przez program docelowy lub przeglądarkę internetową. Do konwersji można wykorzystać dowolny edytor graficzny lub specjalne aplikacje narzędziowe zwane konwerterami. Przykładem takiej aplikacji jest program **XnConvert**.



Rys. 22.4. Okno programu XnConvert

Zrzuty ekranu. W początkowej fazie rozwoju komputerów osobistych wprowadzono opcję wydruku z ekranu (ang. *print screen*) za pomocą klawisza [PrtScr] lub [PrtSc]. Ta metoda jest nadal wykorzystywana, ale zrzut nie jest wysyłany do drukarki, tylko wklejany do schowka systemowego. Można również użyć skrótu klawiszy [Alt] + [PrtScr] w celu zrzutu zawartości aktywnego okna. Wykonany w ten sposób zrzut otwiera się w dowolnym edytorze graficznym.

Dodatkowo bezpośrednio w programie graficznym GIMP można wykonywać zrzuty do dalszej obróbki. Aby tego dokonać, otwieramy opcję **Plik – Utwórz – Zrzut ekranu**.



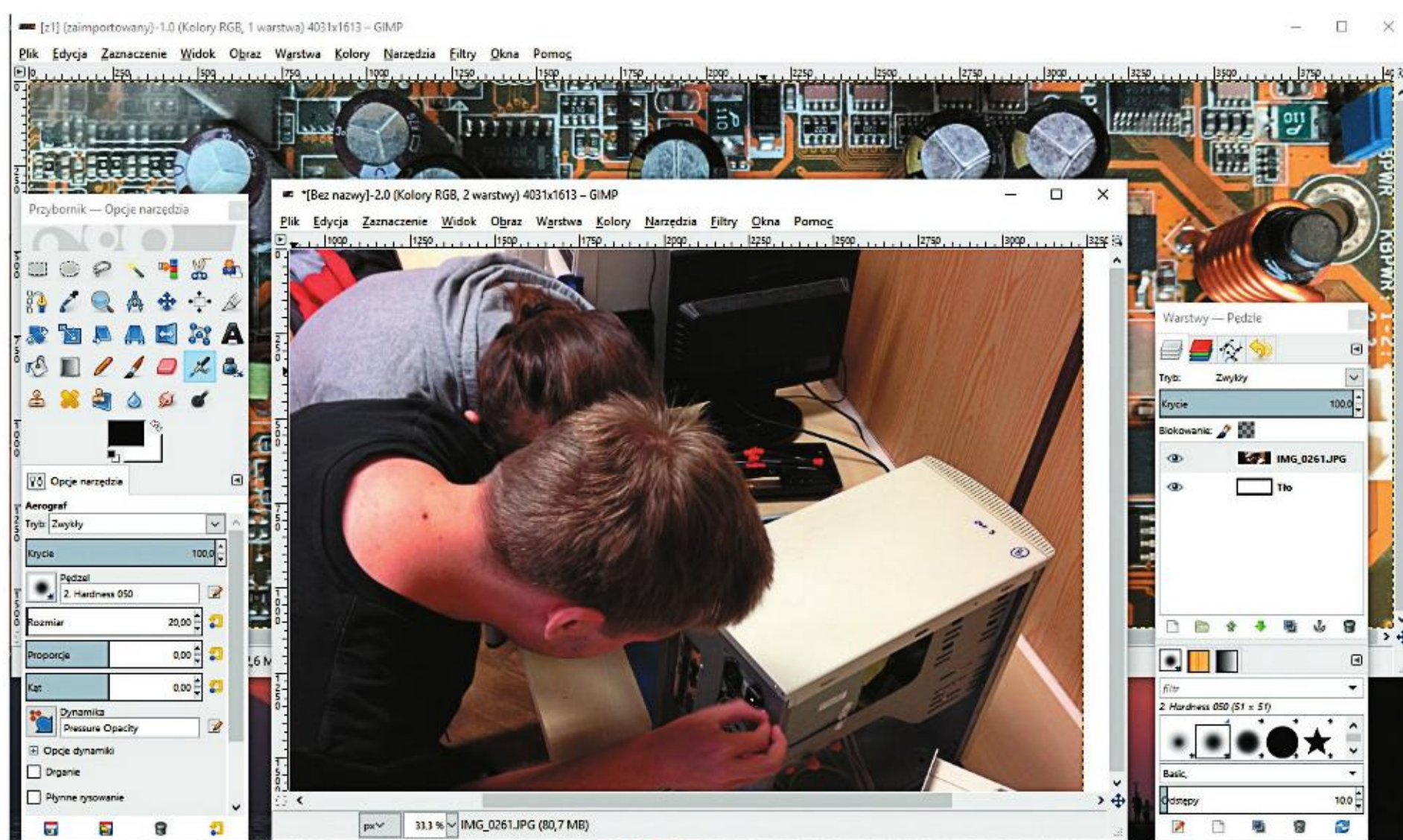
Rys. 22.5. Wykonywanie zrzutów ekranu w programie GIMP

Po utworzeniu zrzutu w programie GIMP lub wklejeniu zrzutu ze schowka systemowego możemy dokonać określonych zmian w rozdzielczości bądź wielkości obrazu.

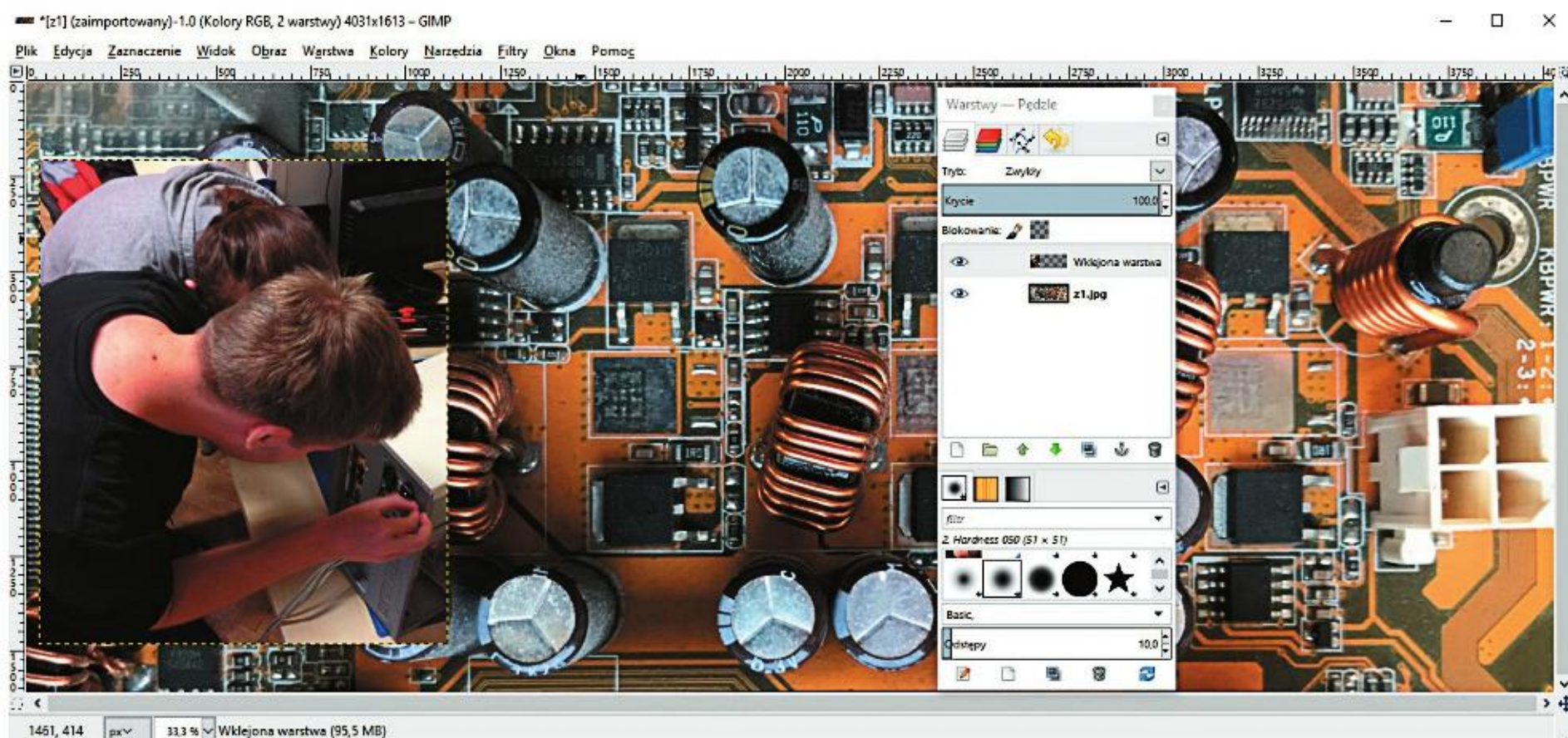
PRZYKŁAD 22.1

Naszym zadaniem jest wykonanie banera na stronę internetową szkoły. Baner ma reklamować jedną z pracowni kształcenia praktycznego. Do wykonania zadania wykorzystamy dwa przygotowane zdjęcia oraz edytor graficzny GIMP. Do wykonania banera wykorzystamy technikę fotomontażu. Aby poprawnie wykonać to zadanie, musimy postępować według poniższych kroków.

Krok 1. Uruchamiamy edytor graficzny GIMP. Następnie kolejno otwieramy dwa przygotowane wcześniej zdjęcia w oddzielnych oknach. Dobieramy zdjęcia tak, aby ich rozmiary, rozdzielczość oraz pozostałe parametry okazały się zbliżone.



Krok 2. Jedno zdjęcie będzie tłem naszego banera. Uczniowie z drugiego zdjęcia będą wklejeni i scaleni z tłem. Zaznaczamy obszar z postaciami uczniów i wycinamy go. Następnie wklejamy do obszaru zdjęcia z tłem.



Wchodzimy do menu warstw. To, co przed chwilą wkleiliśmy, jeszcze nie jest nową warstwą. Aby wklejenie uczynić nową warstwą, musimy wybrać opcję Nowa warstwa. Za pomocą narzędzia Przesunięcie możemy ustawić wklejoną warstwę, tak aby jak najlepiej dopasować grafiki. Na naszym banerze mamy uzyskać wrażenie pracy malutkich ludzi wśród olbrzymich elementów elektronicznych płyty głównej. Dobrą praktyką jest zmniejszenie krycia wklejonej warstwy suwakiem Krycie do ok. 60–70% – w celu lepszego podglądu obu warstw. Jeśli zajdzie taka potrzeba, należy przeskalować wklejoną warstwę. Gdy wykonamy wszystkie czynności i uzyskamy zaplanowany efekt, musimy zmienić krycie na 100%.

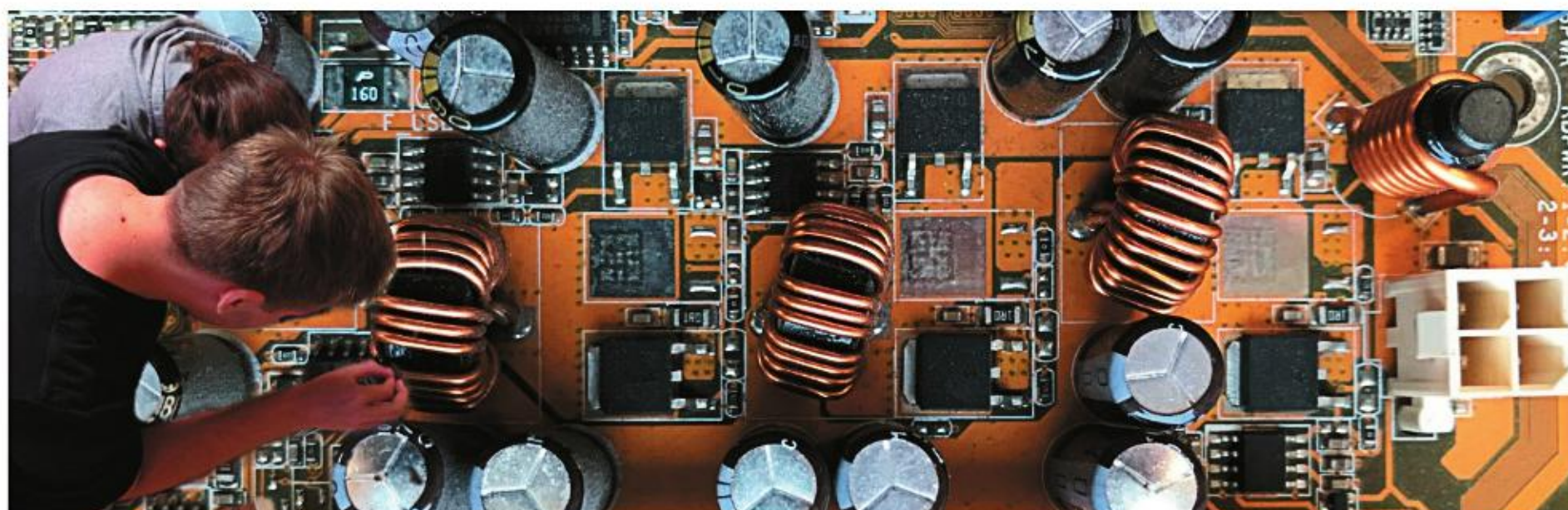
Krok 3. Następnie dodajemy Maskę warstwy (Biała bez przezroczystości) dla wklejonej warstwy. Wykonujemy tę operację w oknie Warstwy – wybieramy z menu uruchomionego prawym klawiszem myszy opcję Dodaj maskę warstwy.



Krok 4. Następnie za pomocą narzędzia Pędzel, wybranego z okna przybornika, oraz koloru czarnego (#000000) rysujemy na masce dodanej do wklejonej warstwy tak, aby zostały tylko te elementy osób, na których nam zależy. Należy tutaj dopasować odpowiednie kształt i wielkość pędzla.



Krok 5. Po zakończeniu usuwania niepotrzebnych elementów wklejonego obrazu nasz baner jest już prawie gotowy. Możemy wykorzystać jeszcze Wygładzanie lub Rozmycie Gaussa w Filtrach do dokładnego wykończenia krawędzi łączących dwa obrazy. Możemy spłaszczyć obraz, likwidując (łącząc) warstwy.



Krok 6. Ostatnim etapem prac będzie dodanie tekstu do naszego banera. Wybieramy narzędzie Tekst oraz czcionkę i jej wielkość. Edytujemy tekst „Pracownia urządzeń techniki komputerowej”. Gotowy baner możemy wyeksportować do dowolnego formatu, aby osadzić go na stronie internetowej.



PRAKTYCZNE WSKAZÓWKI

Do dobrych praktyk stosowanych podczas tworzenia obrazów cyfrowych na potrzeby stron internetowych zalicza się:

- unikanie skalowania obrazów bezpośrednio w kodzie HTML; obrazy powinny być wykonywane w rozmiarze, w jakim ich użyjemy;
- przygotowywanie obrazów z uwzględnieniem rozdzielczości ekranowej urządzeń, na których będą wyświetlane;
- stosowanie modelu barw właściwego dla urządzenia, na którym obraz będzie prezentowany (monitor, drukarka itp.);
- dobieranie formatu pliku graficznego właściwego do zastosowania obrazu na stronie;
- stosowanie kompresji podczas zapisu tylko wtedy, kiedy dany obraz nie będzie poddany zaawansowanej obróbce (np. edycja histogramu);
- pamiętanie o prawach autorskich do obrazów wykorzystywanych na stronie internetowej oraz o ochronie danych osobowych osób znajdujących się na zdjęciach;
- stosowanie w galeriach miniatur obrazów docelowych;
- stosowanie dzielenia na mniejsze elementy w celu przyspieszenia ładowania elementów graficznych mających duże rozmiary.

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj konwersję dowolnego pliku z rozszerzeniem jpg na plik z rozszerzeniem ico. Skorzystaj z zasobów sieci Internet i dowiedz się, jaką funkcję pełnią pliki z rozszerzeniem ico.
2. Zeskanuj dowolne zdjęcie z ostatniego wydarzenia klasowego oraz zapisz je w formacie jpg. Otwórz zdjęcie w programie GIMP i zastosuj na nim trzy dowolnie wybrane filtry dostępne w menu Filtry. Efekt wyeksportuj do pliku z rozszerzeniem png.
3. W programie Inkscape wykonaj logotyp hotelu położonego w nadmorskim kurorcie. Dyrektor hotelu chce, aby głównym motywem był napis „Hotel Bryza SPA” opisany na krzywej imitującej morską falę. Gotowy projekt wyeksportuj do pliku z rozszerzeniem png.

4. W programie GIMP wykonaj baner o rozmiarze 1024×200 na stronę internetową twojej szkoły. Wykorzystaj logo szkoły oraz wykonane przez siebie zdjęcie. Gotowy baner zapisz pod nazwą logo.png.
5. W programie GIMP wykonaj pocztówkę z wakacji. Zastosuj technikę fotomontażu. Gotowy projekt przedstaw do oceny nauczycielowi.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Skorzystaj z sieci internet i znajdź informacje dotyczące matryc światłoczułych najnowszych aparatów cyfrowych. Zapisz w zeszycie parametry obrazów cyfrowych, jakie można wykonać za pomocą tych aparatów.
2. Wyjaśnij, czym różni się kadrowanie od skalowania.
3. Oblicz, ile pikseli zawiera obraz cyfrowy o wymiarach 5,08 cm na 5,08 cm zapisany w rozdzielczości 72 ppi?

VII. Multimedia na stronach internetowych

- Animacja na stronach internetowych
- Publikacja plików wideo na stronach internetowych
- Przygotowanie plików dźwiękowych

23

Animacja na stronach internetowych

ZAGADNIENIA

- Animacja komputerowa
- Formaty plików z animacją
- Rodzaje animacji
- Przygotowanie animacji na potrzeby strony internetowej

Animacja komputerowa

W poprzednim rozdziale opisano zasady tworzenia i edycji **grafiki statycznej**. Charakteryzuje się ona wyświetlaniem obrazu cyfrowego o stałych, określonych w czasie edycji parametrach. W nowoczesnych stronach i witrynach internetowych coraz częściej korzysta się także z grafiki dynamicznej, którą nazywa się **grafiką dynamiczną** lub **animacją**.

Animacja to tworzenie cyfrowych obrazów ruchomych z obrazów statycznych za pomocą zmiany parametrów wyświetlania, odtwarzanych w określonym czasie.

Animacja może obejmować zmiany:

- położenia obiektu;
- kształtu i rozmiaru obiektu;
- wyglądu (koloru, gradientu, tekstury itp.);
- położenia i orientacji kamery;
- parametrów oświetlenia planu.

Każdy obraz, z którego składa się animacja, nazywa się **ramką** (ang. *frame*) lub **klatką**.

Do podstawowych metod tworzenia animacji komputerowej zalicza się:

- **animację poklatkową** – każda ramka animacji tworzona osobno, niezależnie od innych, najczęściej w postaci mapy bitowej;
- **metodę ramek kluczowych** (ang. *key frames*) – generowanie wybranych ramek, które definiują ważne fazy animacji; pozostałe ramki są obliczane (interpolowane) na podstawie ramek kluczowych;
- **animację proceduralną** (ang. *procedural animation*) – poszczególne ramki są obliczane przy użyciu skryptów (algorytmów); skrypty opisują szczegółową zmianę właściwości poszczególnych obiektów;
- **animację szkieletową** – definiuje każdy ruchomy obiekt za pomocą dwóch składowych: szkieletu (ang. *skeleton*) oraz skóry (ang. *skin*); szkielet tworzy hierarchiczną strukturę,

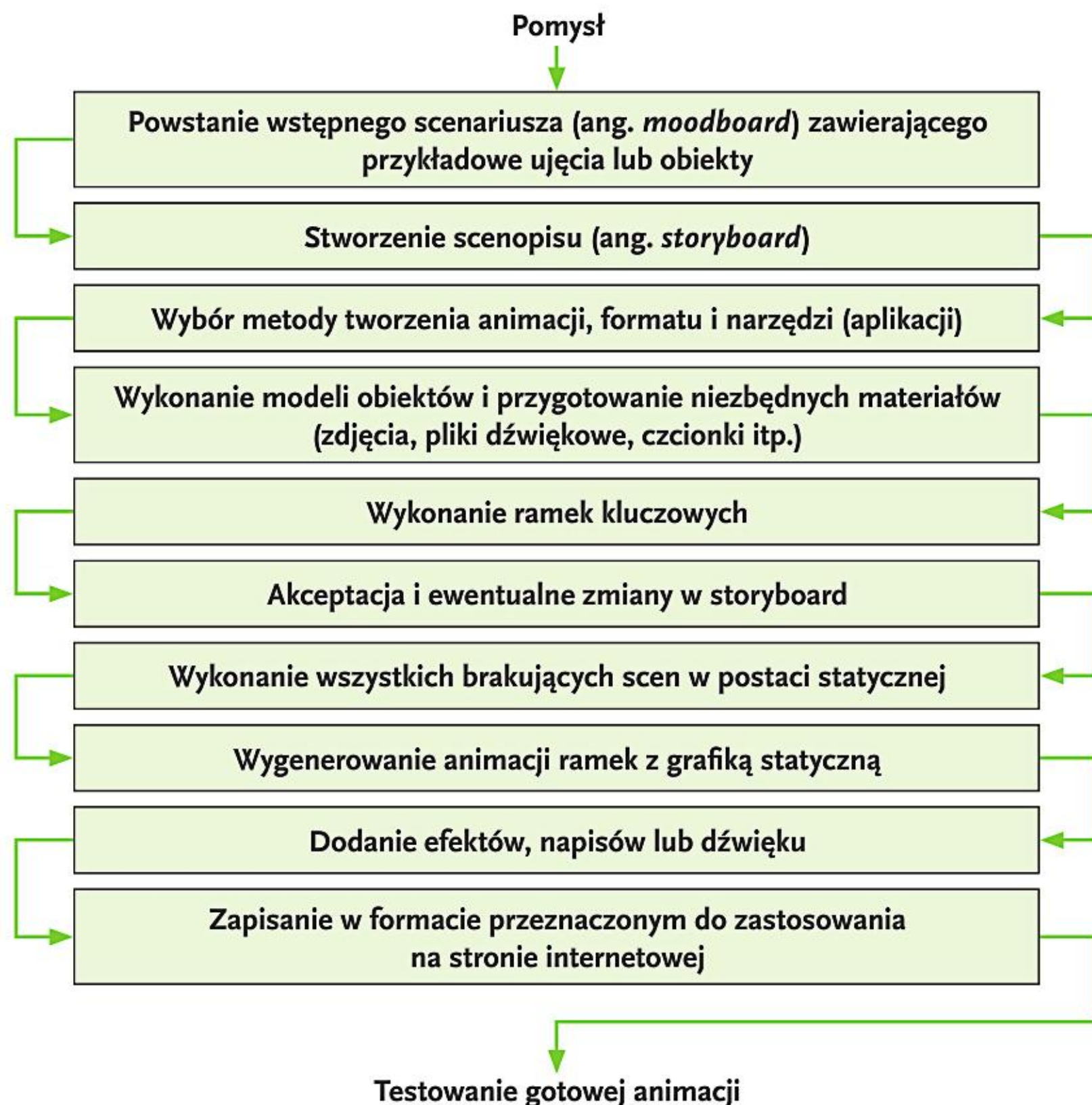
na którą jest nakładana skóra (np. tekstura lub siatka trójkątowa) przenosząca ruch szkieletu, ten zaś jest zgodny z przyjętymi zasadami i modelami kinematycznymi;

- **animację interaktywną** – obiekty są sterowane przez użytkownika.

Poszczególne obrazy w animacji zmieniają się z odpowiednią częstotliwością liczoną w **klatkach na sekundę – fps** (ang. *frames per second*). Animację uznaje się za płynną wtedy, gdy częstotliwość wynosi co najmniej 12 fps. Przykładowo: częstotliwość 24 fps stosuje się w kinematografii. Przyjętym standardem animacji komputerowej jest zakres od 25 fps do 35 fps. W bardzo zaawansowanych graficznie grach komputerowych spotyka się częstotliwość dochodzącą do 60 fps.

Animacja komputerowa może być tworzona w dwóch trybach: offline oraz czasu rzeczywistego. **Animację offline** wykorzystuje się na stronach internetowych. Kolejne ramki są edytowane według określonego **scenopisu**, a efekt prac jest zapisywany do pliku. **Animacja czasu rzeczywistego** polega na generowaniu poszczególnych klatek na bieżąco, np. w grach komputerowych.

Na rys. 23.1 przedstawiono przykładowe etapy tworzenia animacji komputerowej wykorzystywanej na stronach internetowych.



Rys. 23.1. Etapy tworzenia animacji komputerowej

Storyboard to tabela zawierająca rysunkowy oraz słowny opis poszczególnych scen.

Podczas projektowania i tworzenia stron oraz aplikacji internetowych animacje najczęściej stosuje się w następujących miejscach:

- w nawigacji menu i podmenu;
- podczas ładowaniu plików strony;
- w trakcie pobierania lub przesyłania danych;
- podczas akcji wyboru przycisku;
- w banerach, hero image i sliderach;
- w reklamach bądź marketingowych akcjach.

Slider – rodzaj banera, w którym zbiór grafik jest naprzemiennie wyświetlany za pomocą określonego algorytmu. Elementami sliderów są również przyciski pozwalające na przewijanie slajdów.

Hero image – animowany baner, w którym wykorzystujemy główną animowaną grafikę z tekstem. Tego rodzaju baner staje się pierwszym elementem wyświetlanej strony.

Do podstawowych formatów plików, w których zapisuje się dynamiczną grafikę, zalicza się:

- **GIF** (ang. *Graphics Interchange Format*) – popularny format internetowej grafiki statycznej i dynamicznej obsługiwany przez większość przeglądarek internetowych;
- **SWF** – zamknięty format grafiki wektorowej dedykowany aplikacji Adobe Flash, który umożliwia zapisanie i odtworzenie animacji oraz całych struktur stron internetowych.

Animacja skryptowa

Animację elementów dokumentów hipertekstowych umożliwiają skrypty CSS wykorzystujące np. właściwość **animation** oraz reguły **@keyframes**. Na rys. 23.2 pokazano przykład wykorzystania CSS do płynnej zmiany koloru tła w określonym czasie.

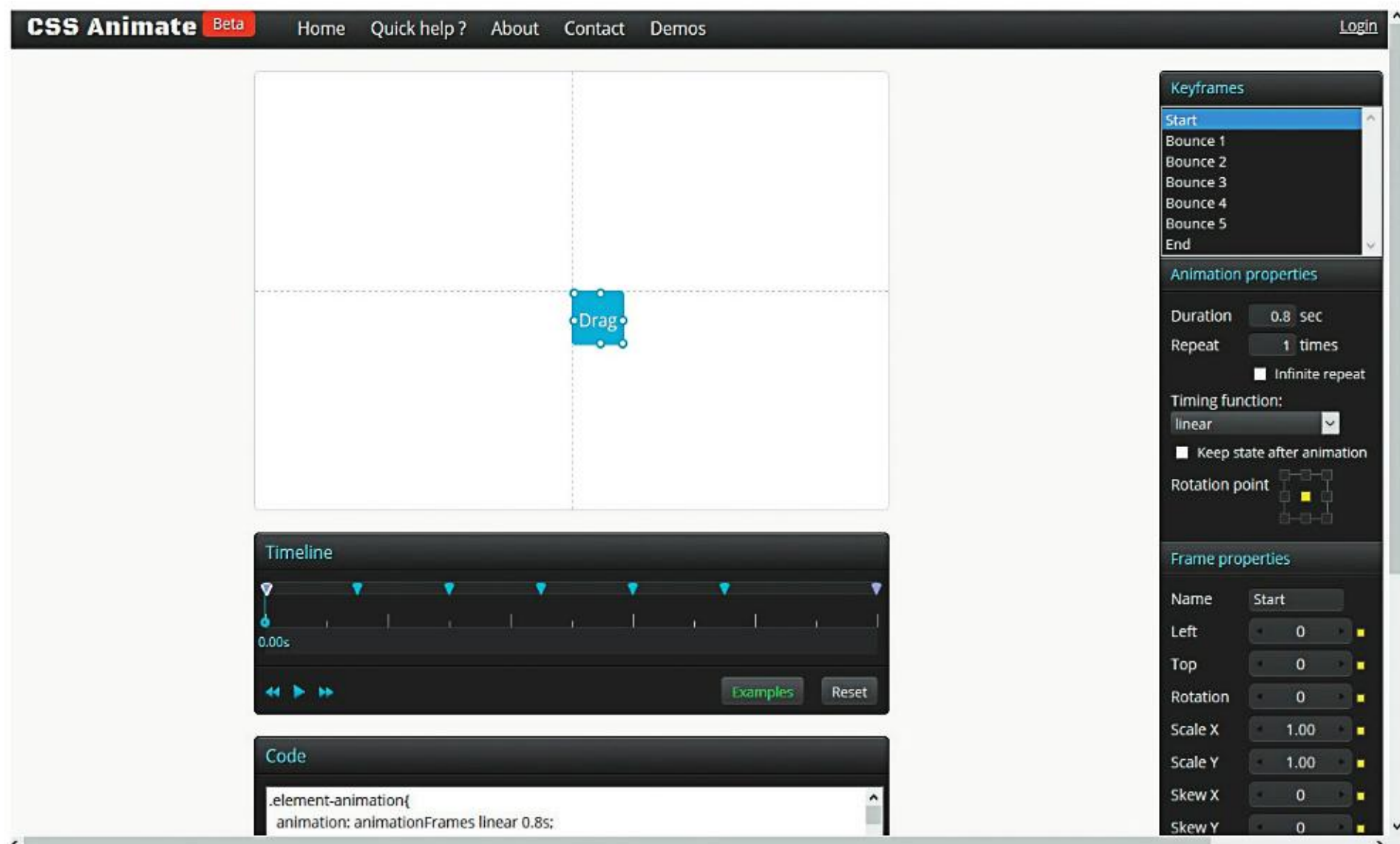
```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5    div {
6      animation: pulse 5s infinite;
7    }
8
9    @keyframes pulse {
10     0% {
11       background-color: #3322FF;
12     }
13     100% {
14       background-color: #FF1245;
15     }
16   }
17 </style>
18 </head>
19 <body>
20 <div>Przykład animacji poprzez zmianę koloru tła</div>
21 </body>
22 </html>

```

Rys. 23.2. Przykład kodu umożliwiający płynną zmianę koloru tła pod tekstem

Coraz więcej aplikacji internetowych służy do budowania animacji na podstawie arkuszy stylów CSS 3. Przykładem jest tu aplikacja online <http://cssanimate.com/>, za której pomocą można utworzyć prostą animację na podstawie kilku udostępnionych przykładów oraz skopiować wygenerowany kod CSS (rys. 23.3).



Rys. 23.3. Okno cssanimate.com

Z powodzeniem można też tworzyć animację skryptową na poziomie skryptów JavaScript i przygotować wrażenie animacji poklatkowej. Przykłady tego rodzaju skryptów zostaną opisane w drugiej części podręcznika.

Dodatkowo skryptów można używać w tworzeniu animacji ramek kluczowych w takich programach, jak **Adobe Flash** lub **Adobe Animate CC**. Do dyspozycji pozostaje język skryptowy **ActionScript**, który służy głównie do kontroli animacji, tworzenia prezentacji danych oraz interaktywnych aplikacji. Skrypty można tworzyć za pomocą panelu **Operacje**, okna **Skrypt** lub edytora zewnętrznego. Skrypty napisane w języku ActionScript uruchamia się z poziomu animacji, kiedy dotrze do niego wskaźnik odtwarzania, lub za pomocą zdarzeń wywołanych przez aktywne elementy, jak np. przyciski. Pełną dokumentację tego języka skryptowego umieszczono na stronie http://help.adobe.com/pl_PL/FlashPlatform/reference/actionscript/3/index.html.

Animowane obrazy GIF

Animowane GIF-y to najpopularniejszy i najprostszy sposób tworzenia animacji poklatkowej. Ten format może przechowywać wiele obrazków w jednym pliku i tworzyć z nich animację. Dodatkowo GIF obsługuje przezroczystość monochromatyczną, co ma znaczenie w przypadku graficznego projektowania stron internetowych.

Do stworzenia animowanych GIF-ów wykorzystuje się liczne oprogramowanie, np. GIF Animator, Free GIF Maker czy QGifer. Prosta animację poklatkową można również wykonać za pomocą popularnych edytorów graficznych, np. GIMP bądź Adobe Photoshop.

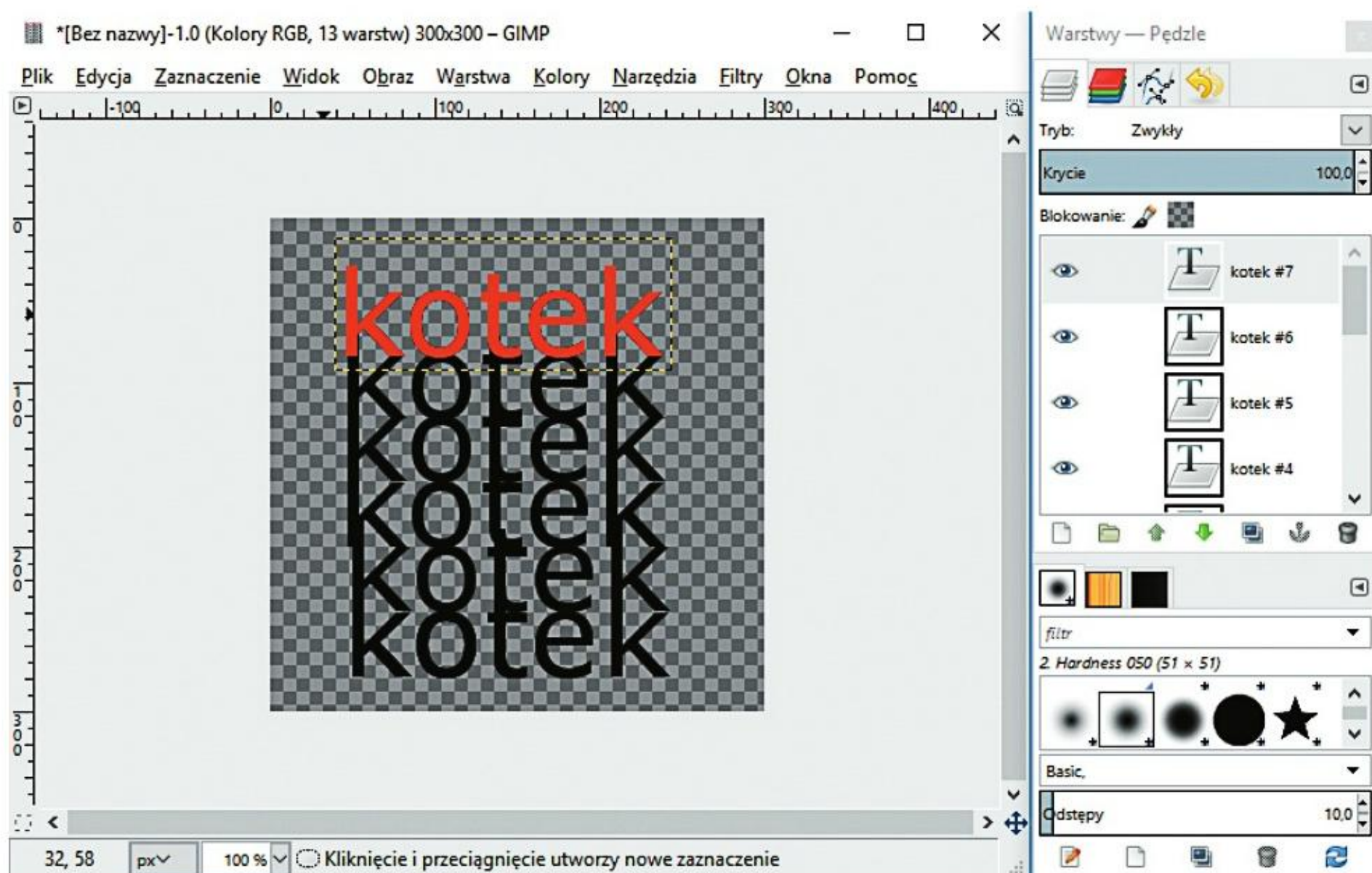
PRZYKŁAD 23.1

Animowany gif w GIMP

Animację poklatkową tworzymy dzięki powielaniu początkowego obrazu statycznego oraz przeprowadzeniu modyfikacji na zdublowanych warstwach. Przykładową animację w programie GIMP wykonamy w czterech krokach.

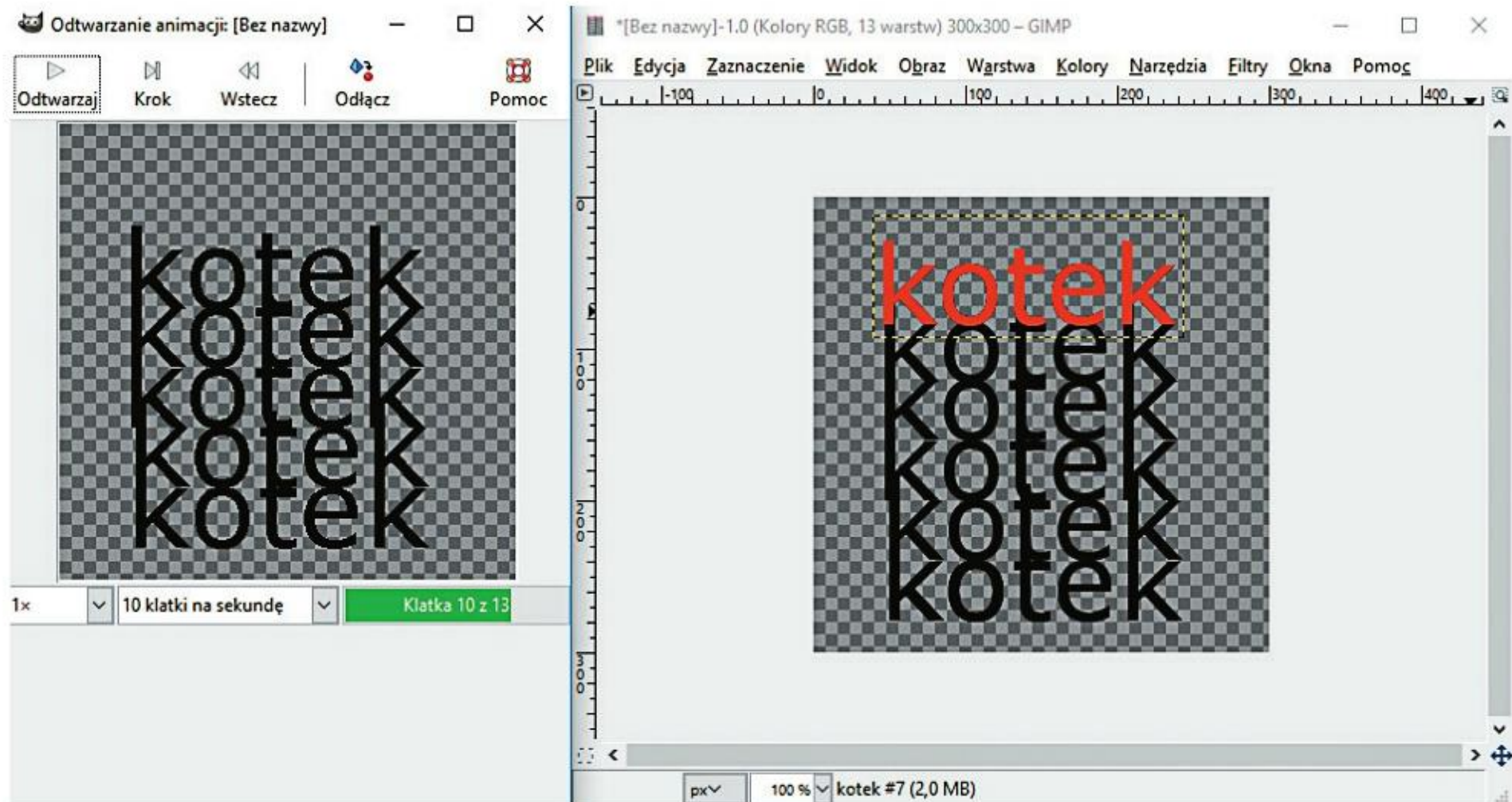
Krok 1. Tworzymy nowy dokument: ustawiamy założone rozmiary obrazu. Animację wykorzystamy jako logo na stronie z własnym tłem, dlatego wybieramy dla naszego obrazu przezroczystość.

Krok 2. Po utworzeniu nowego obrazu tworzymy napis KOTEK. Piszemy literę K, następnie tworzymy duplikat warstwy za pomocą okna warstw. Powtarzamy tę czynność aż do momentu, gdy powstanie cały napis. Dzięki tej samej metodzie kopiujemy napis linia po linii do góry, a w ostatniej linii zmieniamy kolor napisu. Cały czas pamiętamy o wykonywaniu duplikatów warstw po każdej edycji napisu. Powstaje w ten sposób zbiór warstw, w którym każda z nich jest traktowana jako jedna klatka naszej animacji. Efekt końcowy edycji jest widoczny na rys. 23.4.



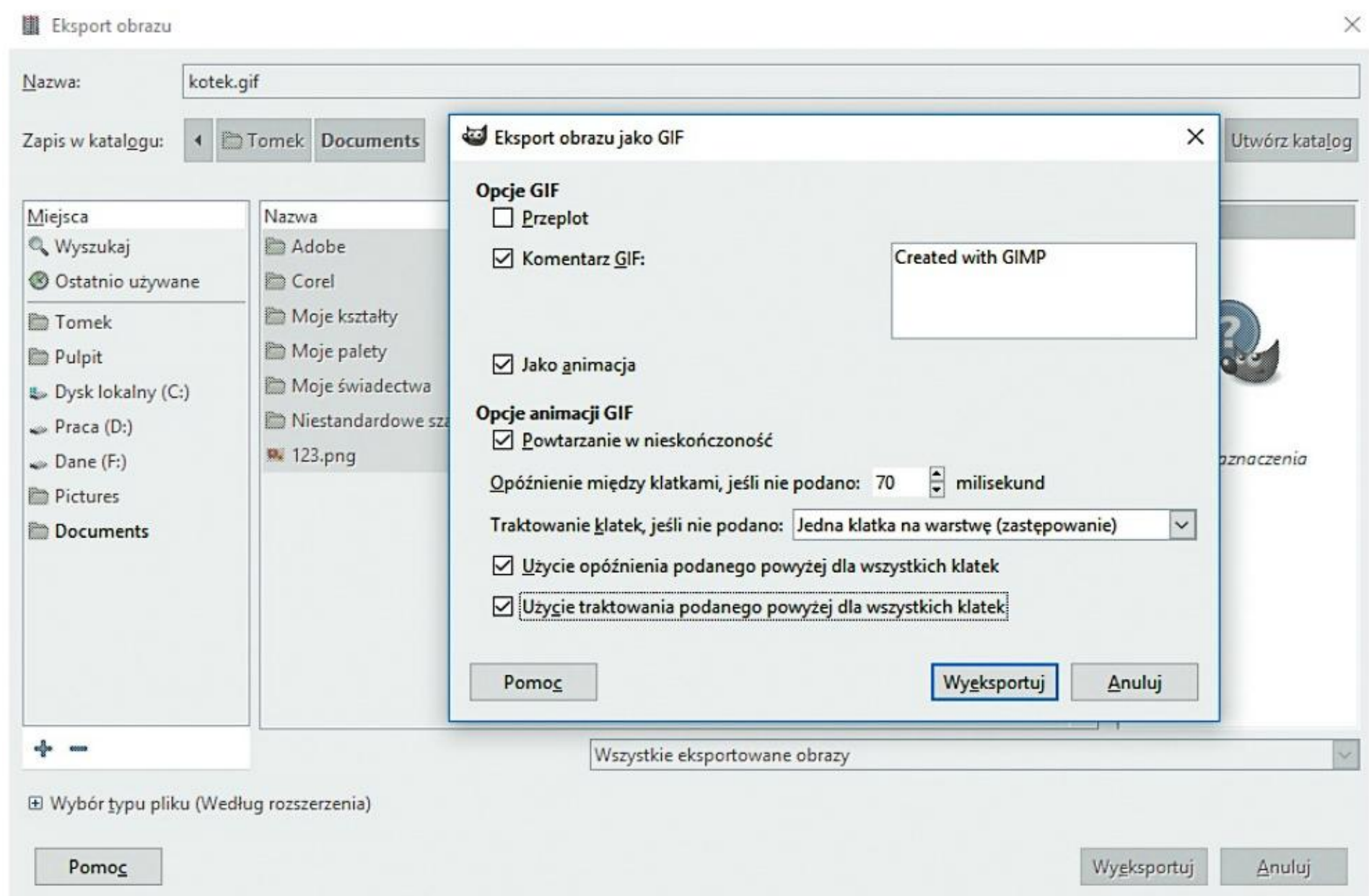
Rys. 23.4. Przykład animacji wykonanej w programie GIMP

Krok 3. Po zakończeniu edycji możemy uruchomić podgląd animacji dzięki poleceniu **Filtry - Animacja - Odtwarzanie**.



Rys. 23.5. Testowanie animacji w programie GIMP

Krok 4. Następnie musimy wyeksportować obraz w formie animacji. W oknie wyeksportowania podajemy nazwę kotek.gif i ustawiamy opcje odtwarzania animacji. Przykład tej czynności pokazano na rys. 23.6.

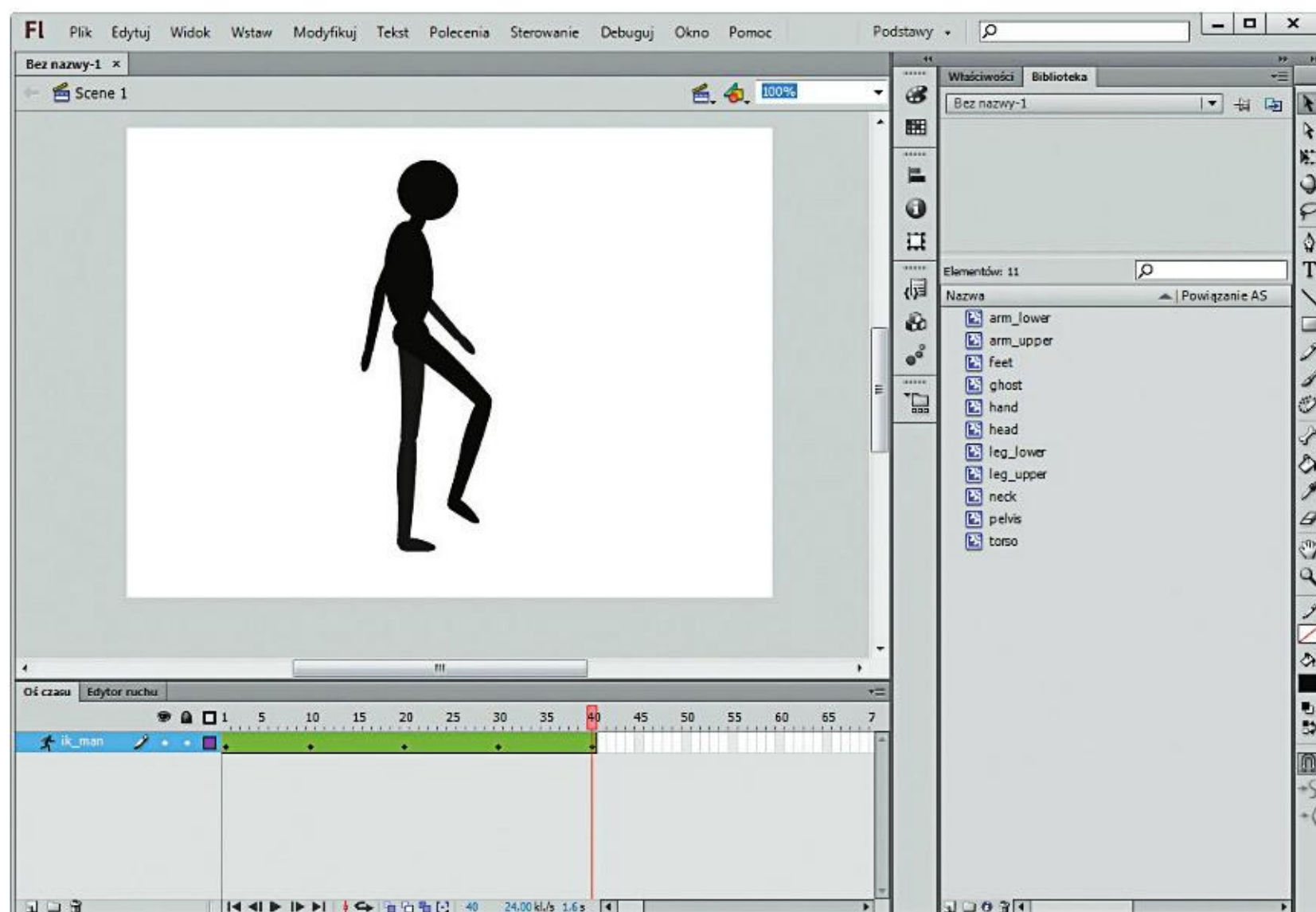


Rys. 23.6. Eksport animacji do formatu GIF

Po zapisaniu obraz można odtworzyć w dowolnej przeglądarce. _____

Animacja SWF

Pliki SWF mogą zawierać animacje dzięki wykorzystaniu metodę ramek kluczowych lub apletów w języku skryptowym. **Ramka kluczowa** to zazwyczaj główna klatka filmu, która musi być opisana w całości, ponieważ jest elementem łączącym klatki pośrednie w procesie animacji. Pliki SWF mogą być używane do tworzenia animowanej grafiki i menu na stronach internetowych. Ten format powstał jako format zamknięty programu **Adobe Flash**.



Rys. 23.7. Okno programu Adobe Flash

Program Adobe Flash (wcześniej Macromedia Flash) przez wiele lat był głównym narzędziem do tworzenia animacji na strony internetowe w formacie SWF. Do jej odtwarzania niezbędne jest zainstalowanie odtwarzacza Flash Player.

Głównym obszarem przeznaczonym do pracy jest stół montażowy, na którym montuje się animację. Poniżej znajduje się oś czasu, którą można indywidualnie modyfikować dla każdej warstwy. Poza tym są dostępne biblioteka animowanych symboli i obrazów oraz paski narzędziowe (np. do rysowania). Na rys. 23.7 przedstawiono okno programu Adobe Flash.

Pracę z programem należy rozpocząć od zaplanowania animacji i przygotowania elementów graficznych lub zdjęć. Ważne jest określenie liczby klatek animacji, rozdzielczości obrazu oraz tempa odtwarzania.

Dzięki dostępności narzędzi do tworzenia grafiki wektorowej oraz edycji tekstu większość potrzebnych elementów animacji wykonuje się w programie. Na koniec pracy można dodać dźwięk i opublikować gotową animację do formatu SWF, GIF lub samo uruchamiających się plików EXE. Gotowy projekt zapisuje się również w wewnętrznym formacie FLA.

Adobe Flash dysponuje także obiektywnym językiem skryptowym do sterowania animacjami **Action Script** w wersjach 2.0 oraz 3.0. Skrypty można tworzyć w trakcie budowy animacji.

Do ciekawych opcji programu należy m.in. wykorzystanie w animacji **ruchu po ścieżce**. W ten sposób można animować obiektem po narysowanej wcześniej ścieżce. Oprócz tego, zarówno w obrazach statycznych, jak i w animacjach wykorzystuje się warstwy i **maskowania**. Polega to na tym, że jeden obiekt może zostać wykorzystany jako maska w celu utworzenia otworu, przez który będą widoczne inne obiekty wyświetlane na niższej warstwie.

W programach typu Adobe Flash często wykorzystuje się animację **metodą klatek pośrednich**. Polega ona na utworzeniu rysunków statycznych w początkowej ramce kluczowej oraz końcowej ramce kluczowej, która pokazuje zamierzony efekt końcowy. Za pomocą opcji **Animacja ruchu** program generuje grupę ramek pośrednich i tworzy płynną animację.

Technologie wprowadzoną przez Adobe Flash coraz częściej zastępują w ostatnich latach HTML 5, WebGL, WebAssembly i CSS. Tylko do końca 2020 r. Flash będzie wspierany przez Microsoft i Google Chrome.

Do tworzenia animacji internetowych korzystających z metody ramki (klatki) kluczowej poleca się aplikacje Adobe Premiere Pro i Adobe Animate CC.

Innymi programami do tworzenia animacji metodą ramki kluczowej są:

- Alligator Flash Designer,
- SwiSH Max,
- Sothink SWF Quicker.

Niestety, wszystkie opisywane programy są udostępniane na licencji komercyjnej, choć umożliwiają pracę również w wersji trial.

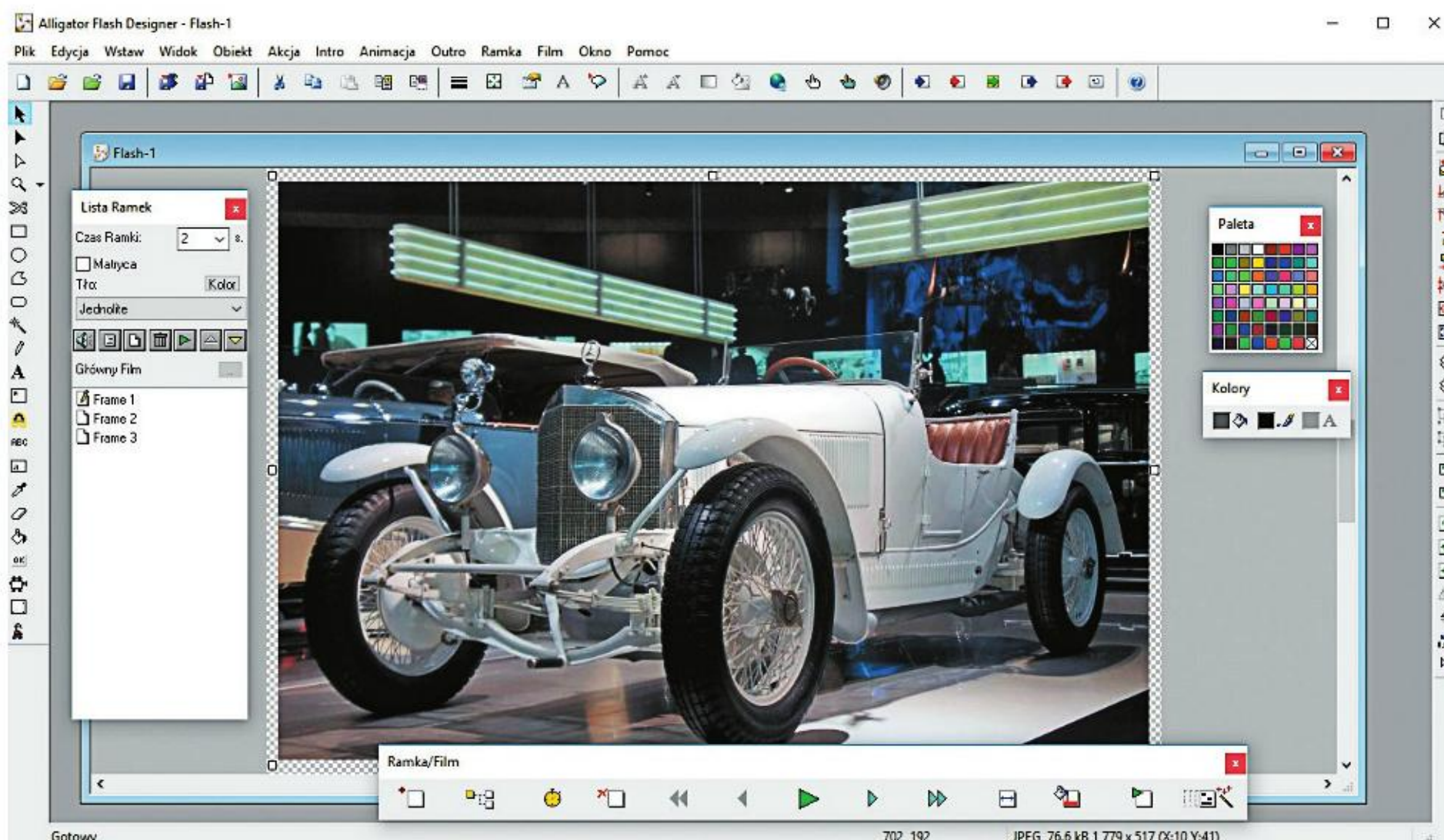
PRZYKŁAD 23.2

Przykład wykorzystania animacji flash pokażemy dzięki posłużeniu się programem **Alligator Flash Designer**. W czterech krokach utworzymy prosty pokaz slajdów do wykorzystania na stronie internetowej.

Krok 1. Na wstępie musimy wyselekcjonować i przygotować zdjęcia do pokazu slajdów. W naszym przykładzie będą to pliki o nazwach 1.jpg, 2.jpg i 3.jpg, poddane obróbce poprawiającej jakość zdjęcia oraz skalowaniu w programie GIMP.

Krok 2. Po uruchomieniu programu Alligator Flash Designer określamy rozdzielczość ekranu na 800 px na 600 px oraz brak tła pokazu ze względu na lepsze dopasowanie takiego obrazu do konkretnej kolorystyki strony, na której animacja będzie publikowana.

Krok 3. Po zatwierdzeniu wstępnych parametrów pojawi się obszar roboczy z pierwszą ramką gotową do pracy. Pierwsza ramka to nasz pierwszy slajd. Wstawiamy tutaj pierwsze zdjęcie. Następnie tworzymy drugą ramkę za pomocą okna **Lista Ramek**. W drugiej ramce wstawiamy drugie zdjęcie. Operację powtarzamy dla trzeciego zdjęcia. Przykład edycji przedstawiono na rys. 23.8.



Rys. 23.8. Przygotowanie pokazu slajdów

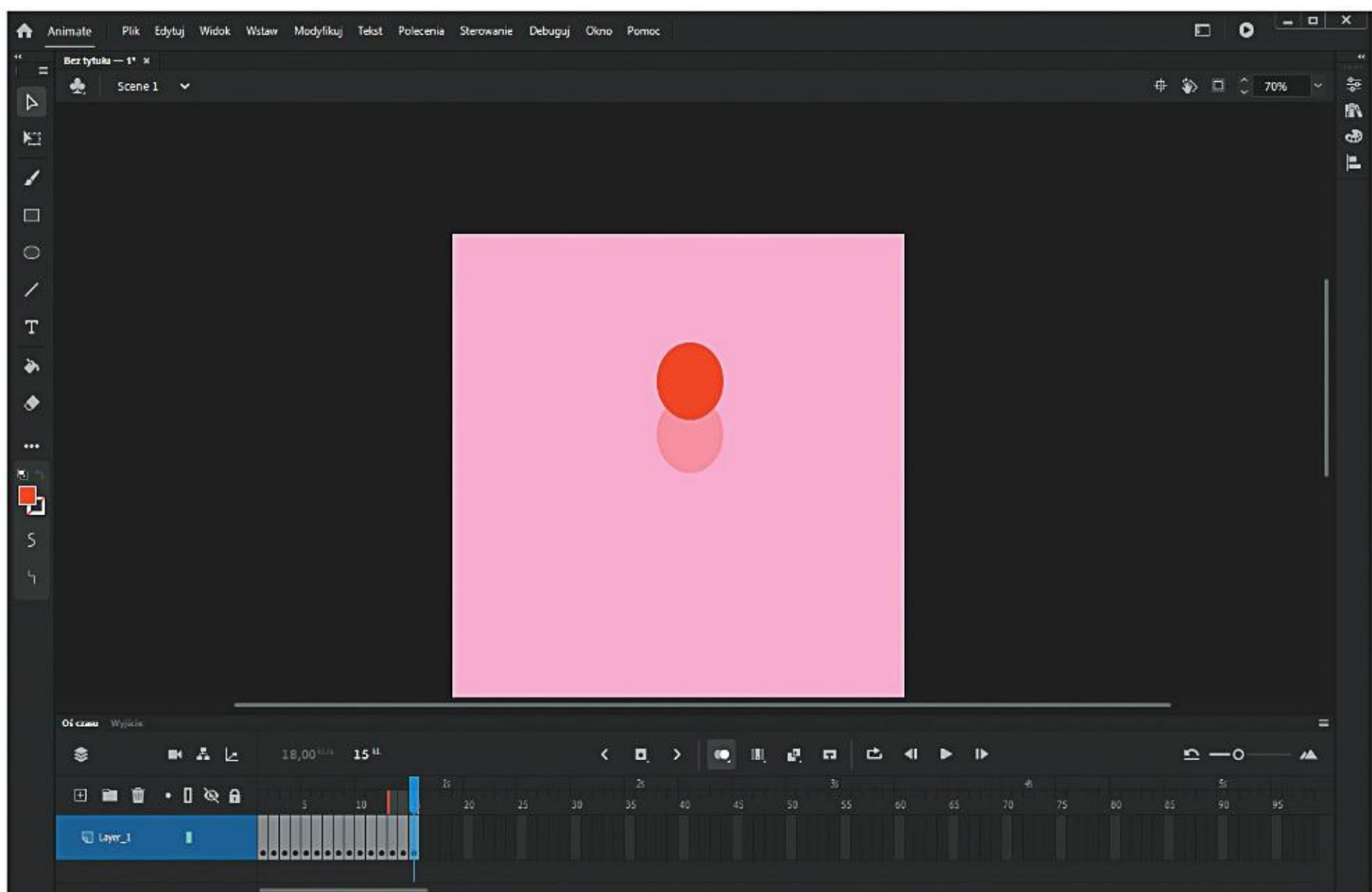
Krok 4. Wystarczy teraz skorzystać z ustawienia parametrów animacji za pomocą funkcji **Animacja – Kreator animacji** i zastosować je do wszystkich ramek. Po opublikowaniu animacji można ją wykorzystać na dowolnej stronie internetowej. Efekt działania animacji pokazano na rys. 23.9.



Rys. 23.9. Przykład gotowej animacji wykonanej w programie Alligator Flash Designer

ĆWICZENIE 1**Tworzenie banera w Adobe Animate CC**

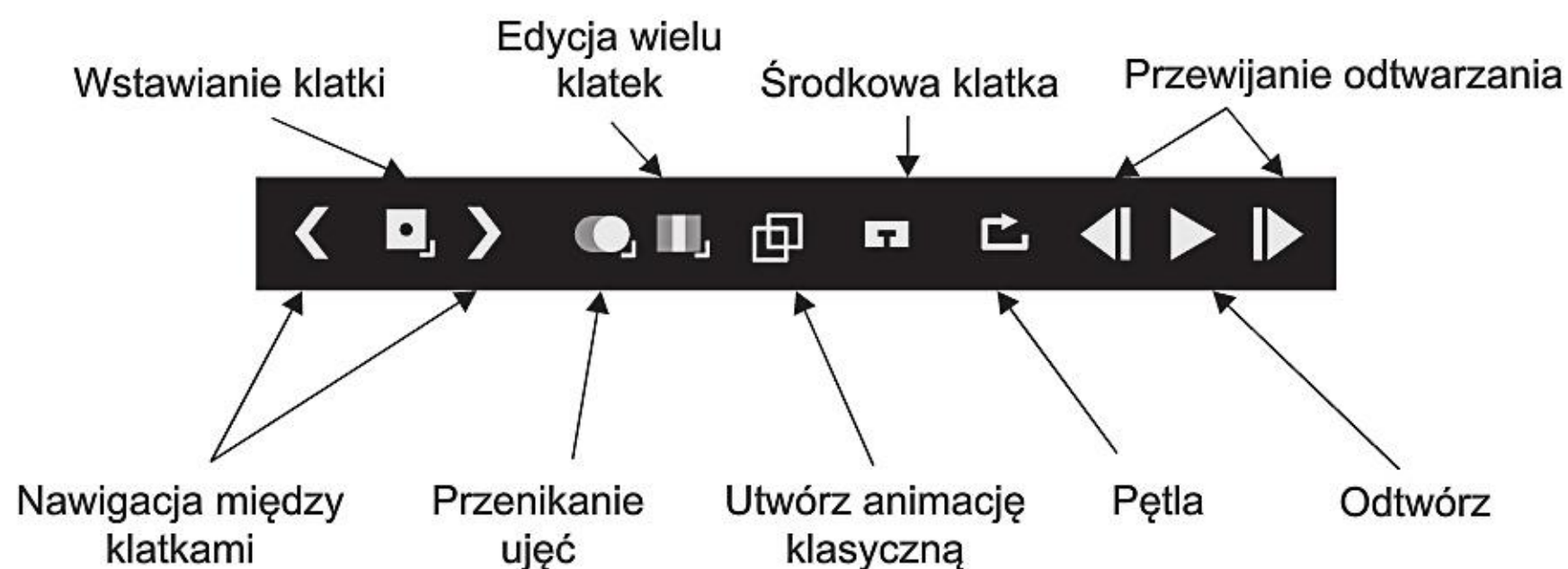
Naszym zadaniem jest utworzenie banera na stronę internetową metodą ramek kluczowych. Tym razem wybierzemy program Adobe Animate CC, dostępny do wypróbowania na stronie Adobe. Adobe Animate CC to bardzo wszechstronna aplikacja do tworzenia zarówno prostych, jak i zaawansowanych projektów animacji komputerowej. Od wersji Animate CC formatem wewnętrznym tworzonych plików FLA jest **XFL**. Dla pliku zapisywanego w formacie programu Animate formatem domyślnym jest FLA, a formatem wewnętrznym pliku – XFL. Na rys. 23.10 przedstawiono okno programu.



Rys. 23.10. Okno programu Adobe Animate CC

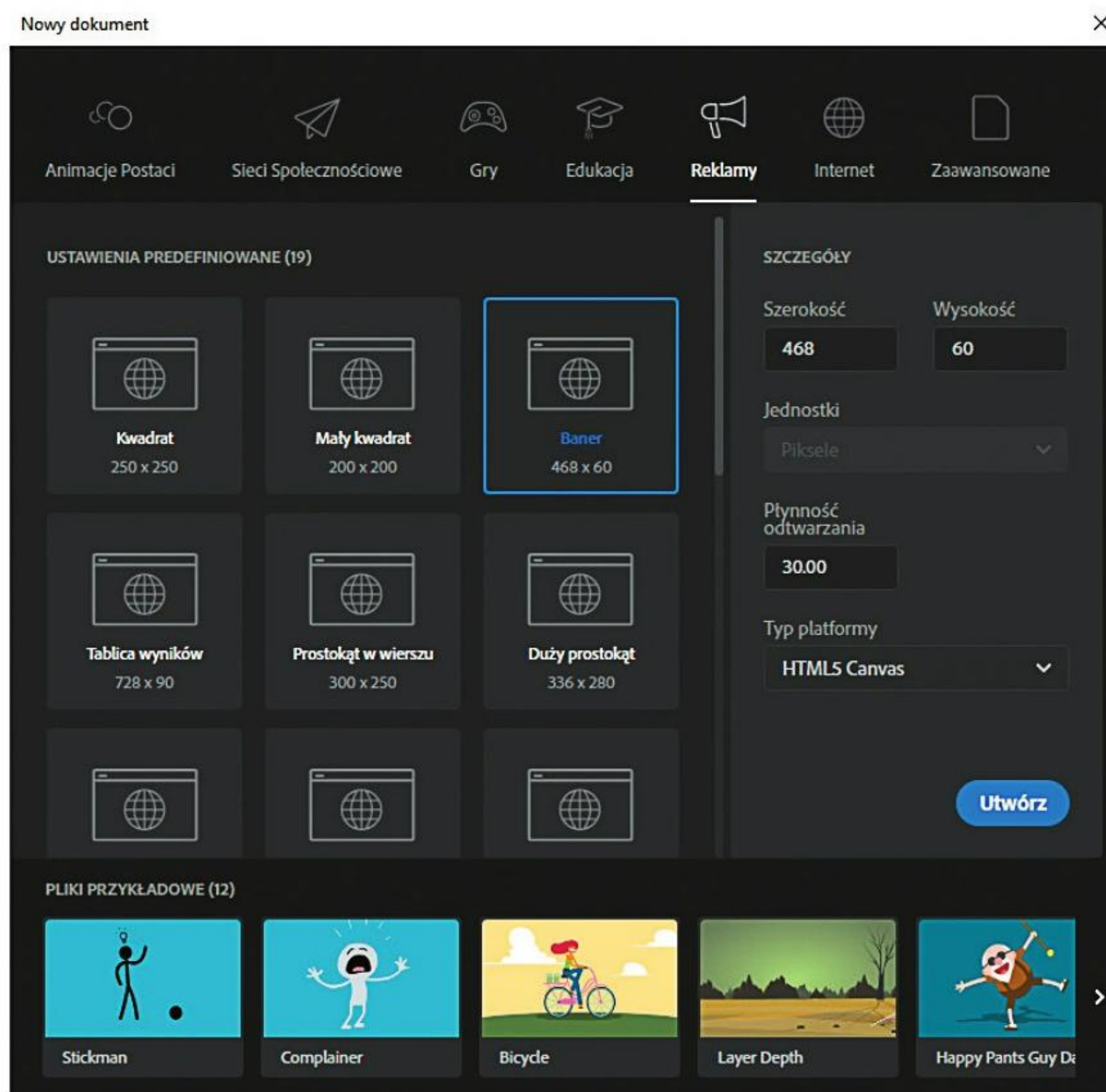
Głównymi elementami okna są obszar roboczy z położonym po lewej stronie przybornikiem oraz umiejscowionym w dolnej części panelem nawigacyjnym oraz oś (lub linia) czasu.

Istotny dla pracy w programie jest panel nawigacyjny. Na rys. 23.11 opisano jego wszystkie elementy.



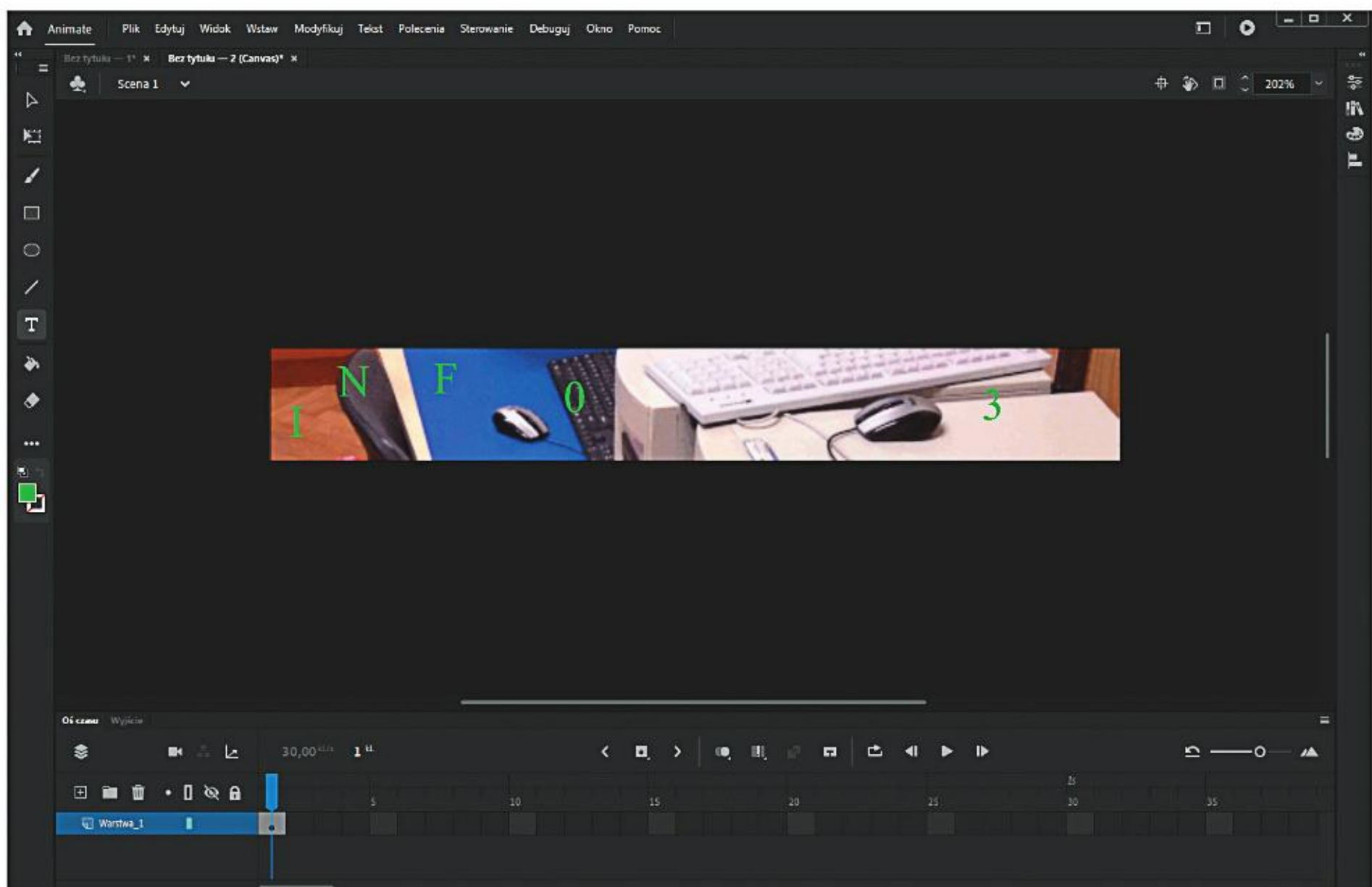
Rys. 23.11. Budowa panelu nawigacyjnego

Otwieramy nowy dokument i wybieramy z wyświetlonych opcji baner reklamowy o wymiarach 468 × 60 px. Ustalamy w tym miejscu płynność odtwarzania oraz platformę, na której baner będzie pracował (rys. 23.12).



Rys. 23.12. Okno tworzenia nowej animacji

W głównym oknie pojawi się utworzony obszar roboczy, a wraz z nim – aktywna pierwsza ramka kluczowa. To, co teraz stworzymy, będzie statycznym rysunkiem w tej ramce. Do utworzonego obszaru roboczego importujemy przygotowany wcześniej rysunek, który będzie odgrywał rolę tła. Pliki tworzone w programach Adobe Illustrator i Adobe Photoshop można importować bezpośrednio do programu Animate. Osobno edytujemy litery INF i cyfry 0 oraz 3: ustawiamy je w różnych miejscach rysunku. Następnie tworzymy nową ramkę – wybieramy przycisk **Wstaw klatkę kluczową** na panelu nawigacyjnym. Wykonane czynności zaprezentowano na rys. 23.13.



Rys. 23.13. Edycja animacji w programie Animate CC

W nowej klatce kluczowej nieznacznie zmieniamy położenie liter i cyfr w taki sposób, aby zbliżały się do siebie. Następnie dodajemy kolejną klatkę kluczową. Proces kończymy, gdy litery i cyfry stworzą jednolity napis INF03.

Wykonaj podobny baner. Pamiętaj, że im więcej klatek, tym animacja będzie bardziej płynna. Pamiętaj również, jaka liczba klatek została zadeklarowana podczas tworzenia projektu. Najlepiej, gdy ta liczba będzie równa liczbie klatek kluczowych na osi czasu.

Po zakończeniu edycji eksportujemy gotowy projekt do postaci animowanego GIF-u. Efektem naszej pracy będzie plik animacji do osadzenia na stronie internetowej (rys. 23.14).

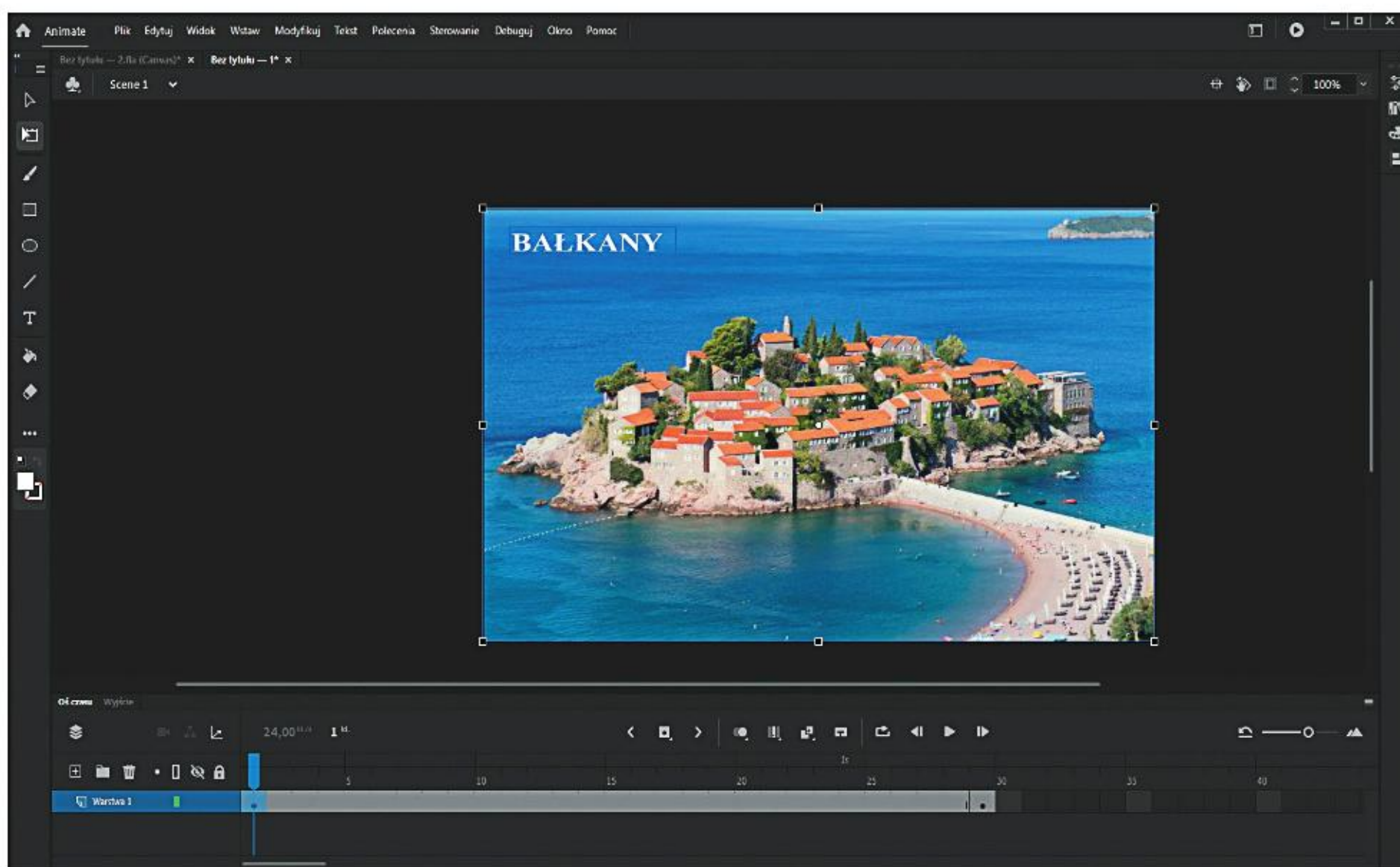


Rys. 23.14. Gotowa animacja na stronę internetową

ĆWICZENIE 2

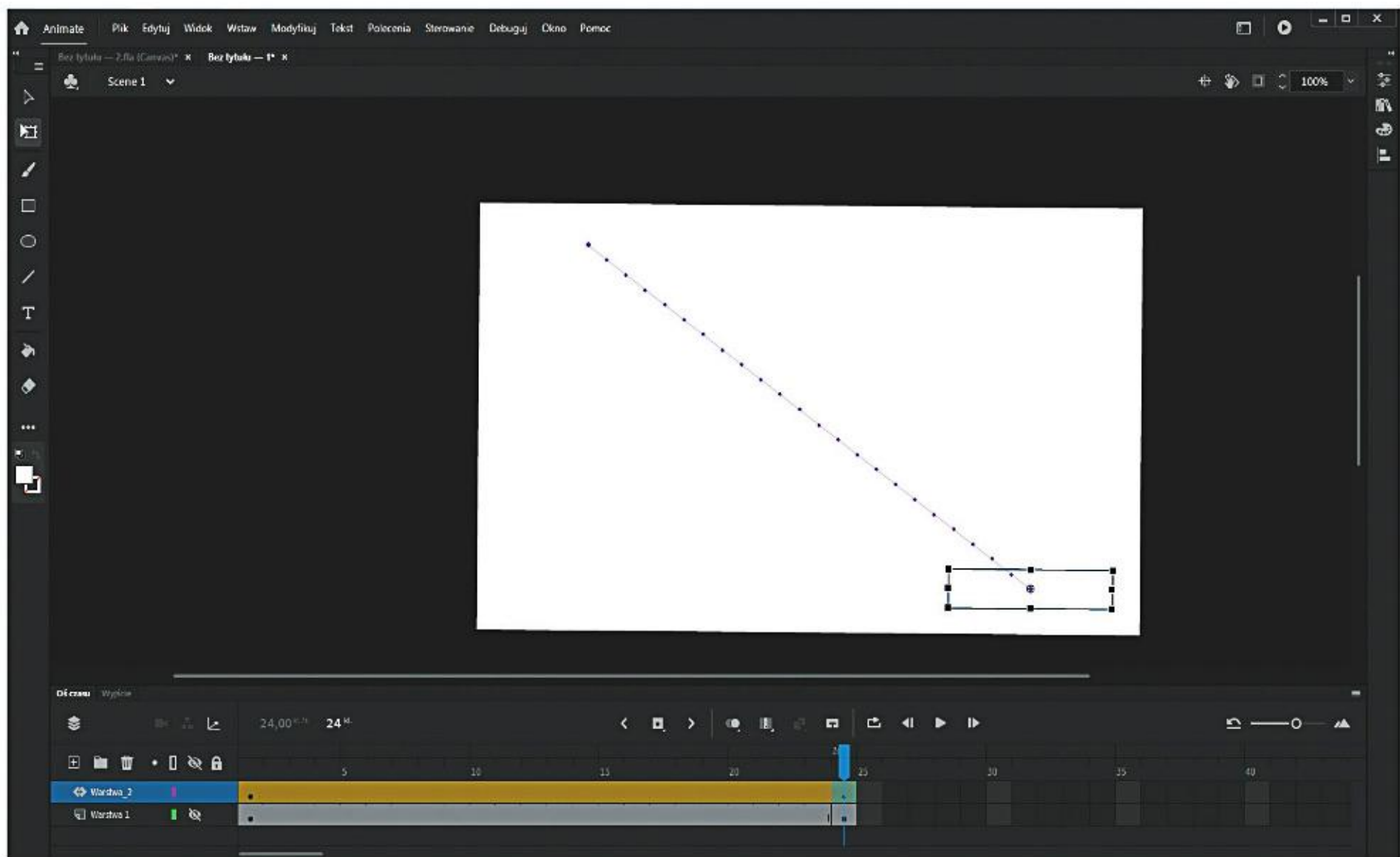
Tworzenie animacji metodą klatek pośrednich w Adobe Animate CC

Metoda ramek pośrednich wykorzystuje dwa rodzaje ramek: pośrednie i kluczowe, warstwy oraz animację ruchu. Utworzymy prostą animację, wykorzystując tę metodę. Przygotowaliśmy zdjęcie, które stanie się tłem naszej reklamy. Importujemy je na stół montażowy. Następnie w górnym lewym rogu edytujemy napis – określamy jego wielkość oraz kolor wypełnienia (rys. 23.15). Zaznaczamy tekst, który będzie naszym obiektem poddawany animacji. Jeżeli zaznaczenie obejmie inne obiekty lub wiele obiektów z danej warstwy, program Animate zaproponuje przekonwertowanie tego zaznaczenia na symbol klipu filmowego.



Rys. 23.15. Początek edycji animacji metodą klatek pośrednich

Następnie uruchamiamy prawym klawiszem myszy menu kontekstowe na pierwszej klatce kluczowej i wybieramy polecenie **Utwórz animację ruchu**. Utworzy się nowa warstwa, po której możemy zaznaczony obiekt przeciągnąć w dowolne miejsce stołu montażowego. Można również przesunąć dowolny koniec zakresu animacji na osi czasu, aby skrócić lub wydłużyć zakres do odpowiedniej liczby klatek. Istniejące klatki kluczowe właściwości zostaną przesunięte proporcjonalnie do przesunięcia końca zakresu (rys. 23.16).

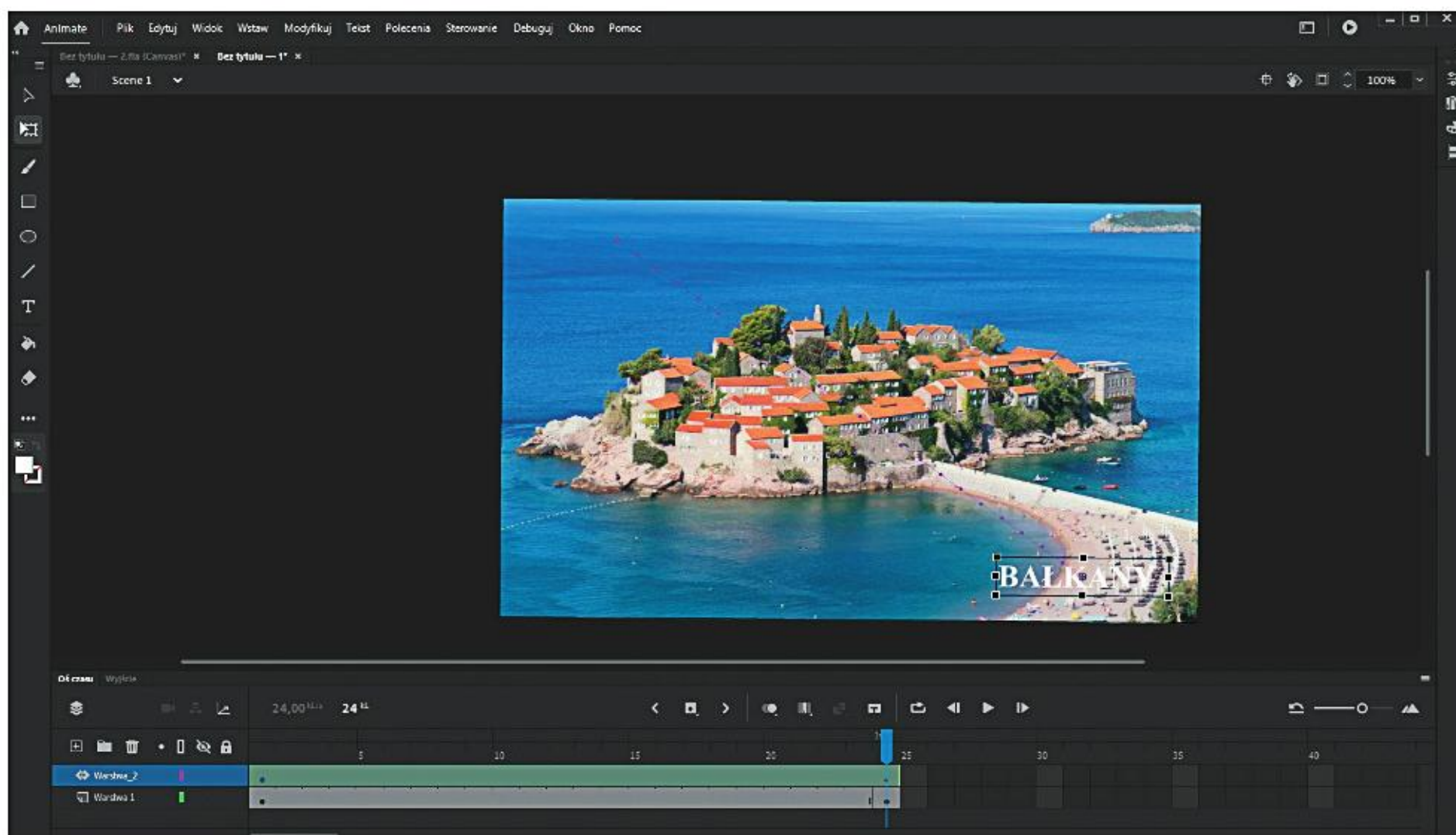


Rys. 23.16. Tworzenie animacji ruchu

Aby przenieść koniec zakresu bez przenoszenia istniejących klatek kluczowych, należy przeciągnąć koniec zakresu, przytrzymując wciśnięty klawisz [Shift].

Po wykonaniu tych czynności na stole montażowym pojawi się ścieżka ruchu biegnąca od położenia w pierwszej klatce zakresu animacji do nowego położenia. W zakresie animacji klatki właściwości są widoczne jako małe romby.

Teraz wystarczy wstawić klatkę kluczową dla warstwy zawierającej tło tak, aby była widoczna przez cały czas trwania animacji obiektu. Po zakończeniu edycji możemy wypróbować nasz projekt za pomocą polecenia **Testuj film** w prawym górnym rogu okna programu.



Rys. 23.17. Gotowa animacja metoda klatek pośrednich

Aby jednocześnie utworzyć wiele animacji, trzeba rozmieścić obiekty przeznaczone do animowania na różnych warstwach, zaznaczyć wszystkie te obiekty i wybrać polecenie **Wstaw – Animacja ruchu**. W ten sam sposób można również stosować ruchy predefiniowane dla wielu obiektów.

Gotową animację można opublikować jako plik SWF lub wyeksportować do animowanego pliku GIF.



SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj animację typu flash za pomocą dowolnej aplikacji, w której narysowany dowolny kwiatek musi się poruszać po spirali. Animacja powinna trwać 10 s.
2. Skorzystaj z dowolnej aplikacji i utwórz animację w postaci banera o wymiarach 600 px na 100 px, spełniającego następujące wytyczne:
 - ustawić tło dla banera w kolorze # e80f0f;
 - animowanym obiektem ma być piłka w kolorze #290fe8, która ma odbijać się od dolnej i górnej krawędzi banera pod kątem ok. 45°: zaczynać od dolnego lewego rogu i kończyć na prawym górnym rogu;
 - animacja ma symulować odbicie piłki od każdej krawędzi w taki sposób, że po dotknięciu do niej obiekt musi być lekko spłaszczony, a następnie wrócić do swojego pierwotnego kształtu;
 - w odbijającym się obiekcie należy wpisać litery oznaczające inicjały autora;
 - obiekt w czasie trwania całej animacji powinien obrócić się o 360°;
 - efekt pracy należy zapisać jako animowany GIF pod nazwą piłka.

3. Skorzystaj z dowolnej animacji i utwórz hero image o wymiarach 1200 × 800 px przedstawiający logo i motto twojej szkoły.
4. Skorzystaj z dowolnej animacji i utwórz animowany GIF imitujący pokaz slajdów złożony z pięciu dowolnych zdjęć promujących zawody technika informatyka i technika programisty.
5. Utwórz animowany baner reklamowy o dowolnych wymiarach, który ma zachęcić internautów do zakupu karty graficznej w promocyjnej cenie.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Jakie znasz metody tworzenia animacji komputerowej?
2. Czym charakteryzuje się animacja GIF?
3. Na czym polega metoda klatek pośrednich?
4. Jakie aplikacje pozwalają na tworzenie animacji SWF?
5. Opisz etapy projektowania i tworzenia animacji na potrzeby stron internetowych.
6. Co to jest Slider?

24

Wideo na stronach internetowych

ZAGADNIENIA

- Metody odtwarzania filmów na stronach WWW
- Standardy obrazu
- Formaty plików wideo

Film na stronie internetowej to już nie ekstrawagancja, lecz konieczność i wymóg marketingu sieciowego. Pierwszym sposobem osadzania plików wideo jest wykorzystanie znaczników HTML. Zalicza się do nich: `<video>` oraz `<embed>`. Dzięki takiej metodzie pokazuje się prosty odtwarzacz umożliwiający nie tylko odtworzenie filmu, lecz także jego pobranie na dysk.

PRZYKŁAD 24.1

```
<video src="plik.mp4" poster="rys.jpg">Film</video>
```

Oprócz wstawiania plików multimedialnych z poziomu kodu HTML, odtwarzanych przez narzędzia przeglądarki, często wykorzystuje się również specjalnie instalowane **kodeki**. Plik wideo jest zazwyczaj skompresowanym formatem zawierającym wiele elementów niezbędnych do odtworzenia filmu. Kodeki pozwalają na sprawne przeprowadzenie procesu kodowania lub dekodowania pliku multimedialnego. Do popularnych kodeków zalicza się: Quick Time Player, Adobe Flash Player i Microsoft Silverlight.

Podczas ładowania pliku o określonym formacie lepiej jest wskazać kodek, w którym zamierza się odtworzyć wideo. Możliwe jest to dzięki zastosowaniu atrybutu **plugin-spase** = "adres URL kodeka". Umożliwia to pobranie i zainstalowanie określonego kodeka.

PRZYKŁAD 24.2

```
<embed type="application/x-mplayer2" src="film.wmv"  
„pluginspage="http://www.microsoft.com/Windows/MediaPlay-  
er/">FILM</embed>
```

Za pomocą atrybutu **codecs** = „nazwa kodeka” można również wskazać wttyczkę, która posłuży jako odtwarzacz do danego pliku.

PRZYKŁAD 24.3

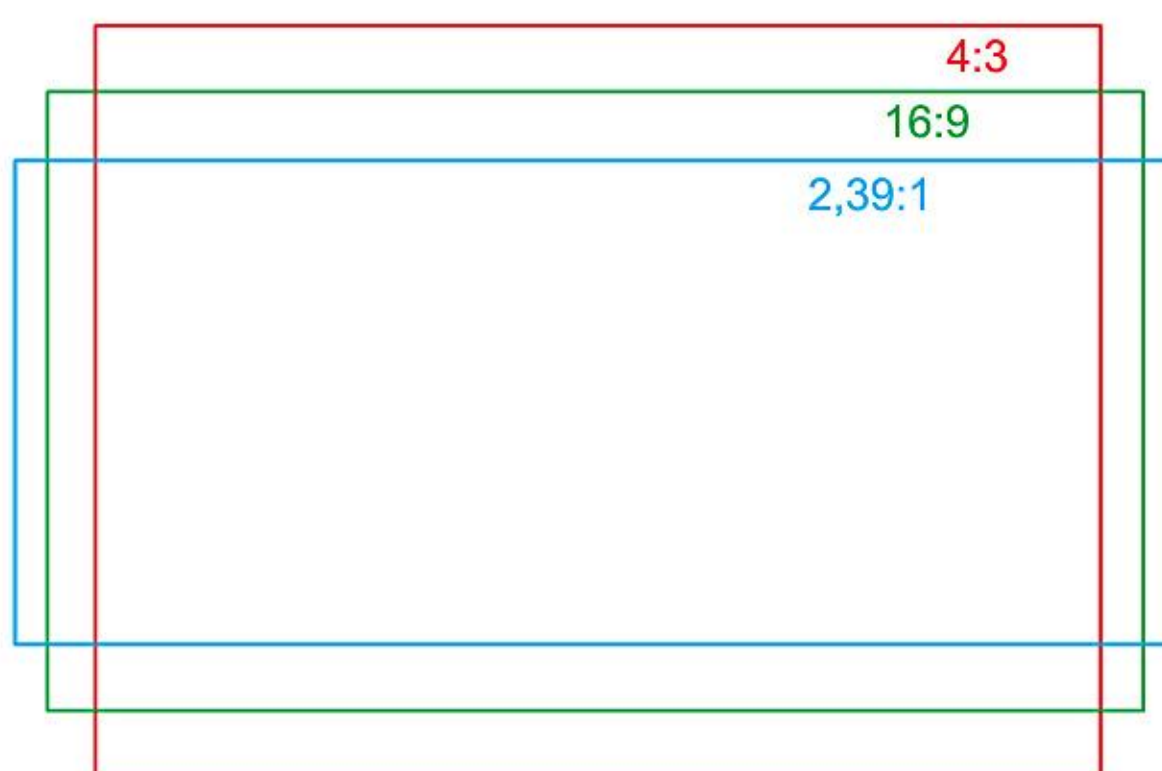
```
<source src="film.mp4" type='video/mp4; codecs="mp4v.20.8, mp4a.40.2"' >
```

Oprócz osadzania plików wideo z zasobów własnego serwera, można też wyświetlić film opublikowany na serwerach publicznych. W ten sposób wstawia się filmy np. z serwisu YouTube, o czym była mowa w rozdziale o języku HTML.

Aby odtworzyć film na stronie internetowej, należy wcześniej przygotować materiał wideo. Za przechowywanie materiałów wideo w postaci cyfrowej są odpowiedzialne odpowiednie formaty plików. Do popularnych formatów plików wideo zalicza się:

- AVI – standardowy format Windows występujący w postaci skompresowanej lub bez kompresji;
- MPEG – format stratnej kompresji obrazu wideo;
- WMV – format kompresji filmów opracowany przez firmę Microsoft;
- FLV – kontenerowy format wideo stworzony przez firmę Adobe; format ten wykorzystują takie serwisy, jak YouTube czy Google Video; zapewnia pełną integrację z projektami Flash;
- Ogg – format kontenera strumieni danych wideo; stworzony, by wspierać inicjatywy związane z rozwojem wolnego oprogramowania do kodowania i dekodowania multimedialnych;
- MOV – format plików wideo firmy Apple.

Istotnym parametrem filmów wideo jest **format obrazu**. Do najczęściej stosowanych formatów obrazów zalicza się 4 : 3 oraz 16 : 9.



Rys. 24.1. Różnice między formatami obrazów

Oprócz formatu istotna jest **rozdzielczość**. Na potrzeby stron internetowych, po upowszechnieniu się formatu 16 : 9, najczęstsze stały się rozdzielczości:

- 1080 p (1920 × 1080, zwyczajowo FullHD) – w tej rozdzielczości należy na urządzenia Desktop publikować materiały, które mają wiele istotnych szczegółów, takich jak tekst, prezentacje, nagrywanie ekranu;

- 720 p (1280 × 720, HD lub HD Ready) – w tej rozdzielczości należy publikować filmy na ekrany Desktop i mobilne, w których poziom szczegółowości nie jest tak istotny;
- 480 p (720 × 480 – DVD) – w tej rozdzielczości można publikować materiały, które nie zawierają tekstu lub szczegółów, których wyrazistość jest kluczowa;
- 360 p (480 × 360) – format przeznaczony na urządzenia mobilne;
- 240 p (352 × 240) – format przeznaczony na urządzenia mobilne, gdy kluczowa jest oszczędność transferu.

Bezpośrednio związanym z rozdzielczością oraz formatem jest **standard jakości obrazu**. Najczęściej stosuje się standardy jakości obrazu:

- VHS (ang. *Video Home System*) – opracowany 1976 r. przez firmę JVC, z rozdzielczością poziomą 240 linii;
- PAL (ang. *Phase Alternating Line*) – stosowany w Europie, z rozdzielczością 704 × 576 px.
- NTSC (ang. *National Television System Committee*) – stosowany w Stanach Zjednoczonych, wyświetlający obraz z rozdzielczością 720 × 480 px;
- HDTV (ang. *High Definition TeleVision*) – z rozdzielczością 1280 × 720 px;
- HDV (ang. *High Definition Video*) – z rozdzielczością 1440 × 1080 px;
- HD (ang. *High Definition*) – z rozdzielczością 1920 × 1080 px;
- UHD TV (ang. *Ultra High Definition Video*) – z rozdzielczością 3840 × 2160 px.

Ostatnim istotnym parametrem wideo jest bit rate – wskaźnik mierzony w liczbie bitów na sekundę (bps). Bit rate to tzw. przepływność lub prędkość transmisji danych, często utożsamiana ze stopniem kompresji obrazu wideo. Im większy bit rate, tym większy rozmiar pliku i szansa, że materiał wideo będzie wysokiej jakości.

Do najczęściej wykonywanych operacji dotyczących przygotowania plików wideo na strony internetowe zalicza się:

- przechwytywanie (ang. *capture*) obrazu wideo z pulpitu lub okna aplikacji;
- konwersja między formatami plików wideo;
- tworzenie krótkich filmów cyfrowych;
- montaż i edycja plików wideo;
- publikowanie plików wideo na stronach internetowych.

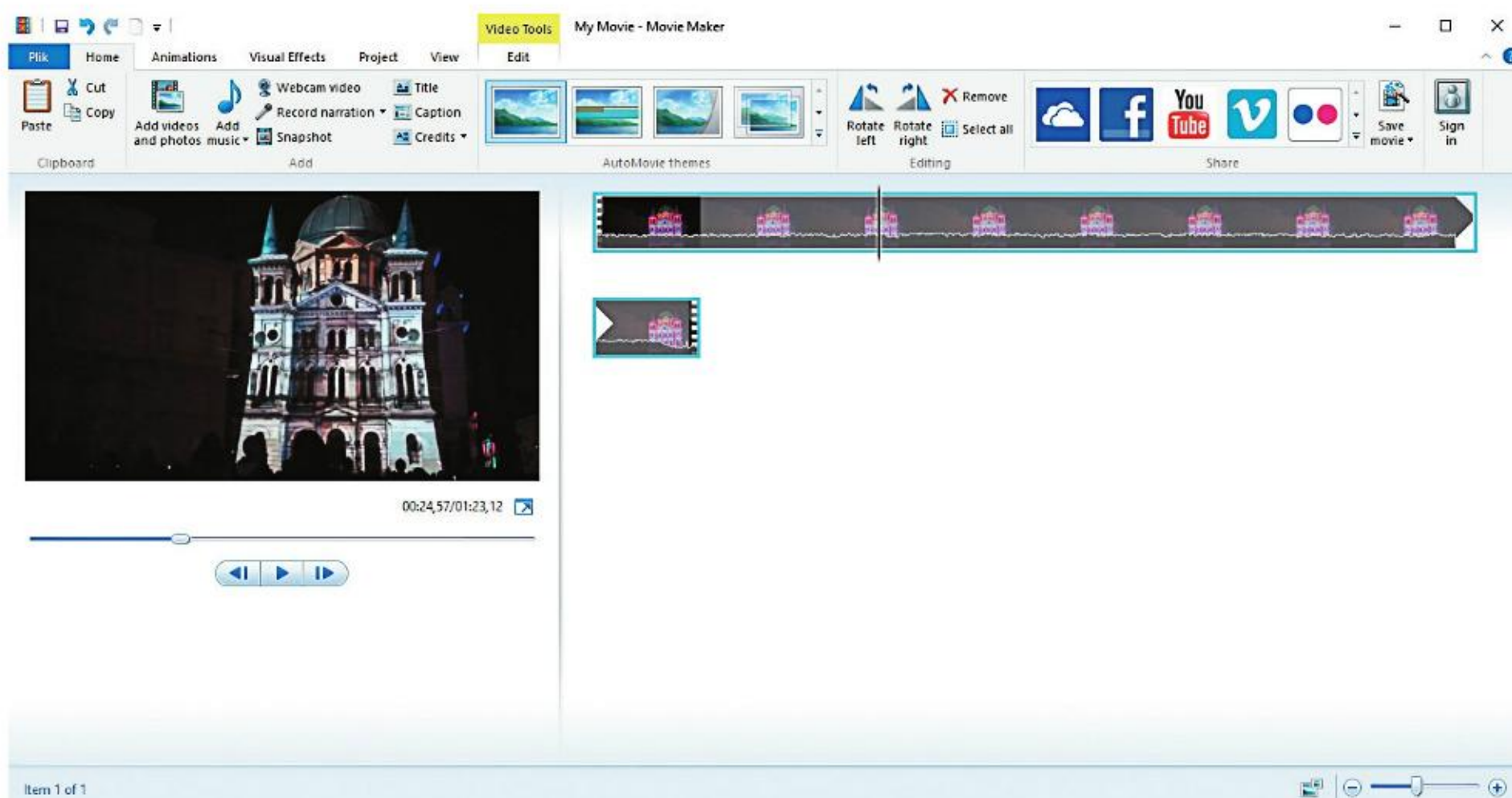
Wśród programów do edytowania plików wideo należy wyróżnić: Corel VideoStudio, Adobe Premiere, Movavi Video Editor lub Windows Movie Maker.

W tym rozdziale zostanie opisana aplikacja Windows Movie Maker (rys. 24.2), której plik instalacyjny można pobrać ze strony <http://www.windows-movie-maker.org>.

Program Windows Movie Maker jest dodatkowym składnikiem systemów MS Windows. Aplikacja oferuje wiele opcji edycyjnych, jak np. przyspieszanie lub spowalnianie nagrań, dodawanie płynnych przejść między jednym a drugim fragmentem czy wstawianie napisów. Doskonale nadaje się do tworzenia prostych filmów, które mogą być publikowane na stronach internetowych.

Po instalacji programu i jego uruchomieniu otworzy się okno robocze z prostym i czytelnym interfejsem użytkownika. Dodanie filmu lub zdjęcia jest możliwe za pomocą opcji

Dodaj wideo i zdjęcia (ang. *Add videos and photos*). Materiał źródłowy można również przechwytywać z kamery internetowej lub kamery telewizji analogowej.



Rys. 24.2. Okno programu Windows Movie Maker

Podczas importu plików źródłowych należy pamiętać, że aplikacja nie obsługuje takich formatów, jak MP4 i MOV. Przed importem tego typu plików trzeba je przekonwertować na inny format.

Zimportowane obiekty można ułożyć w dowolnej kolejności. Za pomocą narzędzi wideo dodaje się napisy początkowe i końcowe, stosuje efekty przejść, obrót zaznaczonych fragmentów bądź dostosowuje szybkość odtwarzania. Jeżeli film zawiera zbędne kadry, można przyciąć określony zakres klatek. Program umożliwia również dodanie podkładu muzycznego lub głosu lektora.

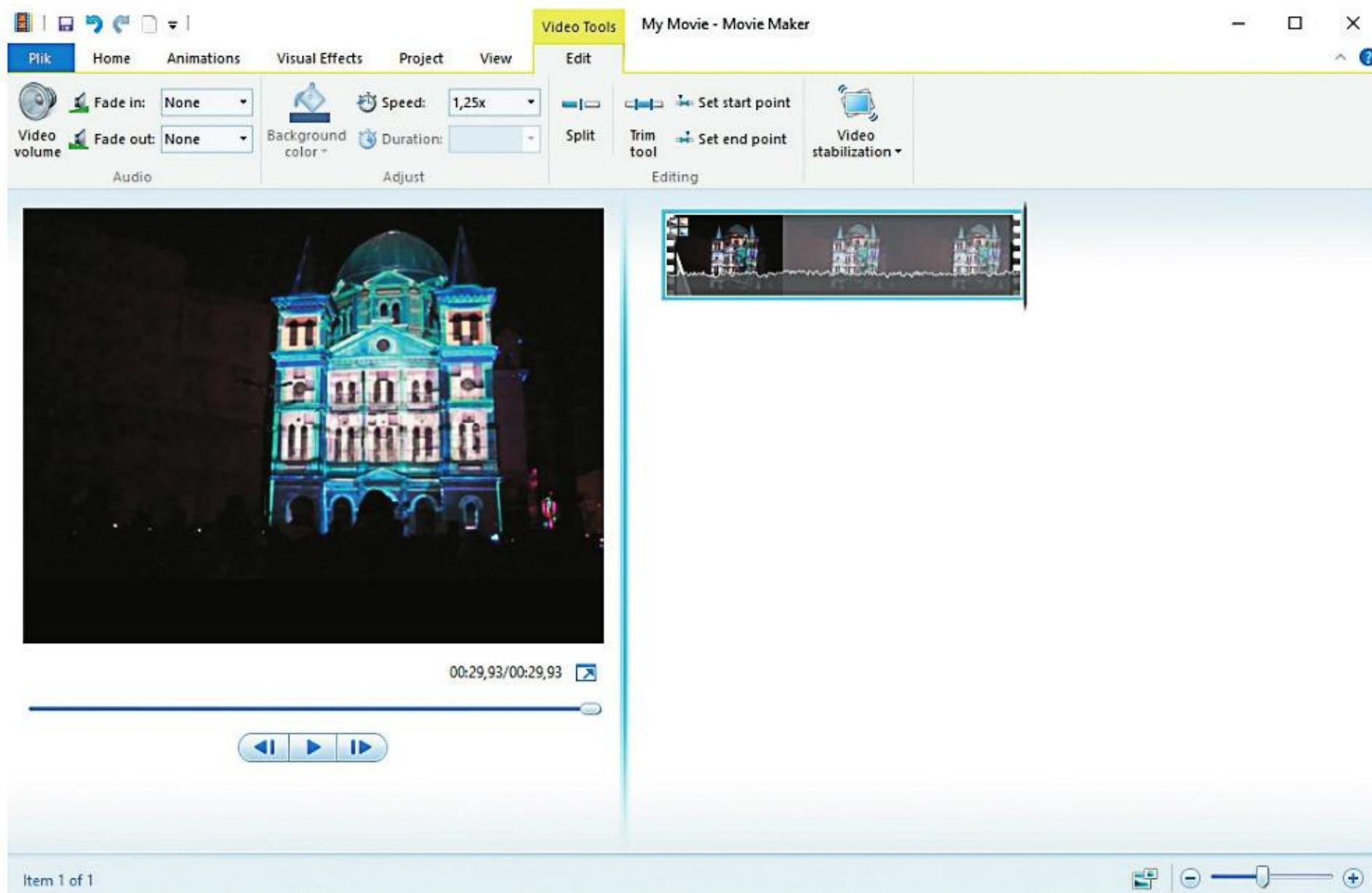
Na koniec utworzony film można opublikować bezpośrednio na serwerach społecznościowych lub zapisać w wybranym formacie.

PRZYKŁAD 24.1

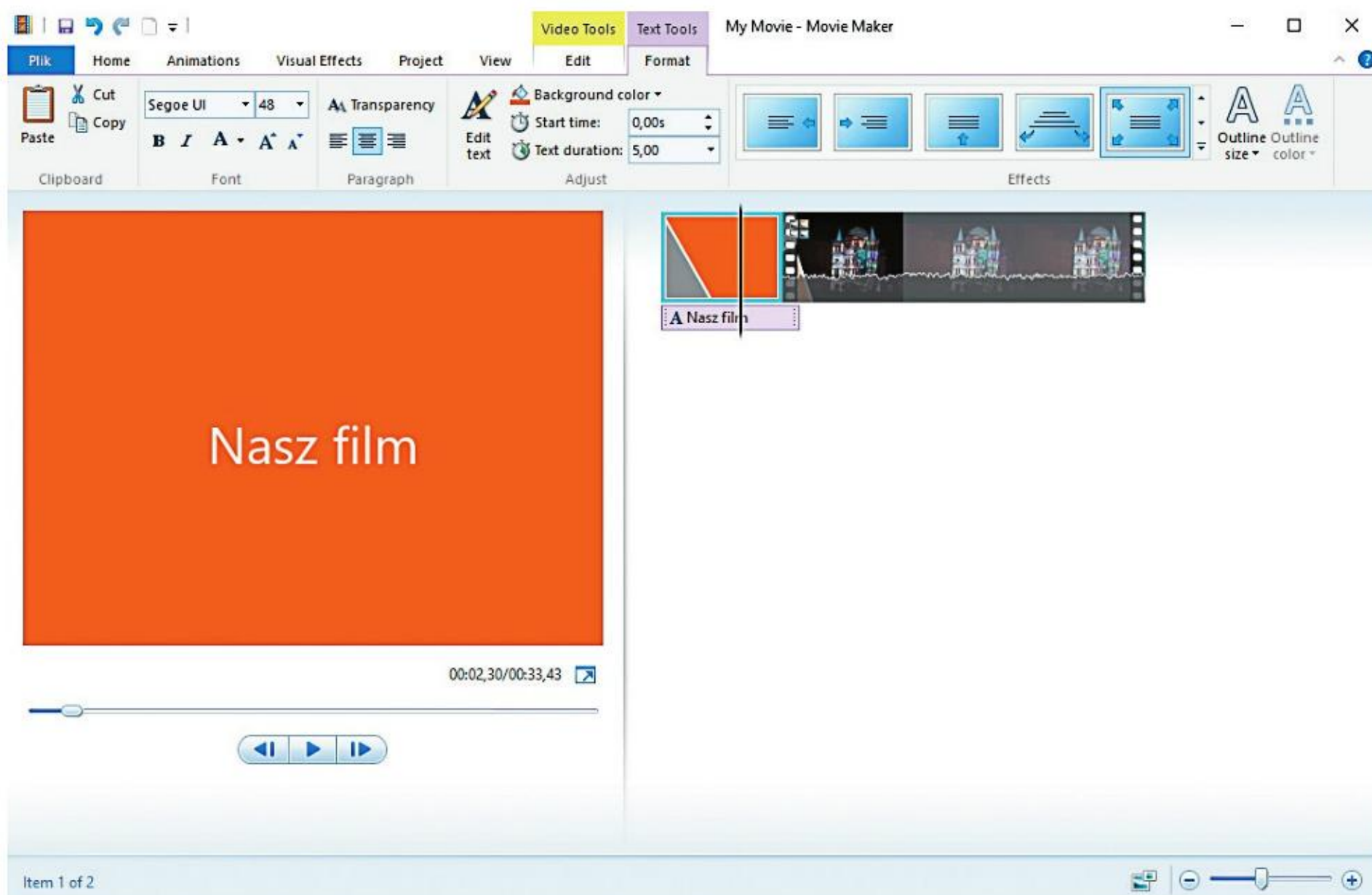
Naszym zadaniem będzie skrócenie filmu do 30 s oraz dodanie napisów początkowych trwających 5 s. Gotowy plik należy zapisać w pliku film.wmv. Do wykonania są potrzebne trzy kroki.

Krok 1. Otwieramy program i wczytujemy plik z materiałem źródłowym. Następnie otwieramy zakładkę **Edycja** i ustalamy punkt początkowy oraz końcowy filmu na poziomie 30 s odtwarzania. Po wskazaniu końcowego punktu film zostanie automatycznie docięty.

Krok 2. Następnie dodajemy napisy początkowe za pomocą narzędzi tekstowych. Edytujemy styl i rodzaj czcionki, kolor tła oraz tekst tytułowy. Na koniec ustalamy czas odtwarzania napisu początkowego na 5 s.



Rys. 24.3. Edycja filmu



Rys. 24.4. Dodawanie efektów do filmu

Krok 3. Końcowy efekt zapisujemy pod nazwą film.wmv. _____

SPRAWDŹ SVOJE UMIEJĘTNOŚCI

1. Za pomocą telefonu komórkowego nagraj krótki film o swojej szkole. Wykonaj montaż przygotowanego materiału za pomocą dowolnego programu do edycji plików wideo. Dodaj do filmu napisy tytułowe – wprowadź swoje imię i nazwisko jako autora filmu. Zastosuj dowolny efekt w trakcie filmu. Wyeksportuj gotowy 30-sekundowy materiał do formatu mov. Utwórz stronę internetową odtwarzającą film za pomocą znacznika **<video>**. Efekt pracy zaprezentuj nauczycielowi.
2. Za pomocą dowolnego programu (np. Corel Video Studio) nagraj film pokazujący, w jaki sposób wykonuje się skalowanie dowolnego obrazu w programie graficznym. Przygotuj z nagrania plik będący samouczkiem do zamieszczenia na stronie internetowej. Dodaj napisy początkowe i końcowe oraz dowolny podkład dźwiękowy. Czas trwania wideo nie może przekroczyć 3 min.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Wymień nazwy popularnych kodeków.
2. Jakie zastosowanie mają kodeki?
3. Wymień i scharakteryzuj formaty plików wideo.
4. Opisz najczęściej stosowane standardy jakości obrazu wideo.
5. Opisz różnice między formatami obrazu.

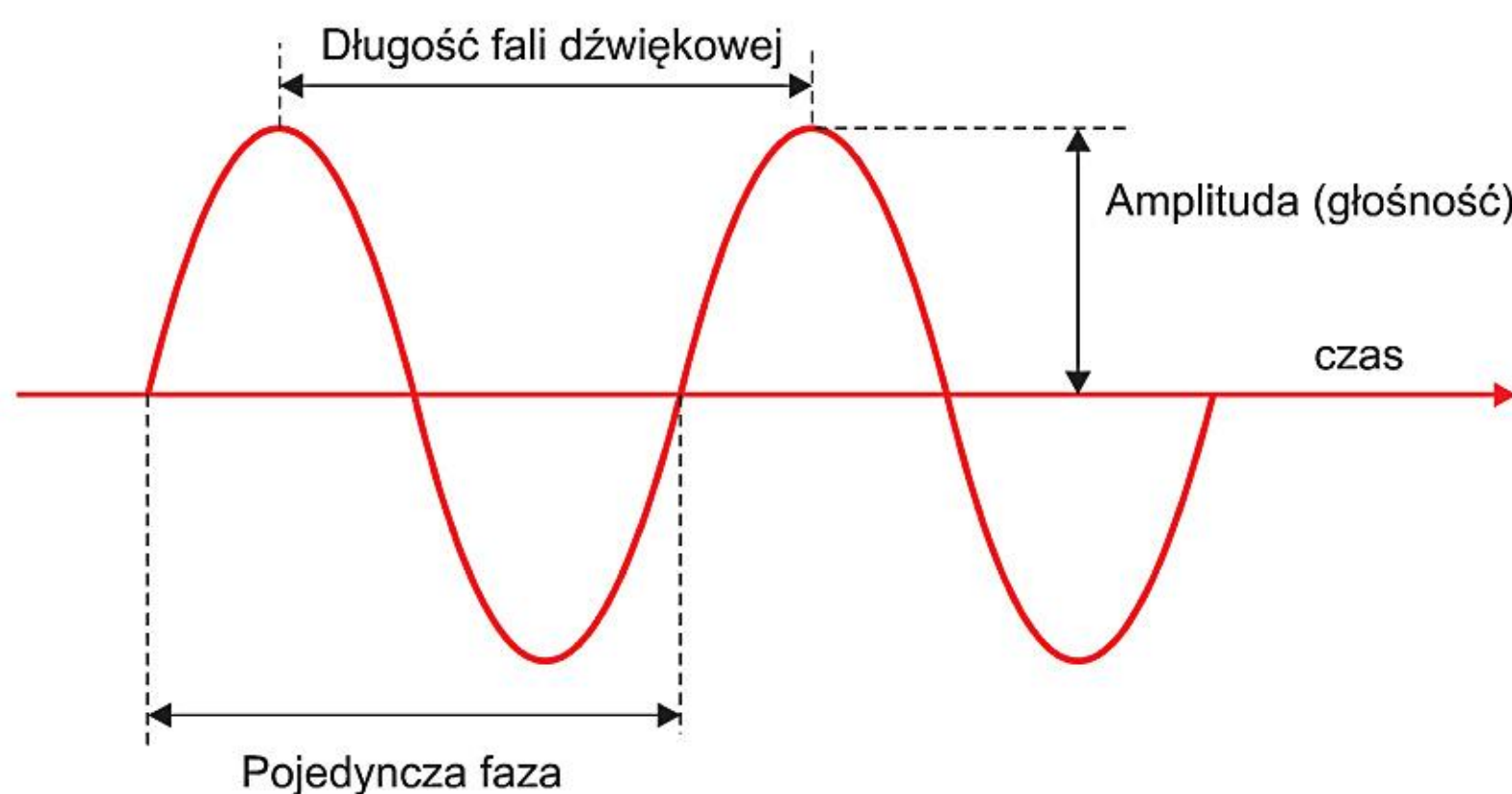
25

Dźwięk na stronach internetowych

ZAGADNIENIA

- Definicja dźwięku
- Formaty plików dźwiękowych
- Edycja ścieżki dźwiękowej

Dźwięk to wrażenie słuchowe wywołane przez falę akustyczną rozchodzącą się w wybranym środowisku (np. w powietrzu). Dźwięk ma swoje: długość fali, częstotliwość i intensywność. Ludzki narząd słuchu wyłapuje dźwięki o częstotliwości fal z przedziału od 16 Hz do 20 kHz.



Rys. 25.1. Wykres fali dźwiękowej

Dźwięk można zapisać w formie cyfrowej. Odwzorowanie fali następuje przez dokonywanie pomiarów jej natężenia wiele razy w ciągu sekundy. Takie pomiary nazywają się próbkowaniem (ang. *sampling*).

Do podstawowych parametrów dźwięku cyfrowego zalicza się:

- częstotliwość próbkowania sygnału dźwiękowego – ma bezpośredni związek z maksymalną częstotliwością dźwięku, jaką można zarejestrować i odtworzyć;
- rozdzielczość bitową dźwięku;
- przepływność danych (bit rate) – liczba informacji w kilobitach przesyłanych w jednostce czasu.

W procesie przetwarzania dźwięku analogowego na cyfrowy rozróżnia się następujące etapy:

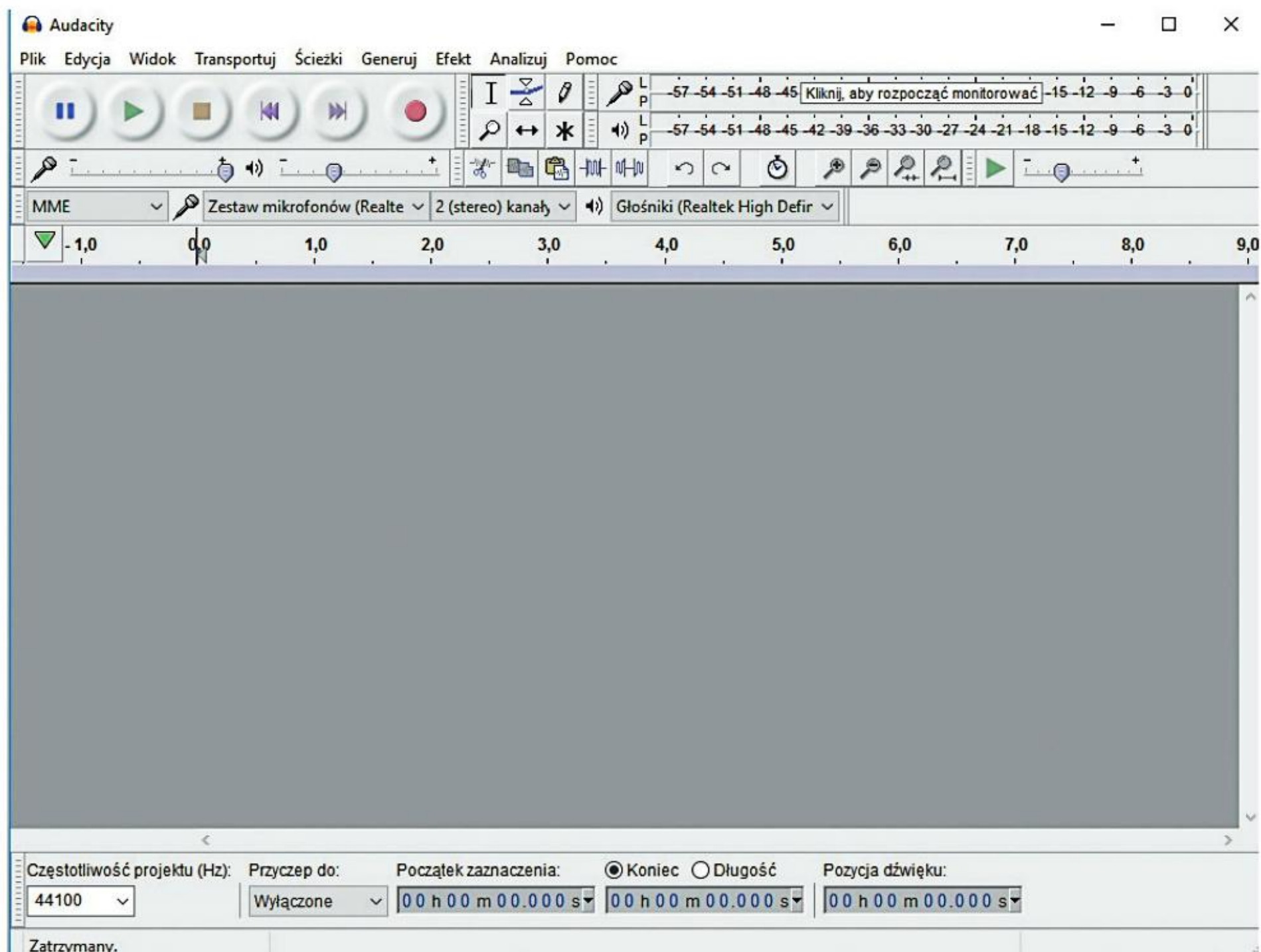
- próbkowanie – prowadzenie z określoną częstością pomiarów amplitudy sygnału dźwiękowego;
- kwantyzacja – przyporządkowanie każdej wartości analogowej z próbkowania sygnału wartości dyskretnych z określonego zbioru;
- kodowanie – zapisanie otrzymanych kodów w formacie cyfrowym, zrozumiałym dla komputera;
- kompresja – zmiana zapisu danych dotyczących sygnału dźwiękowego prowadząca do zmniejszenia rozmiaru pliku; można uzyskać kompresje stratną (usuającą mniej istotne informacje) oraz bezstratną;
- konwersja – zmiana formatu zapisu danych sygnału dźwiękowego.

Jeżeli zamierza się wykorzystać dźwięki na stronie internetowej, to trzeba przygotować określone pliki dźwiękowe. Zależnie od zastosowanej metody osadzenia dźwięku oraz rodzaju przeglądarki internetowej wybiera się odpowiedni format pliku. Do najpopularniejszych formatów plików dźwiękowych zalicza się:

- CD-audio – format audio płyt kompaktowych; częstotliwość próbkowania wynosi 44,1 kHz;
- MIDI – wielościeżkowy zapis dźwięku o niskiej jakości;
- MP3 – format pliku do dużej kompresji; kodowanie MP3 korzysta ze standardu MPEG-I;
- WMA – format pliku o kodowaniu opracowanym przez firmę Microsoft;
- WAV – jeden z prostszych formatów plików dźwiękowych, opracowany jako podstawowy format Microsoft Windows;
- AIFF – dane audio są zapisane w formie nieskompresowanej; wykorzystywany głównie na komputerach Macintosh firmy Apple.

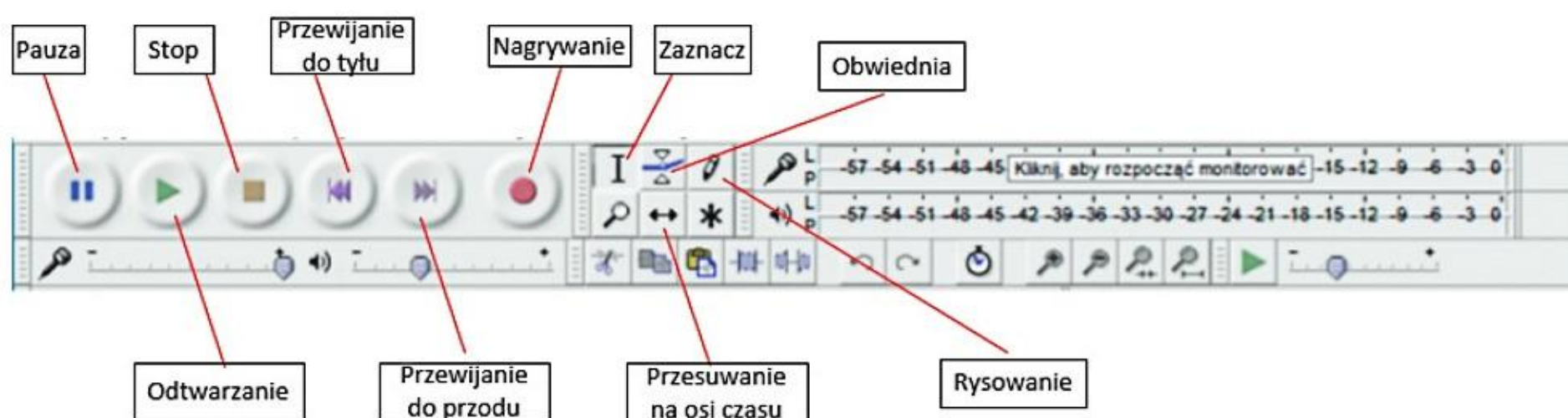
Rodzaje metod oraz przykłady osadzania dźwięku na stronach internetowych opisano w rozdziale dotyczącym znaczników HTML. W tym rozdziale omówimy przygotowanie materiału dźwiękowego.

Jednym z najpopularniejszych darmowych programów do obróbki dźwięku jest Audacity (rys. 25.2). Aplikację można pobrać ze strony internetowej pod adresem <http://www.audacityteam.org/download/>.



Rys. 25.2. Okno programu Audacity

Audacity umożliwia edycję ścieżek dźwiękowych z wykorzystaniem wycinania, kopiowania i wklejania oraz odszumianie. Można również realizować mastering utworów za pomocą wbudowanych filtrów i efektów (echo, usuwanie szumu, normalizacja). Dodatkowo w programie przeprowadza się konwersję formatów dźwiękowych. Ważnym obszarem okna jest górny przybornik. Na rys. 25.3 opisano jego główne elementy.

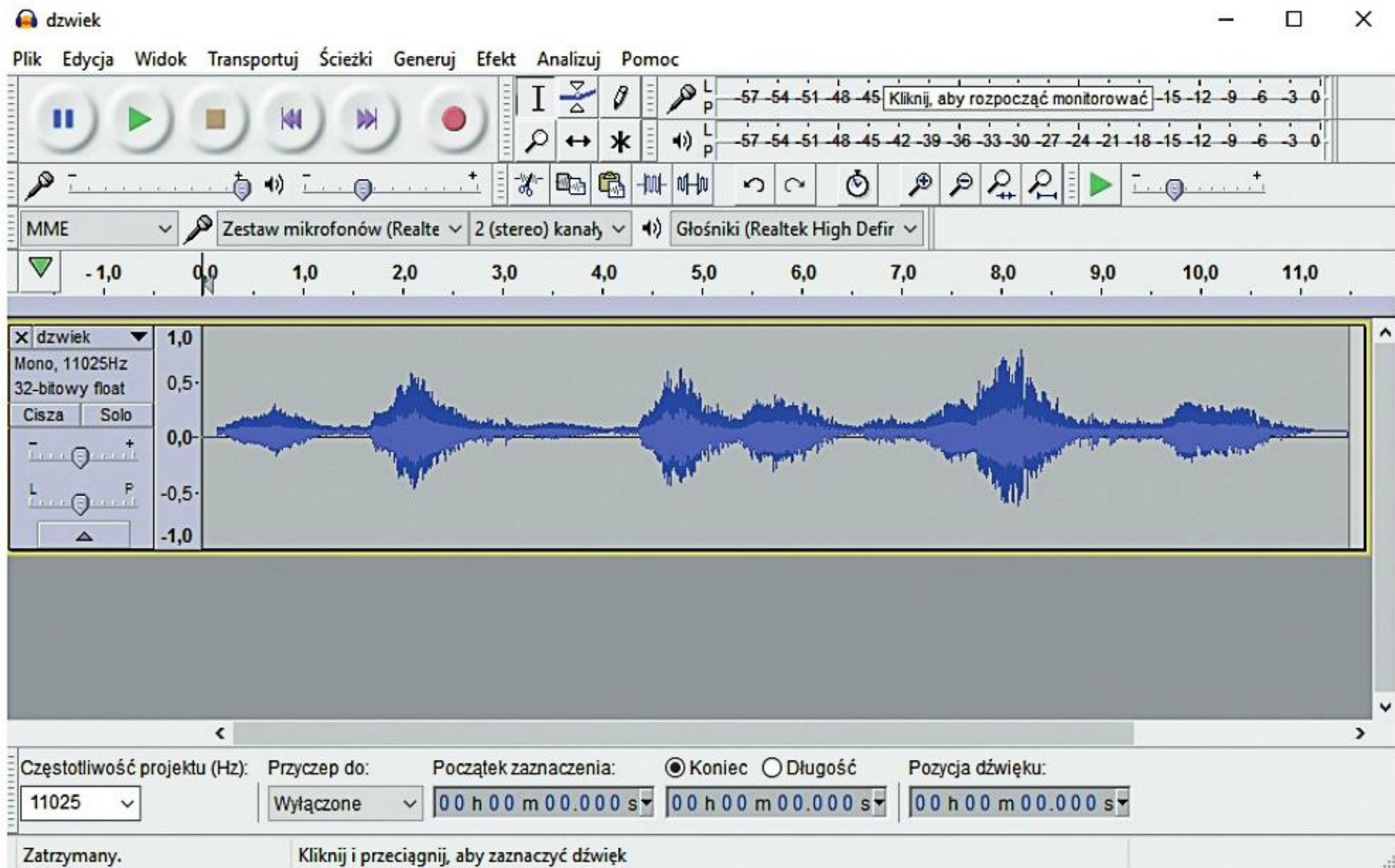


Rys. 25.3. Budowa górnego przybornika

Edycję rozpoczyna się po otwarciu lub importowaniu gotowego pliku dźwiękowego oraz utworzeniu nowego pliku z wykorzystaniem możliwości nagrywania ścieżki dźwiękowej z mikrofonu bądź miksera.

PRZYKŁAD 25.1

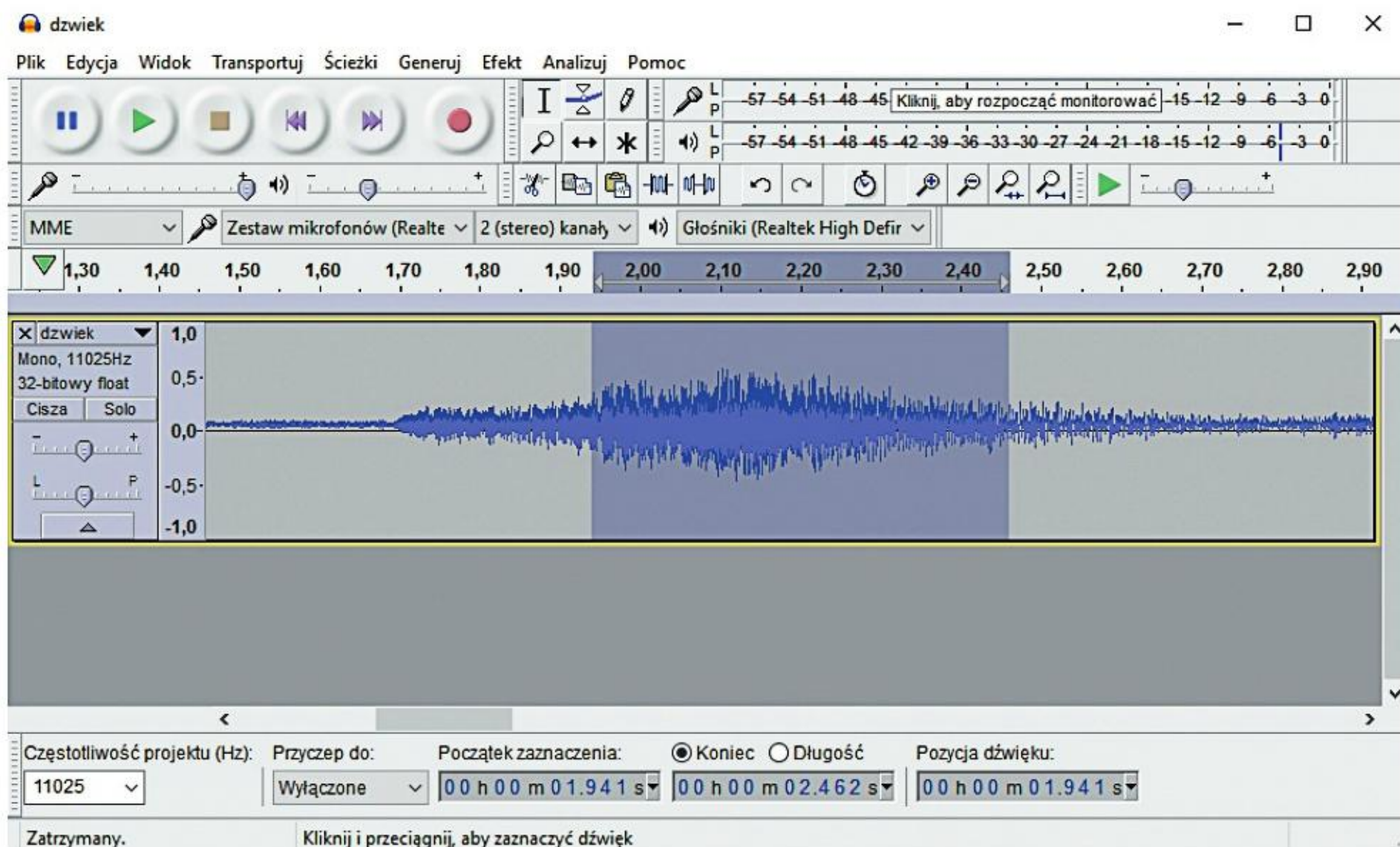
Aby wykonać obróbkę dźwięku do wykorzystania na stronie internetowej, na początek wczytujemy określony plik źródłowy za pomocą polecenia **Otwórz** z menu **Plik**. Po wczytaniu pliku w oknie programu pojawi się przebieg czasowy dźwięku, który zostanie poddany obróbce (rys. 25.4).



Rys. 25.4. Wczytany plik dźwiękowy

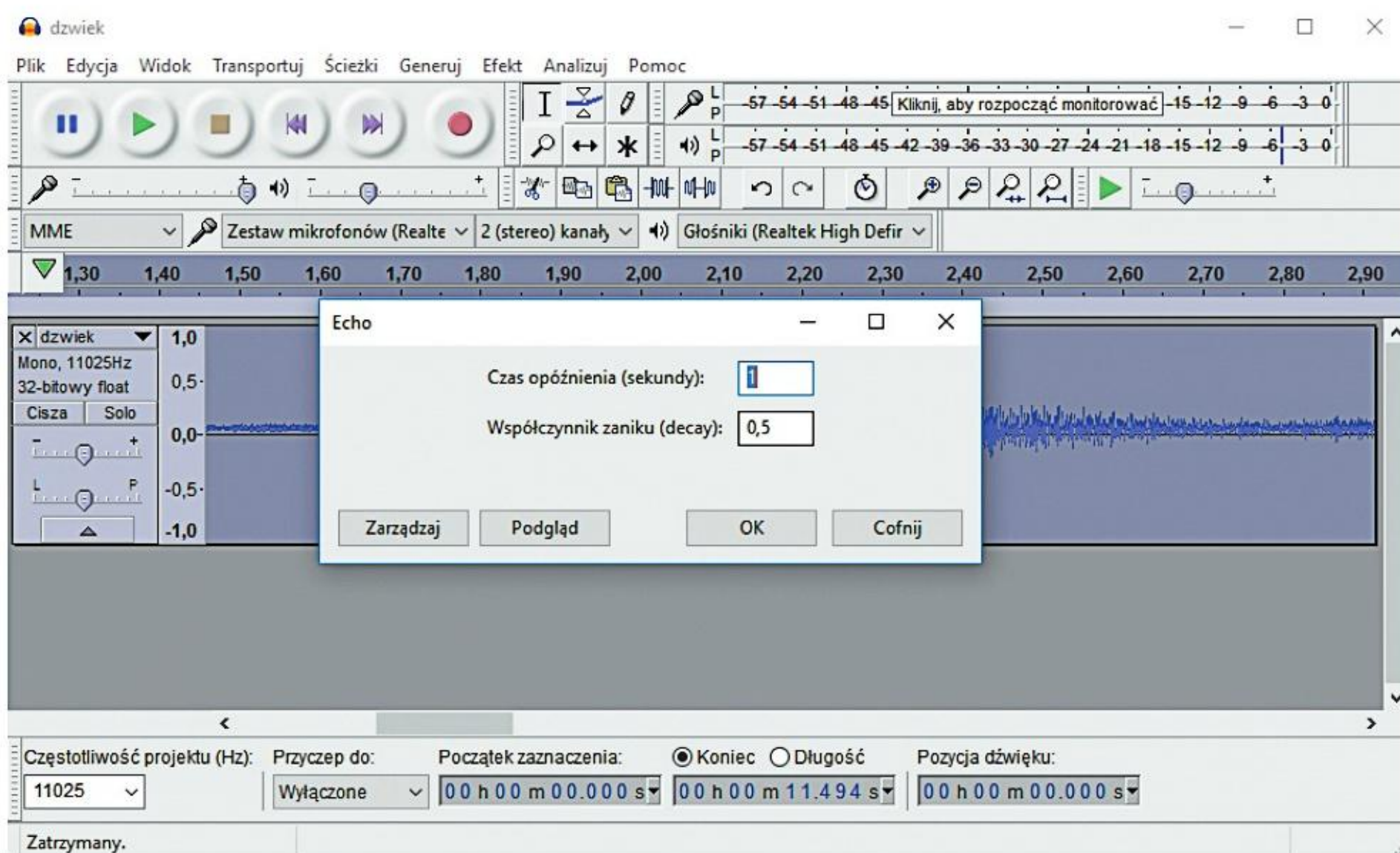
W górnym lewym rogu okna programu znajdują się przyciski sterujące odtwarzaniem dźwięku. Powyżej nich jest widoczne menu główne programu. Oś czasu można sterować, dzięki czemu łatwiej pracuje się nad szczegółowymi elementami utworu.

Do edycji fragmentu utworu używa się narzędzia zaznaczania znajdującego się po lewej stronie przycisku nagrywania. Zaznaczony fragment można wyciąć, powiększyć lub zastosować na nim dowolny efekt.



Rys. 25.6. Zaznaczenie fragmentu utworu

Przykładem stosowania efektu może być dodanie echa do zaznaczonego fragmentu utworu. W tym celu należy wybrać z menu **Efekt** opcję **Echo...** oraz ustawić parametry dotyczące jego działania. Po zatwierdzeniu zmian program przetworzy ścieżkę dźwiękową i doda pogłos dla zaznaczonego efektu.



Rys. 25.7. Efekt Echo

Po wykonaniu wszystkich zaplanowanych czynności zapisujemy efekt pracy i eksportujemy go do wybranego formatu dźwiękowego za pomocą menu **Plik**. Tak zapisany plik dźwiękowy jest gotowy do publikacji na dowolnej stronie internetowej.



SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Zainstaluj program Audacity. Przygotuj do pracy komputer: podłącz słuchawki i mikrofon. Za pomocą przycisków nawigacyjnych programu nagraj swój wokal w dowolnej aranżacji ulubionego utworu muzycznego. Wybierz najlepszy, twoim zdaniem, 10-sekundowy fragment nagrania i usuń pozostałe części. W pozostałym fragmencie usuń szumy. Następnie dla ostatnich 5 s zastosuj efekt echa i wyciszenie. Ścieżkę dźwiękową wyeksportuj do pliku z rozszerzeniem wav. Tak przygotowany plik opublikuj na stronie WWW za pomocą znacznika **<audio>**. Końcowy efekt swojej pracy zaprezentuj nauczycielowi.
2. Za pomocą dowolnego programu do obróbki dźwięku nagraj głos lektora. Wykorzystaj go jako ścieżkę dźwiękową do filmu reklamowego o swojej szkole utworzonego w ćwiczeniu w poprzednim rozdziale.



SPRAWDŹ SWOJĄ WIEDZĘ

1. Co to jest częstotliwość próbkowania i w jakim procesie odgrywa zasadniczą rolę?
2. Scharakteryzuj najważniejsze parametry dźwięku.
3. Wymień etapy przetwarzania dźwięku analogowego na cyfrowy.
4. Scharakteryzuj format MP3.

VIII. Edytory WYSIWYG

- Definicja i rodzaje edytorów WYSIWYG
- Tworzenie dokumentów w edytorze WYSIWYG

26

Rodzaje edytorów
WYSIWYG

ZAGADNIENIA

- Specyfika edytorów WYSIWYG
- Rodzaje edytorów WYSIWYG

Edycja dokumentów hipertekstowych wymaga od pracownika branży informatycznej zarówno wiedzy na temat funkcji poszczególnych elementów, jak i wyobraźni dotyczącej efektu ich działania w oknie przeglądarki internetowej. Tekstowe edytory kodu HTML i CSS pomagają właściwie edytować składnię dokumentu lub wskazać drobne błędy zapisu. Nie umożliwiają jednak wizualizacji pracy programisty. Inaczej działają edytory WYSIWYG.

Akronim **WYSIWYG** pochodzi od słów *What You See Is What You Get*, co w wolnym tłumaczeniu oznacza, że to, co widać podczas edycji dokumentu, będzie widoczne podczas wyświetlenia gotowego pliku w oknie przeglądarki internetowej.

Z aplikacji spełniających założenia WYSIWYG można korzystać w różny sposób. Są bowiem dostępne edytory instalowane w systemie komputerowym jako oprogramowanie desktopowe, można edytować kody źródłowe online, istnieją również edytory WYSIWYG jako część systemu zarządzania treścią (CMS), do edytowania dokumentów w trybie online.

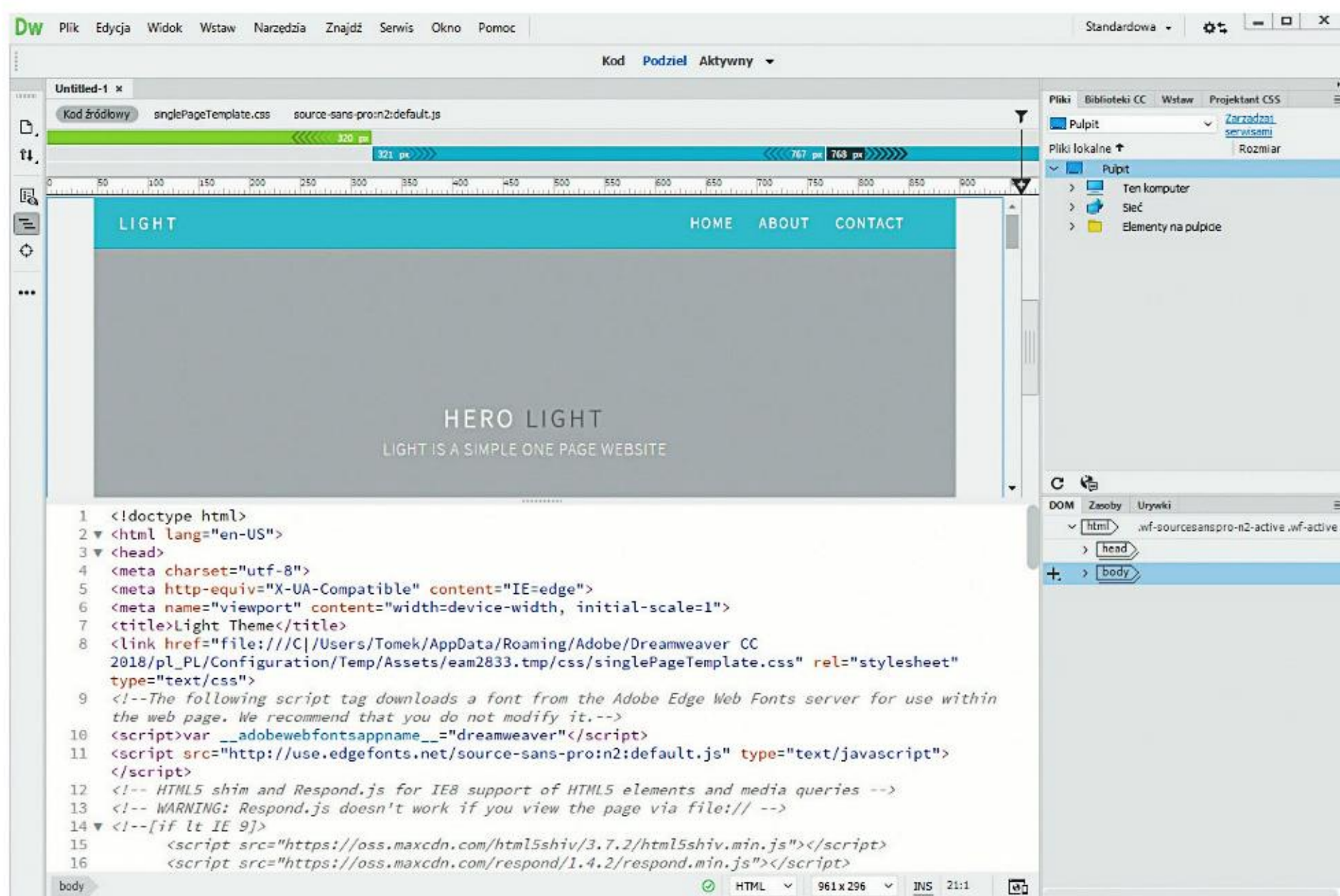
Do podstawowych funkcji edytorów WYSIWYG zalicza się:

- szybką edycję i formatowanie tekstu dokumentu w formie tekstowej oraz z poziomu znaczników języka html;
- intuicyjne wstawianie i pozycjonowanie obiektów graficznych, tabel i odsyłaczy;
- natychmiastową wizualizację efektów pracy webmastera.

Do najbardziej znanych edytorów WYSIWYG zalicza się:

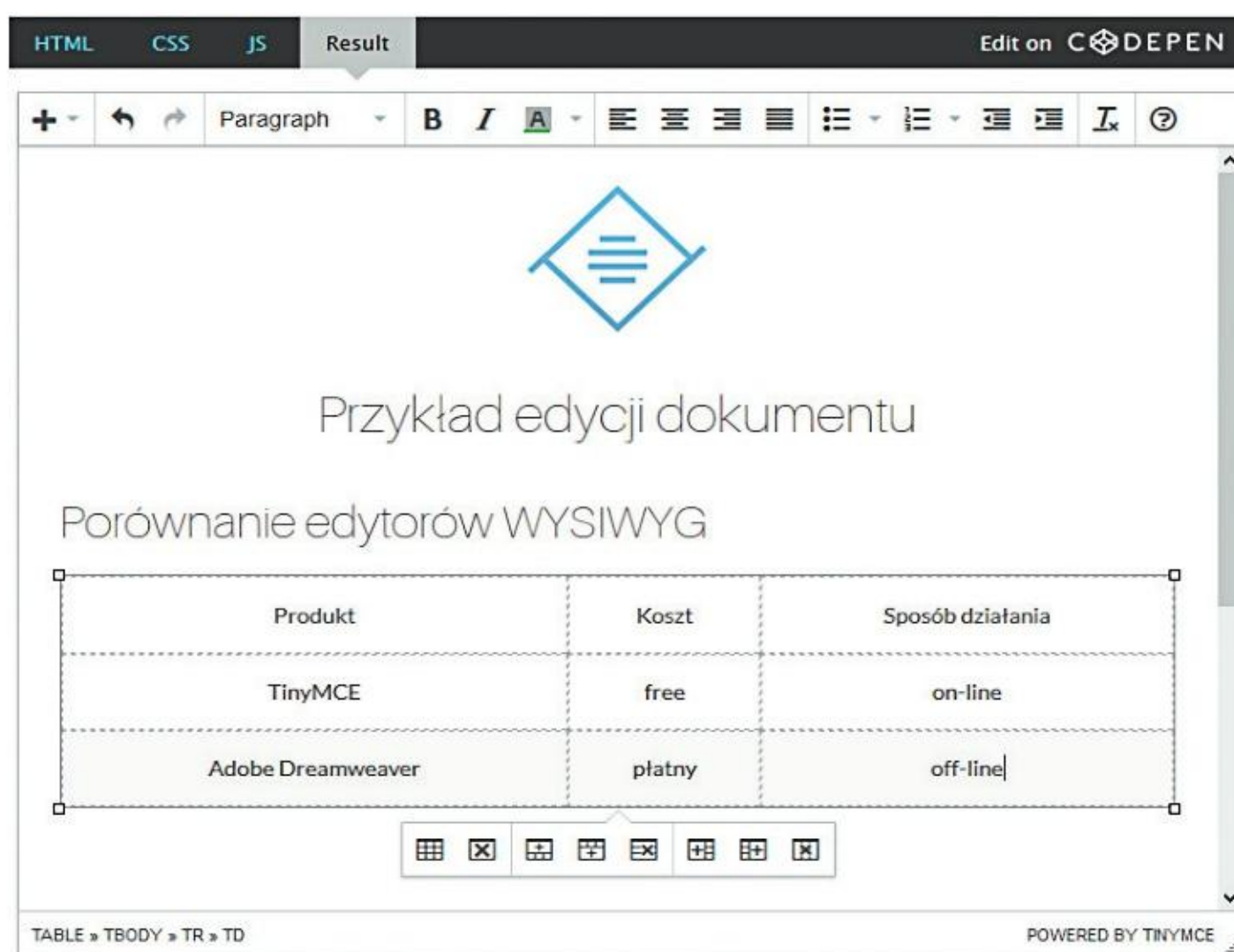
- Adobe Dreamweaver,
- WYSIWYG Web Bulider,
- MS FrontPage,
- CoffeeCup HTML Editor,
- TinyMCE.

Edytor **Adobe Dreamweaver** jest płatną aplikacją do edycji stron internetowych. Można skorzystać z jego darmowej wersji próbnej, żeby przekonać się o walorach programu przed jego zakupem. Plik instalacyjny pobiera się ze strony <http://www.adobe.com/pl/products/dreamweaver.html>. Jest to najbardziej ceniony edytor offline spełniający wymogi WYSIWYG.



Rys. 26.1. Okno programu Adobe Dreamweaver CC

Spośród aplikacji online zostanie opisany edytor **TinyMCE**, który jest klasycznym przykładem aplikacji skryptowej do wykorzystania na stronach internetowych lub w systemach CMS. Strona domowa tego projektu znajduje się pod adresem <https://www.tinymce.com>.



Rys. 26.2. Okno aplikacji TinyMCE

SPRAWDŹ SVOJE UMIEJĘTNOŚCI

1. Sprawdź, który edytor ma CMS WordPress oraz czy spełnia on zasady WYSIWYG.
2. Pobierz ze strony <https://www.coffeecup.com/html-editor/> wersję próbną programu CoffeeCup HTML Editor oraz ją zainstaluj. Opisz jego funkcje i przydatność w edycji stron internetowych. We wnioskach porównaj go z innym edytorem WYSIWYG.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Czym powinny charakteryzować się edytory spełniające wymagania WYSIWYG?
2. Wyszukaj w zasobach sieciowych edytory spełniające wymagania WYSIWYG, a nieopisane w podręczniku. Porównaj je i napisz, który – twoim zdaniem – jest najlepszy.
3. Czy notatnik systemowy można zaliczyć do edytorów spełniających zasady WYSIWYG? Uzasadnij odpowiedź.

27

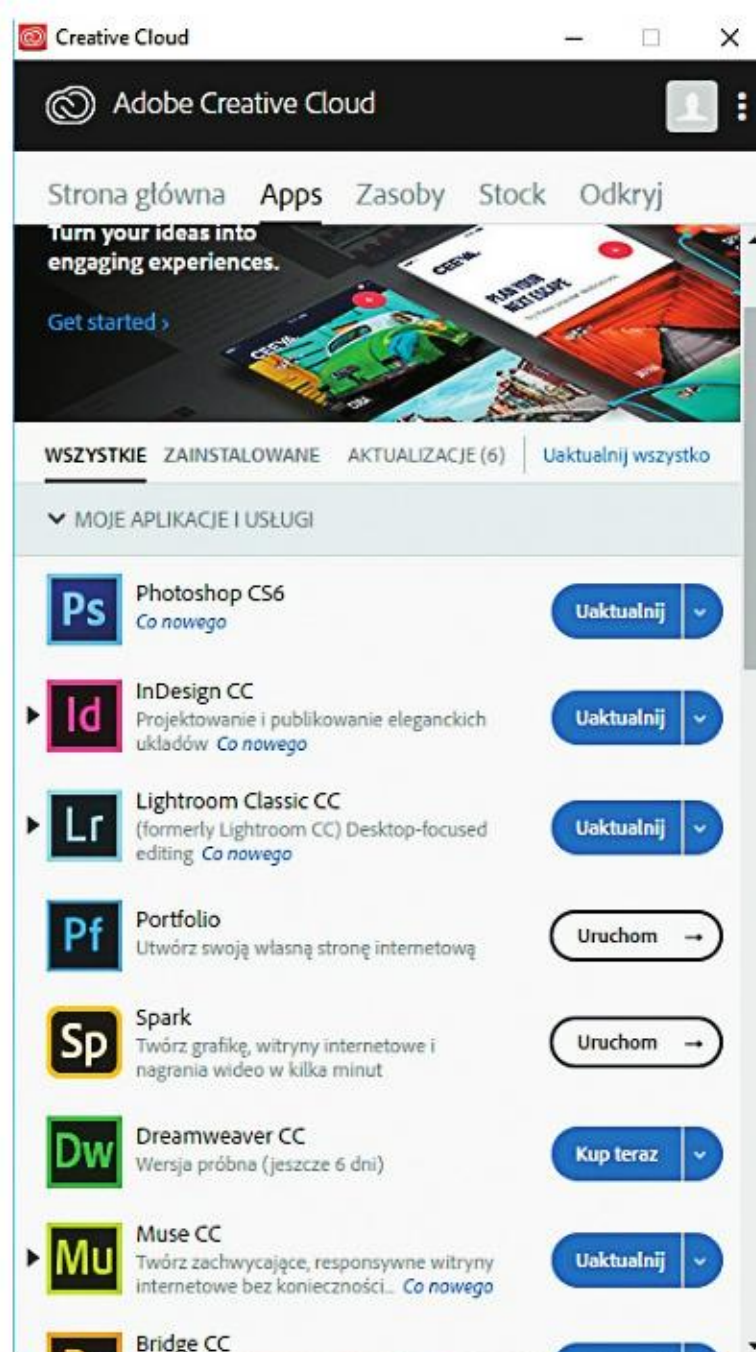
Tworzenie dokumentów w edytorach WYSIWYG

ZAGADNIENIA

- Zasady edycji dokumentów hipertekstowych w edytorach WYSIWYG
- Przykłady wykorzystania edytorów WYSIWYG

Adobe Dreamweaver CC

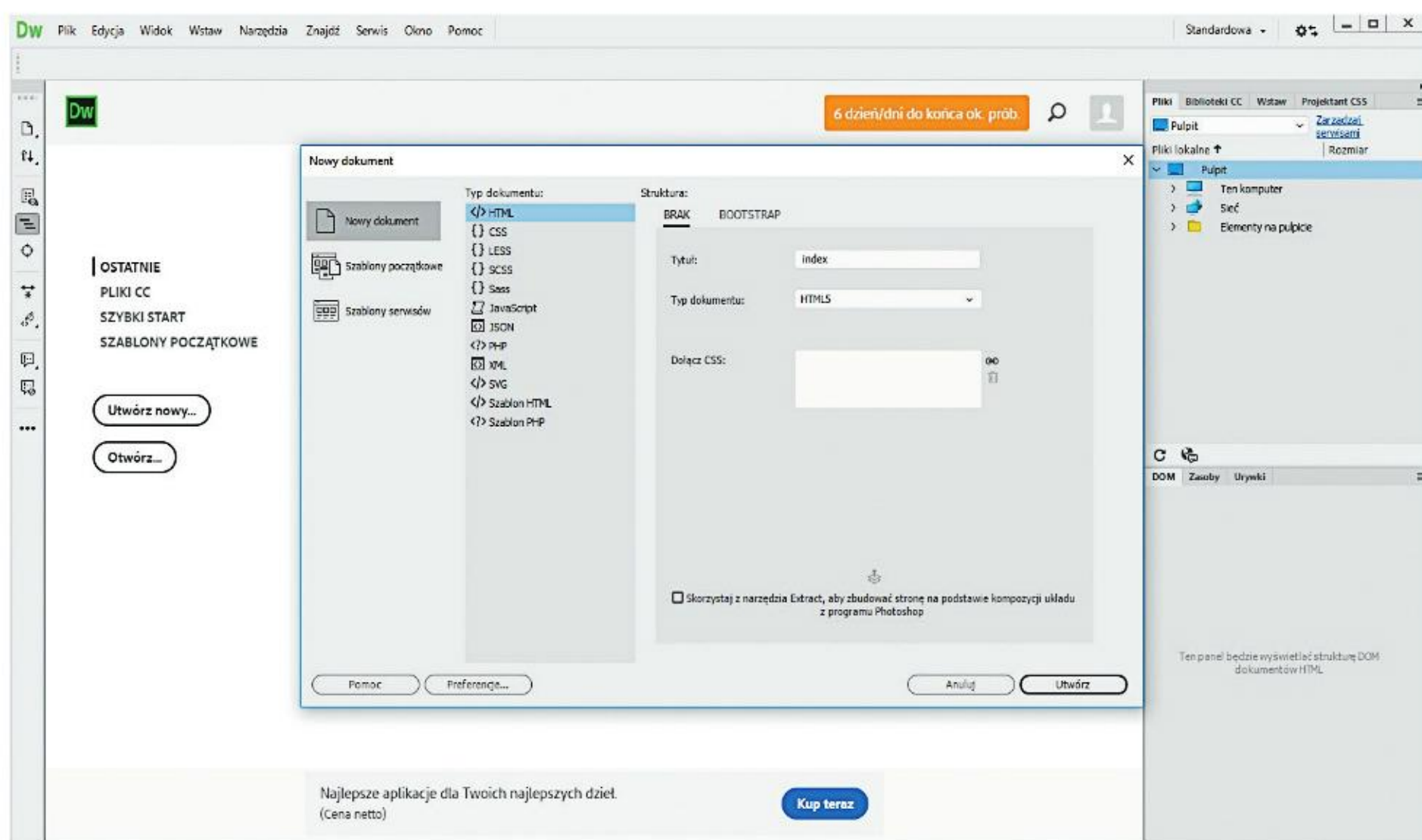
Pobranie wersji testowej tego programu jest możliwe po uprzedniej rejestracji na stronie www.adobe.com. Ciekawym rozwiązaniem jest scentralizowanie przez firmę Adobe wszystkich aplikacji pobieranych przez zarejestrowanego użytkownika w Adobe Creative Cloud. Webmasterowi usługa Creative Cloud zapewnia dostęp do wszystkich aplikacji firmy Adobe na komputery i urządzenia przenośne. Oprócz dostępu do programów i ich aktualizacji, są tam wbudowane szablony ułatwiające rozpoczynanie projektowania stron internetowych, a także samouczki pomagające krok po kroku doskonalić swoje umiejętności.



Rys. 27.1. Okno Adobe Creative Cloud

Po zainstalowaniu aplikacji i jej uruchomieniu pojawi się okno robocze programu. W czasie instalacji użytkownik może skonfigurować parametry interfejsu obsługi. W programie Adobe Dreamweaver CC strony można tworzyć w trybie wizualnym oraz przez edycję kodu źródłowego. Lista dostępnych trybów pracy znajduje się w lewym górnym rogu okna programu.

Pierwszą czynnością po uruchomieniu programu będzie utworzenie nowego dokumentu HTML. Po wyborze opcji **Utwórz nowy...** wybieramy dokument HTML oraz nadajemy mu nazwę.

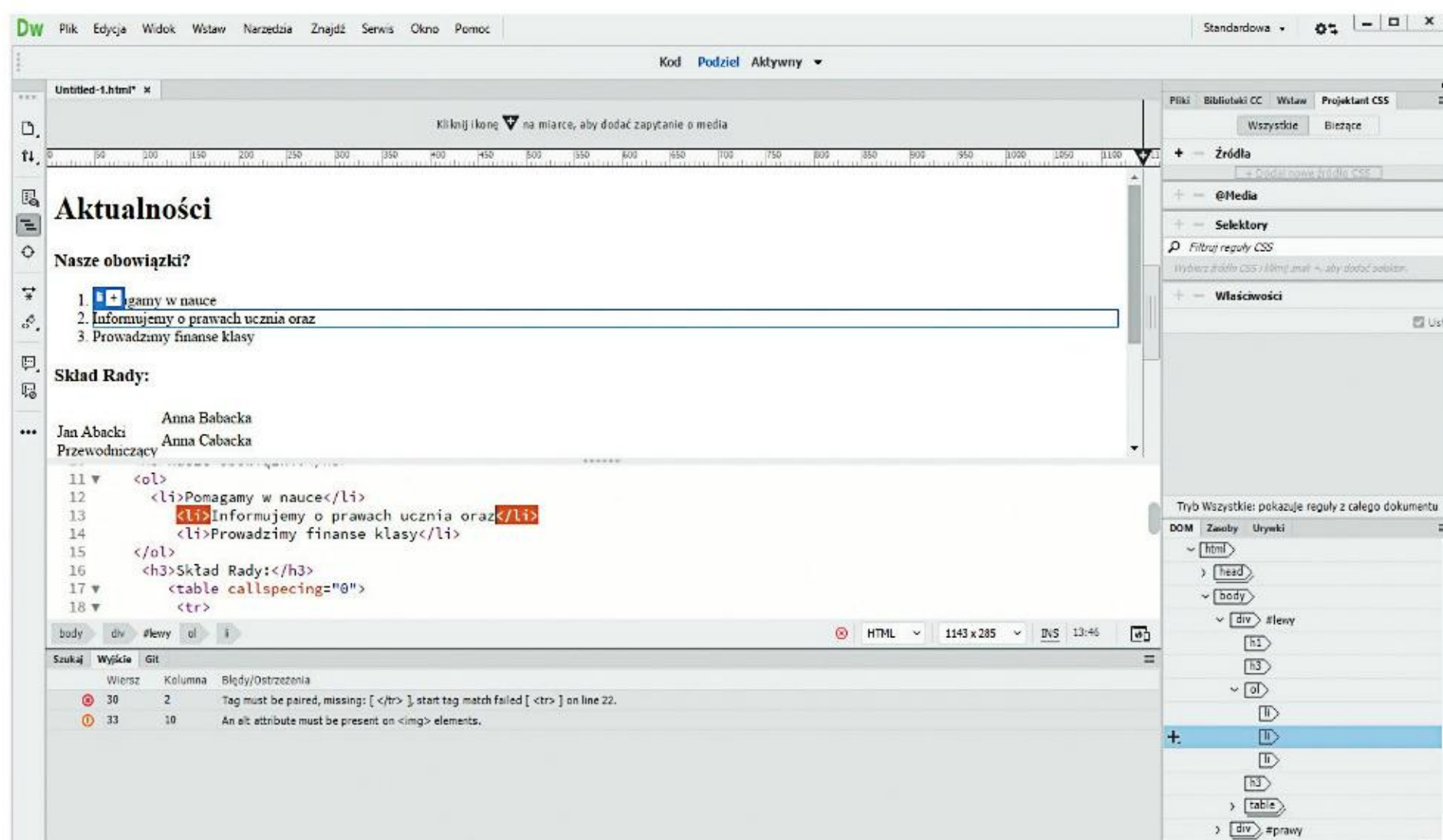


Rys. 27.2. Tworzenie nowego dokumentu

Okno robocze programu możemy dopasować do swoich potrzeb. Do dyspozycji mamy opcje **Kod**, **Podziel** i **Aktywny**. Nasze okno jest ustawione w trybie **Podziel**, dzięki czemu mamy widok zarówno kodu, jak i jego interpretacji w przeglądarce. Po lewej stronie okna roboczego znajduje się pasek narzędzi, natomiast po lewej – grupa zakładek funkcji programu. Pomagają one śledzić wykonywane czynności i modyfikować ich wyniki. Najważniejsze zakładki to **Wstaw**, **Projektant CSS** i panel **Pliki**.

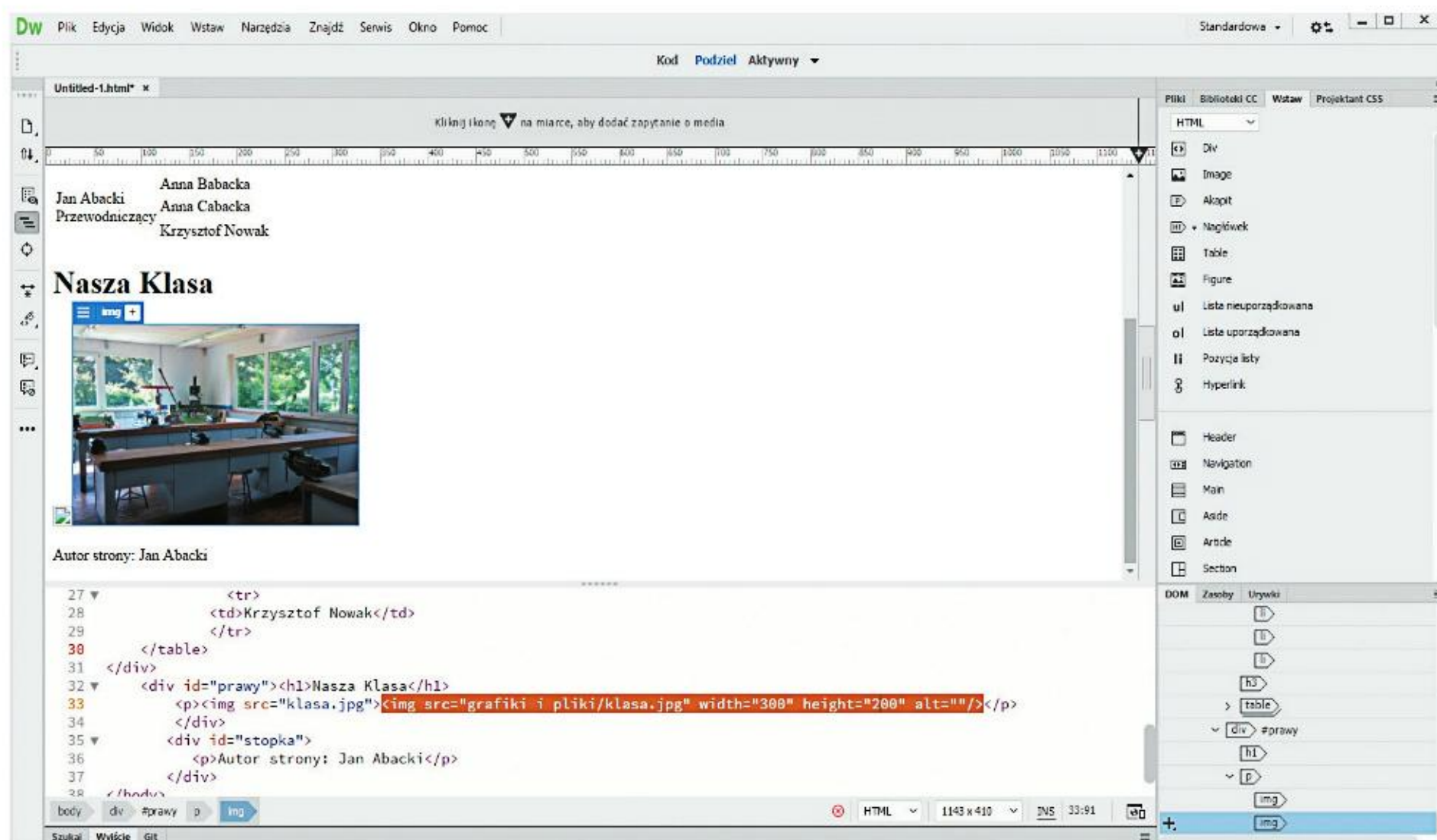
Podczas edycji kodu HTML możemy korzystać z podglądu w czasie rzeczywistym uruchamianym klawiszem [F12]. W opisywanym przypadku podczas pracy w trybie Podziel mamy okno edytora kodu w dolnej części ekranu, a w górnym oknie – stały podgląd rzeczywistej strony. Umożliwia to wprowadzanie zmian w kodzie przez edycję w oknie podglądu. W trakcie edycji kodu źródłowego strony zmiany są od razu widoczne.

Program jest również wyposażony w walidator kodu, więc od razu możemy sprawdzić zgodność struktury dokumentu hipertekstowego ze standardem HTML w wersji wybranej podczas tworzenia nowego dokumentu.



Rys. 27.3. Edycja kodu HTML w programie Adobe Dreamweaver CC

Do dokumentu w prosty sposób możemy wstawiać tabele, obrazy oraz elementy multimedialne. W zakładce **Wstaw** wybieramy element, który chcemy osadzić na stronie, i wskazujemy jego miejsce docelowe. Następnie możemy dowolnie formatować obiekt. Jego wygląd sprawdzamy na podglądzie w górnym oknie.



Rys. 27.4. Wstawianie obiektów

W każdej chwili do naszego pliku HTML możemy dołączyć plik kaskadowych arkuszy stylów. Plik arkusza uda się również utworzyć w programie Adobe Dreamweaver CC.

Na koniec zapisujemy projekt za pomocą opcji **Zapisz jako...** w menu **Plik**.

TinyMCE

Jest to internetowy edytor WYSIWYG napisany w Javascript. TinyMCE wydaje Moxiecode Systems AB jako program darmowy na licencji LGPL. Główną zaletą tej aplikacji jest łatwa integracja z dowolnym systemem zarządzania treścią (CMS). Takie rozwiązanie okazuje się szczególnie przydatne, kiedy po publikacji witryny internetowej docelowy użytkownik projektu chce samodzielnie tworzyć i formatować wpisy na stronie. W normalnym trybie byłaby konieczna ingerencja webmastera, ale wykorzystanie edytora w kodzie rozwiązuje ten problem.

Korzystanie z edytora jest możliwe na dwa sposoby. Pierwszy polega na pobraniu skompresowanych plików programu ze strony producenta, wypakowaniu archiwum w wybranej lokalizacji oraz podlinkowaniu w kodzie naszej strony. Drugi, jeszcze prostszy, polega na podlinkowaniu programu bezpośrednio ze strony producenta.

PRZYKŁAD 27.1

Skrypt umożliwiający podłączenie edytora TinyMCE do kodu naszej strony internetowej bezpośrednio z zasobów twórców programu wygląda następująco:

```
<!DOCTYPE html>
<html>
<head>
  <script src="https://cloud.tinymce.com/stable/tinymce.min.js"></script>
  <script>tinymce.init({ selector:'textarea' });</script>
</head>
<body>
  <textarea>Osadzanie edytora w kodzie strony</textarea>
</body>
</html>
```

Oprócz możliwości wykorzystania gotowego edytora na naszej stronie, dostajemy również narzędzia do zaawansowanej konfiguracji, która ma dopasować aplikację do naszych wymagań i potrzeb. Pełna lista możliwości konfiguracji jest dostępna na stronie <https://www.tinymce.com/docs/configure>.

Edytor pozwala nam na edycję kodu HTML, JS oraz CSS. Możemy podglądać rezultat naszej pracy w czasie rzeczywistym.

Przykład edycji dokumentu hipertekstowego w edytorze TinyMCE pokazano na rys. 26.2.



SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Utwórz stronę internetową z osadzonym edytorem TinyMCE.
2. Zainstaluj wybrany przez siebie edytor WYSIWYG i utwórz z jego pomocą stronę internetową o dowolnym polskim bohaterze narodowym. Strona musi zawierać: tekst, obraz i tabelę.



SPRAWDŹ SWOJĄ WIEDZĘ

1. Jakie funkcje pełni walidator kodu w edytorze Adobe Dreamweaver CC?
2. Wypisz w tabeli zalety i wady edytorów WYSIWYG.
3. Opisz aplikacje dostępne na Adobe Creative Cloud.

IX. Wykonywanie stron internetowych zgodnie z projektem

- Zasady projektowania stron internetowych
- Brief projektu strony internetowej
- Dobór czcionek na stronę internetową
- Szablon strony internetowej
- Layout strony internetowej
- Wykonywanie stron zgodnie z projektami

28

Zasady projektowania stron internetowych

ZAGADNIENIA

- Zasady projektowania stron internetowych
- Etapy projektowania stron internetowych
- Rodzaje struktur stron internetowych

Projektowanie stron internetowych jest procesem złożonym i wymaga przeanalizowania wielu aspektów dotyczących wizualizacji treści. Efekt w postaci witryny musi być optymalnym kompromisem między życzeniami klientów, możliwościami technologicznymi a kosztami projektu. Nie ma jednej recepty na idealną stronę internetową, ponieważ jest to jedna z najbardziej rozwijających się technologicznie dziedzin informatyki. Na podstawie analizy technologii webowych i funkcjonujących rozwiązań webmasterskich można określić zbiór zasad stosowanych obecnie podczas projektowania i tworzenia witryn internetowych. Do najważniejszych należy zaliczyć:

1. **temat przewodni strony** – należy pamiętać, jaki jest cel tworzenia strony i jaką informację trzeba przekazać odwiedzającym;
2. **przejrzystość i minimalizm** – użytkownik musi łatwo i szybko znaleźć interesujące go treści; odchodzi się od nadmiaru treści i elementów na stronie;
3. **hierarchiczność struktury** – w strukturze strony poszczególne bloki muszą mieć określoną wagę w hierarchii strony; przykładowo: stopka i informacje w niej zawarte nie mogą być bardziej wyeksponowane niż blok z informacją główną;
4. **responsywność** – echa umożliwiające dostosowanie się strony do ekranu, na którym jest wyświetlana; polega na tworzeniu płynnych układów struktury strony; oznacza to, że inna struktura może być wyświetlana na laptopie, a inna – na smartfonie;
5. **czytelność** – informacje na stronie muszą być czytelne; nie można stosować zbyt małych czcionek, fontów szeryfowych czy zbyt małego kontrastu między tłem a kolorem czcionki;
6. **szybkość ładowania** – nikt nie lubi czekać na załadowanie się strony do okna przeglądarki; należy zatem ograniczać liczbę elementów i wagę poszczególnych plików; szybkość ładowania strony na urządzeniach mobilnych może również zależeć od spełnienia np. standardów AMP Google;
7. **właściwy dobór palety barw** – paleta barw to jeden z najważniejszych detali w opracowywaniu struktury graficznej strony; najlepiej dobierać barwy z profesjonalnych aplikacji. Należy pamiętać, że zależnie od kontekstu strony, treści czy kultury regionu,

w którym strona będzie publikowana, kolor może mieć określone znaczenie; przykładowo: byłoby nietaktem zastosowanie odcieni koloru czarnego dla strony sklepu sprzedającego suknie ślubne;

8. **dopasowanie do wielu przeglądarek** – należy pamiętać o przetestowaniu strony internetowej na jak największej liczbie przeglądarek, aby zoptymalizować wizualizację treści bez względu na oprogramowanie użytkownika;
9. **przygotowanie do pracy z wyszukiwarkami – pozycjonowanie** – strona internetowa musi być przygotowana do współpracy z robotami największych wyszukiwarek;
10. **dostępność strony** – zbiór cech pozwalających na zapewnienie dostępu do zasobów internetowych możliwie jak największej liczbie użytkowników, bez względu na ich niepełnosprawność, wiek, status majątkowy, sprzęt i oprogramowanie; na dostępność składają się: percepcja, funkcjonalność, zrozumiałość i rzetelność; warunki spełnienia tych cech opisano w standardzie **WCAG 2.0** (ang. *Web Content Accessibility Guidelines*);
11. **zgodność ze standardami** – kod strony powinien być zgodny ze standardami W3C; walidacja i testowanie muszą zostać przeprowadzone na każdym etapie tworzenia strony;
12. **grafika i animacja dobrej jakości** – bez grafiki, animacji czy filmów trudno wyobrazić sobie dobrą stronę internetową; należy jednak zwrócić uwagę na dobór grafiki, maksymalne zmniejszenie rozmiarów plików graficznych, wysoką jakość obrazu oraz animację w formacie obsługiwanym przez większość przeglądarek; zaleca się odejście od animacji flash na rzecz skryptów HTML, CSS i JS; filmy najlepiej umieszczać na stronie w postaci odsyłacza do wydajnego serwera, np. YouTube; trzeba pamiętać, by multimedia nie przesłaniały treści zawartych na stronie.
13. **poprawność językowa** – nie ma nic gorszego jak informacja podana z błędami ortograficznymi, gramatycznymi lub stylistycznymi; tekst umieszczony na stronie powinien być zrecenzowany oraz zredagowany merytorycznie i językowo; jeżeli strona jest kierowana również do obcokrajowców, należy zaplanować tłumaczenie treści i umieszczenie na stronie opcji zmiany języka;
14. **interakcja z użytkownikiem** – wszystkie elementy aktywne, takie jak odnośniki, banery czy pola formularza, powinny mieć wyraźny wizualny fokus (zwykle w postaci ramki widocznej w trakcie nawigacji po stronie – klawiszem [Tab]);
15. **logiczne menu** – nawigacja powinna być spójna, logiczna i niezmienna w obrębie strony; nawigacja w obrębie całego serwisu musi być dostępna z poziomu klawiatury.

Projektowanie witryny internetowej to proces obejmujący następujące etapy:

- analiza wymagań klienta, przeznaczenia witryny oraz możliwości technicznych i finansowych;
- wstępna analiza biznesowa (kosztorys, SWOT itp.);
- dobór hostingu (serwery, domeny i subdomeny) oraz technologii i narzędzi do wykonania poszczególnych elementów witryny;
- określenie elementów polityki pozycjonowania i bezpieczeństwa dotyczących projektowanej witryny internetowej;
- stworzenie struktury logicznej i graficznej witryny oraz poszczególnych podstron.

Ewentualne projektowanie bazy danych dla witryny;

- dobór palety kolorystycznej dla witryny;
- opracowanie layoutu witryny internetowej (czcionka, rozdzielczość, tło, nawigacja, menu itp.);
- zaprojektowanie poszczególnych elementów graficznych;
- stworzenie struktury fizycznej witryny – drzewo URL plików.

W ramach struktury strony internetowej można wyróżnić:

- strukturę logiczną, określającą topologię poszczególnych podstron, zwaną również mapą strony;
- strukturę graficzną, tworzącą szczegółowy layout poszczególnych podstron z projektami elementów graficznych i paletą barw;
- strukturę fizyczną, wskazującą drzewo katalogowe z poszczególnymi plikami składającymi się na zasoby strony internetowej.

Liczba czynności składających się na tworzenie projektu strony czy aplikacji internetowej jest tak duża, że najczęściej pracuje nad nimi zespół fachowców. Pozwala to na szybkie i profesjonalne stworzenie strony internetowej spełniającej wymagania zarówno zleceniodawcy, jak i wszystkich standardów webowych.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Wymień i opisz poszczególne etapy projektowania stron internetowych.
2. Wymień rodzaje struktur strony internetowej.
3. Opisz zasady, których należy przestrzegać w trakcie projektowania strony internetowej.

29

Projekt strony internetowej

ZAGADNIENIA

- Przykład briefu strony internetowej
- Szablon i projekt graficzny strony internetowej
- Dobór palety barw i czcionek

Briefing strony internetowej

Przed przystąpieniem do pracy nad stroną internetową warto mieć plan działania, jak najdokładniej określający wymagania inwestora (klienta) dotyczące przyszłej witryny. Takim planem dla webmastera jest brief. Powstaje on na podstawie ankiet, rozmów i analizy dokumentacji.

Brief jest dokumentem zbierającym analizy, założenia techniczne, wymagania użytkowników oraz wyliczenia finansowe dotyczące projektowanej witryny.

Brief jest często tworzony w postaci tabeli zawierającej najważniejsze czynniki z punktu widzenia webmastera. Jest on podstawą do dalszego projektowania elementów witryny internetowej.

PRZYKŁAD 29.1

Poniżej przedstawiono przykład briefu wstępnego zawierającego elementy niezbędne do planowania prac nad stroną internetową. W prawej kolumnie znajdują się przykładowe odpowiedzi klienta.

Podstawowe informacje o kliencie

Nazwa firmy	Przedsiębiorstwo „Zworka” s.c.
Branża firmy	montaż komputerów, handel urządzeniami elektronicznymi, usługi informatyczne
Oferowane produkty i usługi	urządzenia techniki komputerowej, oprogramowanie, usługi informatyczne, montaż sieci teleinformatycznych, szkolenia komputerowe

Elementy identyfikacji firmy (używane kolory, specyficzne elementy, logo)	logo  Kolory: czarny, brązowy, złoty, beżowy, srebrny, biały Elementy specyficzne: elementy elektroniczne, płytki scalone, urządzenia elektroniczne
Osoba do kontaktu	Andrzej Abacki tel. kom.: 603 000 234 e-mail: prezes@gmail.com
Informacje o stronie WWW	
Cel tworzenia strony WWW	reklama, zapoznanie z ofertą handlową firmy
Charakter witryny (serwis informacyjny, produktowy, korporacyjny, portal branżowy, tematyczny)	serwis informacyjny serwis kontaktowy blog
Adres obecnej strony WWW (jeśli istnieje)	firma nie ma strony internetowej
Struktura logiczna strony WWW	1. strona główna 2. aktualności 3. o firmie • struktura organizacyjna • działalność 4. oferta • grupy produktów a) urządzenia i podzespoły komputerowe b) oprogramowanie c) usługi informatyczne 4. zamówienia 5. współpraca 6. promocje 7. kontakt
Wersje językowe Jakież?	tylko język polski
Czy firma dysponuje materiałami niezbędnymi do wykonania struktury graficznej strony WWW (grafiki, fotografie)?	nie, tylko logo
Przewidywany budżet (opcjonalnie)	10 000 zł

Oczekiwany termin wykonania projektu	5 tygodni
Na jakich urządzeniach ma być wyświetlana strona?	komputery stacjonarne i urządzenia mobilne
Preferowane przeglądarki	Firefox i Chrome
Elementy multimedialne (animacje, filmy, dźwięki, muzyka)	tylko grafika statyczna
Informacje o domenie i hostingu	
Czy firma ma domenę internetową, która będzie adresem strony WWW? (jeżeli nie, proszę podać preferowaną nazwę domeny do sprawdzenia jej dostępności)	nie ma www.zworka.pl www.zworka.lodz.pl
Czy firma ma usługę hostingu?	nie ma
Dodatkowe informacje	
Czy będzie potrzebna usługa opieki i obsługi witryny WWW?	firma powinna mieć możliwość samodzielnego edytowania informacji na stronie
Strony internetowe, które mogą być wzorem w trakcie realizacji strony dla klienta	Inf03.pl
Uwagi	zaliczka w wysokości maksymalnie 10% ceny

Szablon strony internetowej

Jeżeli mamy już opracowany brief projektu strony, przystępujemy do tworzenia szablonu, nazywanego też makietą strony internetowej. Makietą jest w pewnym sensie prototypem strony lub aplikacji internetowej, ponieważ na jej podstawie powstaje finalny produkt.

Zaczynamy tworzyć projekt szablonu strony internetowej od zapoznania się z briefem oraz od analizy funkcjonalności stron internetowych konkurencji z danej branży. Każda branża może mieć specyficzne elementy stron internetowych, które są istotne dla ich klientów. Warto sprawdzić, jak wyglądają działające już witryny. Najlepiej sporządzić kilka szkiców propozycji szablonu strony, w których zostaną określone: położenie, wielkość i kształt takich elementów, jak:

- logotypy i grafiki obowiązkowe,
- banery,
- menu nawigacyjne, dodatkowe,
- bloki informacyjne,
- odsyłacze,
- inne elementy (np. stopka, formularze, wyszukiwarki, dodatkowe treści).

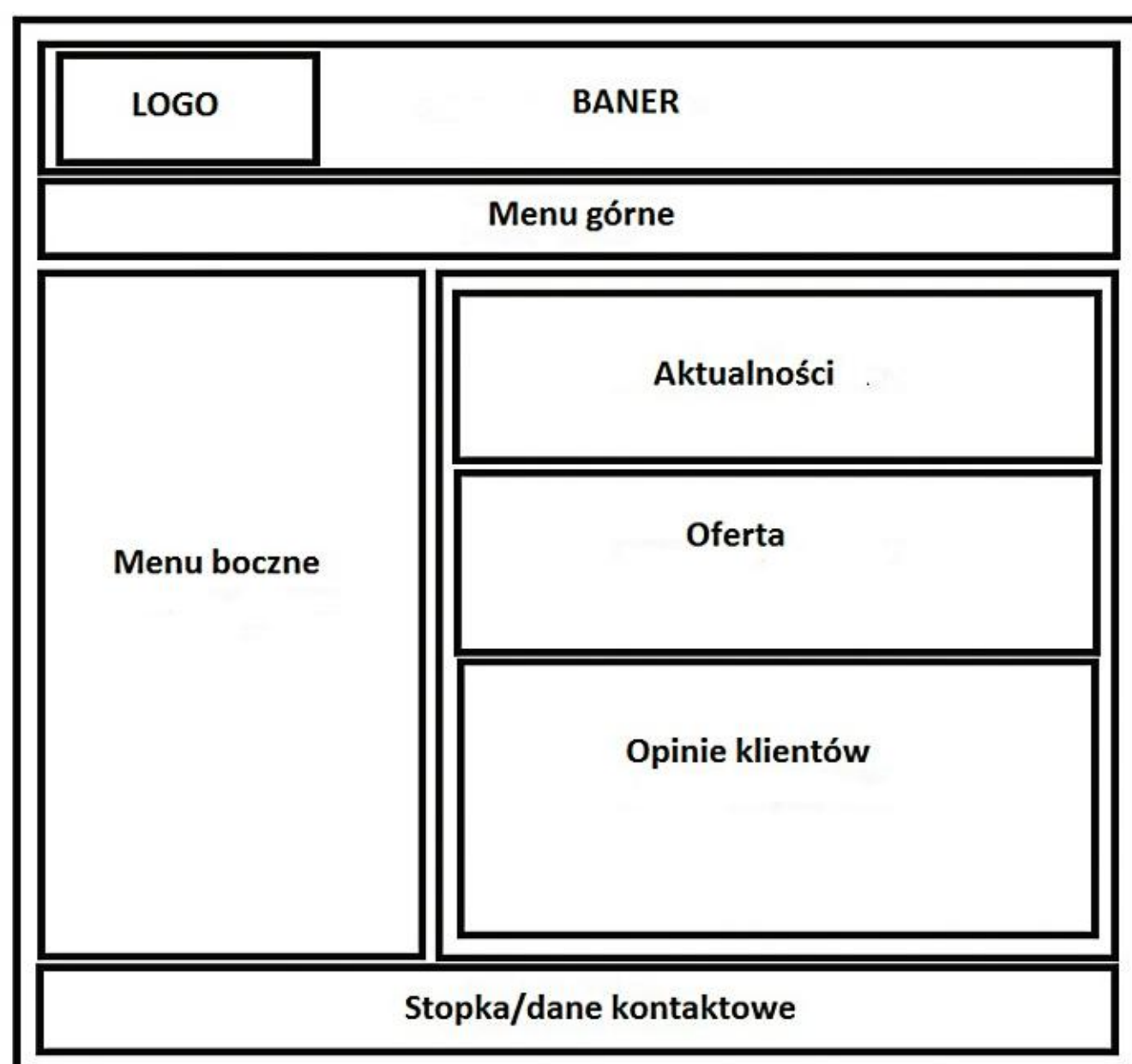
Po konsultacji z klientem można wybrać właściwą strukturę strony, która będzie bazą do tworzenia projektu graficznego.

Ważnym elementem na tym etapie prac jest ustalenie rozdzielczości, w jakiej ma pracować dany szablon. Pozwoli to na określenie proporcji poszczególnych elementów strony. Jeżeli witryna ma pracować na różnych urządzeniach, należy sporządzić makiety stron na każdy typ urządzenia.

Makiety strony można wykonać w jednej z dwóch wersji:

1. **makiety uproszczoną o małym stopniu szczegółowości** (ang. *low-fidelity*), która zawiera podstawowe elementy interakcji oraz pozwala szybko stworzyć kilka opcji;
2. **makiety o dużym stopniu szczegółowości** (ang. *high-fidelity*), która jest bardzo zbliżona do wersji końcowej strony lub aplikacji internetowej.

Makiety uproszczone można wykonać odręcznie na kartce papieru lub skorzystać z dowolnego edytora graficznego (np. MS Paint). W takim szablonie trzeba tylko nanieść kontenery o określonej wielkości na obszar strony internetowej i przypisać im określone funkcje.

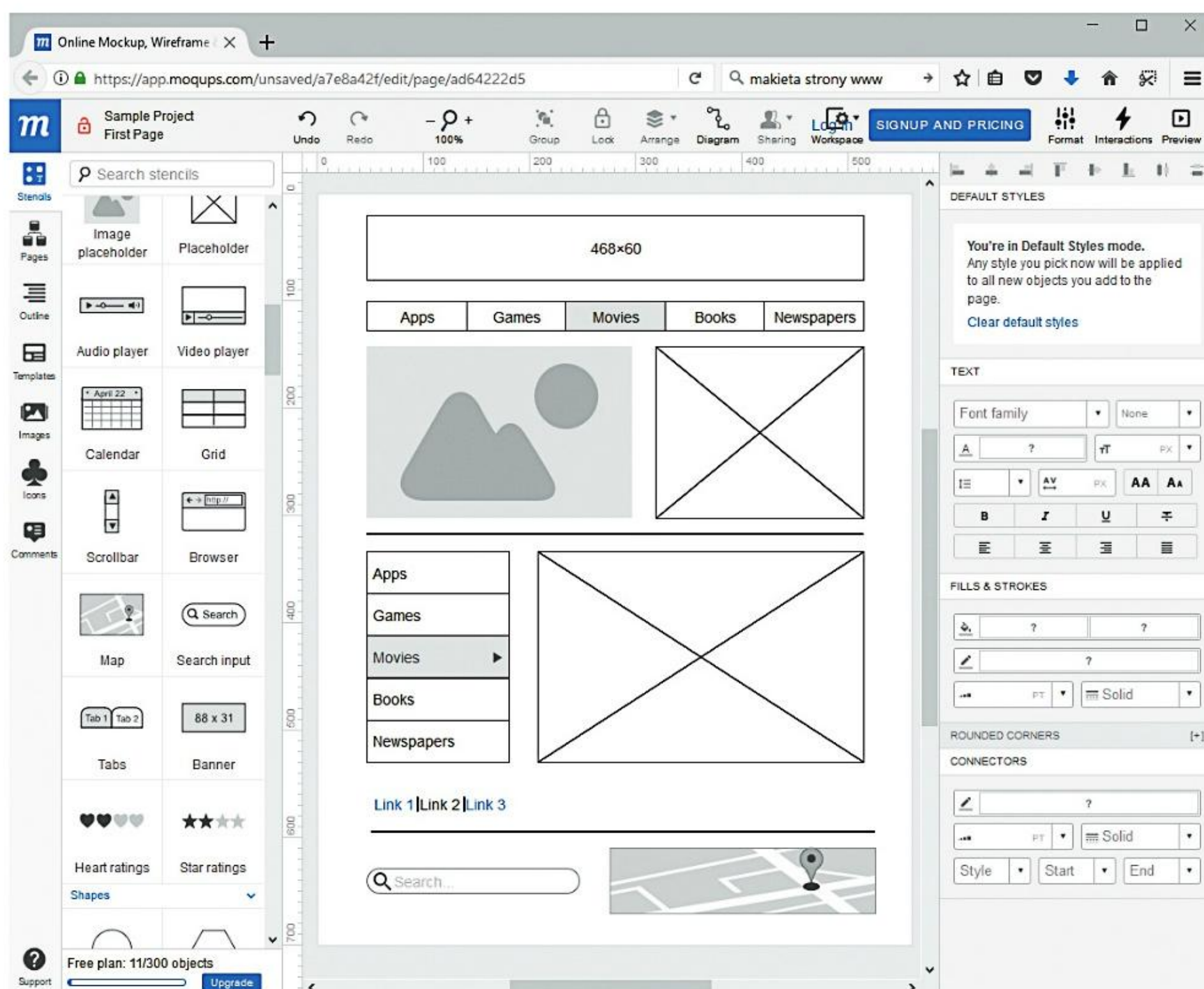


Rys. 29.1. Przykład prostej makiety strony

Makiety o dużym stopniu szczegółowości wykonuje się zazwyczaj za pomocą specjalnie do tego celu tworzonych aplikacji. Mogą one pracować online, co pozwala tworzyć szablony przez edycję projektu w oknie przeglądarki internetowej, lub działają jako samodzielne aplikacje narzędziowe. Przykładem takiej aplikacji jest **Moqups.com**.

Jest to w pełni funkcjonalny program do tworzenia makiet online. Aplikacja udostępnia dużą liczbę funkcji, a jedynym wymogiem korzystania z niej jest rejestracja. Program ma bogaty zestaw narzędzi, umożliwiający tworzenie projektów różnych interfejsów. Dodatkowymi zaletami tej aplikacji są jej przejrzystość i intuicyjność pracy z nią.

Należy pamiętać, że po wykonaniu makiety strony internetowej jest konieczna jej akceptacja przez klienta.



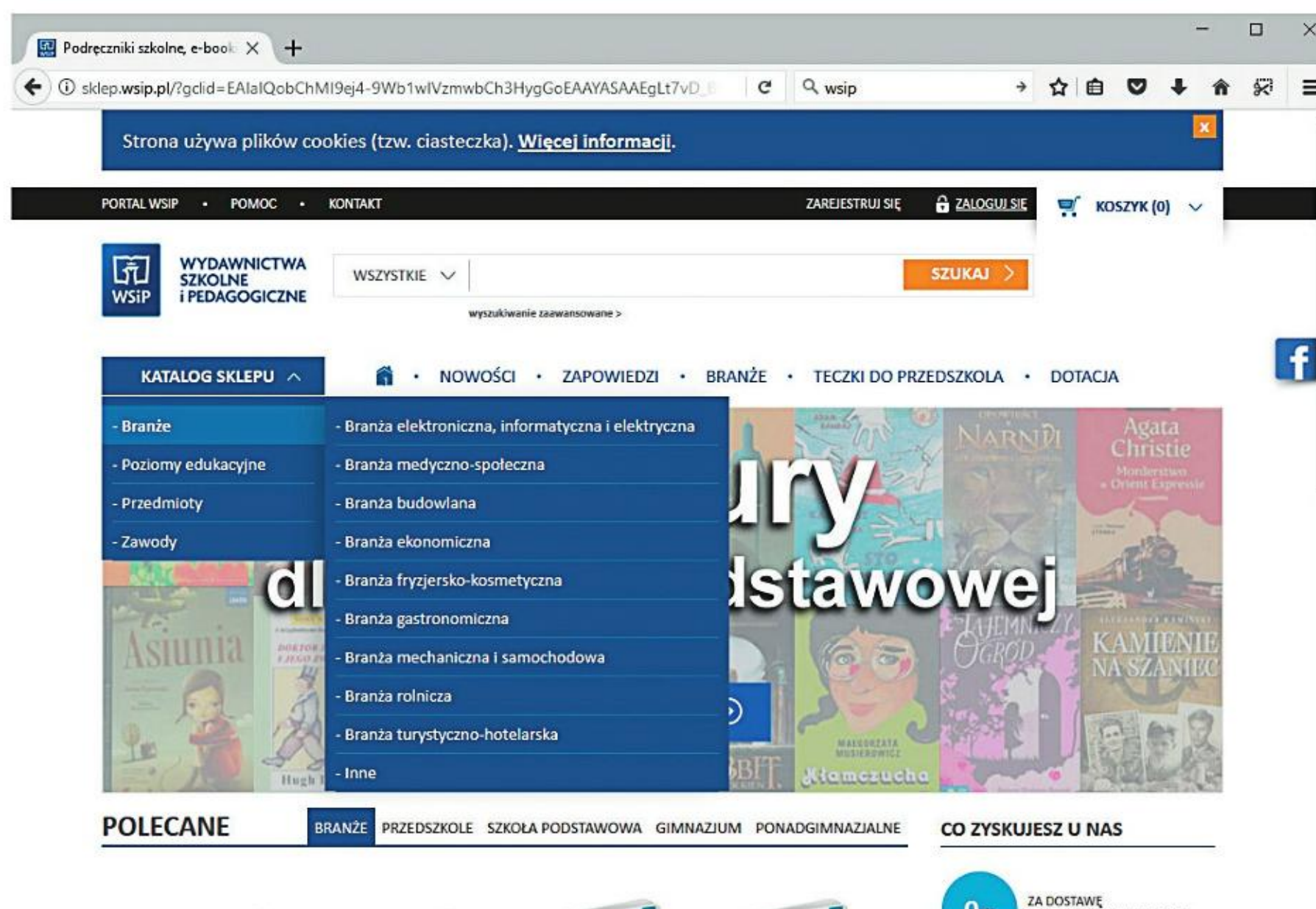
Rys. 29.2. Makieta o dużym stopniu szczegółowości wykonana w Moqups.com

Dobór palety barw dla strony internetowej

Jednym z istotnych elementów każdej strony internetowej jest właściwie dobrana paleta barw. Można ją zdefiniować jako zestaw kolorów dominujących w projekcie całej witryny internetowej.

PRZYKŁAD 29.2

Na stronie internetowej WSiP możemy łatwo zdefiniować kolor dominujący, którym jest odcień błękitu pruskiego. Wyróżnimy także biały kolor tła oraz dodatki w kolorach czarnym i pomarańczowym.



Rys. 29.3. Przykład palety barw strony internetowej WSiP

Kolory zawsze oddziałują na samopoczucie człowieka, a nawet mogą zaważyć na jego opinii dotyczącej danego przedmiotu czy usługi. Symbolika kolorów to potraktowanie kolorów jako wartości emocjonalnych, religijnych lub kulturowych dla określonych grup społecznych bądź narodów.

Poniżej zostanie opisana symbolika niektórych kolorów, ponieważ ich wybór na elementy strony może zaważyć na odbiorze społecznym projektu.

Kolor czarny to symbol elegancji, luksusu oraz szyku. Używa się go tam, gdzie trzeba podkreślić powagę, tajemnicę, kompetencję lub autorytet. Jest również kojarzony z żałobą i śmiercią.

Kolor biały to uniwersalny symbol pokoju i czystości. Kojarzony z prawdomównością, prostotą, dobrem, jasnością lub przestrzenią.

Kolor brązowy jest symbolem spokoju, powagi, ciepła, neutralności i niezawodności.

Kolor czerwony symbolizuje przygodę, energię, niebezpieczeństwo, miłość, pasję, wigor i siłę. To także symbol życia i sprzecznych, skrajnych emocji, piękna i witalności.

Kolor pomarańczowy to barwa młodości, kreatywności, lekkości i aktywności. Symbolizuje zabawę i entuzjazm.

Kolor żółty symbolizuje radość, ciepło, słońce i dobrą pogodę.

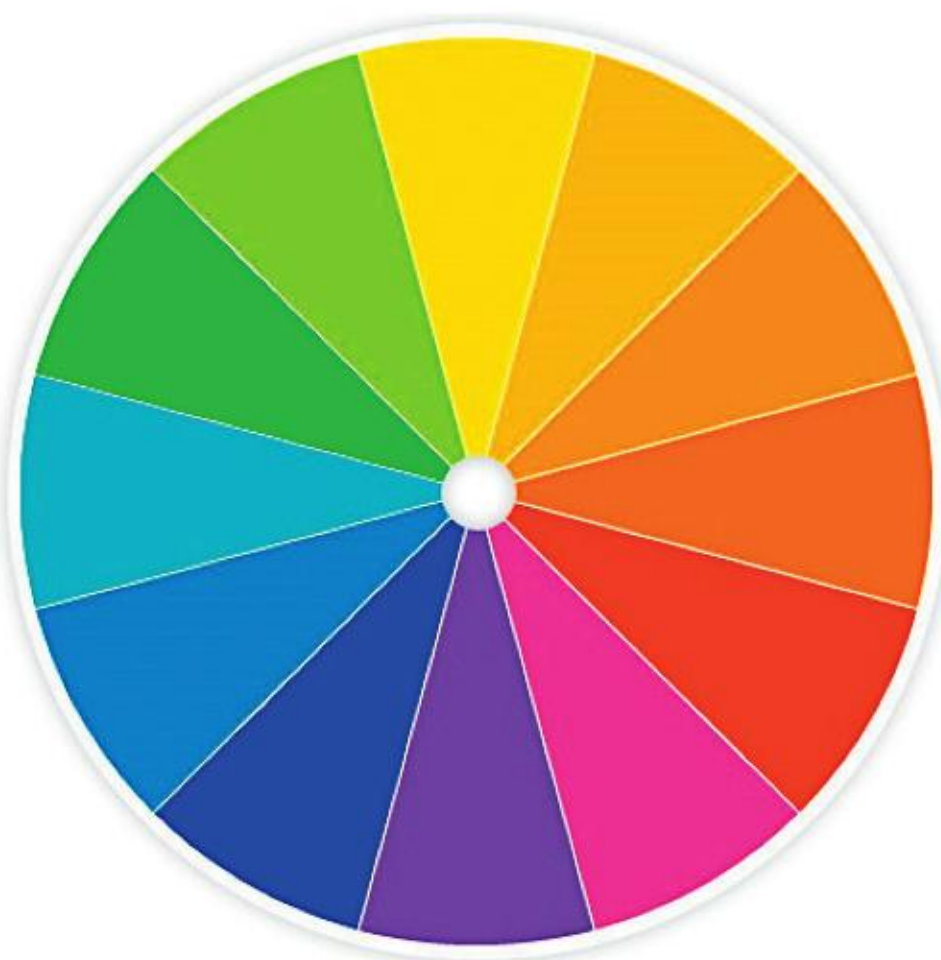
Kolor zielony to barwa natury, kojarzy się z życiem i harmonią. Stanowi też kolor uzdrowienia, nadziei i wolności.

Kolor niebieski przywodzi na myśl spokój, stabilność, sukces, pewność, lojalność i bezpieczeństwo.

Poniżej opisano sześć klasycznych reguł kolorystycznych opartych na tzw. **kole kolorów** (ang. *color wheel*), przedstawionym na rys. 29.4. Dzięki nim można uzyskać harmonijne kompozycje kolorystyczne również na stronach i aplikacjach internetowych. Składają się na nie reguły:

1. dopełniające (komplementarne) – kolory kontrastujące ze sobą; znajdują się po przeciwnych stronach koła kolorów;
2. analogowe – kolory sąsiadujące ze sobą; zwykle dobrze mieszają się ze sobą oraz są harmonijne;
3. monochromatyczne – reprezentujące odcienie jednego koloru; dobrze do siebie pasują i dają uspokajający efekt;
4. dopełniające rozbite – trzy kolory kontrastujące ze sobą;
5. tworzące trójkąt (triada) – trzy kolory znajdujące się w równej odległości od siebie na kole kolorów;
6. tworzące kwadrat (złożone) – kolory znajdujące się po czterech bokach koła kolorów.

Pierwsze koło kolorów powstało w XVII w. i do dziś jest podstawowym wzorcem dla wielu twórców w malarstwie, reklamie czy projektowaniu wnętrz. Cechą charakterystyczną koła kolorów jest to, że po jednej stronie są skumulowane kolory ciepłe, a po przeciwnej – chłodne.

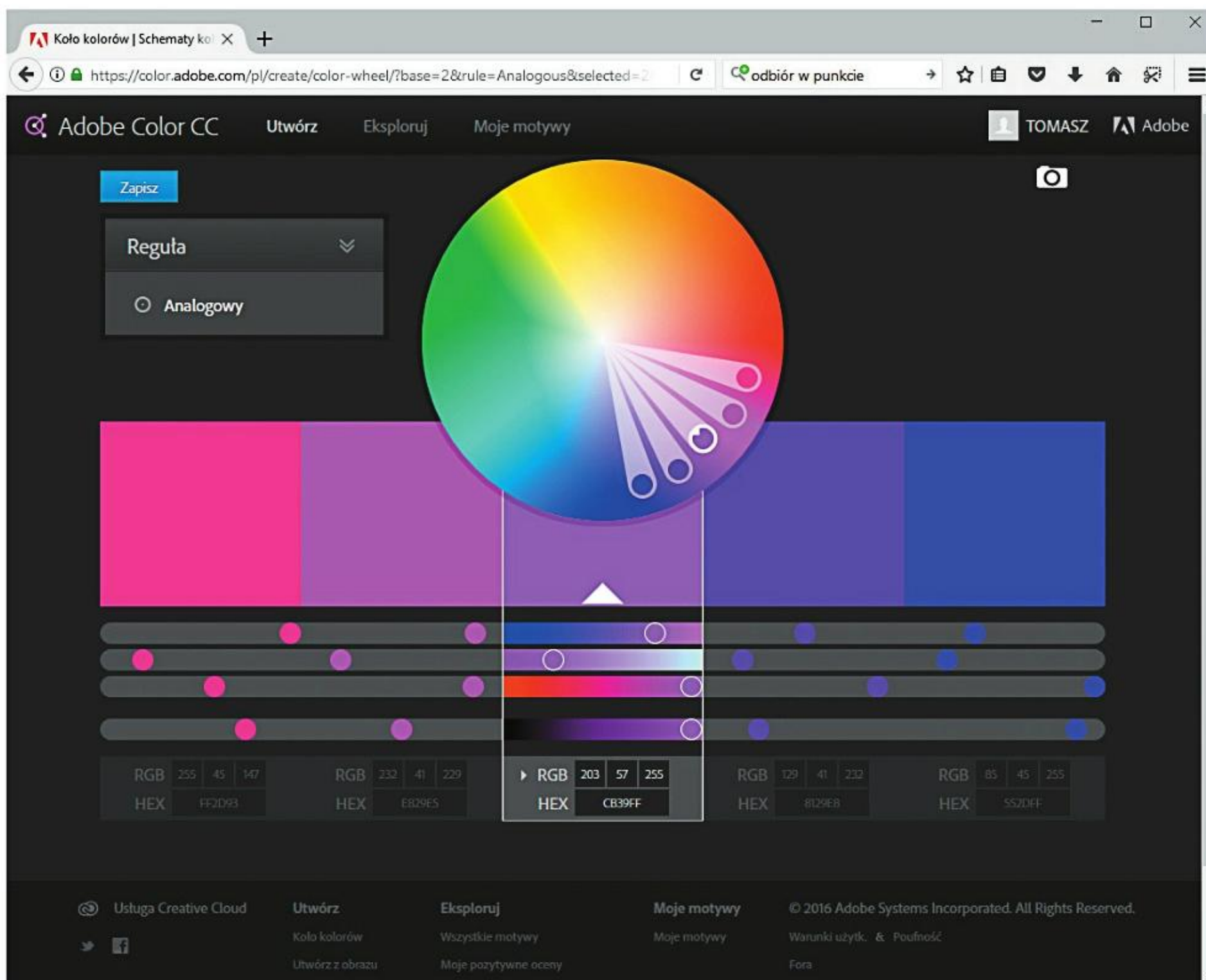


Rys. 29.4. Koło kolorów

Samodzielny dobór palety barw wymaga jednak pewnego doświadczenia i zmysłu artystycznego. Współczesny webmaster może skorzystać z pomocy profesjonalnych aplikacji, które pomogą ustalić najlepszy zbiór barw do określonego zastosowania.

W tego typu aplikacjach motyw kolorystyki strony można utworzyć samodzielnie według określonych reguł lub na podstawie związanych ze stroną obrazów, wczytywanych online. Przykładem takiej aplikacji internetowej uruchamianej w przeglądarce internetowej jest Adobe Color CC dostępny na stronie <https://color.adobe.com>.

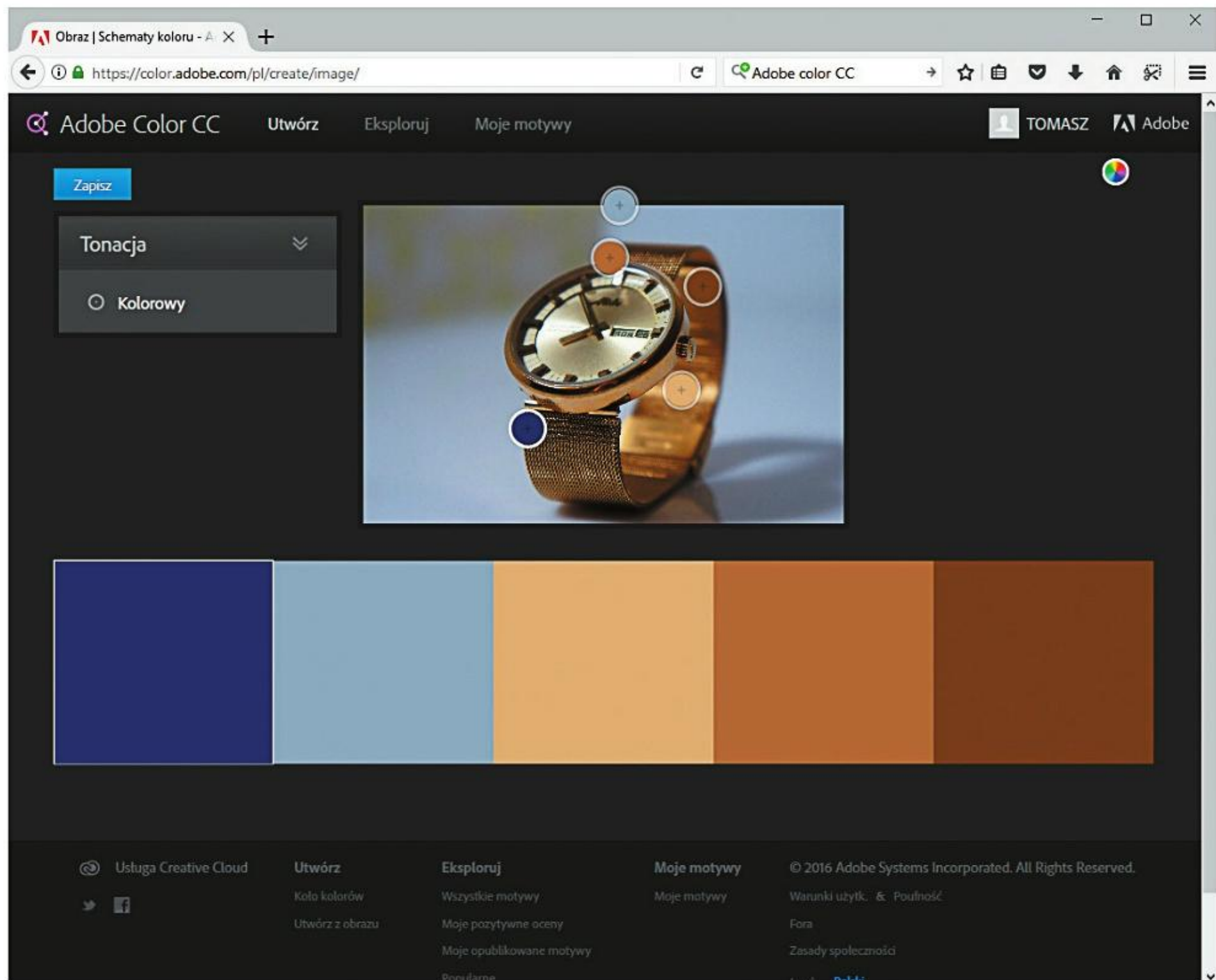
Okno aplikacji jest czytelne, a interfejs obsługi – bardzo intuicyjny. Na górnym pasku znajduje się proste menu nawigacyjne, w centralnym oknie – koło kolorów z punktami wskazującymi barwy wybrane, a pod kołem – próbki palety pięciu barw z możliwością odczytania kodów RGB oraz HEX. W stopce umieszczono dodatkowe menu nawigacyjne użytkownika.



Rys. 29.5. Okno aplikacji Adobe Color CC

Paletę barw można zdefiniować na kilka sposobów:

1. Wybrać z menu **Reguła** właściwy schemat doboru palety oraz określić punktem wskaźnika pierwszy kolor dominujący. Do dyspozycji mamy schematy: analogowy, monochromatyczny, triada, dopełniający, złożony, odcienie oraz własny.
2. Wprowadzić kod RGB lub HEX koloru podstawowego (wiodącego) i dobrać pozostałe według określonej reguły.
3. Wybrać motyw za pomocą opcji **Eksploruj** z biblioteki gotowych palet zapisanych przez użytkowników tej aplikacji.
4. Wczytać główne logo lub grafikę za pomocą opcji **Utwórz z obrazu** i na podstawie ich kolorystyki dobrać pozostałe barwy palety. Dobrać parametry tonacji palety barw z menu **Tonacja**.



Rys. 29.6. Dobór palety kolorów przez obraz

Po zakończeniu procesu definiowania palety barw można zapisać kody wybranych kolorów w notacji RGB lub HEX, jak również zachować wypracowany motyw w bibliotece Adobe Color CC.

Zapisz w bibl.



Nazwij swój motyw.

Wybierz chmurę i bibliotekę.

Szablony Stock ▼

- ✓ Szablony Stock (1)
- Moja bibliot... (0)
- + Nowa biblioteka

Otaguj i opublikuj motyw, aby mogli go

Dodaj tagi

☒ Opublikuj ten motyw w sekcji Eksploruj.

Rys. 29.7. Okno zapisywania motywu w bibliotece

Dobór palety barw jest często związany z innymi materiałami graficznymi, które będą wykorzystywane na projektowanej stronie. W związku z tym do tego etapu należy podejść w momencie, gdy mamy już przygotowane obrazy, zdjęcia i inne materiały multimedialne. Pozwoli to na ujednolicenie standardu kolorów dla wszystkich elementów graficznych strony. Po wyborze palety kolorów może się okazać, że niektóre zdjęcia lub grafiki trzeba poddać dodatkowej obróbce, aby dopasować je do przyjętego motywu barw.

Po ustaleniu kolorystyki elementów strony internetowej oraz przygotowaniu zdjęć, obrazów i multimedialnych materiałów można przystąpić do następnego etapu prac projektowych – do zaprojektowania graficznego layoutu strony.

Dobór czcionek na stronę internetową

Oprócz dopasowania odpowiedniej palety barw na stronę internetową, należy również zaplanować typografię strony, czyli zestaw odpowiednich krojów czcionek. Trzeba wziąć pod uwagę nie tylko estetykę fontów, lecz także zasady dostępności oraz szybkość wczytywania, szczególnie na urządzeniach mobilnych.

Do każdej czcionki, którą chcemy wykorzystać do projektu witryny, powinniśmy nabyć prawa licencyjne. Dzięki temu będziemy mogli stosować ją zarówno lokalnie, jak i w sieci. Dodatkowo licencja pozwala nam wykorzystać więcej opcji językowych (ważne w przypadku stron z różnymi wersjami językowymi) i dodatkowych znaków. Na pewno należy sprawdzić, czy licencja przewiduje polską opcję językową ze wszystkimi znakami diakrytycznymi oraz symbolami, takim jak polskie cudzysłowy (górny i dolny) lub twarde spacja. W sieci Internet jest dostępnych wiele stron oferujących czcionki do różnych zastosowań. Przykładem takiej strony jest <https://www.omnibus-type.com>.

The screenshot shows the Omnibus-Type website interface. At the top, there's a navigation bar with links: FONTS, SERVICES, TEAM, CONTACT, and a search icon. Below the navigation bar, a grid displays various font families: Archivo, Bahianita, Manuale, Saira Extra Condensed, Archivo Black, Barriecito, MuseoModerno, Saira Stencil One, Archivo Narrow, Barrio, Pragati Narrow, Sansita, Asap, Chivo, Rosario, Truculenta, Asap Condensed, Faustina, Saira, Unna, Asap Symbol, Grenze, Saira Semi Condensed, and Bahiana, Jaldi, Saira Condensed. Below the grid, a large, stylized font specimen for 'SAIRA STENCIL ONE' is shown. Underneath the specimen, the font's details are provided: 'Saira Stencil One' is the cut version of Saira, maintaining its primary structure but adding precise slits. It was designed by Héctor Gatti and developed by the Omnibus-Type Team. The font includes 635 characters and 904 glyphs. It is available under the SIL Open Font License, 1.1. A 'FREE DOWNLOAD' button and a 'PDF SPECIMEN' button are visible. A 'DONATE' button is also present.

Rys. 29.8. Strona z darmowymi czcionkami

Oferuje ona wiele darmowych czcionek z dokładnym opisem zastosowania, opcjami licencji i podglądem w formacie pdf. Dzięki temu możemy dokładnie określić możliwość jej zastosowania oraz wygląd.

**A B C Ç D E F G H I J K L M N Ñ O P Q R S T U
V W X Y Z / à â á ä å ã abc ç d e f g h i j k l m
n ñ o p q r s t u v w x y z / 0 1 2 3 4 5 6 7 8 9**

Rys. 29.9. Podgląd wyglądu liter i znaków czcionki

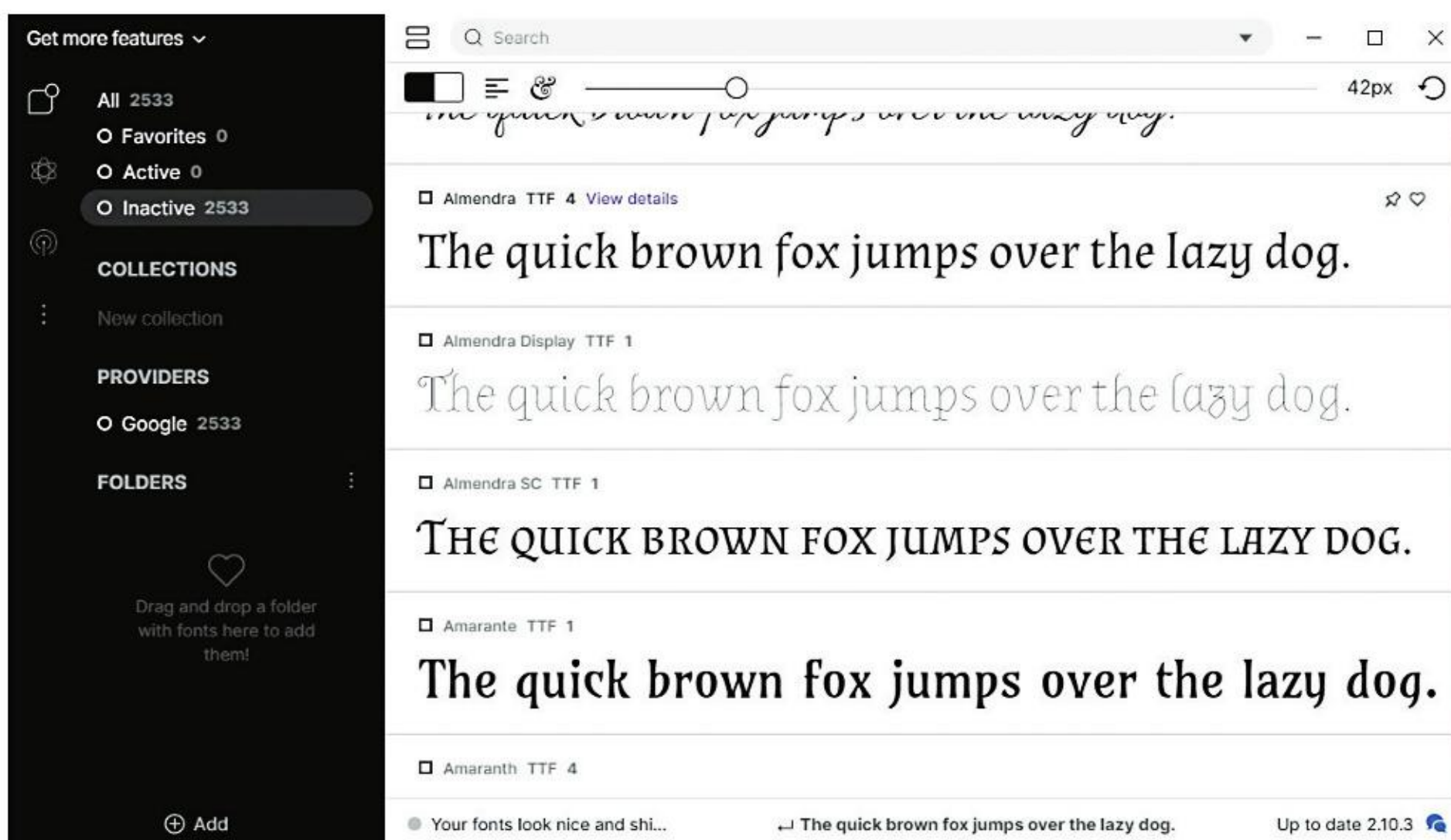
Jeżeli mamy już wybrany font z dobrymi znakami, o odpowiednim wyglądzie oraz z licencją, musimy sprawdzić, jak będzie się on wyświetlać na stronach internetowych. Chodzi przede wszystkim o szybkość wczytywania i brak występowania różnic w trakcie wyświetlania w oknie przeglądarki.

Jednymi z ważniejszych aspektów stron internetowych są: czytelność, dobry kontrast, kolor i odpowiednia wielkość fontu. W tym przypadku, oprócz zagadnień związanych z kompozycją czy interlinią, warto też zwrócić uwagę na taki element, jak wielkość fontu. Przykładowo: narzędzia Google w swoich algorytmach badania użyteczności stron określają odpowiednią wielkość tekstu na 16 px. Każda czcionka może inaczej wyglądać po zastosowaniu tej wielkości. Bardzo istotna jest wysokość linii małych liter i ich stosunek do wydłużeń górnych i dolnych. Należy pamiętać, że nadmierna liczba kolorów, pogrubień i pochyień tekstu nie wpływa dobrze na czytelność i atrakcyjność strony.

Ostatnią kwestią, która powinna wpłynąć na dobór czcionek, jest określenie, czy dany krój pasuje do treści znajdujących się na stronie. Stosujemy następujące zasady:

- **fonty szeryfowe** bardziej pasują do stron eleganckich, o poważnych treściach; musimy jednak pamiętać, że dla osób niedowidzących jest lepiej, gdy czcionka nie jest szeryfowa;
- **fonty bezszeryfowe** sprawiają wrażenie bardziej nowoczesnych, dlatego często są wykorzystywane w treściach nawiązujących do zaawansowanych technologii;
- **fonty pisane i ozdobne** są wykorzystywane na stronach zdecydowanie rzadziej i powinny być używane jedynie jako ozdobniki lub elementy dodatkowe strony; ze względu na cienkie linie i skomplikowaną formę warto stosować większe rozmiary liter.

Aby właściwie zarządzać typografią na stronie internetowej, stworzono specjalne aplikacje. Przykładem takiego oprogramowania jest FontBase (<https://fontbase.se>).



Rys. 29.10. Okno aplikacji do zarządzania czcionkami

FontBase to organizator fontów. Idealnie sprawdza się podczas przeglądania, dobierania i grupowania czcionek do wykorzystania w różnorodnych projektach plastycznych lub webowych. Jego podstawowe funkcje to:

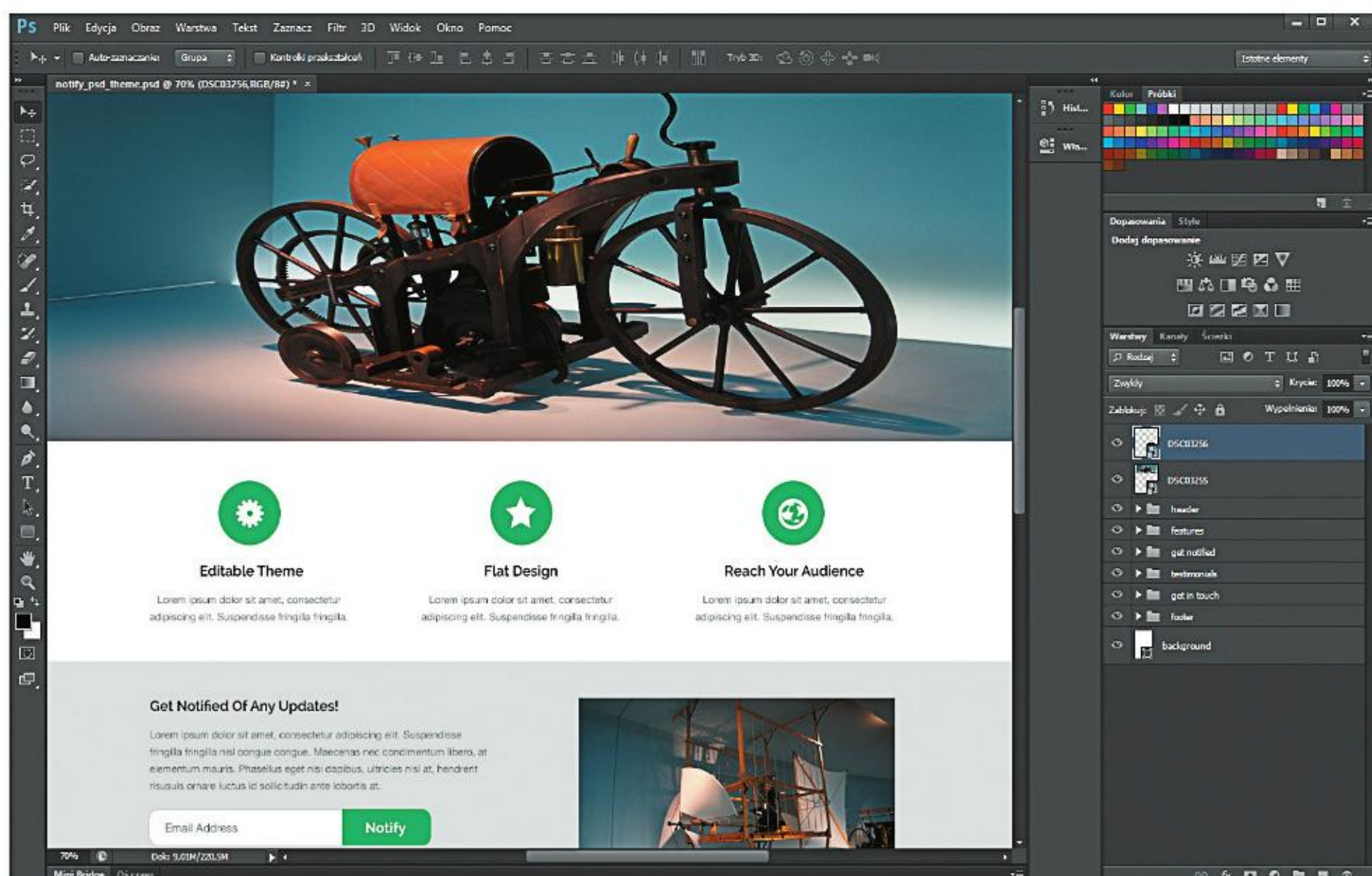
- instalacja i aktywacja fontu lub całej kolekcji za pomocą jednego przycisku;
- dodawanie całych folderów do aplikacji;
- możliwość uporządkowania fontów z zachowaniem określonej kolejności;
- synchronizacja fontów na wielu komputerach za pomocą chmury.

Projekt graficzny strony internetowej

Po zatwierdzeniu makiety strony internetowej należy wykonać jej layout, czyli układ kompozycyjno-graficzny. Określa on kolorystykę, czcionki, ułożenie elementów i grafik, wielkość nagłówków, determinuje również hierarchię elementów w taki sposób, jak będzie to widoczne na gotowej stronie internetowej. Layout przygotowuje się w skali 1 do 1 na przyjętym obszarze o rzeczywistej rozdzielczości ekranu. W fazie projektowej layout ma postać pliku graficznego prezentującego zakładany styl strony. W następnym etapie prac layout jest kodowany i cięty na gotowe elementy graficzne, wykorzystywane później na gotowej stronie bądź aplikacji.

Podczas tworzenia projektu graficznego witryny należy konsekwentnie trzymać się zdefiniowanej palety barw oraz szablonu strony. Wykorzystuje się również przygotowane wcześniej elementy, takie jak:

- grafiki statyczne,
- zdjęcia,
- animacje,
- filmy.



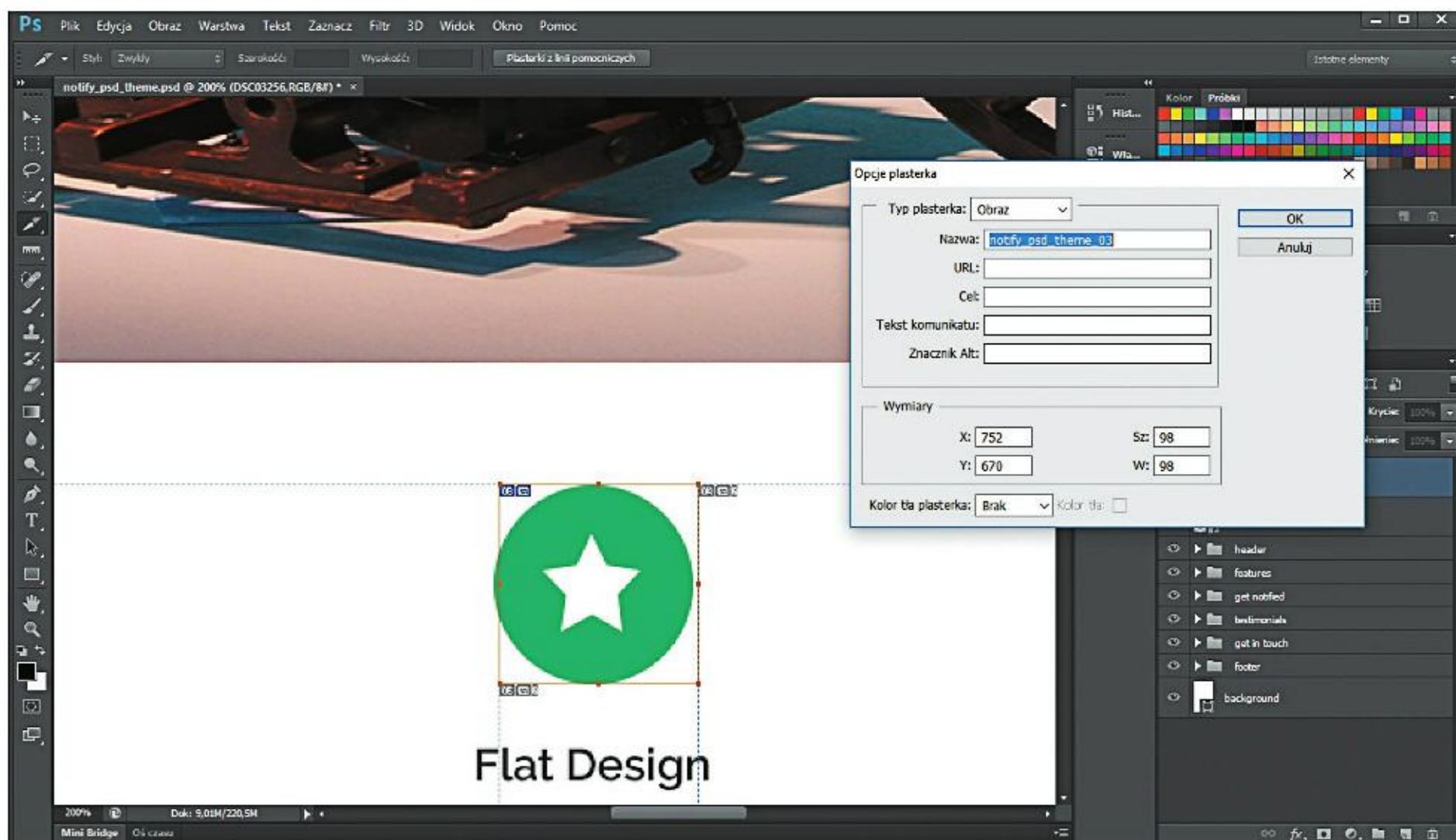
Rys. 29.11. Przykład layoutu strony WWW w Adobe Photoshop

Ponieważ layout strony jest plikiem graficznym, można go wykonać w edytorach grafiki rastrowej. Najczęściej wybierane przez webmasterów są programy: **Adobe Photoshop**, **Adobe XD**, **Adobe Illustrator**, ale można też wykorzystać **GIMP**.

Program Adobe Photoshop oferuje wiele przydatnych funkcji dla webmasterów. Dzięki dostępowi do nieograniczonego obszaru kompozycji ułatwia opracowywanie zawartości dla różnych wielkości ekranów i urządzeń oraz grafiki. **Obszary kompozycji** usprawniają proces projektowania przez udostępnienie nieskończonego obszaru roboczego, na którym można edytować jednocześnie wersje przeznaczone dla różnych urządzeń i ekranów o różnych rozdzielczościach. Podczas tworzenia obszaru kompozycji wybiera się jedno z wielu ustawień predefiniowanych lub określa własny rozmiar. Dodatkowo obszary kompozycji zoptymalizowano do pracy w trybie kolorów RGB.

Obszary kompozycji można eksportować jako obrazy JPEG, GIF, PNG i SVG oraz jako dokumenty PDF.

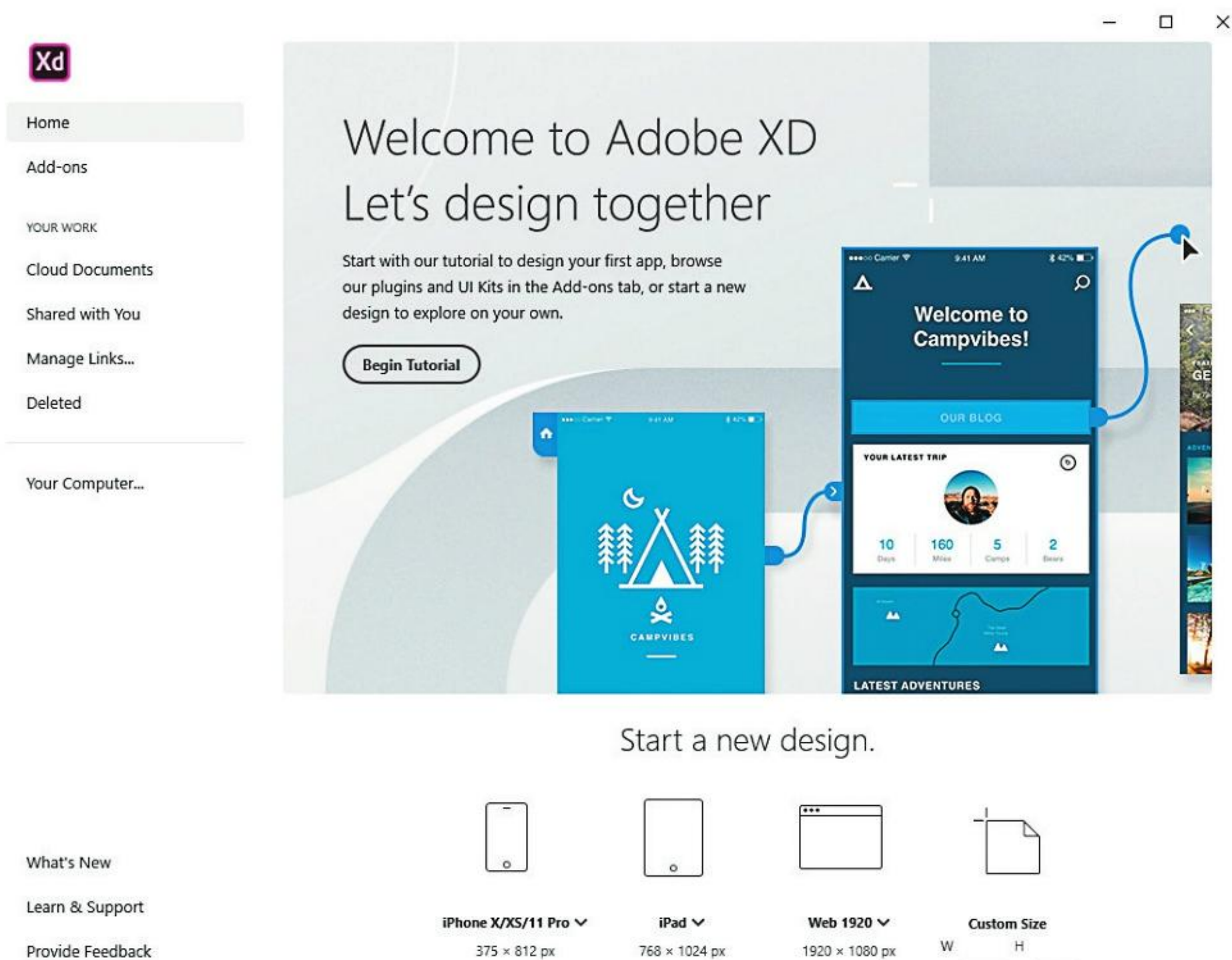
Dodatkowym, nowym pojęciem w projektowaniu layoutu strony WWW jest plasterk. **Plasterki** dzielą obraz na mniejsze obrazy, aby później można było je łączyć w stronę internetową za pomocą języka HTML lub CSS. Plasterki tworzy się na podstawie warstw albo za pomocą narzędzia **Plasterki**. Cięcie projektu graficznego na plasterki jest szybkie i pozwala zapisać je wszystkie za jednym razem na koniec pracy z projektem.



Rys. 29.12. Tworzenie plasterka

Do tworzenia projektu kompozycyjno-graficznego strony coraz częściej wykorzystuje się aplikację Adobe XD. Po pierwsze, w podstawowej wersji jest ona dostępna za darmo. Po drugie, ma nieskomplikowany interfejs. Adobe XD pozwala na proste i szybkie projektowanie małych i średnich stron oraz aplikacji. Pracę usprawniają narzędzia dostępu do symbolów, stylów znakowych oraz kolorów. Wszystkie te możliwości można grupować i wykorzystywać w tym samym momencie, dzięki czemu zarówno drobne, jak i te większe korekty wizualne wprowadza się bardzo sprawnie. Program oferuje duży wybór szablonów roboczych dla komputerów stacjonarnych i urządzeń mobilnych.

Aplikację można pobrać przez Adobe Creative Cloud po uprzedniej rejestracji. Adobe XD w wersji darmowej pozwala na Nielimitowane korzystanie z aplikacji zainstalowanej na komputerze oraz dzielenie się jednym projektem w sieci z innymi członkami zespołu. W programie istnieje możliwość korzystania z tutoriali dla początkujących użytkowników. Okno startowe programu przedstawiono na rys. 29.13.



Rys. 29.13. Okno startowe programu Adobe XD

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Pobierz i zainstaluj program FontBase. Za pomocą zainstalowanej aplikacji zapoznaj się z fontami dostępnymi w systemie operacyjnym. Utwórz nowy folder z czcionkami do nowego projektu o nazwie STRONA i dobierz dwa fonty, które najbardziej ci się podobają.
2. Wyszukaj w sieci Internet strony oferujące darmowe czcionki. Wybierz, pobierz i zainstaluj za pomocą organizera czcionek fonty najlepiej nadające się na stronę o tematyce:
 - historii II wojny światowej na morzu,
 - olimpiady wiedzy humanistycznej,
 - nowej gry komputerowej dla dzieci do 6 lat.
3. Wykonaj projekt briefingu strony internetowej. Wyszukaj przedsiębiorstwo produkcyjne, szkołę, przedszkole, sklep lub zakład usługowy. Przeprowadź wywiad lub ankietę oraz sporządź brief strony według własnego szablonu. Efekt swojej pracy umieść w pliku edytora tekstu i przedstaw nauczycielowi do oceny.

4. Wykonaj projekt makiety strony o rozdzielczości 800 px × 600 px, która ma się składać z następujących elementów:

- nagłówek zawierający logotyp po lewej stronie;
- poziomego menu z pięcioma przyciskami umieszczonego pod nagłówkiem;
- dwóch sekcji z treścią informacyjną pod menu po lewej i prawej stronie;
- stopki na dole strony.

Wykorzystaj do tego celu aplikację online Moqups.com lub inny program.

5. Dobierz odpowiednią paletę kolorów do poniższego zdjęcia (plik otrzymasz od nauczyciela).



6. Otwórz w przeglądarce stronę twojej szkoły. Określ paletę kolorów wykorzystaną do jej wykonania. Wskaż na opisanej palecie kolor dominujący.
7. Pobierz i zainstaluj aplikację Adobe XD. Wykonaj projekt szablonu strony internetowej, który spełni wymagania klienta określone w briefie z zadania 3.
8. Wyszukaj w sieci Internet przykłady stron, na których – twoim zdaniem – niewłaściwie dobrano czcionki i ich style. Przedyskutuj swoje spostrzeżenia z resztą klasy i nauczycielem.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Na czym polega projektowanie makiety witryny internetowej?
2. Co to jest plasterki i do czego służy?
3. Dlaczego projekt graficzny strony wykonuje się w edytorze grafiki, np. Photoshop?
4. Czym charakteryzuje się monochromatyczny standard kolorystyczny oparty na kole kolorów?
5. Wymień podstawowe elementy briefu strony internetowej.
6. Na jakich stronach i aplikacjach internetowych nie należy wykorzystywać czcionek szeryfowych?
7. Wyjaśnij rolę koloru dominującego na stronie internetowej.

30

Wykonywanie stron zgodnie z projektem

ZAGADNIENIA

- Etapy tworzenia stron zgodnie z projektem
- Zasady realizacji projektów stron WWW

Po wykonaniu projektu graficznego strony internetowej musimy określić jej strukturę fizyczną. Na podstawie struktury logicznej założonej w briefie oraz opracowanej struktury graficznej tworzymy projekt nazw poszczególnych folderów i plików składających się na zasoby witryny internetowej. Uporządkuje to pracę podczas przekształcania projektu w dokumenty hipertekstowe, arkusze stylów, skrypty czy graficzne elementy strony.

Struktura logiczna treści to schemat, który pomaga nam ułożyć informacje tak, by były one czytelne dla potencjalnego użytkownika witryny. Może być wykonany jako lista w dokumencie tekstowym lub drzewo katalogowe. Klient, dla którego jest tworzona strona internetowa, powinien brać czynny udział w ustaleniu priorytetów dla struktury fizycznej.

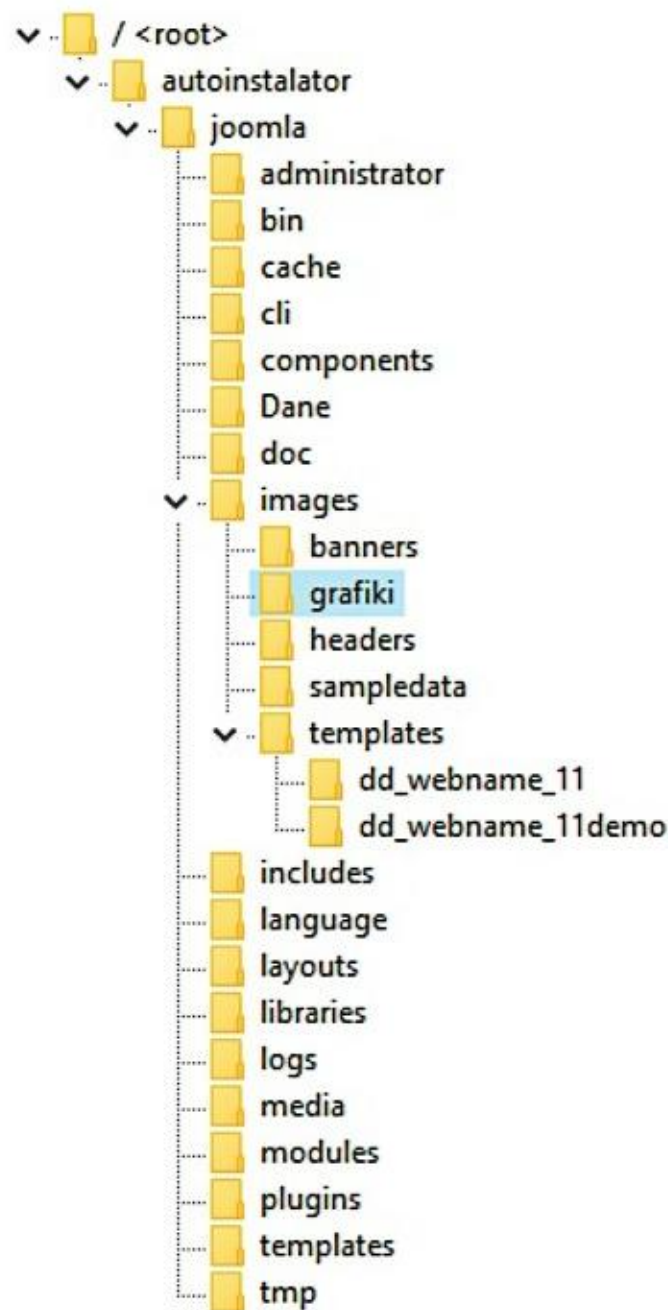
Po zatwierdzeniu struktury fizycznej można przystąpić do cięcia i eksportowania elementów graficznych z projektu graficznego oraz kodowania poszczególnych podstron i arkuszy stylów.

Przyjęta makieta strony pozwoli na zdefiniowanie w języku HTML odpowiednich kontenerów oraz nadaniu im określonych wymiarów i wyglądu w arkuszach CSS.

Plasterki umożliwią przygotowanie grafik przycisków, logo, symboli, banerów, ikon oraz elementów graficznych makiety strony. Kolorystyka, położenie oraz wymiary elementów posłużą do właściwego zakodowania elementów w języku HTML i CSS. Zaprojektowane treści można skopiować do dokumentów hipertekstowych. W miejscach określonych w projekcie kodujemy również osadzenie przygotowanych obiektów graficznych i multimedialnych.

Pocięty i zakodowany projekt zapisujemy w plikach oraz katalogach według przyjętej struktury fizycznej strony. W ten sposób zasoby strony są gotowe do publikacji na serwerze WWW.

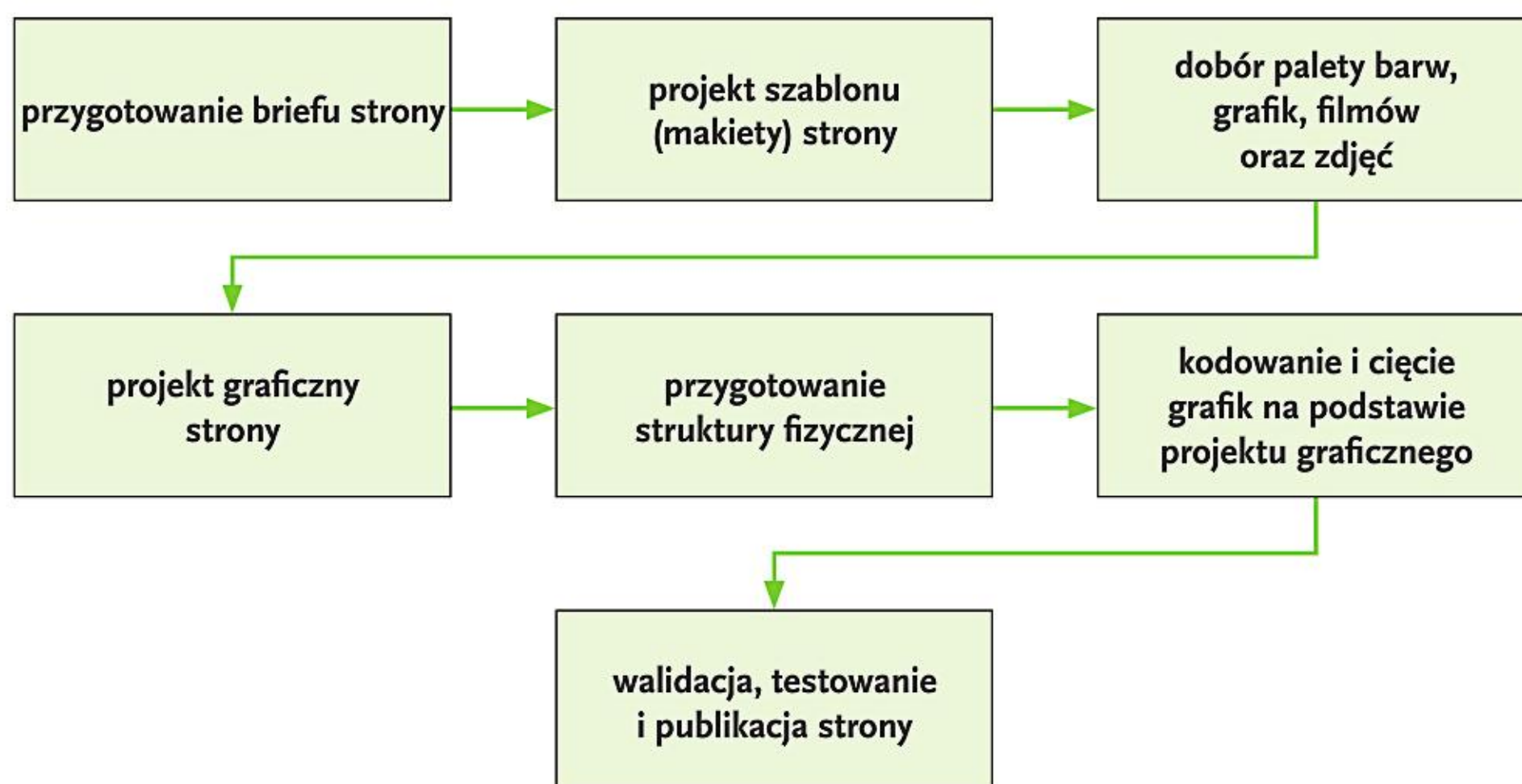
Po skopiowaniu gotowej struktury fizycznej na serwer WWW można rozpocząć proces testowania i optymalizacji witryny internetowej.



Nazwa	Rozmiar	Zmodyfikowany	Prawa	Właścic...
..				
str.png	3 KB	28.07.2016 14:33:50	rw-r--r--	
pismo.jpg	241 KB	27.07.2016 11:07:22	rw-r--r--	
logoFull.png	7 KB	27.07.2016 15:22:23	rw-r--r--	
logobip.png	4 KB	27.07.2016 15:46:10	rw-r--r--	
librus.png	3 KB	28.07.2016 14:29:15	rw-r--r--	
kontakt1.png	3 KB	28.07.2016 14:31:38	rw-r--r--	
kontakt.png	3 KB	28.07.2016 14:30:05	rw-r--r--	
h.jpg	70 KB	27.07.2016 11:07:35	rw-r--r--	
galerie.png	3 KB	28.07.2016 14:32:16	rw-r--r--	
fb.png	3 KB	28.07.2016 14:42:51	rw-r--r--	
curie.png.bmp	44 KB	27.07.2016 15:20:21	rw-r--r--	
bbip1.png	2 KB	12.01.2017 08:52:46	rw-r--r--	
bbip.png	3 KB	12.01.2017 08:49:52	rw-r--r--	

Rys. 30.1. Przykład struktury fizycznej CMS Joomla!

Projektowanie stron i aplikacji internetowych to wieloetapowy proces, który – po pominięciu aspektów technicznych i zapewnieniu hostingu – można podzielić na etapy przedstawione na rys. 30.2.



Rys. 30.2. Etapy procesu projektowania stron i aplikacji internetowych

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Na podstawie briefu strony, wykonanego w ćwiczeniu w poprzednim rozdziale, zaprojektuj i wykonaj logo na stronę klienta. Przedstaw klientowi (dla którego był przeznaczony briefing strony) wykonane logo. Opisz jego uwagi.
2. Na podstawie projektu graficznego strony, utworzonego w poprzednim rozdziale, wykonaj pliki witryny z wykorzystaniem języków HTML i CSS. Porównaj efekt końcowy, wyświetlony w przeglądarce, z projektem.
3. Wykonaj projekt struktury logicznej treści szkolnej strony internetowej.
4. Wykonaj w grupie projekt strony internetowej o dowolnej tematyce według zasad i wiadomości opisanych w rozdziale. Na podstawie projektu zakoduj pliki strony. Projekt wraz z plikami strony przedstaw nauczycielowi do oceny.

Uwaga! Zadania projektowe wymagają wsparcia nauczyciela na każdym etapie prac. Wszystkie szczegóły oraz zasady oceniania określa nauczyciel.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Na czym polega projektowanie struktury fizycznej witryny internetowej?
2. Wymień etapy projektowania i wykonywania strony internetowej.

X. Testowanie i optymalizowanie stron oraz aplikacji internetowych

- Testowanie stron internetowych
- Walidacja dokumentów hipertekstowych
- Publikacja stron i aplikacji internetowych
- Optymalizacja stron i aplikacji internetowych
- Bezpieczeństwo stron i aplikacji internetowych

31

Weryfikacja oraz walidacja stron internetowych

ZAGADNIENIA

- Weryfikacja a walidacja
- Testowanie stron internetowych
- Analiza wyników testów

Weryfikacja projektu, jakim jest proces tworzenia stron i aplikacji internetowych, służy sprawdzeniu jego zgodności z założeniami. Weryfikacja powinna być przeprowadzana na każdym etapie projektu.

Walidacja pomaga sprawdzić zgodność kodu ze standardami W3C oraz możliwościami przeglądarek internetowych lub urządzeń, na których produkt będzie użytkowany. Te dwie czynności pozwalają zyskać pewność, że program, aplikacja internetowa czy strona WWW będzie działać właściwie oraz że będzie spełniała wymagania użytkownika.

Najlepszą metodą weryfikacji jest **testowanie**. Powinno ono odbywać się zgodnie z harmonogramem utworzonym wcześniej. Dokument harmonogramu testów musi zawierać opis procesu testowania, czyli wyjaśnienie, w jaki sposób będzie przeprowadzany oraz kto będzie za niego odpowiedzialny. Plan testów oprogramowania bądź stron internetowych powinien być tak skonstruowany, by umożliwiać śledzenie wymagań użytkownika. Do planu testów trzeba również dodać zakres kryteriów, według których będzie oceniana poprawność produktu.

Z myślą o stronach WWW często stosuje się Wielokryterialną Metodę Oceny Serwisów Internetowych – **WAES** (ang. *Website Attribute Evaluation System*). Polega ona na binarnej weryfikacji kilkunastu kryteriów podzielonych na dwa obszary: przejrzystość oraz interaktywność / dostępność.

W odniesieniu do sposobu oraz liczby osób zaangażowanych w proces można wyróżnić testowanie:

- jednostkowe, gdy osoba testująca jest twórcą kodu;
- integracyjne i systemowe, gdy testuje się współpracę poszczególnych elementów całego projektu;
- wdrożeniowe, w którym uczestniczą końcowi użytkownicy.

Walidacja, czyli sprawdzanie poprawności technologicznej stron WWW, polega na ocenie następujących parametrów:

- czasu ładowania strony;
- weryfikacji działania odnośników wewnętrznych i zewnętrznych;

- weryfikacji działania formularzy;
- zgodności z wyszukiwarkami;
- zgodności z przeglądarkami;
- użycia elementów zdeprecjonowanych;
- poprawności językowej;
- błędów składni kodu.

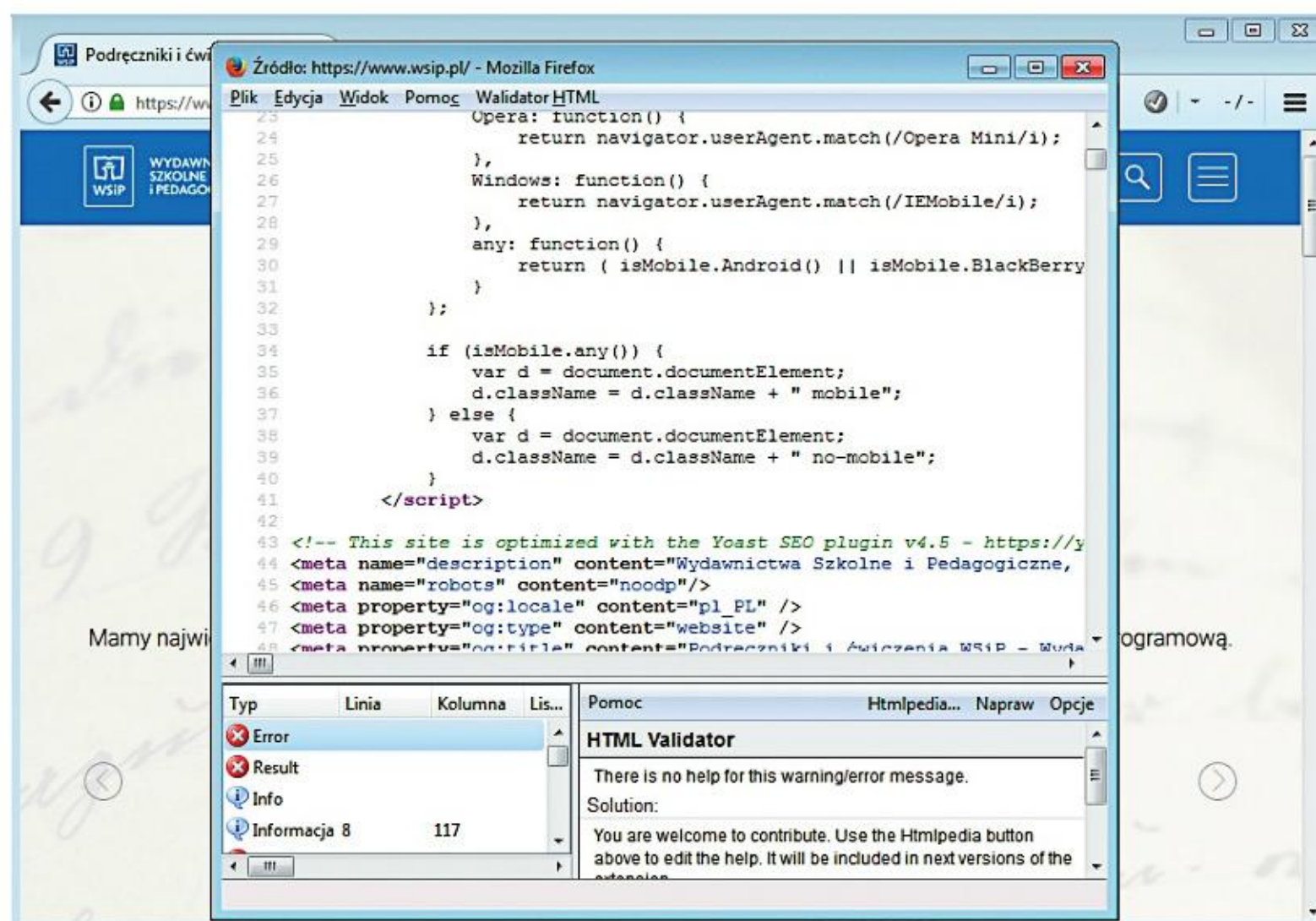
Walidację można przeprowadzać za pomocą skryptów na stronie (np. sprawdzanie poprawności danych wprowadzanych przez użytkownika) lub specjalnie do tego celu przygotowanych aplikacji lokalnych oraz online.

Pierwszy etap walidacji powinien się odbyć w trakcie pisania kodu aplikacji. Dotyczy to przede wszystkim poprawności użycia danych źródłowych w skryptach lub sprawdzania wypełnienia pól dowolnego formularza. Efektem właściwej walidacji powinna być informacja zwrotna wyświetlana na ekranie, informująca użytkownika o błędnym wprowadzeniu danych lub o braku błędów.

Testowanie błędów składni kodu można wykonać za pomocą parserów. **Parser** to program przeprowadzający analizę składniową danych wejściowych w celu określenia ich struktury gramatycznej w odniesieniu do określonej gramatyki wzorcowej. W walidacji stron internetowych można korzystać z aplikacji lokalnych, na serwerach zdalnych (W3C) oraz z wtyczek instalowanych w przeglądarkach internetowych.

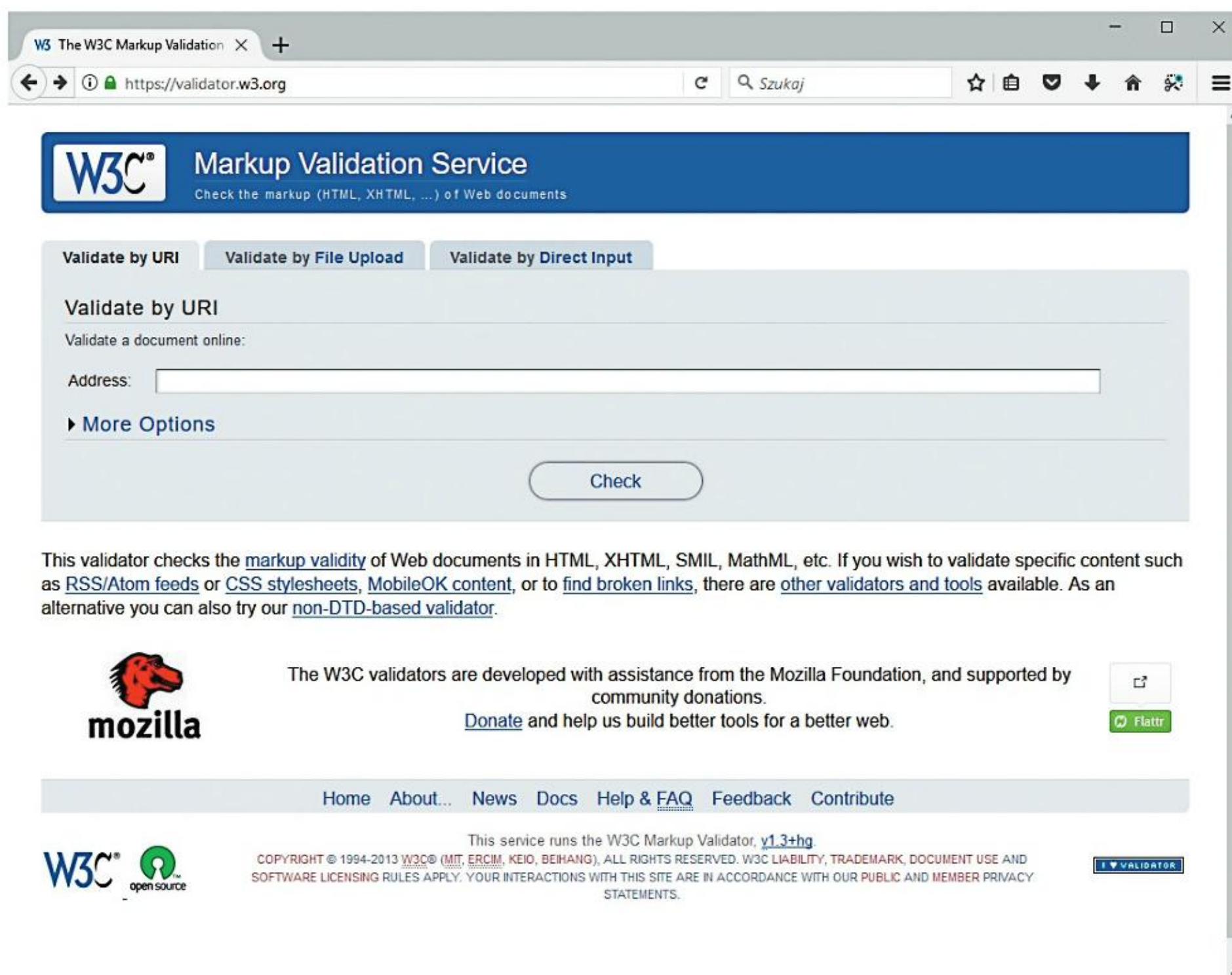
W przeglądarkach Firefox popularnym dodatkiem jest HTML Validator, który podczas instalowania w przeglądarce tworzy nowy przycisk do zarządzania walidacją. Umożliwia on korzystanie z dwóch parserów:

- HTML Tidy, który służy do sprawdzania składni języków HTML i XHTML;
- SGML Parser, który jest oparty na dokumentach DTD.



Rys. 31.1. Okno walidatora przeglądarki Firefox

Oprócz walidacji wykonywanej przez instalowane dodatki do przeglądarek, istnieje możliwość testowania kodu plików składających się na stronę internetową za pomocą walidatorów online W3C. Można w ten sposób testować dokumenty HTML, XHTML, SMIL, MathML i inne. Dodatkowo do dyspozycji są linki do narzędzi walidacji RSS, arkuszy CSS oraz odnośników.



Rys. 31.2. Okno walidatora W3C

Dokumenty można testować online w następujący sposób:

- przez podanie adresu URI dokumentu;
- przez wczytanie pliku przez formularz;
- przez wklejenie fragmentu kodu do okna na stronie.

Jeżeli strona przejdzie pomyślnie walidację przez stronę W3C, można umieścić w kodzie strony ikonę zgodności ze standardem W3C.

Innym sposobem walidacji online jest wykorzystanie aplikacji internetowej Dr. Watson. Jest ona dostępna pod adresem <http://watson.addy.com/>.

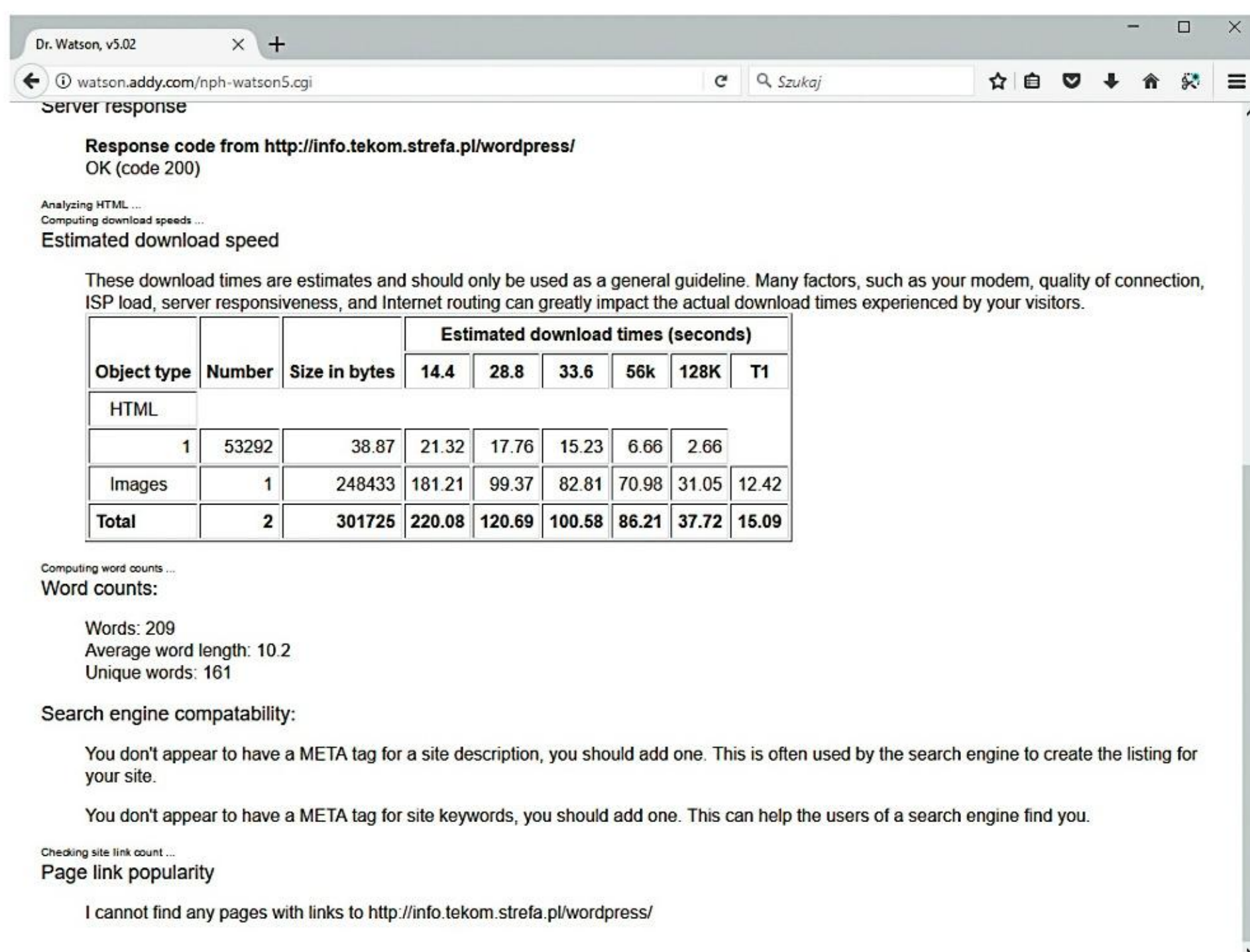
Dr. Watson sprawdza następujące parametry strony:

- czas ładowania strony dla różnych typów łącz;
- składnię kodu HTML;
- statystkę słów;

- liczbę słów, przeciętną długość słowa;
- sprawdzanie pisowni;
- weryfikację odnośników oraz odnośników do plików graficznych;
- poprawność znaczników META (ułatwiająca właściwe indeksowanie przez przeglądarki);
- popularność badanej strony;
- liczbę linków do badanej strony.

Testowanie strony polega na podaniu adresu URL strony oraz zaznaczeniu opcji walidacji.

Poza tym dobre efekty zawsze przynosi testowanie gotowej strony internetowej w różnych przeglądarkach internetowych. Pozwala to na wyłapanie anomalii w ich wyświetlaniu i umożliwia likwidowanie błędów na bieżąco.



Dr. Watson, v5.02

watson.addy.com/nph-watson5.cgi

Server response

Response code from <http://info.tekom.strefa.pl/wordpress/>
OK (code 200)

Analyzing HTML ...
Computing download speeds ...
Estimated download speed

These download times are estimates and should only be used as a general guideline. Many factors, such as your modem, quality of connection, ISP load, server responsiveness, and Internet routing can greatly impact the actual download times experienced by your visitors.

Object type	Number	Size in bytes	Estimated download times (seconds)					
			14.4	28.8	33.6	56k	128K	T1
HTML								
	1	53292	38.87	21.32	17.76	15.23	6.66	2.66
Images	1	248433	181.21	99.37	82.81	70.98	31.05	12.42
Total	2	301725	220.08	120.69	100.58	86.21	37.72	15.09

Computing word counts ...
Word counts:
Words: 209
Average word length: 10.2
Unique words: 161

Search engine compatability:
You don't appear to have a META tag for a site description, you should add one. This is often used by the search engine to create the listing for your site.
You don't appear to have a META tag for site keywords, you should add one. This can help the users of a search engine find you.

Checking site link count ...
Page link popularity
I cannot find any pages with links to <http://info.tekom.strefa.pl/wordpress/>

Rys. 31.3. Wynik walidacji aplikacją Dr. Watson

Istnieje też grupa programów offline do walidacji kodów źródłowych stron internetowych offline. Do takich programów zaliczamy:

- CSE HTML Validator,
- W3C Link Checker,
- Wave Accessibility Tool.

Testowanie stron internetowych na urządzenia mobilne

Na rynku narzędzi dla webmasterów pojawiają się rozwiązania pozwalające zoptymalizować i testować działanie witryn na urządzeniach mobilnych. Jednym z takich rozwiązań są przyspieszone strony mobilne AMP firmy Google.

Przyspieszone strony mobilne AMP (ang. *Accelerated Mobile Pages*) to, opracowany w ramach licencji open source, standard umożliwiający tworzenie stron internetowych, które wyświetlają się właściwie na urządzeniach mobilnych i cechują dużą szybkością wczytywania.

Jeżeli strona ma być zgodna ze standardem AMP, musi zawierać kilka dodatkowych informacji w kodzie HTML. Można nawet powiedzieć, że mamy do czynienia z nowym szablonem dokumentu hipertekstowego AMP HTML. Poniżej przedstawiono wzorzec dokumentu hipertekstowego zgodnego z AMP.

```
<!doctype html>
<html amp lang="en">
  <head>
    <meta charset="utf-8">
    <script async src="https://cdn.ampproject.org/v0.js"></script>
    <title>Hello, AMPs</title>
    <link rel="canonical" href="https://amp.dev/documentation/guides-and-tutorials/start/create/basic_markup/">
    <meta name="viewport" content="width=device-width,minimum-scale=1,initial-scale=1">
    <script type="application/ld+json">
      {
        "@context": "http://schema.org",
        "@type": "NewsArticle",
        "headline": "Open-source framework for publishing content",
        "datePublished": "2015-10-07T12:02:41Z",
        "image": [
          "logo.jpg"
        ]
      }
    </script>
  </head>
  <body>
    <h1>Hello, AMPs</h1>
  </body>
</html>
```



```

</script>
<style amp-boilerplate>body{-webkit-animation:-amp-start
8s steps(1,end) 0s 1 normal both;-moz-animation:-amp-start
8s steps(1,end) 0s 1 normal both;-ms-animation:-amp-start
8s steps(1,end) 0s 1 normal both;animation:-amp-start 8s
steps(1,end) 0s 1 normal both}@-webkit-keyframes -amp-start
{from{visibility:hidden}to{visibility:visible}}@-moz-keyframes
-amp-start{from{visibility:hidden}to{visibility:visible}}@-ms-keyframes
-amp-start{from{visibility:hidden}to{visibility:visible}}@-o-keyframes
-amp-start{from{visibility:hidden}to{visibility:visible}}@keyframes
-amp-start{from{visibility:hidden}to{visibility:visible}}</style>
<noscript><style amp-boilerplate>body{-webkit-animation:none;
-moz-animation:none;-ms-animation:none;animation:none}</style></noscript>
</head>
<body>
<h1>Welcome to the mobile web</h1>
</body>
</html>

```

Zgodnie z tym standardem każda strona powinna zawierać m.in.:

1. deklarację `<!doctype html>`
2. identyfikację AMP `<html amp>`
3. deklarację w nagłówku

```
<script async src="https://cdn.ampproject.org/v0.js"></script>
```

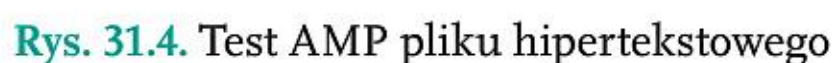
4. znacznik viewport, który pozwala ustawić opcje skalowania stron responsywnych

```
<meta name="viewport" content="width=device-width,minimum-scale=1">
```

5. wskazanie źródła zwykłej wersji HTML dokumentu

```
<link rel="canonical" href="$SOME_URL">
```

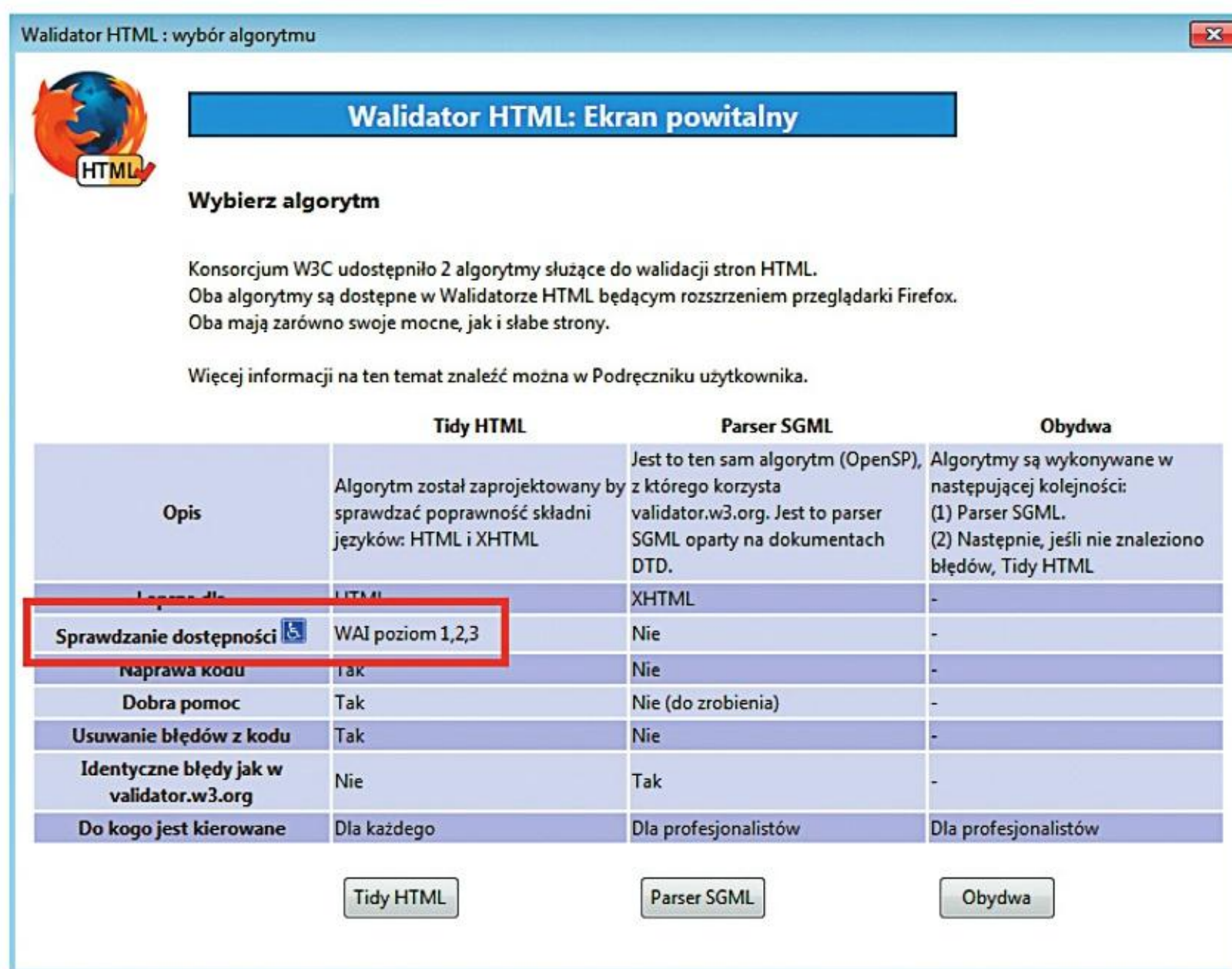
Testowanie witryn lub kodu plików hipertekstowych spełniających standard AMP jest możliwe w specjalnych walidatorach online, udostępnionych przez Google – Test AMP (<https://search.google.com/test/amp>). Na rys. 31.4 przedstawiono przykład testowania kodu strony w aplikacji Test AMP.



Rys. 31.4. Test AMP pliku hipertekstowego

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Pobierz i zainstaluj dowolny program do walidacji kodów źródłowych. Za jego pomocą sprawdź pliki twojej strony. Opisz otrzymane wyniki.
2. W opcji walidacji dokumentu hipertekstowego za pomocą parsera przeglądarki internetowej mamy do wyboru kilka opcji. Na rysunku zaznaczono element sprawdzany przez jeden z parserów. Skorzystaj z zasobów sieci Internet i wyjaśnij, jakie obszary kodu będą przez niego sprawdzane na każdym z trzech poziomów.



3. Wykonaj testowanie plików hipertekstowych strony internetowej za pomocą walidatora W3C online.
4. Wykonaj test AMP plików dowolnej strony internetowej. Opisz wyniki przeprowadzonych testów.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Jakie funkcje spełnia walidator?
2. Co oznacza skrót WAES?
3. Czym różni się walidacja od weryfikacji plików strony internetowej?
4. Wymień nazwy trzech programów do walidacji kodu stron internetowych.
5. Wymień znaczniki, który powinien mieć plik strony internetowej zgodny ze standardem AMP.

32

Pozycjonowanie, optymalizacja oraz dostępność stron i aplikacji internetowych

ZAGADNIENIA

- Pozycjonowanie strony WWW
- Optymalizacja kodu
- Wymagania dotyczące osób z niepełnosprawnościami

Głównym celem tworzenia stron internetowych jest dotarcie z określoną informacją lub ofertą do właściwego odbiorcy. Specyfika usługi WWW opiera się na wpisaniu odpowiedniego adresu URL w pasek przeglądarki internetowej. Dopiero wtedy mamy szansę na wczytanie i otwarcie plików interesującej nas witryny. Jeżeli użytkownik nie dysponuje adresem strony, ale zna jej tematykę lub zasoby, może wykorzystać wyszukiwarki internetowe do znalezienia właściwej witryny.

Wraz z rozwojem sieci Internet przybywa wyników w wyszukiwarkach. Każdy twórca witryny internetowej stara się zrobić wszystko, aby jego strona znalazła się na jednym z pierwszych miejsc wśród wyszukiwanych zasobów. Czynności, które mają doprowadzić do takiego celu, nazywa się **pozycjonowaniem strony** i **optymalizowaniem kodu** pod wyszukiwarki internetowe, w skrócie: **SEO** (ang. *Search Engine Optimization*).

Pozycjonowanie strony internetowej to zespół działań prowadzących do osiągnięcia najwyższej pozycji w wynikach wyszukiwarek internetowych dla wybranych słów i fraz kluczowych.

Przeglądanie zasobów sieci, zbieranie i indeksowanie wyszukanych informacji jest zadaniem specjalistycznych aplikacji zwanych robotami. Wyniki pracy robotów trafiają do bazy współpracującej z nimi wyszukiwarki i są prezentowane w postaci listy wyszukiwania. Na liście wyświetlają się informacje o znalezionych stronach internetowych w określonej kolejności, która jest wynikiem przydzielonej wagi. Waga jest przyznawana przez algorytm wyszukiwarki, który ocenia zgodność zawartości strony z wpisaną frazą do wyszukania.

Obecnie algorytmy wyszukiwarek rozpatrują ponad 200 niepowtarzalnych wskaźników, dzięki którym można odgadnąć, czego szuka użytkownik. Wśród nich są takie elementy, jak: wyrażenia używane na stronach, aktualność treści, region, a także PageRank. **PageRank** to metoda nadawania indeksowanym stronom internetowym określonej wartości liczbowej, oznaczającej ich jakość.

Rzadko się zdarza, aby użytkownik przeglądał wszystkie strony z listy wyszukiwania. Dlatego należy dążyć do tego, by nasza strona znalazła się możliwie najwyżej w rankingu wyszukiwania.

Do czynników wpływających na dobre pozycjonowanie strony internetowej należą m.in.:

- zastosowanie dużej liczby dobrej jakości hiperłączy wewnętrznych i zewnętrznych;
- wykonanie płynnej komunikacji między poszczególnymi stronami danej witryny;
- trafnie dobrane słowa i frazy kluczowe oraz ich odpowiednie rozmieszczenie w treści poszczególnych stron;
- optymalizacja kodu źródłowego poszczególnych stron;
- łączenie odsyłaczami naszej strony z innymi witrynami o innych adresach IP;
- zastosowanie złożonego tytułu strony internetowej, np.:

<title> Wydawnictwa Szkolne i Pedagogiczne WSiP – podręczniki, ćwiczenia, książki, sklep internetowy</title>;

- marketing naszej strony w mediach społecznościowych;
- stosowanie opisów hiperłączy.

Aby poprawić indeksowanie stron, można wykorzystać narzędzia oferowane webmasterom przez same wyszukiwarki. Jednym z nich jest **Google Search Console**. Jest to darmowe rozwiązanie pozwalające na dodanie nazwy domeny lub **mapy strony** do algorytmu indeksowania wyszukiwarki Google. Można dodać domenę lub prefix adresu URL. Jeżeli wybierzemy pierwszą metodę, trzeba wpisać nazwę domeny, a następnie dodać wygenerowany rekord tekstowy do opcji zarządzania wpisami DNS w panelu administratora domeny usługi hostingowej (rys. 32.1).

DOMENY

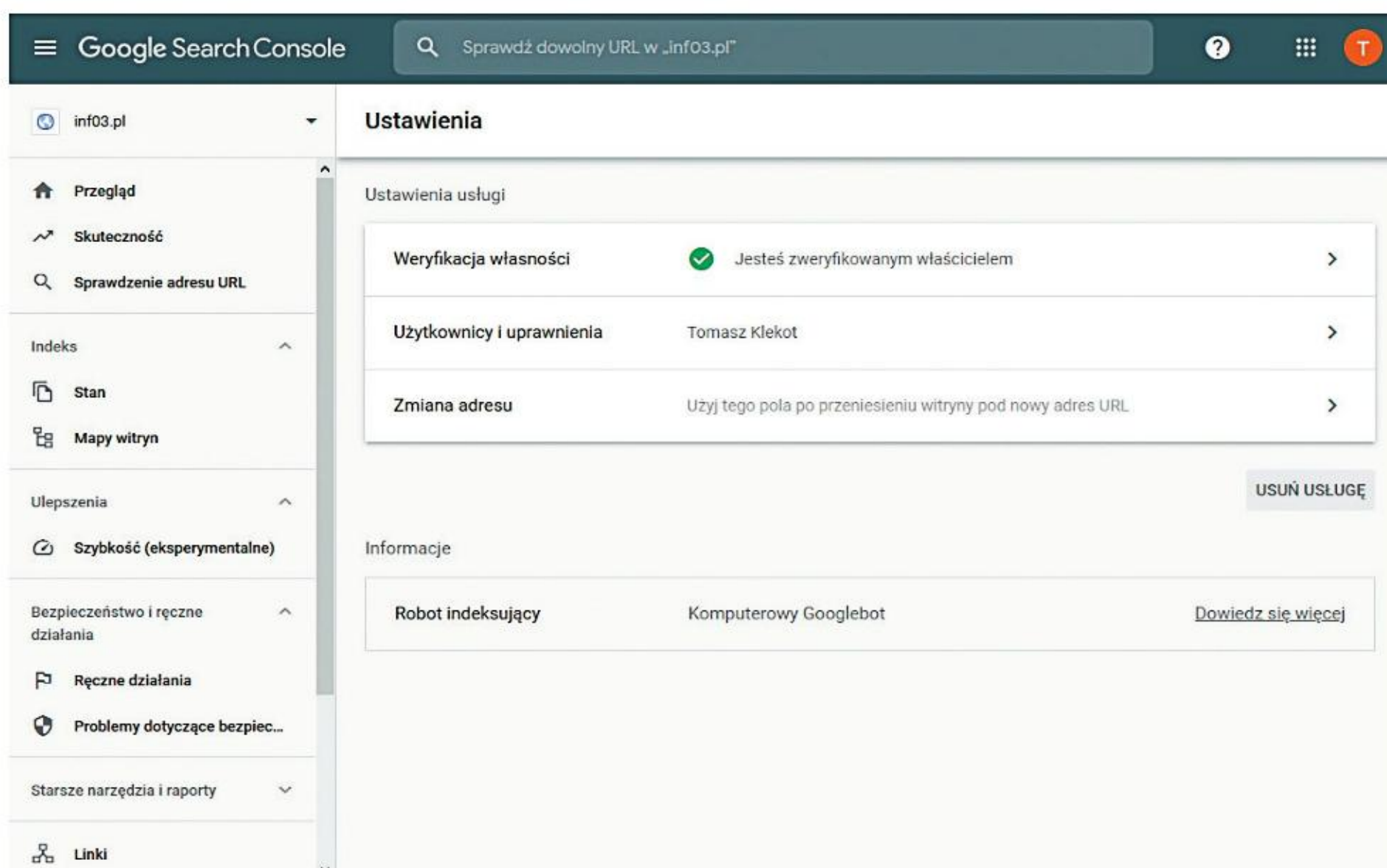
Zarządzaj rekordami DNS dla domeny inf03.pl

[Dodaj nowy rekord](#) [Usuń](#)

☐ Typ rekordu	Host	Dane	TTL	
☐ MX	inf03.pl.	10 inf03.pl.	3600	Opcje
☐ A	inf03.pl.	46.242.239.147	3600	Opcje
☐ TXT	inf03.pl.	google-site-verification=U1v471hsek_pl3118Bk5vBcWukfdR4cgsY6vUYD6OA8	3600	Opcje
☐ TXT	inf03.pl.	v=spf1 mx a ptr ~all	3600	Opcje

Rys. 32.1. Dodanie rekordu weryfikacji domeny w Google

Następnie należy dokończyć weryfikację domeny w Google Search Console za pomocą przycisku **Weryfikuj**. Na rys. 32.2 przedstawiono ostateczny wynik weryfikacji domeny inf03.pl w Google Search Console.



Rys. 32.2. Proces dodania i weryfikacji domeny przez rekord DNS

Mapa witryny to plik, w którym podaje informacje o podstronach, grafikach i innych plikach w danej witrynie. Wyszukiwarki, takie jak Google, odczytują ten plik, by skuteczniej indeksować strony. Mapa witryny informuje robota, które pliki w witrynie są dla twórcy ważne.

W systemach zarządzania treścią CMS można również zauważyć narzędzia wspomagające twórców stron internetowych w pozycjonowaniu własnych projektów. Na rys. 32.3 przedstawiono okno konfiguracji metadanych witryny. Metadane są niezbędne do właściwej interpretacji treści strony internetowej przez roboty wyszukiwarek internetowych. W związku z tym takie dane trzeba umieścić w odpowiednim miejscu strony. W zapleczu CMS Joomla! mamy możliwość konfiguracji następujących opcji:

- **opis witryny** – umieszczamy tu informacje dotyczące treści znajdujących się na naszej stronie;
- **słowa kluczowe witryny** – te słowa, które będą według autora najczęściej używane w artykułach i zgodne z tematyką strony;
- **wyszukiwarki** – ustawienia dotyczą sposobu indeksowania przez roboty wyszukiwarek treści na naszej stronie; aby robot wyszukiwarki indeksował stronę, należy wybrać odpowiednią opcję (widoczną na rysunku); poprawi to pozycjonowanie witryny;
- **prawa do treści** – można tu podać licencję, której treść jest rozpowszechniana na stronie, lub dane autora;
- **autor w metadanych** – można wybrać opcje „tak” lub „nie”, zależnie od tego, czy autor artykułu ma być widoczny dla robotów indeksujących, czy – nie.

Metadane

Opis witryny	
Słowa kluczowe witryny	
Wyszukiwarki	Indeksuj strony i indeksuj ws... ▼
Prawa do treści	
Autor w metadanych	☒ Tak ☐ Nie
Pokaż wersję Joomla	☐ Tak ☒ Nie

Rys. 32.3. Okno konfiguracji metadanych w CMS Joomla!

Optymalizacja strony internetowej może również dotyczyć jej responsywności. **Responsywny design** (ang. *responsive web design*) określa stronę internetową, która dostosowuje się automatycznie do wielkości ekranu, na którym jest wyświetlana.

Do zalet responsywnej strony internetowej należą:

- łatwość oglądania, ponieważ wielkość liter i obrazy dopasowują się do wyświetlacza;
- potencjalnie większa liczba odwiedzin, ponieważ każdy może otworzyć stronę w dowolnym miejscu i czasie na dowolnym urządzeniu;
- oszczędności dla projektu strony, ponieważ nie trzeba dodatkowo inwestować w kosztowną wersję mobilną lub specjalne aplikacje.

Responsywność można wprowadzać już od etapu projektowania, aż po etap tworzenia kodu strony w języku HTML i kaskadowych arkuszy stylów CSS.

W dokumentach hipertekstowych istnieje opcja użycia metaznaczników zapewniających właściwe wyświetlanie strony na urządzeniach mobilnych.

PRZYKŁAD 32.1

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

Poza tym w odpowiedni sposób należy sformatować poszczególne elementy strony za pomocą arkuszy CSS. Przykładowo: szerokość bloków powinno podawać się w procentach. Trzeba też korzystać z selektorów i właściwości pozwalających na automatyczną zmianę stylu zależnie od rozdzielczości ekranu.

PRZYKŁAD 32.2

Na rysunku przedstawiono przykład kodu strony internetowej, która wybiera styl zależnie od zmiany wielkości okna przeglądarki. Przy dużym oknie wyświetla się obraz o większych rozmiarach, natomiast gdy okno przeglądarki będzie mniejsze niż 400 px, zmieni się również rozmiar wyświetlanego obrazu.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta name="viewport" content="width=device-width, initial-scale=1.0">
5  <style>
6    /* Dla rozdzielczości poniżej 400px: */
7    body {
8      background-repeat: no-repeat;
9      background-image: url('mały_kotek.jpg');
10   }
11
12   /* Dla rozdzielczości powyżej 400px: */
13   @media only screen and (min-width: 400px) {
14     body {
15       background-image: url('kotek.jpg');
16     }
17   }
18 </style>
19 </head>
20 <body>
21   <p style="margin-top:300px;">Przykład responsywności grafiki</p>
22 </body>
23 </html>

```

Rys. 32.4. Przykład kodu, który dopasuje obraz do rozdzielczości ekranu przeglądarki

Optymalizacja strony to także dopasowanie jej treści do aktualnych wytycznych prawnych lub obsługi przez osoby z niepełnosprawnościami.

Aspekty prawne mogą dotyczyć elementów, logotypów lub odsyłaczy, które muszą się pojawić na stronach internetowych określonych jednostek organizacyjnych. Przykładowo: na stronach internetowych instytucji podległych rządowi lub samorządom lokalnym muszą znaleźć się takie elementy, jak godło państwowe i odsyłacz do biuletynu informacji publicznej (BIP). Ich brak może spowodować konsekwencje prawno-finansowe dla właściciela strony.

Ważnym aspektem interfejsu obsługi publicznych stron internetowych jest uwzględnienie elementów wspomagających korzystanie z niej przez osoby z niepełnosprawnościami. Często takie ułatwienia nazywa się dostępnością strony internetowej.

Dostępność strony internetowej jest spełniona wtedy, gdy wszyscy użytkownicy mają pełny dostęp do jego treści, mogą te treści zrozumieć, a wygodna nawigacja umożliwia logiczną i intuicyjną interakcję z witryną.

W Rozporządzeniu Rady Ministrów z dnia 12 kwietnia 2012 r. w sprawie Krajowych Ram Interoperacyjności, minimalnych wymagań dla rejestrów publicznych i wymiany informacji w postaci elektronicznej oraz minimalnych wymagań dla systemów teleinformatycznych określono, że od czerwca 2015 r. serwisy publiczne muszą spełniać wymagania dostępności.

Powołano się przy tym na specyfikację **WCAG** (ang. *Web Content Accessibility Guidelines*) w wersji 2.0. Tę specyfikację, przygotowaną przez organizację W3C, uznaje się za najważniejszy dokument określający zasady dostępności (DzU z 2019 r., poz. 848; DzU UE L 327 z 2.12.2016).

Przykładem dostępności na stronie internetowej jest umieszczanie narzędzi na stronie do dopasowywania sposobu wyświetlania treści przez użytkownika oraz dodanie wskaźnika fokusu lub odtwarzanie audio treści artykułów.



Rys. 32.5. Przykład elementów dla osób z niepełnosprawnościami oraz informacja publiczna BIP

Wskaźnik fokusu to wizualne oznaczenie elementu interaktywnego (np. odnośnika bądź przycisku), który w danym momencie ma fokus. Na ogół wskaźnik fokusu ma postać obramowania otaczającego dany element. Jest niezbędny dla osób korzystających wyłącznie z klawiatury.



Rys. 32.6. Przykład zastosowania wskaźnika fokusu

Niedostosowanie stron do wymagań osób słabowidzących może być spowodowane również przez niewystarczający kontrast między tłem a tekstem.

PRZYKŁAD 32.3

Aby otrzymać wskaźnik fokusu, można przeprowadzić formatowanie elementu w kaskadowym arkuszu stylów CSS według wzoru:

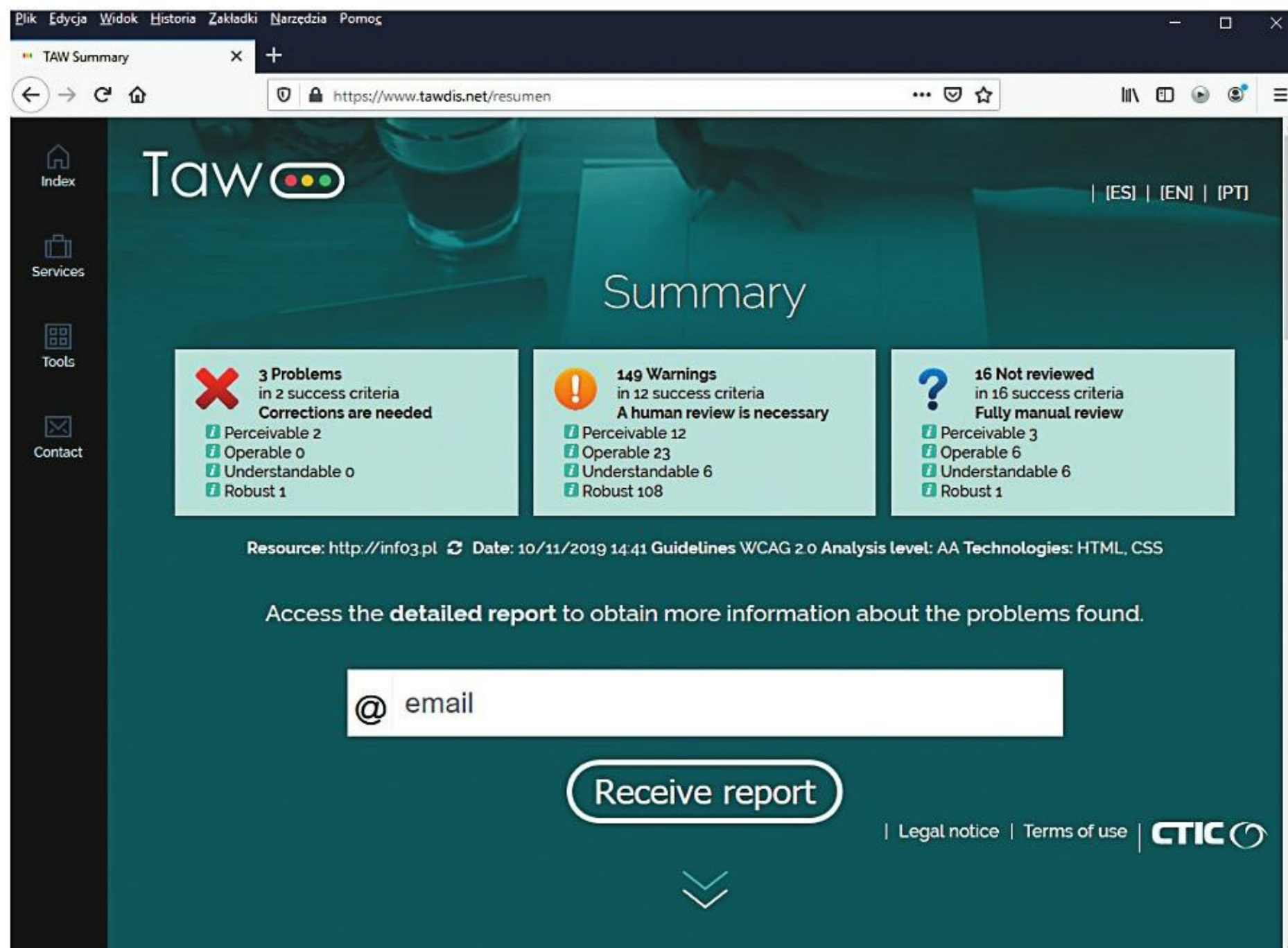
```
button {
  color: #FFF100;
  background: #000000;
  border: 2px solid #000000;
}
button:focus {
  color: #000000;
  background: #FFF100;
  border-style: dashed;
}
```


Tego typu elementy należy wziąć pod uwagę na etapie wstępnej analizy wytycznych do projektu strony internetowej.

Pozostałe zalecenia mające wprowadzać zalecenia specyfikacji WCAG 2.0 stanowią, że:

- wszystkie elementy graficzne powinny mieć tekst alternatywny;
- wszystkie pliki wideo z dźwiękiem powinny być uzupełnione napisami;
- nawigacja w obrębie całego serwisu powinna być dostępna z poziomu klawiatury;
- kontrast kolorystyczny wszystkich elementów przekazujących treść musi mieć stosunek jasności tekstu do tła co najmniej 4,5 do 1;
- wszystkie tytuły podstron muszą być unikalne i informować o treści podstrony;
- wszystkie podstrony powinny być oparte o nagłówki;
- do porządkowania treści w tekstach bądź elementów nawigacji należy wykorzystywać listy nieuporządkowane i uporządkowane;
- wszystkie skrypty i aplety powinny być dostępne dla osób niewidomych i osób korzystających wyłącznie z klawiatury;
- wszystkie pola formularzy i przyciski stosowane w formularzach muszą zostać właściwie opisane;
- strona powinna być dostępna w przeglądarkach i urządzeniach z wyłączoną obsługą arkuszy CSS.

W celu sprawdzenia zgodności strony ze specyfikacją WCAG przeprowadza się specjalistyczny audyt. W tym celu można wykorzystać gotowe narzędzia do walidacji kodu.



Rys. 32.7. Walidacja kodu strony pod względem zgodności ze specyfikacją WCAG 2.0

Przykładowo: **T.A.W. Validator WCAG 2.0** to darmowe narzędzie online do analizy dostępności stron internetowych. Jest ono intuicyjne w obsłudze, a wyniki z przeprowadzanych analiz prezentuje się w formie przystępnej graficznie. Wśród funkcji istnieje także możliwość sprawdzenia strony pod kątem urządzeń mobilnych. Analizę strony można przeprowadzić pod adresem <https://www.tawdis.net>. Szczegółowy raport z testu warto przesłać na podany adres e-mailowy. Na rys. 32.7 przedstawiono przykład audytu WCAG 2.0 strony INF03.pl.

Dodatkowo w *Ustawie z dnia 4 kwietnia 2019 r. o dostępności cyfrowej stron internetowych i aplikacji mobilnych podmiotów publicznych* w art. 10 zobowiązano podmioty publiczne do sporządzania i publikowania **deklaracji dostępności**. Celem publikowania na stronach internetowych deklaracji dostępności jest umożliwienie zapoznania się z informacjami dotyczącymi dostępności podmiotu publicznego dla osób z niepełnosprawnościami. Deklaracja zawiera informacje nie tylko o stronie internetowej, lecz także o urzędzie bądź innej jednostce publicznej, dostępie do tłumacza języka migowego i koordynatorze dostępności.

SPRAWDŹ SVOJE UMIEJĘTNOŚCI

1. Uzupełnij metadane w konfiguracji swojego systemu zarządzania treścią CMS Joomla! (lub innego).
2. Zapisz trzy propozycje nazwy adresu domenowego drugiego poziomu dla strony internetowej twojej szkoły. Sprawdź na stronie internetowej Krajowego Rejestru Domen, czy są one dostępne.
3. Otwórz stronę internetową swojej szkoły. Wykonaj jej audyt dotyczący zgodności ze specyfikacją WCAG 2.0. Wykorzystaj do tego celu narzędzia online. Końcowy raport wyślij na swój adres e-mail. Podziel się wnioskami z audytu z nauczycielem oraz pozostałymi uczniami w grupie.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Jakie znaczenie w pozycjonowaniu strony mają słowa kluczowe?
2. Dlaczego stosowanie wskaźnika fokusu może zwiększyć dostępność strony internetowej?
3. Co to jest PageRank?
4. Wymień czynniki, które mogą wpłynąć na lepsze pozycjonowanie strony internetowej.
5. Na czym polega audyt dostępności strony internetowej?
6. Jakie obowiązki nakłada na strony organizacji publicznych (np. szkoły) rozporządzenie dotyczące krajowych ram interoperacyjności?
7. Wymień 10 zasad, których należy przestrzegać podczas budowy dostępnej strony internetowej.

33

Bezpieczeństwo stron internetowych

ZAGADNIENIA

- Metody zabezpieczania stron internetowych
- Certyfikaty SSL

Zasady bezpieczeństwa stron internetowych

Na każdym etapie prac związanych z tworzeniem stron internetowych kluczową rolę odgrywa ich bezpieczeństwo. Z jednej strony jest to spowodowane czasochłonną i kosztowną naprawą witryny po ataku hakerskim lub po uszkodzeniu sprzętu serwera. Z drugiej strony wynika to z wytycznych prawnych dotyczących ochrony danych osobowych lub transakcyjnych.

Rozważając aspekty bezpieczeństwa stron internetowych, musimy zwrócić uwagę na kilka zasad.

- **Polityka haseł.** Nigdy nie należy dopuścić do tego, aby hasła używane przez administratorów, moderatorów czy użytkowników, którzy mogą ingerować w działanie witryny, nie spełniały zasad silnych haseł. Można korzystać z pomocy generatorów i menedżerów haseł.
- **Polityka poziomów dostępu.** Należy umiejętnie organizować pracę poszczególnych grup użytkowników przez odpowiednie dostosowywanie uprawnień do zasobów.
- **Wysokiej jakości usługi hostingowe.** Należy korzystać wyłącznie ze sprawdzonych rozwiązań sprzętowych i usług hostingowych najwyższej jakości.
- **Kopia bezpieczeństwa.** Bieżące wykonywanie kopii bezpieczeństwa jest obowiązkiem każdego odpowiedzialnego administratora. Najlepiej byłoby, gdyby archiwalne pliki znajdowały się na dyskach, które nie mają bezpośredniego połączenia z siecią lokalną lub globalną.
- **Aktualizacja technologii webowych.** W trakcie eksploatacji strony należy zadbać o aktualizowanie poszczególnych elementów serwera WWW i innych wykorzystywanych technologii webowych (interpretera PHP, systemu CMS, oprogramowania antywirusowego itp.).
- **Zastosowanie protokołów szyfrowania informacji.** Dobrą praktyką jest wykorzystywanie metod szyfrowania połączeń podczas korzystania ze stron i aplikacji internetowych. Przykładem tego typu jest stosowanie protokołów HTTPS i certyfikacji SSL.
- **Stosowanie narzędzi diagnozujących pliki strony.** Można realizować tę zasadę przez stosowanie walidatorów, programów antywirusowych czy innych narzędzi testujących poprawność działania strony (np. VirusTotal).

Certyfikat SSL dla własnej domeny

W 1994 r. przedsiębiorstwo Netscape stworzyło protokół służący do bezpiecznej transmisji zaszyfrowanych danych za pomocą algorytmów szyfrujących – SSL (ang. *Secure Socket Layer*). Ta technologia szyfruje i chroni dane przesyłane za pomocą protokołu **HTTPS**. Aby skorzystać z tej technologii, należy wygenerować **certyfikat SSL**. Certyfikacja SSL pełni trzy zasadnicze funkcje w procesie zabezpieczania komunikacji:

- zapewnia poufność przesyłanych informacji przez ich szyfrowanie;
- pozwala zweryfikować tożsamość stron komunikacji;
- gwarantuje integralność otrzymanych danych.

Certyfikat to zbiór danych jednoznacznie identyfikujących jednostkę pozwalający stwierdzić, czy osoba, która się nim legitymuje, jest rzeczywiście tym, za kogo się podaje. Jest on potwierdzony przez zaufaną organizację, zwaną w protokole SSL – CA (ang. *certificate authority*). Każdy certyfikat zawiera:

- nazwę certyfikowanego obiektu (osobę lub komputer),
- identyfikator obiektu,
- klucz publiczny obiektu,
- czas ważności certyfikatu,
- nazwę wystawcy certyfikatu,
- identyfikator oraz podpis wystawcy.

Uwierzytelnienie certyfikatu SSL powoduje utworzenie dwóch kluczy niezbędnych do ustanowienia połączenia szyfrowanego i transmisji danych. Są to:

- **klucz prywatny**, który jest instalowany na serwerze; jest odpowiedzialny za utworzenie pieczęci certyfikacji na domenę lub stronę;
- **klucz publiczny**, który tworzy drugi element certyfikatu SSL; pozwala szyfrować dane użytkowników; dane te są szyfrowane przed ich wysłaniem.

Klucz prywatny jest kluczem lustrzanym i tylko on może odszyfrować informacje.

Na rys. 33.1 przedstawiono sposób działania certyfikatu SSL.



Rys. 33.1. Działanie połączenia szyfrowanego SSL

Zabezpieczenie stron i aplikacji internetowych podłączonych do danej domeny przekłada się nie tylko na zwiększone bezpieczeństwo danych i kontrolę ich transmisji, lecz także na zaufanie internautów do treści witryny oraz autorów witryny. Co więcej, witryna zabezpieczona certyfikatem SSL jest lepiej pozycjonowana w wyszukiwarkach.

Instalacja certyfikatu SSL w domenie

Aby korzystać z certyfikatu SSL na swojej stronie internetowej, należy go najpierw wygenerować, a następnie zainstalować i aktywować na serwerze. W komercyjnych usługach hostingowych certyfikaty SSL często są płatne. Wtedy dochodzi proces zamówienia oraz opłaty za certyfikat. Obecnie na rynku są dostępne następujące typy certyfikatów SSL:

- **certyfikat z walidacją domeny DV** (ang. *Domain Validation*); to podstawowy typ certyfikatu, który gwarantuje pełne bezpieczeństwo, poufność i integralność transmisji danych zapewniane przez technologię SSL, ale nic nie mówi o właścicielu, który się nim posługuje; w certyfikacie DV jest zapisana jedynie informacja o domenie, którą chroni;
- **certyfikat z walidacją organizacji OV** (ang. *Organization Validation*); potwierdza, że informacje o właścicielu, dla którego wystawiono certyfikat, zostały potwierdzone przez zaufane centrum certyfikacyjne; dzięki temu dane organizacji mogą być zawarte w certyfikacie, a użytkownik może je po prostu wyświetlić i zweryfikować;
- **certyfikat z rozszerzoną walidacją podmiotu EV** (ang. *Extended Validation*); jest on stawiany poziom wyżej; gwarantuje, że podmiot wnioskujący został poddany jeszcze bardziej szczegółowej weryfikacji; nazwa firmy, dla której wystawiono certyfikat, zostanie wyświetlona na zielonym polu w pasku adresu przeglądarki internetowej.

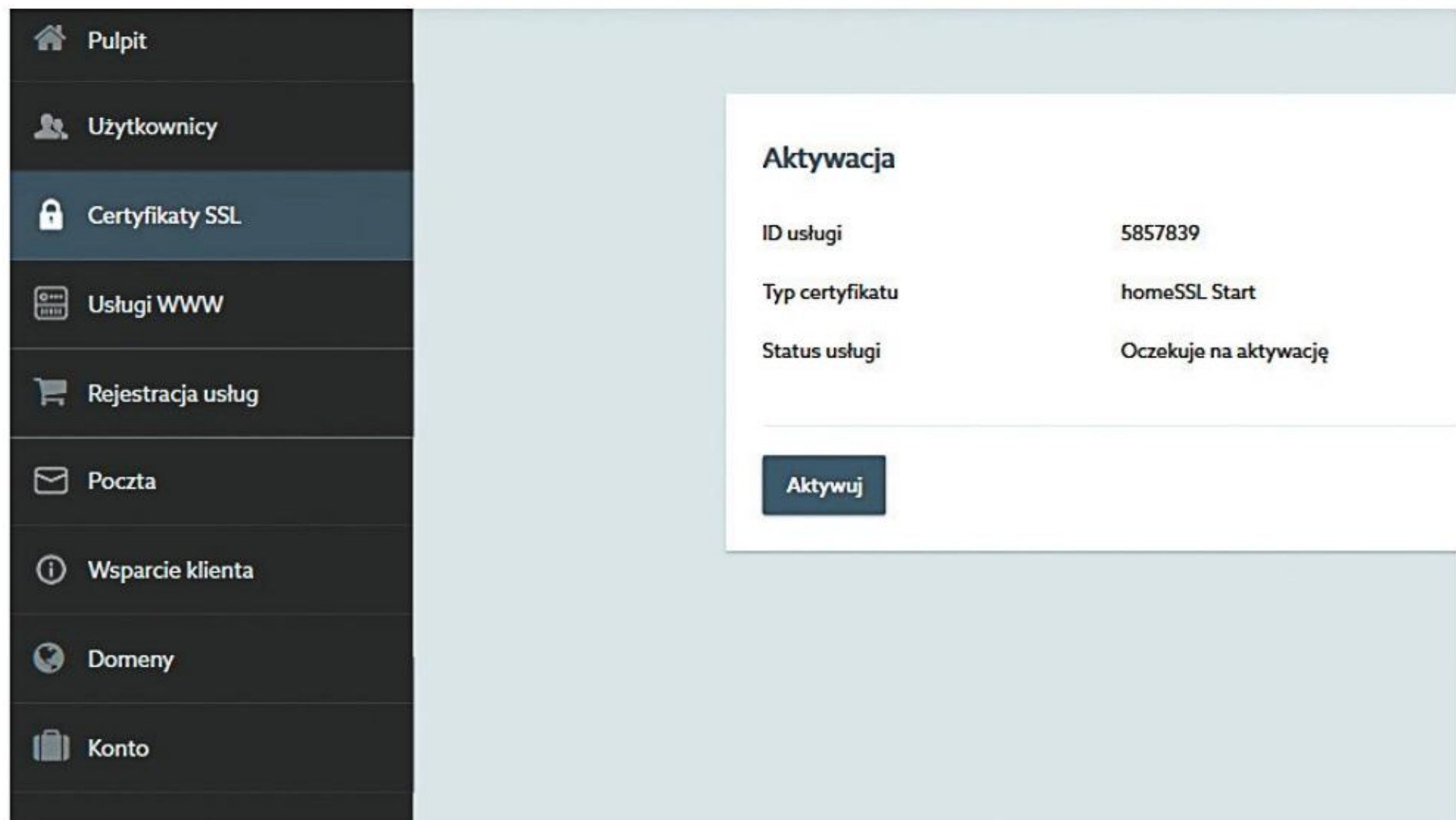
PRZYKŁAD 33.1

Na potrzeby naszego przykładu wykupiliśmy standardowy certyfikat DV w usłudze hostingowej, w której instalowaliśmy system CMS Joomla!. Strona przed instalacją certyfikatu dla domeny, po uruchomieniu w przeglądarce, wyświetlała się z przekreśloną kłódką na pasku adresu (rys. 33.5).

Po złożeniu i opłaceniu zamówienia w panelu administratora serwera pojawiła się informacja o certyfikacie SSL przygotowanym do aktywacji (rys. 33.2).

Po rozpoczęciu aktywacji certyfikatu zostaje na adres e-mail właściciela usługi wysłana wiadomość od organizacji certyfikującej z prośbą o potwierdzenie tożsamości (rys. 33.3).

Po weryfikacji domeny pojawi się w panelu serwera usługi hostingowej wygenerowany certyfikat, gotowy do instalacji w domenie. Po wybraniu procesu instalacji certyfikat zostanie dołączony do naszej domeny. Zakończony proces będzie można zweryfikować po zmienionym kolorze okna certyfikatu. Na rys. 33.4 po lewej stronie pokazano okno przed instalacją certyfikatu na serwerze domeny INF03.pl. Po prawej stronie jest widoczne okno po instalacji certyfikatu.



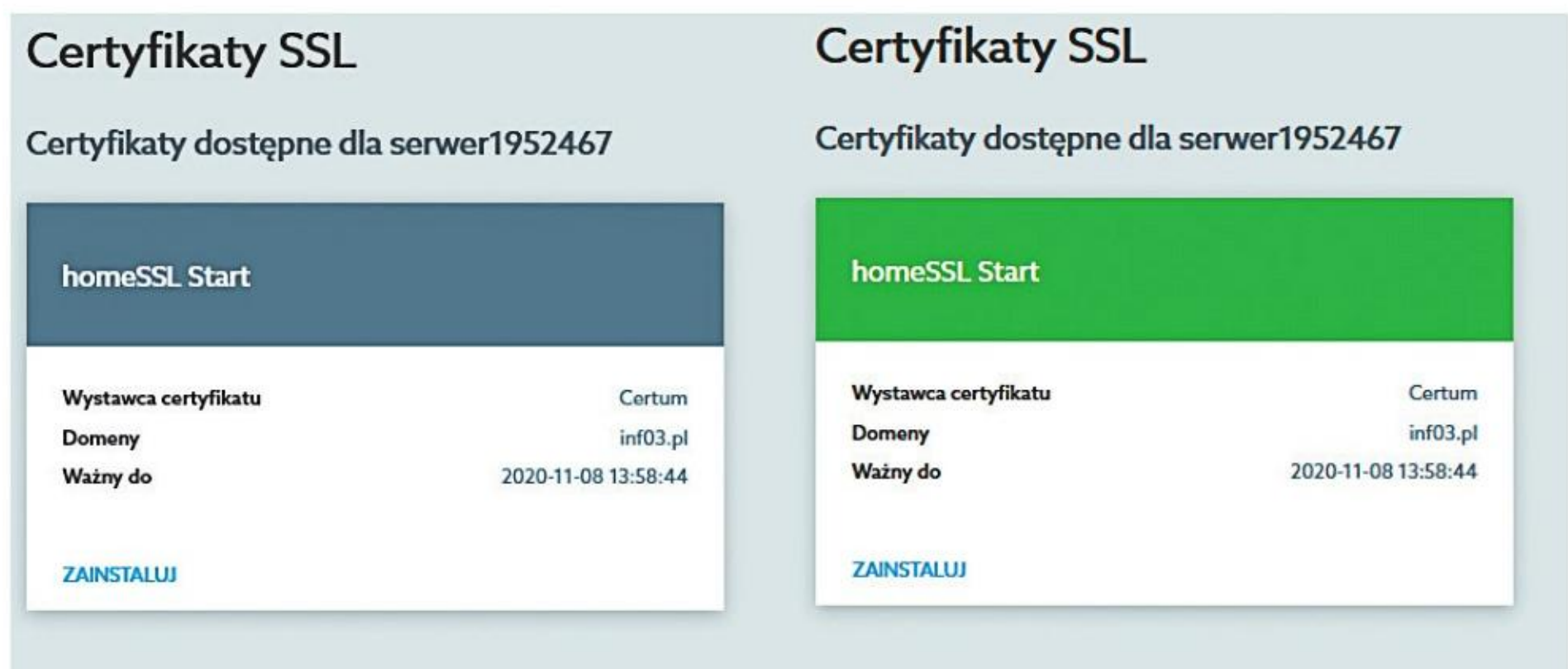
Rys. 33.2. Okno certyfikatów SSL



 Domena została zweryfikowana

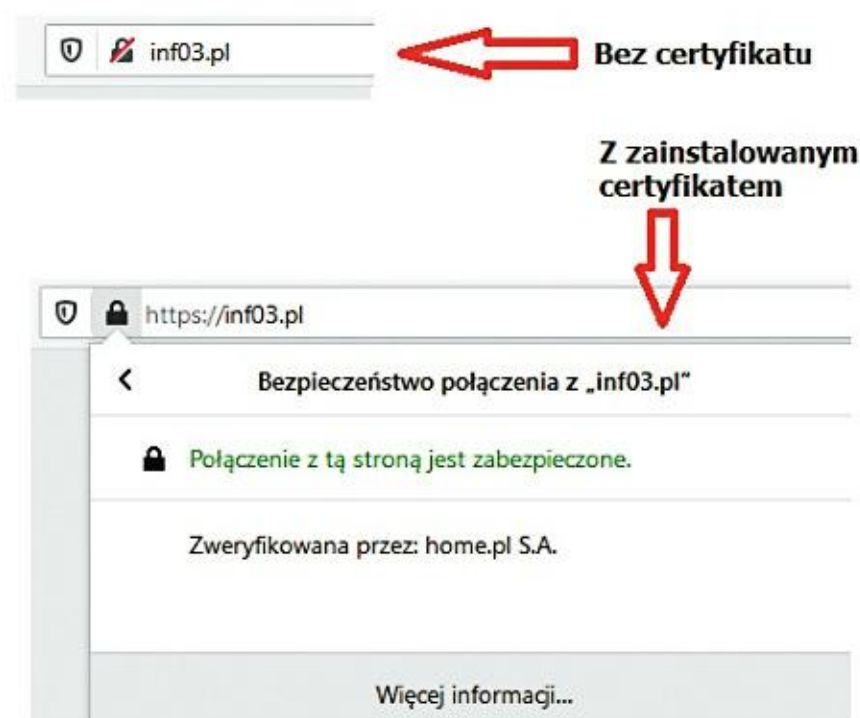
© 2010-2018 by Asseco Data Systems S.A. | Projekt [Ideacto](#) | Wdrożenie [Divante](#)

Rys. 33.3. Weryfikacja właściciela domeny przez organizację certyfikującą



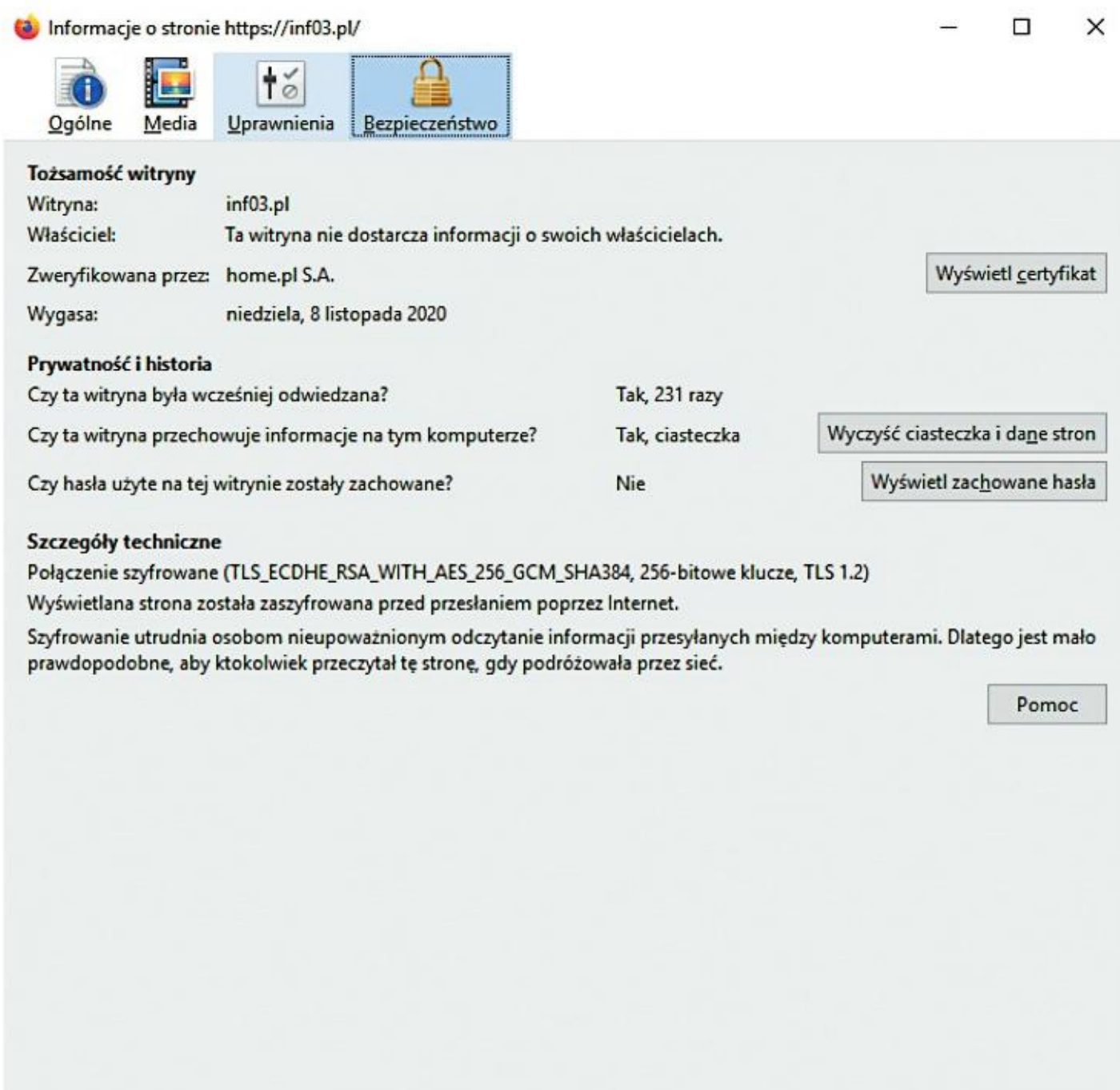
Rys. 33.4. Instalacja certyfikatu SSL

Po zakończeniu instalacji certyfikatu zmieni się informacja na temat strony, która jest wyświetlana w pasku adresu. Na rys. 33.5 przedstawiono różnicę w trakcie wyświetlania adresu strony bez certyfikatu oraz z zainstalowanym certyfikatem.



Rys. 33.5. Różnica wyświetlania adresu strony bez certyfikatu oraz z zainstalowanym certyfikatem SSL

W informacjach o bezpieczeństwie przeglądarki internetowej można uzyskać informacje na temat statusu połączenia szyfrowanego oraz wyświetlić certyfikat domeny (rys. 33.6).



Rys. 33.6. Informacje o stronie w przeglądarce internetowej

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Przeskanuj pliki dowolnej strony internetowej za pomocą serwisu internetowego VirusTotal.
2. Wyszukaj w zasobach sieci Internet oferty certyfikatów SSL. Porównaj ich parametry oraz ceny. Sporządź plik tekstowy zawierający wyniki porównania w formie tabeli. Określ, która oferta jest, według ciebie, najbardziej korzystna. Wybór uzasadnij.
3. Przeprowadź proces instalacji certyfikatu SSL na dowolnej domenie. Niezbędne zasoby otrzymasz od nauczyciela lub skorzystaj z usługi generowania darmowych certyfikatów SSL dostępnych w sieci.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Scharakteryzuj rodzaje certyfikatów SSL.
2. Wyjaśnij, w jaki sposób dysponowanie certyfikatem SSL może wpłynąć na pozycjonowanie strony w wyszukiwarkach internetowych.

Cascading Style Sheets (CSS 3)

BACKGROUND		BORDER		BOX MODEL	
background	<i>background-image</i> <i>background-position</i> <i>background-size</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-origin</i> <i>background-clip</i> <i>background-color</i>	border-top	<i>border-top-width</i> <i>border-style</i> <i>border-color</i>	float	left right none
background-attachment	scroll fixed	border-top-color	<i>border-color</i>	height	auto <i>length</i> %
background-break	bounding-box each-box continuous	border-top-style	<i>border-style</i>	max-height	none <i>length</i> %
background-clip	<i>length</i> % border-box padding-box content-box no-clip	border-top-width	thin medium thick <i>length</i>	max-width	none <i>length</i> %
background-color	<i>color</i> transparent	border-width	thin medium thick <i>length</i>	min-height	none inherit <i>length</i> %
background-image	<i>url</i> none	border-radius	<i>border-top-right-radius</i> <i>border-bottom-right-radius</i> <i>border-bottom-left-radius</i> <i>border-top-left-radius</i>	min-width	none inherit <i>length</i> %
background-origin	border-box padding-box content-box	border-top-right-radius	<i>length</i>	width	auto % <i>length</i>
background-position	top left top center top right center left center center center right bottom left bottom center bottom right <i>x</i> -% <i>y</i> -% <i>x-pos</i> <i>y-pos</i>	border-bottom-right-radius	<i>length</i>	margin	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>
background-repeat	repeat repeat-x repeat-y no-repeat	border-bottom-left-radius	<i>length</i>	margin-bottom	auto <i>length</i> %
background-size	<i>length</i> % auto cover contain	border-top-left-radius	<i>length</i>	margin-left	auto <i>length</i> %
BORDER		FONT		margin-right	auto <i>length</i> %
border	<i>border-width</i> <i>border-style</i> <i>border-color</i>	font	<i>font-style</i> <i>font-variant</i> <i>font-weight</i> <i>font-size/line-height</i> <i>font-family</i> caption icon menu message-box small-caption status-bar	margin-top	auto <i>length</i> %
border-break	<i>border-width</i> <i>border-style</i> <i>color</i> close	font-family	<i>family-name</i> <i>generic-family</i> inherit	padding	<i>padding-top</i> <i>padding-right</i> <i>padding-bottom</i> <i>padding-left</i>
border-bottom	<i>border-bottom-width</i> <i>border-style</i> <i>border-color</i>	font-size	xx-small x-small small medium large x-large xx-large smaller larger inherit <i>length</i> %	padding-bottom	<i>length</i> %
border-bottom-color	<i>border-color</i>	font-size-adjust	none inherit <i>number</i>	padding-left	<i>length</i> %
border-bottom-style	<i>border-style</i>	font-stretch	normal wider narrower ultra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded inherit	padding-right	<i>length</i> %
border-bottom-width	thin medium thick <i>length</i>	font-style	normal italic oblique inherit	padding-top	<i>length</i> %
border-collapse	collapse separate	font-variant	normal small-caps inherit	marquee-direction	forward reverse
border-color	<i>color</i>	font-weight	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit	marquee-loop	infinite <i>number</i>
border-image	<i>image</i> [<i>number</i> / % <i>border-width</i> stretch repeat round] none	BOX MODEL		marquee-play-count	infinite <i>integer</i>
border-left	<i>border-left-width</i> <i>border-style</i> <i>border-color</i>	clear	left right both none	marquee-speed	slow normal fast
border-left-color	<i>border-color</i>	display	none inline block inline-block list-item run-in compact table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption ruby ruby-base ruby-text ruby-base-group ruby-text-group	marquee-style	scroll slide alternate
border-left-style	<i>border-style</i>			overflow	visible hidden scroll auto no-display no-content <i>overflow-x</i> <i>overflow-y</i>
border-left-width	thin medium thick <i>length</i>			overflow-style	auto marquee-line marquee-block
border-right	<i>border-right-width</i> <i>border-style</i> <i>border-color</i>			overflow-x	visible hidden scroll auto no-display no-content
border-right-color	<i>border-color</i>			overflow-y	visible hidden scroll auto no-display no-content
border-right-style	<i>border-style</i>			rotation	<i>angle</i>
border-right-width	thin medium thick <i>length</i>			rotation-point	<i>position</i> (<i>paired value</i> off-set)
				visibility	visible hidden collapse

Cascading Style Sheets (CSS 3)

TEXT	
direction	ltr rtl inherit
hanging-punctuation	none [start end end-edge]
letter-spacing	normal <i>length</i> %
punctuation-trim	none [start end adjacent]
text-align	start end left right center justify
text-align-last	start end left right center justify
text-decoration	none underline overline line-through blink
text-emphasis	none [[accent dot circle disc] [before after]?]
text-indent	<i>length</i> %
text-justify	auto inter-word inter-ideograph inter-cluster distribute kashida tibetan
text-outline	none <i>color</i> <i>length</i>
text-shadow	none <i>color</i> <i>length</i>
text-transform	none capitalize uppercase lowercase
text-wrap	normal unrestricted none suppress
unicode-bidi	normal embed bidi-override
white-space	normal pre nowrap pre-wrap pre-line
white-space-collapse	preserve collapse preserve-breaks discard
word-break	normal keep-all loose break-strict break-all
word-spacing	normal <i>length</i> %
word-wrap	normal break-word
COLUMN	
column-count	auto <i>number</i>
column-fill	auto balance
column-gap	normal <i>length</i>
column-rule	<i>column-rule-width</i> <i>column-rule-style</i> <i>column-rule-color</i>
column-rule-color	<i>color</i>
column-rule-style	<i>border-style</i>
column-rule-width	thin medium thick <i>length</i>
columns	<i>column-width</i> <i>column-count</i>
column-span	1 all
column-width	auto <i>length</i>
COLOR	
color	inherit <i>color</i>
opacity	inherit <i>number</i>

TEMPLATE LAYOUT	
box-align	start end center base-
box-direction	normal reverse
box-flex	<i>number</i>
box-flex-group	<i>integer</i>
box-lines	single multiple
box-orient	horizontal vertical inline-axis block-axis
box-pack	start end center justify
box-sizing	content-box padding-box border-box margin-box
tab-side	top bottom left right
TABLE	
border-collapse	collapse separate
border-spacing	<i>length</i> <i>length</i>
caption-side	top bottom left right
empty-cells	show hide
table-layout	auto fixed
SPEECH	
cue	<i>cue-before</i> <i>cue-after</i>
cue-before	<i>uri</i> [silent x-soft soft medium loud x-loud] none inherit <i>number</i> %
cue-after	<i>uri</i> [silent x-soft soft medium loud x-loud] none inherit <i>number</i> %
mark	<i>mark-before</i> <i>mark-after</i>
mark-before	<i>string</i>
mark-after	<i>string</i>
pause	<i>pause-before</i> <i>pause-after</i>
pause-before	none x-weak weak medium strong x-strong inherit <i>time</i>
pause-after	none x-weak weak medium strong x-strong inherit <i>time</i>
phonemes	<i>string</i>
rest	<i>rest-before</i> <i>rest-after</i>
rest-before	none x-weak weak medium strong x-strong inherit <i>time</i>
rest-after	none x-weak weak medium strong x-strong inherit <i>time</i>
speak	none normal spell-out digits literal-punctuation no-punctuation inherit
voice-balance	left center right leftwards rightwards inherit <i>number</i>
voice-duration	<i>time</i>

SPEECH	
voice-family	inherit [<specific-voice, age, generic-voice, number>]
voice-rate	x-slow slow medium fast x-fast inherit %
voice-pitch	x-low low medium high x-high inherit <i>number</i> %
voice-pitch-range	x-low low medium high x-high inherit <i>number</i>
voice-stress	strong moderate none reduced inherit
voice-volume	silent x-soft soft medium loud x-loud inherit <i>number</i> %
LIST & MARKERS	
list-style	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i>
list-style-image	none <i>uri</i>
list-style-position	Inside outside
list-style-type	none asterisks box check circle diamond disc hyphen square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek lower-latin upper-latin hebrew armenian georgian cjk-ideographic hiragana katakana hiragana-iroha katakana-iroha footnotes
marker-offset	auto <i>length</i>
ANIMATIONS	
animation	<i>animation-name</i> <i>animation-duration</i> <i>animation-timing-function</i> <i>animation-delay</i> <i>animation-iteration-count</i> <i>animation-direction</i>
animation-delay	<i>time</i>
animation-direction	normal alternate
animation-duration	<i>time</i>
animation-iteration-count	inherit <i>number</i>
animation-name	none IDENT
animation-play-state	running paused
animation-timing-function	ease linear ease-in ease-out ease-in-out cubic-Bezier (<i>number</i> , <i>number</i> , <i>number</i> , <i>number</i>)
TRANSITIONS	
transition	<i>transition-property</i> <i>transition-duration</i> <i>transition-timing-function</i> <i>transition-delay</i>
transition-delay	<i>time</i>
transition-duration	<i>time</i>
transition-property	none all
transition-timing-function	ease linear ease-in ease-out ease-in-out cubic-Bezier (<i>number</i> , <i>number</i> , <i>number</i> , <i>number</i>)

Cascading Style Sheets (CSS 3)

GRID POSITIONING	
grid-columns	none inherit [length percentage relative length]
grid-rows	none inherit [length percentage relative length]
OUTLINE	
outline	outline-color outline-style outline-width
outline-color	color invert
outline-offset	inherit length
outline-style	None dotted dashed solid double groove ridge inset outset
outline-width	thin medium thick length
3D / 2D TRANSFORM	
backface-visibility	visible hidden
perspective	none number
perspective-origin	[[[percentage > <length> left center right] [[<percentage> <length> top center bottom]?] <length>] [[[left center right] [top center bottom]]] <length>]
transform	none matrix matrix3d translate3d translateX translateY translateZ scale scale3d scaleX scaleY scaleZ rotate rotate3d rotateX rotateY rotateZ skewX skewY skew perspective
transform-origin	[[[<percentage> <length> left center right] [[<percentage> <length> top center bottom]?] <length>] [[[left center right] [top center bottom]]] <length>]
transform-style	flat preserve-3d
GENERATED CONTENT	
bookmark-label	content attr string
bookmark-level	none integer
bookmark-target	self uri attr
border-length	auto length
content	normal none inhibit uri
counter-increment	none identifier number
counter-reset	none identifier number
crop	auto shape
display	normal none list-item
float-offset	length length

GENERATED CONTENT	
hyphenate-after	auto integer
hyphenate-before	auto integer
hyphenate-character	auto string
hyphenate-lines	no-limit integer
hyphenate-resource	none uri
hyphens	none manual auto
image-resolution	normal auto dpi
marks	[crop cross] none
move-to	normal here identifier
page-policy	start first last
quotes	none string string string string
string-set	identifier content-list
text-replace	none [<string> <string>]+
LINE BOX	
alignment-adjust	auto baseline before-edge text-before-edge middle central after-edge text-after-edge ideographic alphabetic hanging mathematical length %
alignment-baseline	baseline use-script before-edge text-before-edge after-edge text-after-edge central middle ideographic alphabetic hanging mathematical
baseline-shift	baseline sub super length %
dominant-baseline	auto use-script no-change reset-size alphabetic hanging ideographic mathematical central middle text-after-edge text-before-edge
drop-initial-after-align	alignment-baseline
drop-initial-after-adjust	central middle after-edge text-after-edge ideographic alphabetic mathematical length %
drop-initial-before-align	caps-height alignment-baseline
drop-initial-before-adjust	before-edge text-before-edge central middle hanging mathematical length %
drop-initial-value	initial integer
drop-initial-size	auto integer % line
inline-box-align	initial last integer

LINE BOX	
line-height	normal number length %
line-stacking	line-stacking-strategy line-stacking-ruby line-stacking-shift
line-stacking-strategy	inline-line-height block-line-height max-height grid-height
line-stacking-ruby	exclude-ruby include-ruby
line-stacking-shift	consider-shifts disregard-shifts
text-height	auto font-size text-size max-size
vertical-align	Baseline sub super top text-top middle bottom text-bottom length %
HYPERLINK	
target	target-name target-new target-position
target-name	current root parent new modal string
target-new	window tab none
target-position	above behind front back
POSITIONING	
bottom	auto % length
clip	shape auto
left	auto % length
position	static relative absolute fixed
right	auto % length
top	auto % length
z-index	auto number
RUBY	
ruby-align	auto start left center end right distribute-letter distribute-space line-edge
ruby-overhang	auto start end none
ruby-position	before after right inline
ruby-span	attr(x) none

Cascading Style Sheets (CSS 3)

PAGED MEDIA	
fit	fill hidden meet slice
fit-position	[top center bottom] [left center right] <i>length</i> %
image-orientation	auto <i>angle</i>
orphans	<i>integer</i>
page	auto <i>identifier</i>
page-break-after	auto always avoid left right
page-break-before	auto always avoid left right
page-break-inside	auto avoid
size	auto landscape portrait <i>length</i>
windows	<i>integer</i>
UI	
appearance	normal inherit [icon window desktop workspace document tooltip dialog button push-button hyperlink radio-button checkbox menu-item tab menu menubar pull-down-menu pop-up-menu list-menu radio-group checkbox-group outline-tree range field combo-box signature password]
cursor	auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help <i>url</i>
icon	auto inherit <i>url</i>
nav-index	auto inherit <i>number</i>
nav-up	auto inherit <id> [<i>current</i> <i>root</i> <target-name>]
nav-right	auto inherit <id> [<i>current</i> <i>root</i> <target-name>]
nav-down	auto inherit <id> [<i>current</i> <i>root</i> <target-name>]
nav-left	auto inherit <id> [<i>current</i> <i>root</i> <target-name>]
resize	none both horizontal vertical inherit

- Values in italics are place holders for an actual value (like 1px, 1em, 1%), values in normal text are values that can be used as the actual value
- CSS Properties in **Dark Red** are shorthand properties and each value must be defined. The exception is where the property can define from one to four of the sides of a box element property (Top-Right-Bottom-Left) - i.e. border-width

Cascading Style Sheets (CSS 3)

PSEUDO-CLASS		UNITS		SELECTOR TYPES		
		ABSOLUTE MEASUREMENT		Name	Info	Example
:active	an activated element	%	percentage	Universal	Any element	* { font: 10px Arial; }
:focus	an element while the element has focus	cm	centimeter	Type	Any element of that type	h1 { text-decoration: underline; }
:visited	a visited link	in	inch	Grouping	Multiple elements of different types	h1, h2, h3 { font-family: Verdana; }
:hover	an element when you mouse over it	mm	millimeter	Class	Multiple elements of different types when you don't want to affect all instances of that type	.sampleclass { text-decoration: underline; }
:link	an unvisited link	pc	pica (1p = 12 points)	Id	A single element type when you don't want to affect all instances of that type	#sampleid { text-decoration: underline; }
:disabled	an element while the element is disabled	pt	point (1pt = 1/72 inch)	Descendant	An element that is below (in the document tree) another element—no matter how many levels below	#gallery h1 { text-decoration: underline; }
:enabled	an element while the element is enabled	RELATIVE MEASUREMENT		Child	An element that is directly below (in the document tree) another element	#title > p { font-weight: bold; }
:checked	an element (form element control) that is checked	ch	width of the "0" glyph found in the font for the font size used to render	Adjacent Sibling	All elements that share the same parent and elements are in the same immediate sequence	h1 + p { font-style: italic; }
:selection	an element that is currently selected of highlighted by the user	em	1em = current font size of current element	General Sibling	All elements that share the same parent and elements are in the same sequence (not necessarily immediate)	h1 ~ p { font-style: italic; }
:lang	Allows the author to specify a language to use in a specified element	ex	x-height of the element's font	Attribute	An element with that matches the attribute listed	E[selected] - att whatever the value E[att="val"] - att with a specific value E[rel~="next"] - att with a value is a whitespace separated list *[lang="en"] - att value either being exactly "val" or beginning with "val" immediately followed by "_" E[att^="val"] - att value that begins with the prefix "val" E[att\$="val"] - att value that end with the suffix "val" E[att*="val"] - att value contains at least one instance of the substring "val"
:nth-child(n)	an element that is the n-th sibling	gd	the grid defined by 'layout-grid'			
:nth-last-child(n)	an element that is the n-th sibling counting from the last sibling	px	pixel of the viewing device			
:first-child	an element that is the first sibling	rem	the font size of the root element			
:last-child	an element that is the last sibling	vh	the viewport's height			
:only-child	an element that is the only child	vw	the viewport's width			
:nth-of-type(n)	an element that is the n-th sibling of its type.	vm	viewport's height or width, whichever is smaller of the two			
:nth-last-of-type(n)	an element that is the n-th sibling of its type counting from the last sibling	ANGLES				
:last-of-type	an element that is the first sibling of its type	deg	degrees			
:first-of-type	an element that is the last sibling of its type	grad	grads			
:only-of-type	an element that is the only child of that type	rad	radians			
:empty	an element that has no children	turn	turns			
:root	root element within the document	TIME				
:not(x)	an element not represented by the argument 'x'	ms	milli-seconds			
:target	a target element as specified by a target in a URI	s	seconds			
PSEUDO-ELEMENT		FREQUENCY				
::first-letter	Adds special style to the first letter of a text	Hz	hertz			
::first-line	Adds special style to the first line of a text	kHz	kilo-hertz			
::before	Inserts some content before an element	COLORS				
::after	Inserts some content after an element	color name	red, blue, green, dark green			
		rgb(x,y,z)	red = rgb(255,0,0)			
		rgb(x%,y%,z%)	red = rgb(100%,0,0)			
		rgba(x,y,z, alpha)	red = rgba(255,0,0)			
		#rrggbb	red = #ff0000 (or shorthand = #f00)			
		hsl(hue, saturation, lightness)	red = hsl(0, 100%, 50%)			
		hsla(hue, saturation, lightness, alpha)	red = hsl(0, 100%, 50%)			
		flavor	An accent color (typically chosen by the user) to customize the user interface of the user agent itself			
		currentColor	computed value of the 'currentColor' keyword is the computed value of the 'color' property			

WYKAZ PODSTAWOWYCH POJĘĆ W JĘZYKACH: POLSKIM, ANGIELSKIM I NIEMIECKIM

JĘZYK POLSKI	JĘZYK ANGIELSKI	JĘZYK NIEMIECKI
animacja poklatkowa	stop-motion animation	Stop-Motion
argument	argument	das Argument / der Operand
atrybut	attribute	das Attribut
blok preformatowany	preformatted block	der <pre>-Block / der vorformatierte Block
blok tekstu	block of text	der Textblock
deklaracja	declaration	die Deklaration
dokument hipertekstowy	hypertext document	der Hypertext
dynamiczna strona internetowa	dynamic website	die dynamische Webseite
dźwięk	sound	der Klang
element liniowy	linear element	das Inline-Element
filtr	filter	der Filter
formularz	form	das Formular
fotomontaż	photomontage	die Fotomontage
funkcja	function	die Funktion
głębia barwy	color depth	die Farbtiefe
grafika rastrowa	raster graphics	die Bitmapgrafik / die Rastergrafik / die Pixelgrafik
grafika wektorowa	vector graphics	die Vektorgrafik
hipertekstowy język znaczników	hypertext markup language	die Hypertext-Auszeichnungssprache / die Web-Programmiersprache
implementacja	implementation	die Implementierung
instrukcja sterująca	control instruction	die Steuerungsanweisung
internet wszechrzeczy	internet of all things	das Internet der Dinge
kadrowanie	cropping	das Cropping / das Abschneiden
kaskadowy arkusz stylów	cascading style sheet	der Kaskadenstilentwurf
kodek	codec	der Codec
kodowanie polskich znaków	Polish character encoding	die Codierung der polnischen Sonderzeichen
koło kolorów	color wheel	der Farbenkreis
kompresja obrazu	image compression	die Bildkompression / die Bildkomprimierung
konkatenacja	concatenation	die Verkettung

JĘZYK POLSKI	JĘZYK ANGIELSKI	JĘZYK NIEMIECKI
konstruktor	constructor	der Konstruktor
kontener grafiki	graphic container	der Grafikcontainer
konwersja formatu	format conversion	die Formatkonvertierung
kwantyzacja	quantization	die Quantisierung
mapa odsyłaczy	map of links	die Imagemap / die verweissensitive Grafik
media strumieniowe	streaming media	die Streaming Medien
menu nawigacyjne	navigation menu	das Navigationsmenü
metoda	method	die Methode
model barw	color model	das Farbmodell
model blokowy	block model	das Boxmodell
motyw	theme	das Motiv
obiekt	object	das Objekt
obraz cyfrowy	digital image	das digitale Bild
obszar kompozycji	area of the composition	die Zeichenfläche
odnośnik	reference	der Link / der Verweis
okno decyzyjne	decision window	das Confirm-Fenster / das Bestätigungsfeld
okno dialogowe	dialogue window	das Dialogfeld / das Dialogfenster / die Dialogbox
okno informacyjne	information window	das Informationsfenster
okno tekstowe	text window	das Textfeld / das Textfenster
operator	operator	der Operator
pętla	loop	die Schleife
piksel	pixel	das Pixel / das Bildelement/ der Bildpunkt / der Rasterpunkt
plaster	plaster	der Bildteil
pozycjonowanie	positioning	die Positionierung
protokół sieciowy	network protocol	das Netzwerkübertragungs-proto- koll / das Netzwerk- protokoll
próbkiowanie	sampling	die Abtastung
przetłamanie wiersza	break the line	der Zeilenumbruch
ramka kluczowa	key frame	das Schlüsselbild

JĘZYK POLSKI	JĘZYK ANGIELSKI	JĘZYK NIEMIECKI
retusz	retouch	die Retusche
rozdzielczość	resolution	die Auflösung
ruch po ścieżce	movement along the path	die Pfad-Animation
selektor	selector	der Selektor
skalowanie	calibration	das Skalieren / die Skalierung
statyczna strona internetowa	static website	die statische Webseite
struktura fizyczna	physical structure	die physikalische Topologie
System Zarządzania Treścią	content Management System	das Inhaltsverwaltungssystem
technologia webowa	web technology	die Webtechnologie
templatka	template	das Webtemplate / die Webvorlage
testowanie	testing	der Softwaretest
urządzenie mobilne	mobile device	das Mobilgerät
walidacja	validation	die Validierung
warstwa	layer	die Schicht
weryfikacja	verification	die Verifizierung / die Nachprüfung
witryna internetowa	website	die Website
wskaźnik fokusa	focus indicator	das aktive Fenster/ das fokussierte Fenster
wykaz	list	die Liste
zdarzenie	happening	das Ereignis
zmienna	variable	die Variable

Źródła ilustracji

Okładka: (witryny) nmedia/Shutterstock.com

Tekst główny: s. 12 (biurowy zasilacz awaryjny) AlexLMX/Shutterstock.com

Pozostałe grafiki: Tomasz Klekot, Agnieszka Klekot

Wydawnictwa Szkolne i Pedagogiczne oświadczają, że podjęły starania mające na celu dotarcie do właścicieli i dysponentów praw autorskich wszystkich zamieszczonych utworów. Wydawnictwa Szkolne i Pedagogiczne, przytaczając w celach dydaktycznych utwory lub fragmenty, postępują zgodnie z art. 27¹ ustawy o prawie autorskim. Jednocześnie Wydawnictwa Szkolne i Pedagogiczne oświadczają, że są jedynym podmiotem właściwym do kontaktu autorów tych utworów lub innych podmiotów uprawnionych w wypadkach, w których twórcy przysługuje prawo do wynagrodzenia.