



DO NOWEJ  
PODSTAWY PROGRAMOWEJ

# Część 1

## Projektowanie stron internetowych

### Kwalifikacja INF.03

Tworzenie i administrowanie stronami i aplikacjami internetowymi oraz bazami danych



**Podręcznik do nauki zawodu  
technik informatyk i technik programista**

Jolanta Pokorska

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiekolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicielami.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Joanna Zaręba

Projekt okładki: Jan Paluch

Fotografia na okładce została wykorzystana za zgodą Shutterstock.

Wydawnictwo HELION  
ul. Kościuszki 1c, 44-100 GLIWICE  
tel. 32 231 22 19, 32 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)  
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!  
Jeżeli chcesz ocenić tę książkę, zatrzymaj pod adres  
[http://helion.pl/user/opinie?inf031\\_ebook](http://helion.pl/user/opinie?inf031_ebook)

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-6439-4

Copyright © Helion 2019

Printed in Poland.

- [Poleć książkę na Facebook.com](#)
- [Kup w wersji papierowej](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

# Spis treści

<b>Wstęp</b> .....	7
<b>Rozdział 1.</b> Wprowadzenie .....	9
<b>1.1.</b> Podstawowe informacje .....	9
<b>1.2.</b> Języki HTML, XML, XHTML .....	11
<b>1.3.</b> Kodowanie polskich znaków .....	16
<b>1.4.</b> Edytory stron WWW .....	17
<b>Rozdział 2.</b> Język HTML5 .....	21
<b>2.1.</b> Struktura strony internetowej .....	21
<b>2.2.</b> Elementy składowe strony WWW .....	27
<b>2.3.</b> Obrazy i odsyłacze .....	58
<b>2.4.</b> Formularze .....	70
<b>2.5.</b> Multimedia .....	91
<b>2.6.</b> Inne elementy strony internetowej .....	99
<b>2.7.</b> Pytania i zadania .....	101
<b>Rozdział 3.</b> Kaskadowe arkusze stylów (CSS) .....	107
<b>3.1.</b> Wstawianie stylów .....	108
<b>3.2.</b> Składnia języka CSS .....	118
<b>3.3.</b> Selektory .....	122
<b>3.4.</b> Właściwości elementów .....	140
<b>3.5.</b> Model blokowy CSS .....	159
<b>3.6.</b> Inne elementy .....	171
<b>3.7.</b> Menu w języku CSS .....	198
<b>3.8.</b> Zasady projektowania szablonu strony internetowej .....	214
<b>3.9.</b> Responsywny układ strony .....	236
<b>3.10.</b> Pytania i zadania .....	253
<b>Rozdział 4.</b> Grafika na stronie internetowej .....	257
<b>4.1.</b> Rodzaje grafiki .....	257
<b>4.2.</b> Grafika wektorowa .....	257
<b>4.3.</b> Grafika rastrowa .....	260

<b>4.4.</b> Modele przestrzeni barw .....	266
<b>4.5.</b> Ogólne zasady tworzenia grafiki na potrzeby strony internetowej .....	269
<b>4.6.</b> Program GIMP .....	271
<b>4.7.</b> Program Adobe Photoshop .....	286
<b>4.8.</b> Program Inkscape .....	303
<b>4.9.</b> Zapisywanie obrazów na potrzeby witryny internetowej .....	310
<b>4.10.</b> Pytania i zadania .....	311
 <b>Rozdział 5.</b> Graficzny projekt strony internetowej .....	315
<b>5.1.</b> Opracowanie projektu .....	315
<b>5.2.</b> Projektowanie strony .....	317
<b>5.3.</b> Struktura strony internetowej .....	321
<b>5.4.</b> Dobór palety barw .....	321
<b>5.5.</b> Dobór czcionki .....	325
<b>5.6.</b> Ułatwienia dostępu .....	328
<b>5.7.</b> Pytania i zadania .....	331
 <b>Rozdział 6.</b> Animacja na stronie internetowej .....	333
<b>6.1.</b> Metody tworzenia animacji .....	333
<b>6.2.</b> Tworzenie animacji GIF .....	336
<b>6.3.</b> Tworzenie animacji za pomocą CSS i HTML .....	339
<b>6.4.</b> Pytania i zadania .....	359
 <b>Rozdział 7.</b> Dźwięk i wideo .....	361
<b>7.1.</b> Dźwięk .....	361
<b>7.2.</b> Edytowanie i obróbka dźwięku .....	364
<b>7.3.</b> Pliki wideo .....	372
<b>7.4.</b> Planowanie filmu .....	375
<b>7.5.</b> Tworzenie filmu w programie Adobe Premiere .....	375
<b>7.6.</b> Pytania i zadania .....	382
 <b>Rozdział 8.</b> Systemy zarządzania treścią (CMS) .....	385
<b>8.1.</b> Wprowadzenie .....	385
<b>8.2.</b> WordPress .....	386
<b>8.3.</b> Joomla! .....	406
<b>8.4.</b> Pytania i zadania .....	430

<b>Rozdział 9.</b> Testowanie i publikowanie strony .....	431
<b>9.1.</b> Sprawdzanie poprawności strony .....	431
<b>9.2.</b> Walidacja strony .....	433
<b>9.3.</b> Testowanie strony w przeglądarkach .....	436
<b>9.4.</b> Publikowanie strony internetowej .....	437
<b>9.5.</b> Testy szybkości wczytywania strony .....	441
<b>9.6.</b> Testy funkcjonalności strony internetowej .....	442
<b>9.7.</b> Optymalizacja strony internetowej .....	444
<b>9.8.</b> Pozycjonowanie strony .....	445
<b>9.9.</b> Pytania i zadania .....	449
<b>Bibliografia</b> .....	452
<b>Skorowidz</b> .....	453

d<sub>6</sub>

# Wstęp

Otaczająca nas rzeczywistość zmienia się w zawrotnym tempie. Zawdzięczamy to m.in. intensywnemu rozwojowi branży związanej z technologią informacyjną.

W dobie niezwykle dynamicznego rozwoju technologii internetowych posiadanie wiedzy i praktycznych umiejętności z zakresu projektowania i programowania aplikacji webowych oraz tworzenia responsywnych i multimedialnych witryn internetowych z dostępem do baz danych to klucz do znalezienia dobrej i ciekawej pracy. Zawody takie jak technik informatyk i technik programista stają się bardzo atrakcyjne i poszukiwane na rynku pracy.

Podręcznik *Kwalifikacja INF.03. Tworzenie i administrowanie stronami i aplikacjami internetowymi oraz bazami danych. Część 1. Projektowanie stron internetowych. Podręcznik do nauki zawodu technik informatyk i technik programista* zawiera treści przeznaczone dla osób kształcących się w zawodach technik informatyk i technik programista. Został on opracowany zgodnie z nową podstawą programową, obowiązującą od 1 września 2019 r. i obejmuje materiał kwalifikacji *INF.03. Tworzenie i administrowanie stronami i aplikacjami internetowymi oraz bazami danych* w zakresie projektowania i tworzenia stron internetowych.

W podręczniku zostały omówione zagadnienia teoretyczne z zakresu projektowania i tworzenia stron internetowych. Zawiera on też liczne ćwiczenia, które ułatwiają nabycie umiejętności praktycznych, a proponowane rozwiązania pomogą w ugruntowaniu zdobytej wiedzy.

Podręcznik składa się z dziewięciu rozdziałów. Ich budowa pozwala na realizację treści programowych oraz umożliwia samodzielne rozwijanie tych umiejętności.

Rozdział 1., „Wprowadzenie”, zawiera omówienie podstawowych zagadnień związanych z tworzeniem stron internetowych, takich jak posługiwanie się hipertekstowymi znacznikami i tworzenie stron internetowych za pomocą tych języków.

Rozdział 2., „Język HTML5”, zawiera omówienie zagadnień związanych z tworzeniem stron internetowych w języku HTML5. Opisuje zasady stosowania znaczników edycji tekstu, tabel, różnego rodzaju list, organizacji strony, definiowania hiperłączy i mapy od-сылaczy, używanie znaczników osadzania obrazów i znaczników definiowania formularzy. Przedstawia sposoby planowania rozmieszczenia elementów na stronie internetowej oraz wstawiania obiektów zewnętrznych, takich jak obraz i dźwięk.

Rozdział 3., „Kaskadowe arkusze stylów (CSS)”, zawiera omówienie zagadnień związanych z użyciem kaskadowych arkuszy stylów do tworzenia układu graficznego strony internetowej. Przedstawia zasady tworzenia oraz wykorzystywania kaskadowych arkuszy stylów do opisu formy prezentacji strony internetowej. Pokazuje sposoby definiowania struktury arkusza CSS i tworzenia różnych arkuszy stylów. Omawia atrybuty stylów oraz

sposoby ich stosowania do opisu wyglądu strony internetowej. Przedstawia sposoby definiowania atrybutów responsywnej strony internetowej.

Rozdział 4., „Grafika na stronie internetowej”, zawiera omówienie zagadnień związanych z tworzeniem, modyfikowaniem i umieszczaniem grafiki na stronie internetowej. Opisuje zasady stosowania różnych modeli barw, przestrzegania zasad cyfrowego zapisu obrazu, tworzenia grafiki statycznej. Definiuje terminy „grafika rastrowa” i „grafika wektorowa”. Pokazuje metody tworzenia tekstur, ikon, obrazów statycznych, przycisków i menu na potrzeby stron internetowych. Omawia sposoby modyfikowania obiektów graficznych i zmieniacia ich atrybutów oraz dobierania formatów plików graficznych do publikacji w internecie.

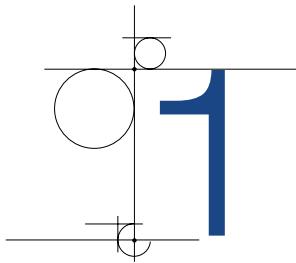
Rozdział 5., „Graficzny projekt strony internetowej”, zawiera omówienie zagadnień związanych z tworzeniem projektu graficznego strony internetowej. Opisuje zasady projektowania struktury witryny internetowej i pokazuje metody tworzenia projektu graficznego witryny.

Rozdział 6., „Animacja na stronie internetowej”, zawiera omówienie zagadnień związanych z tworzeniem animacji na potrzeby strony internetowej. Opisuje i pokazuje metody tworzenia animacji na potrzeby stron internetowych, metody tworzenia animowanych przycisków i menu strony internetowej.

Rozdział 7., „Dźwięk i wideo”, zawiera omówienie zagadnień związanych z edytowaniem i obróbką dźwięku oraz planowaniem i montażem filmu na potrzeby stron internetowych. Opisuje metody przetwarzania obrazu i dźwięku i przygotowania ich do publikacji w internecie. Przedstawia sposoby dobierania formatów plików multimedialnych do publikacji w internecie oraz ich modyfikowania na potrzeby publikacji internetowych.

Rozdział 8., „Systemy zarządzania treścią CMS”, zawiera omówienie zagadnień związanych z tworzeniem stron internetowych z wykorzystaniem systemów zarządzania treścią. Opisuje sposoby tworzenia aplikacji internetowych opartych na szablonach systemów WordPress i Joomla!.

Rozdział 9., „Testowanie i publikowanie strony”, zawiera omówienie zagadnień związanych z walidacją, testowaniem i publikowaniem witryny internetowej. Opisuje zasady stosowania reguł walidacji stron internetowych oraz testowania i publikowania witryny internetowej. Pokazuje metody walidacji strony internetowej. Przedstawia sposoby wykorzystywania narzędzi do testowania stron internetowych i analizowania wyników ich walidacji. Opisuje sposoby publikowania stron i witryn na serwerach WWW oraz korzystania z klientów FTP.



# Wprowadzenie

## 1.1. Podstawowe informacje

Pierwsze strony internetowe pisane w języku HTML były bardzo ograniczone i zawierały jedynie podstawową strukturę w postaci nagłówków i paragrafów. Dawały także możliwość używania odnośników (linków). Klikając odnośniki, internauci mogli łatwo poruszać się między różnymi stronami.

W miarę rozwoju internetu język HTML zmieniał się, zapewniając większą złożoność i elastyczność. Pozwalał na dodawanie do stron różnego rodzaju obiektów, np. obrazów, tabel. Tabele zaczęto wykorzystywać jako elementy struktury i wyglądu strony. Następnym krokiem było wprowadzenie kaskadowych arkuszy stylów (CSS), co pozwoliło na oddzielenie warstwy strukturalnej i treści strony od warstwy prezentacyjnej. Dzięki kaskadowym arkuszom stylów można poprawić wygląd strony, jak również zwiększyć jej dostępność na urządzeniach mobilnych.

Obecnie standardem są strony bogate w grafikę, zdjęcia cyfrowe, animacje, elementy interaktywne oraz w mniej czy bardziej profesjonalne filmy wideo.

Każda ze stron dostępnych w internecie jest zwykłym dokumentem tekstowym, który można stworzyć przy użyciu dowolnego edytora tekstu. Aby się przekonać, że strona internetowa to rzeczywiście dokument tekstowy, wystarczy w dowolnej przeglądarce internetowej otworzyć wybraną stronę, a następnie kliknąć prawym przyciskiem myszy i wybrać z menu opcję *Pokaż źródło*. Spowoduje to wyświetlenie strony w postaci dokumentu tekstowego. Teksty dostępne na stronie internetowej są otoczone znacznikami (tagami), czyli elementami języka HTML (ang. *HyperText Markup Language*), za pomocą którego tworzy się strony WWW.

Dokumenty HTML to pliki tekstowe, które zawierają proste polecenia dotyczące składu tekstu, a także dynamiczne odnośniki do innych dokumentów oraz obiekty postscriptowe, preformatowane i multimedialne (animacje, dźwięki, sekwencje wideo). Dokumenty te są przesyłane w sieci w postaci źródłowej — dopiero przeglądarka internetowa dostępna na komputerze użytkownika interpretuje kod strony, dzięki temu możliwe jest uniezależnienie się od systemu operacyjnego i komputera, na którym wyświetlana jest strona.

Dokumenty HTML można przygotowywać, korzystając z dowolnego edytora tekstu.

Edytory możemy podzielić następująco:

- Edytory tekstowe — umożliwiają bezpośrednią pracę z kodem HTML. Często są wyposażone w dodatkowe funkcje, np.: automatyczne wstawianie znacznika zamkajającego, sprawdzanie składni języka HTML, proste dodawanie atrybutów, łatwe tworzenie formularzy, tabel i ramek.
- Edytory graficzne (wizualne) — umożliwiają pracę w trybie WYSIWYG (ang. *What You See Is What You Get*). Pozwalają na tworzenie stron bez wpisywania kodu źródłowego.

### 1.1.1. Podstawowe pojęcia

#### Strona internetowa

Dokument utworzony w języku HTML lub XHTML, zapisany w pliku i umieszczony na serwerze, odczytywany na komputerze użytkownika za pomocą przeglądarki internetowej.

#### Witryna internetowa

Zbiór stron internetowych powiązanych tematycznie i umieszczonych na jednym serwerze, stanowiący rodzaj serwisu informacyjnego.

#### Portal internetowy

Serwis informacyjny zawierający informacje na różne tematy, zwykle wyposażony w dodatkowe funkcje, takie jak: poczta elektroniczna, katalogi stron internetowych, wyszukiwarki, czat, pobieranie plików.

#### Wortal internetowy

Portal wertykalny (ang. *vertical portal*), czyli wyspecjalizowany, zawierający informacje z jednej dziedziny, tematycznie ze sobą powiązane. Nazwa „wortal” się nie przyjęła.

#### Serwer internetowy

Określenia tego używa się w odniesieniu do komputera podłączonego do internetu i świadczącego różne usługi w internecie oraz w odniesieniu do oprogramowania uruchamianego na tym komputerze. Przykłady: serwery FTP, DNS, HTTP.

#### Statyczna strona internetowa

Strony statyczne to takie, których zawartość i wygląd nie zmieniają się przy każdym odwołaniu do nich. Aby wprowadzić zmiany na stronie, programista musi zmienić jej kod.

#### Dynamiczna strona internetowa

Strony dynamiczne są generowane na bieżąco przez serwer HTTP na podstawie zmiennych i parametrów przekazanych przez przeglądarkę internetową. Takie strony zmieniają zawartość i wygląd w zależności od interakcji z użytkownikiem.

Zmiany zawartości strony mogą odbywać się w dwojakim sposobie. Można wprowadzić te zmiany po stronie użytkownika, na jego komputerze (ang. *client-side*), przy użyciu języków skryptowych takich jak JavaScript, dokonujących zmian bezpośrednio na obiektach. Drugim sposobem jest komplikacja dynamicznej zawartości na serwerze (ang. *server-side*) z wykorzystaniem języków programowania takich jak PHP, ASP, Perl, Java, Python.

## 1.2. Języki HTML, XML, XHTML

### 1.2.1. HTML (ang. HyperText Markup Language)

Podstawowym językiem projektowania stron internetowych jest HTML. Język ten rozwijał się intensywnie, aż stał się potężnym narzędziem do tworzenia stron WWW. W kolejnych wersjach język HTML został wzbogacony o mechanizmy: korzystania z zewnętrznych arkuszy stylów, pisania skryptów, wstawiania obiektów, animowania tekstów. Obecnie najczęściej stosowaną specyfikacją jest HTML w wersji 5. Aby zrozumieć nowe funkcje HTML5, trzeba jednak poznać podstawowe funkcje HTML.

Podstawowym elementem języka HTML jest znacznik. Znaczniki są polecaniami umieszczonymi w nawiasach ostrokątnych, np.: `<body>`, `<head>`. Informują one przeglądarkę o wyglądzie otwieranej strony oraz o strukturze umieszczonego na niej tekstu. Nazwy polecen mogą być pisane małymi lub wielkimi literami, nie ma to znaczenia. Wyróżnia się znaczniki otwierające, np. `<p>`, i zamykające, np. `</p>`. Znacznik zamykający zawsze rozpoczyna się znakiem / i ma taką samą nazwę jak znacznik otwierający. Pomiędzy tymi znacznikami może zostać umieszczony tekst, którego znaczniki dotyczą, np. wpisanie kodu `<p>Formatowanie tekstu</p>` spowoduje umieszczenie tekstu zapisanego między znacznikami w nowym akapicie. Prawie wszystkie znaczniki otwierające trzeba zamknąć, stosując znaczniki zamykające. Wyjątki to np.:

- `<br>` — znacznik łamania wiersza,
- `<hr>` — znacznik linii poziomej,
- `<img>` — znacznik wstawiania obrazka.

Warto jednak pamiętać o tym, że znaczniki, które formalnie nie posiadają oddzielnego elementu domykającego, możemy zapisać w taki sposób, aby to domknięcie się pojawiło. Wymóg taki pojawił się w języku XHTML, ale zapis taki został również dopuszczony w HTML5 i warto go stosować. Wspomniany zapis od czasu do czasu jest używany w niniejszym podręczniku. Dla wskazanych powyżej wyjątków może on mieć następującą postać:

- `<br />`,
- `<hr />`,
- `<img />`.

Znaczniki mają dodatkowe elementy, zwane atrybutami, które definiują ich działanie. Jeden znacznik może posiadać wiele atrybutów. Atrybuty mogą przyjmować różne wartości, nazywane parametrami. Wartość atrybutu powinna zostać zapisana w apostrofach lub cudzysłowie, ale język HTML dopuszcza zapisywanie wartości atrybutu bez cudzysłowu (nie jest to jednak zalecane rozwiązanie):

```
<nazwa_znacznika atrybut="wartość"> zawartość </nazwa_znacznika>
```

lub

```
<nazwa_znacznika atrybut="wartość"> zawartość </nazwa_znacznika>
```

Serwery rozróżniają wielkość liter w nazwach plików. Jeżeli wartością parametru jest nazwa pliku, to wielkość liter może mieć znaczenie przy interpretacji zapisu znacznika, np. `` to coś zupełnie innego niż ``.

### Przykład 1.1

```
<p><a href="http://www.pxp.com"></a></p>
```

W języku HTML znaczniki mogą być zagnieżdżane, czyli umieszczane w innych znacznikach. Jednak nie wszystkie znaczniki mogą zawierać w sobie inne znaczniki.

Znaczniki powinny być zamknięte w kolejności odwrotnej do tej, w jakiej były otwierane. Najpierw powinien zostać zamknięty znacznik, który został otwarty jako ostatni, a na końcu znacznik, który został otwarty jako pierwszy. Kolejność wypisywania znaczników i atrybutów jest dowolna.

W dokumencie tworzonym w języku HTML powinna być wstawiona deklaracja typu dokumentu (ang. *Document Type Definition* — DTD). Określa ona, która wersja języka HTML jest stosowana na stronie.

### Przykład 1.2

Deklaracja w HTML5:

```
<!DOCTYPE html>
```

Deklaracja w HTML 4.01:

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

## 1.2.2. XML (ang. Extensible Markup Language)

XML jest metajęzykiem opartym na zwykłym tekście, przeznaczonym do definiowania języków znakowania. Jest to coś więcej niż tekstowy format do opisywania dokumentów. XML daje dostęp do wielu technologii konstruowania, przetwarzania i przeszukiwania danych. Został stworzony w celu definiowania nowych formatów dokumentów w internecie. Dostarcza mechanizmów opisywania struktury dokumentu za pomocą znaczników.

XML jest powszechnie używany do opisu nowych formatów dokumentów na potrzeby internetu oraz do opisu danych posiadających wyraźną lub słabą wewnętrzną strukturę (arkuszy kalkulacyjnych, protokołów sieciowych, plików konfiguracyjnych).

Ze względu na to, że składnia XML jest oparta na zwykłym tekście, dokumenty XML mogą być czytane i edytowane przy użyciu najprostszych edytorów tekstu.

Język XML nie ma ustalonego słownika, dlatego można tworzyć słowniki przeznaczone dla określonych aplikacji lub branż. Opracowano już słowniki XML dla różnych gałęzi przemysłu i wielu zastosowań (np. dla dokumentów internetowych opracowano standard XHTML, a dla protokołów sieciowych — standard SOAP (ang. *Simple Object Access Protocol*) — protokół wywoływanego zdalnego dostępu do obiektów). Aplikacje wykorzystujące formaty XML są bardziej odporne na zmiany w strukturze danych niż aplikacje oparte na innych formatach.

XML nie jest związany z żadnym językiem programowania ani z żadnym systemem operacyjnym. Dzięki temu dokumenty można tworzyć i edytować, korzystając z różnych języków programowania, a XML może współpracować z różnymi systemami operacyjnymi.

Zapisane w języku XML dokumenty biznesowe i bazy danych mogą być udostępniane jako źródło wymiany informacji pomiędzy firmami lub są formą prezentacji danych wszędzie tam, gdzie wymagana jest współpraca różnych platform.

## Składnia XML 1.0

Specyfikacja XML 1.0 określa tekstowy format do opisu danych przy użyciu składni podobnej do składni języka HTML.

Dokument jest zbudowany z elementów, a każdy element składa się ze znacznika otwierającego (np. `<table>`), znacznika zamk傢acego (np. `</table>`) oraz z informacji znajdujących się pomiędzy tymi znacznikami. Elementy mogą być opatrywane atrybutami zawierającymi metadane opisujące element i jego zawartość.

### 1.2.3. XHTML (ang. Extensible HyperText Markup Language)

Język XHTML to język, który dostosowuje zasady języka HTML4 do specyfikacji języka XML. Opiera się na specyfikacji języka HTML 4.01, ale równocześnie wprowadza obostrzenia wynikające ze składni języka XML. Dokumenty pisane w tym języku są zgodne ze składnią XML, dlatego mogą być przeglądane, edytowane i sprawdzane za pomocą standardowych narzędzi XML. Strony pisane w XHTML są wyświetlane przez przeglądarki obsługujące zarówno język HTML, jak i XHTML. W XHTML jest stosowana taka sama składnia języka jak w HTML, ale zostały zdefiniowane bardziej rygorystyczne zasady, które dla projektantów stron i twórców przeglądarek internetowych stały się obowiązującym standardem:

- Znaczniki muszą być zamkane w kolejności odwrotnej do tej, w której były otwierane.
- Nazwy znaczników i atrybutów muszą być pisane małymi literami.
- Znaczniki zamk傢ce są obowiązkowe.
- Wartości atrybutów muszą być ujęte w cudzysłowy.
- Nie można pomijać atrybutów logicznych.
- Elementy puste muszą mieć znacznik zamk傢cy lub ich znacznik otwierający musi się kończyć sekwencją `>`, np. `<br />`.

Do dokumentu utworzonego w języku XHTML musi zostać wstawiona deklaracja typu dokumentu (DTD). Określa ona, która wersja języka HTML jest stosowana na stronie.

W XHTML 1.0 można wykorzystywać jedną z trzech wersji tej deklaracji.

**Wersja Strict (ścisła)** zawiera znaczniki i atrybuty, które nie są zdeprecjonowane i nie pojawiają się w dokumentach z ramkami; nie obejmuje również atrybutu `target=" "` oraz znacznika `<iframe>`.

Postać deklaracji dla wersji Strict:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

**Wersja Transitional (przejściowa)** zawiera wszystkie elementy Strict, elementy zdeprecjonowane oraz znacznik `<iframe>` i atrybut `target=" "`.

Postać deklaracji dla wersji Transitional:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

**Wersja Frameset (ramkowa)** zawiera wszystkie elementy Transitional oraz elementy odnoszące się do ramek (`frame`, `frameset`, `noframes`).

Postać deklaracji dla wersji Frameset:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

W wersji XHTML 1.1 stosowana jest tylko jedna wersja deklaracji:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

Deklaracja XML zarówno dla wersji XHTML 1.0, jak i dla wersji XHTML 1.1 ma postać:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Użycie jej nie jest wymagane, ale zaleca się umieszczanie tej deklaracji szczególnie wtedy, gdy kodowanie znaków jest inne niż UTF-8.

**UWAGA**

Znaczniki i atrybuty zdeprecjonowane (dotyczy to języków HTML i XHTML) to elementy, które są uznawane za przestarzałe. Zaleca się nieużywanie ich i korzystanie z nowszych technik, np. z arkuszy stylów CSS. Zdeprecjonowane znaczniki i atrybuty są dozwolone w wersji Transitional języków HTML 4.01 i XHTML 1.0, ale nie są dozwolone w wersji Strict.

## 1.2.4. HTML5

Język HTML5 jest rozwinięciem języków HTML4 i XHTML1. Jest on kompatybilny z poprzednimi wersjami języka HTML. W nowej wersji dopracowano specyfikację języka HTML4 (przede wszystkim w zakresie obsługi błędów przez przeglądarki) oraz usprawniono narzędzia tworzenia serwisów i aplikacji internetowych. Pojawiły się nowe znaczniki i typy wejść oraz nowe atrybuty elementów formularzy. Wprowadzono obsługę wideo i dźwięku. Język HTML5 został skonstruowany tak, że starsze przeglądarki bez problemu mogą ignorować jego nowe polecenia. Brak wsparcia ze strony przeglądarek internetowych i brak kompatybilności z wcześniejszymi wersjami spowodował odwrót od XHTML, więc wielkość liter w znacznikach i atrybutach nie ma znaczenia; również domykanie wszystkich znaczników nie jest konieczne. Nie ma znaczników i atrybutów deprecjonowanych, ale niektóre elementy są klasyfikowane jako przestarzałe i nie zaleca się ich używania. Mimo to wiele dokumentów pisanych w HTML zachowuje składnię typową dla XHTML ze względu na jasno określone reguły tego języka.

Standard HTML5 upraszcza zapis deklaracji kilku elementów.

Deklaracja typu dokumentu ma postać:

```
<!DOCTYPE html>
```

Sposób kodowania znaków ma postać:

```
<meta charset="UTF-8" />
```

Dołączenie arkusza stylów nie wymaga atrybutu type:

```
<link rel="stylesheet" href="file.css">
```

Deklaracja JavaScript również nie wymaga atrybutu type:

```
<script src="file.js"></script>
```

Zgodnie z zasadami obowiązującymi w HTML5 następuje rozdzielenie struktury strony, za którą odpowiada HTML, od jej wizualizacji, za którą to odpowiadają arkusze stylów CSS. Układ stron buduje się za pomocą specjalnych pojemyńników (znacznik `<div>` w HTML4), tzw. pojemyńników semantycznych (znaczniki `<header>`, `<nav>`, `<footer>`, `<section>`, `<article>`, `<aside>`), które są formatowane przy użyciu stylów CSS. Nowe elementy nie tylko poprawiają strukturę strony internetowej, powodując, że

struktura ta staje się zrozumiała i logiczna, ale określają też przeznaczenie zapisanych w nich treści. Dzięki temu zapisane w elementach semantycznych treści mogą być wykorzystywane przez różnego rodzaju usługi i narzędzia (wyszukiwarki, validatory stron, pozycjonowanie stron).

## 1.3. Kodowanie polskich znaków

Język HTML umożliwia obsługę znaków różnych języków narodowych, w tym znaków języka polskiego. Aby było to możliwe, musi obsługiwać standardy kodowania tych znaków.

Jednym z pierwszych standardów, który obsługiwał polskie litery, był standard ISO 8859-2. Jest on zgodny z podstawowym standardem ASCII, ale poza zakresem zgodnym z ASCII występują konflikty między znakami reprezentującymi każdy bajt. Obecnie istnieje 15 wariantów tego standardu. Niektóre wersje Microsoft Windows przy kodowaniu polskich znaków korzystają ze standardu Windows-1250, który jest podobny do standardu ISO 8859-2 (występują niewielkie różnice, ale obydwa standardy zachowują zgodność z ASCII).

Powstały jeszcze inne standardy kodowania znaków narodowych, np. znaków koreańskich, japońskich, cyrylicy. Dodatkowo różne systemy operacyjne wykorzystywały inne systemy kodowania dla tych samych języków. Stosowanie różnych sposobów kodowania oraz często występujący brak zgodności standardów z ASCII uniemożliwiały komunikację.

Aby temu zaradzić, opracowano nowy sposób kodowania znaków — Unicode — który dla 17 zbiorów 16-bitowych pozwala zapisać 1 114 112 znaków. Jeden zbiór 16-bitowy zawiera 65 536 znaków, czyli więcej, niż może wykorzystać przeciętny użytkownik.

Najpopularniejsze wersje Unicode to UTF (ang. *Unicode Transformation Format*) i UCS (ang. *Universal Character Set*).

UTF-8 jest najpopularniejszą wersją Unicode ze względu na swoją zgodność z kodowaniem 8-bitowym. Wykorzystuje kodowanie ze zmienną długością znaków. Oznacza to, że każdy znak może być opisany przez 1–4 bajty. Pierwszy bajt używany jest do kodowania ASCII, a pozostałe bajty są wykorzystywane w zależności od stosowanego alfabetu.

UTF-8 to standard, który pozwala na pracę w środowisku wielojęzycznym i jest najprostszym sposobem przesyłania znaków spoza ASCII przez internet.

Obecnie w języku HTML powszechnie używa się kodowania UTF-8, co rozwiązuje problem wyświetlania różnych międzynarodowych znaków diakrytycznych. Także edytory tekstu zapisują dokumenty z wykorzystaniem Unicode.

W wielobajtowym kodowaniu znaków na początku strumienia bajtów może zostać dopisany znak niedrukowany BOM (ang. *byte order mark*). Informuje on, w jakiej kolejności powinny zostać ustalone bajty, aby poprawnie odczytać kod znaku. W kodowaniu

UTF-8 nie jest on konieczny, ponieważ kolejność bajtów jest jednoznaczna. Niektóre edytory tekstu przy kodowaniu wielobajtowym dodają znacznik BOM jako znak niedrukowany na początku pliku, co może prowadzić do błędnej interpretacji zawartości skryptów zapisanych w kodzie UTF-8.

W nagłówku dokumentu HTML można określić sposób kodowania znaków, definiując znacznik `<meta>`, np.:

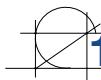
```
<meta charset="UTF-8">
```

Można również stosować kodowanie ISO albo Windows:

```
<meta charset="iso-8859-2">
```

lub

```
<meta charset="windows-1250">
```



## 1.4. Edytory stron WWW

Do pisania kodu strony internetowej wystarczy najprostszy edytor tekstu, należy jednak pamiętać o kontrolowaniu składni języka HTML. W tworzonym dokumencie istotne znaczenie ma każdy nawias czy cudzysłów. Ich brak uniemożliwi przeglądarce prawidłowe zinterpretowanie zapisanego kodu. Aby ułatwić i przyspieszyć pracę z dokumentem, można skorzystać z wyspecjalizowanych edytorów tekstu, które mają wbudowane różne narzędzia, np.: autouzupełnianie, podpowiedzi, schematy kolorowania, pomoc kontekstową.

Najprostszym i zawsze dostępnym w systemie Windows edytorem tekstu jest **Notatnik**. Podobny edytor tekstu dla systemu Linux to **Pico**, a dla systemu Mac OS — **TextEdit**.

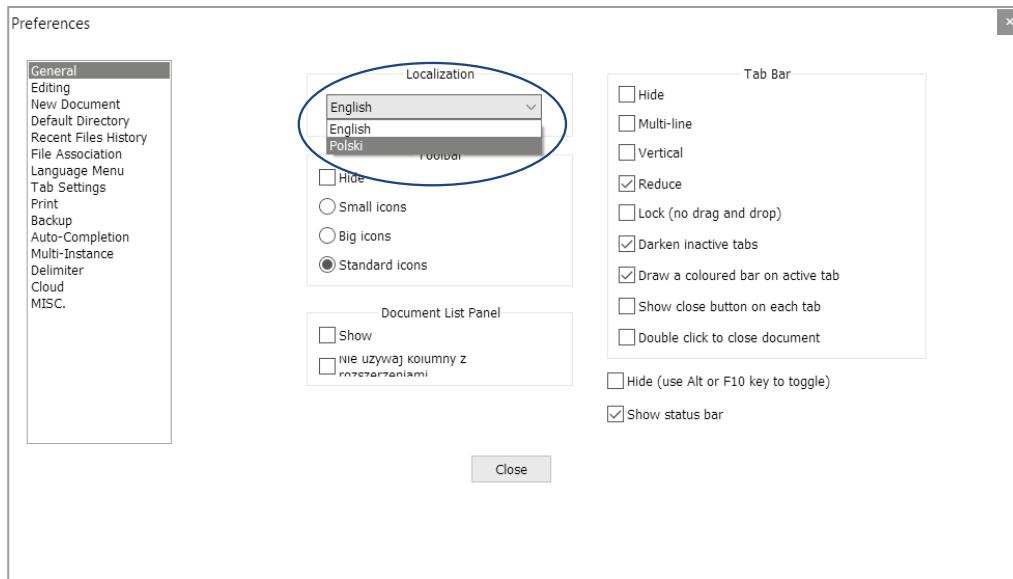
Inny edytor tekstu to **Notepad++**. Jest to bardzo dobry, bezpłatny edytor kodów źródłowych, dostępny w języku polskim i dający możliwość kodowania polskich znaków.

Prosty edytor tekstu jest najlepszym narzędziem do tworzenia dokumentów w językach HTML i CSS, ponieważ wprowadzony przez projektanta kod nie zostanie automatycznie zmieniony ani sformatowany.

Do testowania działania kodu można używać dowolnej przeglądarki internetowej, należy jedynie pamiętać o jej uaktualnieniu.

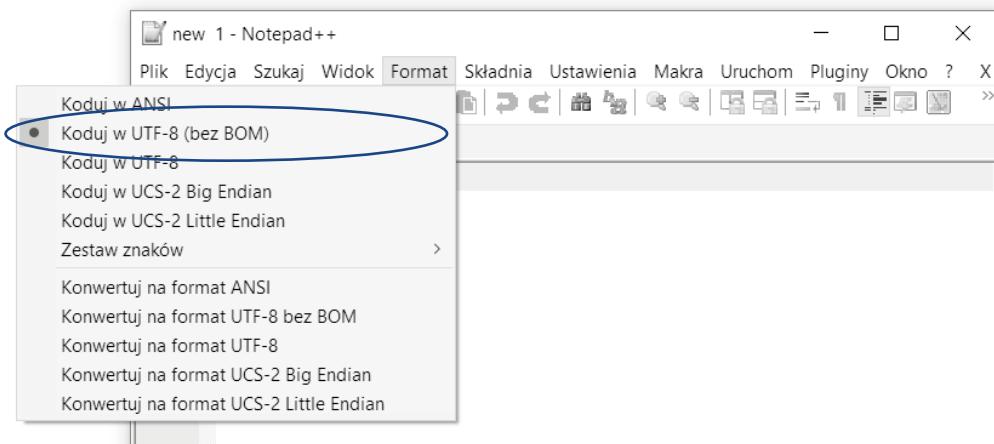
### 1.4.1. Notepad++

Plik instalacyjny programu Notepad++ można pobrać ze strony <http://notepad-plus-plus.org/>. Po zainstalowaniu programu należy ustawić jego polską wersję językową — wybrać **Settings/Preferences**, po czym w opcji **General** wskazać język **Polski** (rysunek 1.1).



**Rysunek 1.1.** Ustawienie polskiej wersji językowej w edytorze Notepad++

Aby polskie znaki były prawidłowo interpretowane, należy ustawić odpowiedni format kodowania. W tym celu trzeba w menu *Format* wybrać z dostępnej listy żądaną opcję. Najlepiej wybrać opcję *Koduj w UTF-8 (bez BOM)* (rysunek 1.2). Jest to standardowy i zalecany format kodowania polskich znaków.



**Rysunek 1.2.** Ustawienie opcji kodowania znaków

## Zapisywanie plików

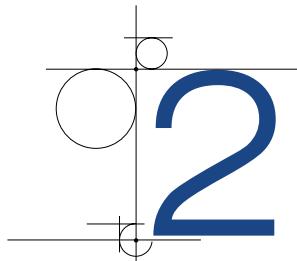
Pliki zawierające dokumenty HTML powinny być zapisywane z rozszerzeniem `.html` lub `.htm`.

Plik o nazwie `index.html` (lub `index.htm`) domyślnie jest traktowany jako strona startowa dla danej witryny. Jeśli plik o takiej nazwie znajduje się w folderze z witryną, to właśnie jego zawartość zostanie automatycznie wyświetlona po wywołaniu jej adresu. Z tego powodu należy stronę startową witryny zawsze nazywać `index.html` (lub `index.htm`) i umieścić ją w wybranym katalogu konta (zwyczajowo jest to `public_html`). Nazwa pliku nie powinna zawierać znaków spacji ani polskich liter. Warto także pamiętać o tym, że małe i duże litery w nazwach plików są rozróżniane i mogą być przyczyną licznych problemów. Dlatego dobrym zwyczajem jest przyjęcie zasad, w myśl której wszystkie nazwy plików (wraz z rozszerzeniem) zapisujemy za pomocą małych liter.

### UWAGA

Jeżeli w folderze z witryną znajdują się dwa pliki — jeden o nazwie `index.html`, drugi o nazwie `index.htm` — to dla domyślnej konfiguracji serwera HTTP plik `index.html` ma pierwszeństwo wyświetlania przed plikiem `index.htm`, a więc będzie pełnił funkcję pliku startowego.





# Język HTML5



## 2.1. Struktura strony internetowej

Każdy dokument HTML składa się ze znaczników, które mówią przeglądarce, w jaki sposób ma wyświetlać teksty, obrazy, animacje, filmy.

### 2.1.1. Znaczniki

Znaczniki są słowami lub znakami umieszczonymi wewnątrz nawiasów < i >. Elementów umieszczonych poza nawiasami przeglądarka nie interpretuje (są to te elementy, które będą wyświetlane na ekranie). Znaczniki są poleceniami wysyłanymi do przeglądarki i składają się z dwóch części: polecenia otwierającego <*nazwa polecenia*> i polecenia zamkijającego </*nazwa polecenia*>, a pomiędzy nimi może znajdować się odpowiednia treść wyświetlana na stronie. Istnieją pojedyncze znaczniki, które nie posiadają oddzielnych elementów zamkijających. Taki znacznik zapisujemy w postaci <*nazwa polecenia*> lub <*nazwa polecenia* />.

#### Przykład 2.1

```
<p>Podróż dookoła świata</p>
```

Wpisanie podanego fragmentu kodu spowoduje wyświetlenie w nowym akapicie tekstu zawartego między znacznikami <p> i </p>.

Znaczniki mogą zawierać atrybuty.

#### Przykład 2.2

```

```

Jeśli obrazek z jakiegoś powodu nie może zostać wyświetlony na stronie, atrybut alt="góry" spowoduje wyświetlenie tekstu *góry*, a atrybuty height i width określą wielkość wyświetlonego obrazka.

Każdy dokument tekstowy powinien zaczynać się od znacznika <html> i kończyć się znacznikiem </html>. Znaczniki te informują przeglądarkę, że wszystko zawarte

między nimi ma być interpretowane jako strona internetowa. Poza tym każda strona internetowa zawiera nagłówek (`<head>` i `</head>`) oraz ciało (`<body>` i `</body>`), w którym zawarta jest treść dokumentu. W sekcji `<head>` powinien pojawić się tytuł strony umieszczony w znacznikach `<title>` i `</title>`. Przed sekcją `<html>` powinien zostać umieszczony jeszcze znacznik zawierający informacje na temat standardu języka HTML, na podstawie którego tworzymy stronę internetową, a w sekcji `<head>` powinien się znajdować znacznik `<meta>`, który przekaże do przeglądarki informację, że to, co ona odczyta z sekcji `<body>`, będzie dokumentem HTML oraz że kodowanie znaków zostanie sformatowane w standardzie **UTF-8**, co oznacza, że gdy pojawią się polskie litery, przeglądarka prawidłowo je zinterpretuje.

### Przykład 2.3

```
<!DOCTYPE html>

<html lang="pl-PL">

<head>

    <title>

        ...

    </title>

    ...

</head>

<body>

    ...

</body>

</html>
```

Podany przykład pokazuje elementy, z których składa się każda strona internetowa. Należy wiedzieć, jak jest zbudowana każda strona internetowa i jak zinterpretować treść dokumentu tekstowego. Po sprawdzeniu kodu tworzonej strony można dokonać modyfikacji jej zawartości i ręcznie dopisać nowe fragmenty.

W pierwszej linii kodu została umieszczona deklaracja typu dokumentu. Określa ona, że zawartość utworzonego dokumentu powstała na bazie struktury HTML5, co oznacza, że strona będzie wyświetlana w dowolnej przeglądarce internetowej zgodnie ze standardami zdefiniowanymi dla HTML5.

### Przykład 2.4

```
<!DOCTYPE html>

<html lang="pl-PL">

<head>

    <title>Dookoła świata</title>

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

```
</head>
<body>
  <p>Podróż dookoła świata</p>
</body>
</html>
```

## 2.1.2. Nagłówek

Nagłówek dokumentu jest zawarty między znacznikami `<head>` i `</head>`. W nim znajduje się informacja o dokumencie HTML.

Najważniejsze polecenia umieszczane w nagłówku to znacznik `<title>`, określający tytuł dokumentu, oraz znacznik `<meta>`, który przechowuje informacje o dokumencie i umożliwia wyszukiwarkom indeksowanie stron.

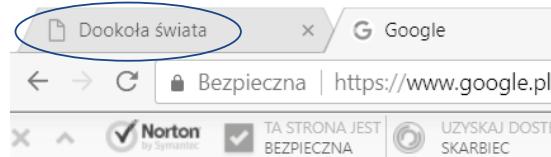
### Tytuł dokumentu

Znacznik `<title>` służy do wyświetlania tytułu strony w pasku zadań przeglądarki (rysunek 2.1) i ma postać:

```
<title>Dookoła świata</title>
```

**Rysunek 2.1.**

Tytuł strony wyświetlany na karcie przeglądarki



Zaleca się nadawanie tytułów, które bezpośrednio informują o zawartości strony, ponieważ wyszukiwarki internetowe umieszczają te informacje w swoich katalogach czy indeksach. Podany tytuł jest również widoczny w zakładkach przeglądarki.

### Znacznik `<meta>`

Znaczniki `<meta>` informują: kto jest autorem strony, kiedy strona została utworzona, jakie zastosowano kodowanie znaków, ponadto decydują o miejscu strony w wynikach wyszukiwania. Występują z trzema atrybutami: `http-equiv`, `name`, `content`, które mogą przyjmować różne wartości.

Atrybut `http-equiv` dostarcza przeglądarce internetowej informacji na temat otwieranej strony i steruje jej zachowaniem. Może przyjmować wartości:

- `Content-Type` — określa sposób kodowania znaków.
- `Content-Language` — określa język zawartości dokumentu.
- `Reply-to` — podaje adres e-mail do kontaktowania się z autorem.

Atrybut `name` pozwala umieszczać niezbędne dla wyszukiwarek informacje dotyczące strony internetowej. Może przyjmować wartości:

- `Title, page-topic` — ustawia tytuł i opis strony.
- `Creation_Date` — ustawia datę utworzenia strony.
- `Keywords` — definiuje słowa kluczowe, które opisują zawartość strony.
- `Description` — definiuje opis strony widoczny w wynikach wyszukiwania.
- `Robots` — określa, czy dana strona powinna być indeksowana.

### Strona kodowa dokumentu

Dołączenie do kodu znacznika w postaci `<meta charset="UTF-8">` informuje o sposobie kodowania strony. Ale aby polskie znaki były prawidłowo interpretowane przez przeglądarkę, należy jeszcze ustawić odpowiedni format kodowania w edytorze tekstu podczas zapisywania strony.

Na przykład w **Notatniku** (Windows) w opcji *Zapisz jako* z listy sposobów kodowania u dołu okna należy wybrać opcję **UTF-8**, w **NotePad++** — *Format/Koduj w UTF-8 (bez BOM)*, w **TextEdit** (Mac OS) — *Format/Make Plain Text* w celu zapisania w zwykłym pliku tekstowym, a następnie należy wybrać w menu *Pain Text Encoding* i w otwartym oknie *Save As* wybrać opcję **Unicode (UTF-8)**.

### Opis strony

```
<meta name="description" content="Strona poświęcona ciekawym miejscom na  
świecie">
```

Zapis znacznika `<meta>` w podanej postaci opisuje zawartość strony. Wyszukiwarki internetowe, wyszukując strony o określonej tematyce, informację zawartą w tym znaczniku wyświetlają jako opis zawartości znalezionej strony. Treść zamieszczona w opisie nie powinna przekraczać 230 znaków.

### Wyrazy kluczowe

```
<meta name="keywords" content="świat, kontynenty, podróże">
```

Użycie znacznika w podanej postaci przekaże wyszukiwarkom informacje o istotnych elementach strony określonych przez podane słowa kluczowe i ułatwi jej wyszukanie.

### Język dokumentu

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

Zapis znacznika w podanej postaci poinformuje przeglądarki internetowe o języku strony. Pierwsze dwie litery, **pl**, są deklaracją używanego języka, kolejne litery, **PL**, są deklaracją kraju, w którym ten język jest używany.

## Inne przykłady użycia znacznika <meta>

```
<meta http-equiv="refresh" content="x">
```

Takie użycie znacznika spowoduje automatyczne odświeżanie dokumentu co x sekund.

```
<meta http-equiv="refresh" content="x; url=adres">
```

Tak zastosowany znacznik spowoduje przekierowanie użytkownika po upływie x sekund do podanego adresu URL.

```
<meta name="title" content="tytuł strony">
```

Tak użyty znacznik definiuje tytuł strony lub jej opis.

## Znacznik <link>

Znacznik <link> wykorzystywany jest do dołączania do strony zewnętrznych plików. Zwykle stosuje się go do dołączenia zewnętrznego pliku CSS i wtedy ma postać <link rel="stylesheet" type="text/css" href="...">.

Atrybut `rel` określa, w jaki sposób zewnętrzny plik jest powiązany ze stroną. Wartość atrybutu `stylesheet` oznacza plik ze stylami.

Zapis znacznika w postaci:

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

dołączy do strony internetowej plik `styl.css` znajdujący się w folderze `style` w katalogu równorzędnym.

Znacznik <link> może być również stosowany do dołączenia do strony ikony pomagającej w identyfikacji strony. Jest to tzw. `favicon` (ikona ulubionych). Ikona ta zostanie wyświetlona na karcie strony przed jej tytułem.

Znacznik zapisany w postaci:

```
<link rel="icon" href="../img/favicon.png" type="image/png">
```

wyświetli zawartość pliku `favicon.png` znajdującego się w folderze `img` w katalogu równorzędnym.

## Adres bazowy

Znacznik <base> określa adres bazowy dla wszystkich adresów na stronie.

W dokumentach strony internetowej znajdują się odsyłacze do innych stron na serwerze. Gdy dokumenty są przenoszone w inne miejsce, odsyłacze mogą stracić swoją aktualność i przestać działać. Aby zapewnić podczas przenoszenia poprawność działania relatywnych odsyłaczy, można zdefiniować za pomocą znacznika <base> bazowy adres dokumentu. Przeglądarka internetowa będzie się kierowała bazowym adresem przy odnajdowaniu adresu odsyłacza.

## Przykład 2.5

```
<body>  
      
</body>
```

Podany zapis jest równoważny zapisowi:

```
<head>  
    <base href="http://www.pxp.com/">  
</head>  
  
<body>  
      
</body>
```

Dla obrazu określony został tylko adres względny. Ponieważ w sekcji `<head>` podany został adres bazowy, przeglądarka wyszuka obraz w <http://www.pxp.com/img/europa.jpg>.

## Przykład 2.6

Po uwzględnieniu wszystkich podanych elementów składni dokumentu HTML powinien wyglądać następująco:

```
<!DOCTYPE html>  
  
<html lang="pl-PL">  
  
<head>  
    <title>Dookoła świata</title>  
    <meta charset="UTF-8">  
    <link rel="stylesheet" type="text/css" href="style/styl.css">  
</head>  
  
<body>  
    <p>Podróż dookoła świata</p>  
</body>  
</html>
```

### 2.1.3. Ciało dokumentu

Ciało dokumentu to jego część zawarta między znacznikami `<body>` i `</body>`. Zawiera właściwą treść dokumentu, pokazywaną użytkownikowi przez przeglądarkę.

Układ i zawartość tego bloku definiujemy za pomocą specjalnych znaczników HTML5, natomiast wygląd tej części strony dokumentu HTML definiujemy przy użyciu stylów CSS.



## 2.2. Elementy składowe strony WWW

Specyfikacja HTML5 oraz CSS3 to standardy, które stanowią platformę do tworzenia aplikacji internetowych. Za wygląd interfejsu użytkownika odpowiadają style CSS. Natomiast za pomocą znaczników kodu HTML w głównej mierze definiujemy układ dokumentu. Dlatego w HTML5 większość znaczników odpowiedzialnych za prezentację została unieważniona. Należą do nich m.in.: <basefont>, <big>, <center>, <font>, <s>, <strike>, <tt>, <u>. Poza tym zostało unieważnionych także wiele atrybutów związanych z prezentacją treści. Dotyczy to atrybutów: align, link, vlink, alink i text, a dla znacznika <body> dodatkowo atrybutów: bgcolor, height, width, valign, hspace i vspace.

### 2.2.1. Elementy blokowe dokumentu HTML

#### Akapit (paragraf)

Przeglądarki internetowe ignorują umieszczany w tekście znak końca akapitu (naciśnięcie klawisza *Enter*) i wyświetlają tekst, który został podzielony na akapyty, w jednej ciągłej linii. Aby tekst umieszczony na stronie był pogrupowany w akapyty (paragrafy), powinien zostać otoczony znacznikami <p> i </p>.

Akapyty oznaczone w ten sposób będą oddzielane w tekście dodatkowym odstępem, który można kontrolować za pomocą CSS. Zapisywanie tekstu w postaci akapitów jest wygodną metodą formatowania jego fragmentów. Jeżeli na stronie umieszczać większy fragment tekstu, to każdy akapit musi zostać otoczony znacznikiem <p>.

#### Przykład 2.7

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Dookoła świata</title>
    <meta charset="UTF-8">
</head>
<body>
    <p>Wybieramy się w podróż dookoła świata.</p>
    <p>Będziemy odwiedzali po kolej wszystkie kontynenty.</p>
</body>
</html>
```

## Nowa linia

Inną metodą podziału wyświetlanego tekstu na oddzielne wiersze jest użycie znacznika `<br>`.

Znacznik `<br>` umieszczony na końcu wiersza przeniesie kolejny tekst do nowego wiersza bez wstawienia interlinii. Tego znacznika należy używać tylko do wprowadzania podziału wierszy, nie należy go stosować do oddzielania akapitów.

### Przykład 2.8

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Dookoła świata</title>
    <meta charset="UTF-8">
</head>
<body>
    Zwiedzanie Europy rozpoczniemy od Madrytu.<br>Zwiedzanie Azji
    rozpoczęliśmy od Pekinu. Najciekawszym krajem w Afryce
    jest Kenia.<br>Australię planujemy zwiedzać na końcu.
</body>
</html>
```

Znacznik łamania wiersza jako jeden z nielicznych nie wymaga znacznika zamykającego.

## Nagłówki

Język HTML umożliwia tworzenie nagłówków w sześciu rozmiarach. Do definiowania nagłówka używa się znacznika `<hx>`, gdzie `x` może przyjmować wartości z zakresu 1–6.

### Przykład 2.9

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Dookoła świata</title>
    <meta charset="UTF-8">
</head>
<body>
    <h1>Europa</h1>
    <h2>Azja</h2>
```

```

<h3>Afryka</h3>
<h4>Ameryka Północna</h4>
<h5>Ameryka Południowa</h5>
<h6>Australia</h6>
</body>
</html>

```

Znaczniki od `<h1>` do `<h6>` powinny być używane w dokumencie tylko do definiowania nagłówków. Nie należy ich używać do pisania tekstu pogrubionego lub do zmiany wielkości tekstu. Do tego celu powinny być stosowane inne znaczniki.

Aby zachować spójność dokumentu przy dzieleniu tekstu na rozdziały, podrozdziały i punkty, należy stosować nagłówki w odpowiedniej kolejności, np. nagłówek `<h2>` powinien zostać poprzedzony nagłówkiem `<h1>`, a nagłówek `<h3>` — nagłówkiem `<h2>`.

## Pogrubienie i kursywa

W HTML4 za pogrubienie czcionki odpowiadał znacznik `<b>`, a znacznik `<i>` służył do wypisania tekstu kursywą. W HTML5 znaczenie tych znaczników zostało zmienione.

Znacznikiem `<b>` otacza się tekst, na który należy zwrócić uwagę, ale nie nadaje się oznaczonej treści dodatkowej wagi. Może być on stosowany np. do oznaczania słów kluczowych w tekście.

Z pomocą znacznika `<i>` otacza się tekst, który powinien zostać wyróżniony. Może być on stosowany np. do zaznaczenia terminu technicznego czy zwrotu w obcym języku.

### Przykład 2.10

```

<!DOCTYPE html>
<html lang="pl-PL">
<head>
  <title>Dookoła świata</title>
  <meta charset="UTF-8">
</head>
<body>
  <b>Podróże kształca.</b>
  <i>Najprzyjemniej podróżować w towarzystwie.</i>
</body>
</html>

```

Znaczniki `<b>` oraz `<i>` nie są w standardzie HTML5 uznawane za przestarzałe, ale powinny być używane tylko wtedy, gdy nie ma innego elementu, za pomocą którego można uzyskać lepszy efekt formatowania.

## Pozioma linia

W HTML4 do narysowania i wyświetlenia na ekranie poziomej linii używany był znacznik `<hr>`. W HTML5 działanie tego znacznika nie uległo zmianie, natomiast zostało zmienione jego znaczenie. Jest on używany do zaznaczenia podziału tematycznego.

### Przykład 2.11

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Dookoła świata</title>
    <meta charset="UTF-8">
</head>
<body>
    <h1>Europa</h1>
    <p>Europa leży na półkuli północnej.</p>
    <hr>
    <h1>Afryka</h1>
    <p>Afryka to drugi pod względem wielkości kontynent na Ziemi.</p>
</body>
</html>
```

## Komentarze

W pracy z dokumentem HTML powinny być używane komentarze, które ułatwiają analizowanie jego kodu. Komentarze są wyświetlane tylko w kodzie dokumentu i nie są widoczne dla użytkownika strony internetowej. Umieszcza się je w następujący sposób:

```
<!--Komentarz do mojej strony internetowej-->
```

## Znaki specjalne

Jeżeli na stronie internetowej będą wyświetlane znaki specjalne, powinny one zostać wpisane w projekcie strony jako nazwy lub numery znaków. Wpisanie znaku bezpośrednio z klawiatury może zostać błędnie zinterpretowane przez przeglądarkę. Najważniejsze znaki specjalne to: `<`, `>` oraz `&`. Przeglądarka odczyta je jako kod języka HTML. Nazwy popularnych znaków specjalnych zostały podane w tabeli 2.1.

**Tabela 2.1.** Tabela znaków specjalnych

Znak specjalny	Nazwa	Opis
<code>&amp;</code>	<code>&amp;amp;</code>	znak „ampersand”
<code>&lt;</code>	<code>&amp;lt;</code>	znak mniejszości

Znak specjalny	Nazwa	Opis
>	&gt;	znak większości
§	&sect;	znak sekcji lub paragrafu
™	&trade;	zastrzeżony znak towarowy
©	&copy;	znak praw autorskich
®	&reg;	zastrzeżony znak towarowy
€	&euro;	znak „euro”
	&ampnbsp	znak zwarcej spacji

### Przykład 2.12

<p>Zastrzeżony znak towarowy może mieć symbol &trade; lub &reg;.</p>

### Znacznik <div>

Aby pogrupować wiele różnych elementów strony albo wydzielić większy fragment dokumentu, można wykorzystać znacznik <div>, który tworzy oddzielny blok. Znacznik ten jest podstawowym elementem, na którym opierają się nowoczesne układy stron. Za jego pomocą można podzielić stronę internetową na takie bloki jak: nagłówek, panele (np. panel lewy, panel prawy), paski nawigacji, stopka. Bloki utworzone przy użyciu znacznika <div> mogą być formatowane i pozycjonowane za pomocą arkuszy stylów CSS, w wyniku czego otrzymamy wymagany wygląd strony (layout).

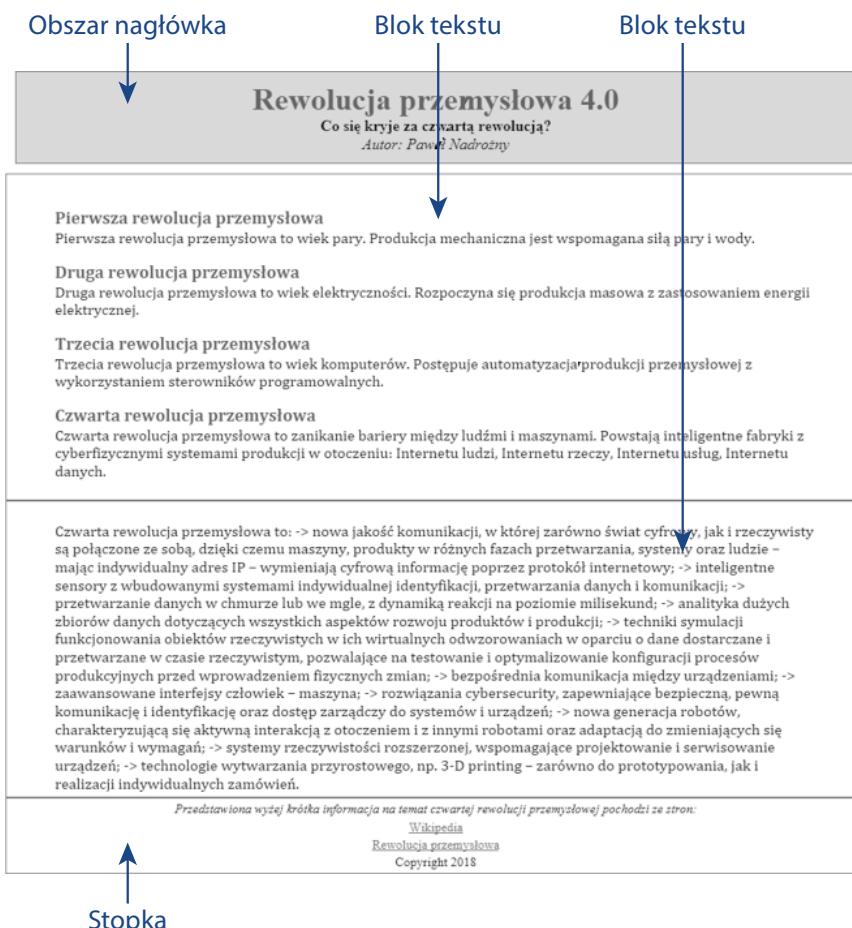
### Przykład 2.13

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Dookoła świata</title>
    <meta charset="UTF-8">
</head>
<body>
    <h2>Podróż dookoła świata.</h2>
    <div style="color: #000FFF">
        <h3>W maju wyruszamy w podróż dookoła świata.</h3>
        <p>W czerwcu odwiedzimy Nowy Jork.</p>
    </div>
    <h3>Koniec podróży nastąpi w październiku.</h3>
</body>
</html>
```

Tekst zapisany w bloku otoczonym znacznikiem `<div>` zostanie wyświetlony w kolorze zadeklarowanym przez atrybut "color: #000FFF".

W HTML5 znacznik `<div>` nadal służy do grupowania elementów w bloki, ale pojawiła się nowa grupa znaczników, za pomocą których można tworzyć szkielet strony.

Schemat strony utworzonej przy użyciu znaczników `<div>` został pokazany na rysunku 2.2.



**Rysunek 2.2.** Schemat strony utworzonej za pomocą znaczników `<div>`. Formatowanie strony zostało dołączone przy użyciu zewnętrznego arkusza stylów

## Ćwiczenie 2.1

Z pomocą poznanych znaczników języka HTML zaprojektuj stronę internetową. Strona nie powinna różnić się wyglądem od strony widocznej na rysunku 2.3. Na stronie tej nie powinny pojawić się znaczniki formatujące.

Na stronie należy zastosować standard kodowania polskich znaków (UTF-8). Tytuł strony *Nowe technologie* powinien być widoczny w karcie przeglądarki. Strona powinna zostać podzielona na bloki za pomocą znacznika `<div>`. Widoczne na rysunku obramowania określają bloki istniejące na stronie (nagłówek, dwa bloki opisu, stopka). Tekst bloku *obszar nagłówka — Rewolucja przemysłowa 4.0* powinien zostać umieszczony w nagłówku pierwszego stopnia, tekst *Co się kryje...* — w nagłówku trzeciego stopnia, a tekst *Autor...* — w akapicie. Teksty wyróżnione w pierwszym bloku opisu powinny zostać umieszczone w nagłówku drugiego stopnia, a pozostałe teksty — w akapicie. Tekst zapisany w drugim bloku opisu powinien znaleźć się w akapicie. Teksty z bloku *stopka* powinny zostać zapisane w oddzielnych akapitach.

## Rewolucja przemysłowa 4.0

Co się kryje za czwartą rewolucją?

Autor: Paweł Nadrożny

### Pierwsza rewolucja przemysłowa

Pierwsza rewolucja przemysłowa to wiek pary. Produkcja mechaniczna jest wspomagana siłą pary i wody.

### Druga rewolucja przemysłowa

Druga rewolucja przemysłowa to wiek elektryczności. Rozpoczyna się produkcja masowa z zastosowaniem energii elektrycznej.

### Trzecia rewolucja przemysłowa

Trzecia rewolucja przemysłowa to wiek komputerów. Postępuje automatyzacja produkcji przemysłowej z wykorzystaniem sterowników programowalnych.

### Czwarta rewolucja przemysłowa

Czwarta rewolucja przemysłowa to zanikanie bariery między ludźmi i maszynami. Powstają inteligentne fabryki z cyberfizycznymi systemami produkcji w otoczeniu: Internetu ludzi, Internetu rzeczy, Internetu usług, Internetu danych.

Czwarta rewolucja przemysłowa to: -> nowa jakość komunikacji, w której zarówno świat cyfrowy, jak i rzeczywisty są połączone ze sobą, dzięki czemu maszyny, produkty w różnych fazach przetwarzania, systemy oraz ludzie – mając indywidualny adres IP – wymieniają cyfrową informację poprzez protokół internetowy; -> inteligentne sensory z wbudowanymi systemami indywidualnej identyfikacji, przetwarzania danych i komunikacji; -> przetwarzanie danych w chmurze lub we mgle, z dynamiką reakcji na poziomie milisekund; -> analityka dużych zbiorów danych dotyczących wszystkich aspektów rozwoju produktów i produkcji; -> techniki symulacji funkcjonowania obiektów rzeczywistych w ich wirtualnych odwzorowaniach w oparciu o dane dostarczane i przetwarzane w czasie rzeczywistym, pozwalające na testowanie i optymalizowanie konfiguracji procesów produkcyjnych przed wprowadzeniem fizycznych zmian; -> bezpośrednią komunikację między urządzeniami; -> zaawansowane interfejsy człowiek – maszyna; -> rozwiązania cybersecurity, zapewniające bezpieczną, pewną komunikację i identyfikację oraz dostęp zarządcy do systemów i urządzeń; -> nowa generacja robotów, charakteryzująca się aktywną interakcją z otoczeniem i z innymi robotami oraz adaptacją do zmieniających się warunków i wymagań; -> systemy rzeczywistości rozszerzonej, wspomagające projektowanie i serwisowanie urządzeń; -> technologie wytwarzania przyrostowego, np. 3-D printing – zarówno do prototypowania, jak i realizacji indywidualnych zamówień.

Przedstawiona wyżej krótka informacja na temat czwartej rewolucji przemysłowej pochodzi ze stron:

[Wikipedia](#)

[Rewolucja przemysłowa](#)

Copyright 2018

**Rysunek 2.3.** Podział dokumentu na bloki

## Rozwiązanie

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Nowe technologie</title>
    <meta charset="UTF-8">
</head>
<body>
    <div>
        <h1>Rewolucja przemysłowa 4.0</h1>
        <h3>Co się kryje za czwartą rewolucją?</h3>
        <p>Autor: Paweł Nadrożny</p>
    </div>
    <div>
        <h2>Pierwsza rewolucja przemysłowa</h2>
        <p>Pierwsza rewolucja przemysłowa to wiek pary. Produkcja mechaniczna jest wspomagana siłą pary i wody.</p>
        <h2>Druga rewolucja przemysłowa</h2>
        <p>Druga rewolucja przemysłowa to wiek elektryczności. Rozpoczyna się produkcja masowa z zastosowaniem energii elektrycznej.</p>
        <h2>Trzecia rewolucja przemysłowa</h2>
        <p>Trzecia rewolucja przemysłowa to wiek komputerów. Postępuje automatyzacja produkcji przemysłowej z wykorzystaniem sterowników programowalnych.</p>
        <h2>Czwarta rewolucja przemysłowa</h2>
        <p>Czwarta rewolucja przemysłowa to zanikanie bariery między ludźmi i maszynami. Powstają inteligentne fabryki z cyberfizycznymi systemami produkcji w otoczeniu: Internetu ludzi, Internetu rzeczy, Internetu usług, Internetu danych.</p>
    </div>
    <div>
        <p>Czwarta rewolucja przemysłowa to:
            -> nowa jakość komunikacji, w której zarówno świat cyfrowy, jak i rzeczywisty są połączone ze sobą, dzięki czemu maszyny, produkty w różnych fazach przetwarzania, systemy oraz ludzie – mając indywidualny adres IP – wymieniają cyfrową informację poprzez protokół internetowy;
            -> inteligentne sensory z wbudowanymi systemami indywidualnej identyfikacji, przetwarzania danych i komunikacji;
    </div>

```

- > przetwarzanie danych w chmurze lub we mgle, z dynamiką reakcji na poziomie milisekund;
- > analityka dużych zbiorów danych dotyczących wszystkich aspektów rozwoju produktów i produkcji;
- > techniki symulacji funkcjonowania obiektów rzeczywistych w ich wirtualnych odwzorowaniach w oparciu o dane dostarczane i przetwarzane w czasie rzeczywistym, pozwalające na testowanie i optymalizowanie konfiguracji procesów produkcyjnych przed wprowadzeniem fizycznych zmian;
- > bezpośrednią komunikację między urządzeniami;
- > zaawansowane interfejsy człowiek – maszyna;
- > rozwiązania cybersecurity, zapewniające bezpieczną, pewną komunikację i identyfikację oraz dostęp zarządczy do systemów i urządzeń;
- > nowa generacja robotów, charakteryzująca się aktywną interakcją z otoczeniem i z innymi robotami oraz adaptacją do zmieniających się warunków i wymagań;
- > systemy rzeczywistości rozszerzonej, wspomagające projektowanie i serwisowanie urządzeń;
- > technologie wytwarzania przyrostowego, np. 3-D printing – zarówno do prototypowania, jak i realizacji indywidualnych zamówień.</p>

</div>

<div>

<p>Przedstawiona wyżej krótka informacja na temat czwartej rewolucji przemysłowej pochodzi ze stron:</p>

<p>Wikipedia</p>

<p>Rewolucja przemysłowa</p>

<p>Copyright 2018</p>

</div>

</body>

</html>

## Ćwiczenie 2.2

Przygotuj krótkie teksty opisujące kontynenty. Za pomocą poznanych znaczników języka HTML zaprojektuj stronę internetową podobną do strony pokazanej na rysunku 2.4. Widoczne na rysunku bloki utwórz, stosując znacznik <div>. Utworzone na stronie tytuły otocz odpowiednimi nagłówkami <h1> ... <h6>. Umieść przygotowane wcześniej teksty w odpowiednich blokach i podziel je na akapity. Zaproponuj teksty, które znajdą się w nagłówku i stopce strony.

## Podróż dookoła świata

Wybieramy się w podróż dookoła świata

Przewodnikiem będzie Adam Bilik

### Zwiedzamy Afrykę

Afryka to drugi pod względem wielkości kontynent na Ziemi. Zajmuje 30,37 mln km<sup>2</sup>, czyli ponad 20,3% ogólnej powierzchni lądowej świata. Przechodzą przez niego południk 0°, obydwa zwrotniki i równik.

### Jedziemy do Azji

Azja razem z Europą tworzą Eurazję, największy kontynent na Ziemi. Z powodów historycznych i kulturowych sama Azja była również nazywana kontynentem. Nazywana jest często kontynentem wielkich kontrastów geograficznych.

### Płyniemy do Australii

Nazwa "Australia" pochodzi od łacińskiego określenia "Terra Australis", czyli "Ziemia Południowa". W dawnej Europie wierzano, że na półkuli południowej znajduje się naprawdę duży kontynent. Ostatecznie nazwa ta przypadła kontynentowi o wiele mniejszemu niż zakładano.

### Teraz odwiedzamy Amerykę Południową

Ameryka Południowa to kontynent o powierzchni 17,8 mln km<sup>2</sup>, leżący na półkuli zachodniej oraz w większej części na półkuli południowej, a w mniejszej – na półkuli północnej. Niekiedy uważana jest również za subkontynent Ameryki.

### Podążamy do Ameryki Północnej

Ameryka Północna to kontynent o powierzchni 24 242 000 km<sup>2</sup> (co stanowi 16,3% całkowitej powierzchni lądów na kuli ziemskiej), położony na półkulach: północnej i zachodniej. Do Ameryki Północnej należy Ameryka Środkowa.

### Wracamy do Europy

Europa to kontynent o powierzchni 10,2 mln km<sup>2</sup>, co stanowi ok. 2% całej powierzchni kuli ziemskiej. Europa zajmuje 1/5 Eurazji i jako jej największy półwysep jest najbardziej wysunięta na zachód.

Podróż dookoła świata pozwoli Ci zwiedzić większość kontynentów, zawitać do wielu państw i zobaczyć dziesiątki miast. Umożliwi dotarcie do miejsc, które ze względu na duże odległości są rzadko odwiedzane.

Przedstawiona wyżej krótka informacja na temat kontynentów pochodzi ze strony:

[Wikipedia](#)

Copyright 2018

**Rysunek 2.4.** Układ strony do ćwiczenia 2.2

## 2.2.2. Znaczniki semantyczne podziału strony

Konstruowanie strony za pomocą znacznika `<div>` pozwala zdefiniować dowolny układ strony. Za jego pomocą można podzielić stronę internetową na odpowiednie fragmenty i przydzielić im określone funkcje. Czasami zdarza się, że nadużywanie znaczników powoduje, że strony stają się mało zrozumiałe. W specyfikacji HTML5 została wprowadzona nowa grupa poleceń, których zadaniem jest dokładne precyzowanie fragmentów tworzzonego dokumentu. Są to znaczniki `<header>`, `<nav>`, `<article>`, `<section>`, `<aside>`, `<footer>`, przeznaczone do opisywania zawartej na stronie treści. Wymienione znaczniki są elementami semantycznymi, które same nic nie robią, ale „mówią” przeglądarkom, jaką rolę w układzie dokumentu odgrywa określony fragment. Za ich pomocą można nadać stronie zrozumiałą i logiczną strukturę.

### Znacznik `<header>`

Znacznik `<header>` powinien zawierać część nagłówkową strony lub sekcji. Często jest to złożony element, w którym występują śródtytuły, część nawigacyjna, formularz

wyszukiwarki lub logo. Może być umieszczany w innych elementach, może też zawierać inne elementy. Każda sekcja strony lub każdy artykuł może posiadać swój własny nagłówek. Dlatego warto jest definiować dla każdego nagłówka dodatkowy atrybut w postaci identyfikatora *id*. Ułatwi on później zlokalizowanie wybranego elementu, a także definiowanie stylów w kaskadowym arkuszu stylów.

### **Przykład 2.14**

```
<div>
  <header id="nag_s">
    <h1>Czwarta rewolucja przemysłowa</h1>
    <h3>Co się kryje za czwartą rewolucją?</h3>
    <p>Autor: Paweł Nadrożny</p>
  </header>
  <p>Czwarta rewolucja przemysłowa to zanikanie bariery między ludźmi i maszynami. Powstają inteligentne fabryki z cyberfizycznymi systemami produkcji w otoczeniu: Internetu ludzi, Internetu rzeczy, Internetu usług, Internetu danych.</p>
</p>
</div>
```

### **Znacznik <nav>**

Znacznik *<nav>* powinien zawierać zbiór odnośników, które pozwolą odnosić się do różnych fragmentów na tej samej stronie lub do innych stron. Jest wykorzystywany jako blok nawigacyjny dokumentu. W tworzonym dokumencie może wystąpić wiele znaczników *<nav>*. Omawiany znacznik może być umieszczany w innych elementach, może też zawierać inne elementy. Często element ten jest widoczny w sekcji nagłówka i w stopce.

### **Przykład 2.15**

```
<header id="nag_s">
  <h2>Kontynenty</h2>
  <nav>
    <ul>
      <li>Europa</li>
      <li>Afryka</li>
      <li>Azja</li>
      <li>Australia</li>
    </ul>
  </nav>
</header>
```

W przykładzie zostały wykorzystane znaczniki listy `<ul>` oraz `<li>`, które zostaną omówione dalej.

## Znacznik `<section>`

Sekcje są logicznymi częściami strony. Tworzy się je za pomocą znacznika `<section>`. Sekcja powinna zawierać treści o określonej tematyce i służyć do ich grupowania. W treści powinien wystąpić tytuł, mogą się też pojawić śródtytuły. Jest to uniwersalny znacznik, który powinien być używany, gdy inne znaczniki nie pasują w danym miejscu konstruowanej strony. Może być umieszczany w innych elementach, może też zawierać inne elementy.

Sekcja może składać się z wielu akapitów, ale jej tematyka powinna być spójna. Nie powinna być stosowana jako pojemnik na elementy różnego rodzaju.

### Przykład 2.16

```
<header>
    <h1>Podróż dookoła świata</h1>
</header>
<section>
    <h2>Europa</h2>
    <p>Europa... </p>
    <h2>Afryka</h2>
    <p>Afryka... </p>
</section>
```

## Znacznik `<article>`

Znacznik `<article>` powinien być stosowany do treści, które stanowią spójną całość, niezależną od innych treści, i mogą być samodzielną częścią dokumentu, np. komentarem pod artykułem, wpisem na blogu. W artykule powinna być umieszczana właściwa treść strony. W dokumencie może pojawić się więcej niż jeden znacznik `<article>`. Może on być umieszczony w innym znaczniku `<article>`, może też zawierać inne elementy. Każdy artykuł powinien posiadać nagłówek, treść i stopkę.

### Przykład 2.17

```
<article id="wpis">
    <header>
        <h2>Jak zwiedzać świat?</h2>
        <p>Wpis Janka</p>
    </header>
```

<p>Skończyły się czasy, kiedy podróż dookoła świata była rozrywką tylko dla najbogatszych albo dla wytrawnych podróżników. Dziś mogę pojechać, gdzie chcę i kiedy chcę. Wystarczy tylko trochę zaoszczędzić i kupić bilet dookoła świata.</p>

<p>Bilety dookoła świata są oferowane przez różne linie lotnicze. Takie bilety dają wiele możliwości i pozwalają zmniejszyć koszty lotów. Dzięki nim można polecieć dookoła świata, a cena zależy od liczby przystanków. Trasę układamy według własnych potrzeb i możliwości.</p>

```
<footer>
  <p>Podróże</p>
</footer>
</article>
```

Czym różni się artykuł od sekcji? Artykuł to rzeczywista zawartość strony, a sekcja to logiczna część dokumentu. W sekcji może się znajdować wiele artykułów, ale w artykule też mogą wystąpić kolejne sekcje, jeżeli artykuł zostanie rozbudowany.

## Znacznik `<aside>`

Znacznik `<aside>` oznacza zamknięty fragment, zawierający różne treści, który został odseparowany od głównej zawartości strony, ale może odwoływać się do tejże zawartości. Może być używany do tworzenia panelu bocznego, który zawiera treść uzupełniającą, np. cytat, dygresję lub przypis. Może być umieszczany w innych elementach, ale sam może też zawierać inne elementy.

### Przykład 2.18

```
<article>
  <h2>Grecja</h2>
  <p>Zwiedzając Europę, planujemy odwiedzić Grecję z jej pięknym położeniem i ciekawą historią.</p>
  <aside>
    <h3>Porada</h3>
    <p>Gdy odwiedzasz Grecję, uważaj na upały.</p>
  </aside>
</article>
```

## Znacznik `<footer>`

Znacznik `<footer>` powinien reprezentować stopkę dokumentu lub sekcji. Element ten najczęściej zawiera informacje dotyczące dokumentu, np. informację o prawach autorskich, dane kontaktowe, grupę odnośników lub elementy nawigacyjne. Może być umieszczany w innych elementach, ale może też zawierać inne elementy. Ponieważ

stopka może być wiele, podobnie jak dla innych znaczników, dla tworzonej stopki można zdefiniować atrybut identyfikatora `id`.

### Przykład 2.19

```
<footer id="stopka_s">  
    <p>kontakt: grupa_ama@gmail.com, autor: Czarek Wilk</p>  
</footer>
```

## 2.2.3. Układ strony budowany za pomocą znaczników semantycznych

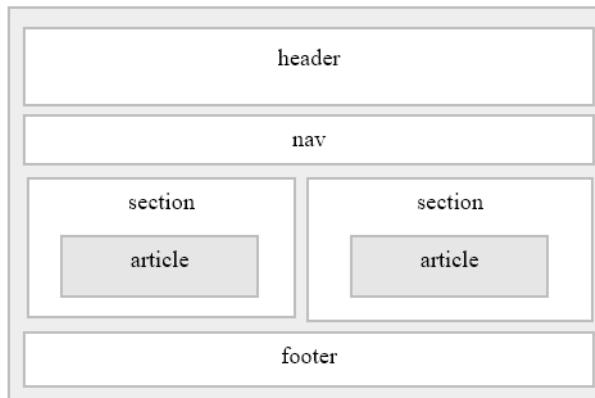
Z pomocą zdefiniowanych w HTML5 nowych znaczników można zbudować strukturę strony internetowej podobną do tej, która została zbudowana z zastosowaniem znaczników `<div>`.

Na rysunku 2.5 pokazany został przykładowy schemat strony zbudowany za pomocą znaczników semantycznych. Podobny schemat strony może zostać utworzony przy użyciu kodu z przykładu 2.20. Formatowanie strony i układ widoczny na rysunku uzyskamy po dołączeniu zewnętrznego arkusza stylów.

### Przykład 2.20

```
<!DOCTYPE html>  
<html lang="pl-PL">  
    <head>  
        <title>Układ</title>  
        <meta charset="UTF-8">  
        <link rel="stylesheet" type="text/css" href="styl1.css">  
    </head>  
    <body>  
        <div>  
            <header>  
                <p>znacznik header</p>  
            </header>  
            <nav>  
                <p>znacznik nav</p>  
            </nav>  
            <section>  
                <h2>znacznik section1</h2>
```

```
<article>
    <h3>nagłówek article1</h3>
    <p>znacznik article</p>
</article>
</section>
<section>
    <h2>znacznik section2</h2>
    <article>
        <h3>nagłówek article2</h3>
        <p>znacznik article</p>
    </article>
</section>
<footer>
    <p>znacznik footer</p>
</footer>
</div>
</body>
</html>
```



**Rysunek 2.5.** Schemat strony zbudowanej na bazie znaczników semantycznych

### Ćwiczenie 2.3

Przeanalizuj kod strony utworzonej w ćwiczeniu 2.1. Zamień bloki utworzone za pomocą znacznika `<div>` na odpowiednie znaczniki semantyczne HTML5.

## Rozwiązanie

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <meta charset="UTF-8">
    <title>Nowe technologie</title>
    <link rel="stylesheet" href="styl1.css">
</head>
<body>
    <div>
        <header>
            <h1>Rewolucja przemysłowa 4.0</h1>
            <h3>Co się kryje za czwartą rewolucją?</h3>
            <p>Autor: Paweł Nadrożny</p>
        </header>
        <nav>NAWIGACJA</nav>
        <section>
            <h2>Rewolucja przemysłowa</h2>
            <article>
                <h3>Pierwsza rewolucja przemysłowa</h3>
                Pierwsza rewolucja przemysłowa to wiek pary.<br> Produkcja mechaniczna jest wspomagana siłą pary i wody.
            </article>
            <article>
                <h3>Druga rewolucja przemysłowa</h3>
                <p>Druga rewolucja przemysłowa to wiek elektryczności.</p>
                <p>Rozpoczyna się produkcja masowa z zastosowaniem energii elektrycznej.</p>
            </article>
            <article>
                <h3>Trzecia rewolucja przemysłowa</h3>
                <p>Trzecia rewolucja przemysłowa to wiek komputerów. Postępuje automatyzacja produkcji przemysłowej z wykorzystaniem sterowników programowalnych.</p>
            </article>
            <article>
```

```
<h3>Czwarta rewolucja przemysłowa</h3>
<p>Czwarta rewolucja przemysłowa to zanikanie bariery między ludźmi i maszynami. Powstają inteligentne fabryki z cyberfizycznymi systemami produkcji w otoczeniu: Internetu ludzi, Internetu rzeczy, Internetu usług, Internetu danych.</p>
</article>
</section>
<section>
<h2>Czwarta rewolucja przemysłowa to:</h2>
<p>
-> nowa jakość komunikacji,
<!--uzupełnić tekst-->
-> technologie wytwarzania przyrostowego, np. 3-D printing – zarówno do prototypowania, jak i realizacji indywidualnych zamówień.
</p>
</section>
<footer>
<p>Przedstawiona wyżej krótka informacja na temat czwartej rewolucji przemysłowej pochodzi ze stron:</p>
<p>Wikipedia</p>
<p>Rewolucja przemysłowa</p>
<p>Copyright 2018</p>
</footer>
</div>
</body>
</html>
```

W podanym rozwiążaniu dodany został nowy blok przeznaczony do nawigacji (znacznik `<nav>`), w pierwszym bloku `<section>` teksty zostały umieszczone w wyodrębnionych blokach `<article>`. W pierwszym bloku `<article>` tekst został złamany za pomocą znacznika `<br>`, w drugim każde zdanie zostało otoczone akapitem, a w trzecim cały tekst został otoczony akapitem.

## Ćwiczenie 2.4

W podobny sposób przeanalizuj kod z ćwiczenia 2.2. Zamień bloki utworzone za pomocą znacznika `<div>` na odpowiednie znaczniki semantyczne HTML5. Poniżej nagłówka `<header>` utwórz blok nawigacji `<nav>`.

## 2.2.4. Listy

Występujące w języku HTML listy służą do tworzenia różnego rodzaju zestawień, uporządkowania ważnych treści, wyliczenia elementów. Można definiować listy nieuporządkowane w postaci punktów, listy uporządkowane w postaci kolejnych numerów oraz listy definicji. Dodatkowo listy te mogą zawierać listy zagnieździone.

### Listy uporządkowane (numerowane)

W listach numerowanych istotna jest kolejność elementów. Do zdefiniowania takiej listy stosowany jest znacznik `<ol>`, a elementy listy są definiowane za pomocą znacznika `<li>`.

Znacznik `<ol>` może zawierać atrybut `type="x"`, określający sposób wyliczania listy (tabela 2.2), oraz atrybut `start="n"`, określający wartość rozpoczynającą wyliczanie. Znaczniki `<li>` mogą zawierać atrybut `type="x"`, określający sposób wyliczania danego elementu listy, oraz atrybut `value="n"`, określający wartość, od której rozpocznie się wyliczanie elementów. Tabela 2.2 zawiera wykaz dostępnych wartości.

Atrybut `type` może być zastąpiony w niektórych przypadkach licznikami stylów CSS (wtedy gdy stanowi element prezentacyjny), ale w innych dokumentach może być częścią semantyki dokumentu (np. gdy w treści dokumentu występują odwołania do elementów listy), dlatego pozostał częścią języka HTML. Powinien być stosowany tylko wówczas, gdy licznik listy pełni funkcję semantyczną.

**Tabela 2.2.** Wartości atrybutu type dla listy uporządkowanej

Type	Opis
<code>type="1"</code>	Pozycje listy zostaną ponumerowane liczbami (domyślny).
<code>type="A"</code>	Pozycje listy zostaną ponumerowane wielkimi literami.
<code>type="a"</code>	Pozycje listy zostaną ponumerowane małymi literami.
<code>type="I"</code>	Pozycje listy zostaną ponumerowane wielkimi cyframi rzymskimi.
<code>type="i"</code>	Pozycje listy zostaną ponumerowane małymi cyframi rzymskimi.

Atrybut `start` jest wykorzystywany, gdy tworzona lista jest podzielona na kilka części i w kolejnej chcemy kontynuować numerowanie z poprzedniej listy.

### Przykład 2.21

```
<ol type="I">
  <li>Europa</li>
  <li>Azja</li>
  <li>Afryka</li>
  <li>Australia</li>
</ol>
```

```
<p>Australia i Oceania są jedną częścią świata, lecz kontynentem jest tylko Australia.</p>  
<ol type="I" start="5">  
    <li>Ameryka Południowa</li>  
    <li>Ameryka Północna</li>  
</ol>
```

Wynik interpretacji kodu został pokazany na rysunku 2.6.

I. Europa  
II. Azja  
III. Afryka  
IV. Australia

Australia i Oceania są jedną częścią świata, lecz kontynentem jest tylko Australia.

V. Ameryka Południowa  
VI. Ameryka Północna

**Rysunek 2.6.** Lista ponumerowana

Ustawienie atrybutów `type` i `value` dla znacznika `<li>` unieważnia atrybuty `type` i `start` ustawione dla znacznika `<ol>`.

### Listy nieuporządkowane (wypunktowane)

Listy nieuporządkowane służą do tworzenia wykazów, w których kolejne elementy będą wyróżnione za pomocą punktorów. Listy nieuporządkowane tworzy się przy użyciu znacznika `<ul>`, a elementy listy, podobnie jak w liście uporządkowanej, są tworzone za pomocą znacznika `<li>`. Formatowanie punktorów jest realizowane z zastosowaniem stylów CSS.

#### Przykład 2.22

```
<ul>  
    <li>Europa</li>  
    <li>Azja</li>  
    <li>Afryka</li>  
    <li>Australia</li>  
    <li>Ameryka Południowa</li>  
    <li>Ameryka Północna</li>  
</ul>
```

Wynik interpretacji kodu został pokazany na rysunku 2.7.

- Europa
- Azja
- Afryka
- Australia
- Ameryka Południowa
- Ameryka Północna

**Rysunek 2.7.** Lista nieuporządkowana

## Listy zagnieżdżone

Zagnieżdżanie list to umieszczanie jednej listy wewnątrz innej. Takie listy tworzy się najczęściej jako listy wielopoziomowe. Przy zagnieżdżaniu można stosować kilka poziomów, jednak każdy kolejny poziom musi zawierać własną definicję listy.

### Przykład 2.23

```
<h3>Kontynenty - państwa - miasta</h3>
<ul>
  <li>Europa
    <ul>
      <li>Francja</li>
      <li>Grecja
        <ol>
          <li>Ateny</li>
          <li>Saloniki</li>
          <li>Patras</li>
        </ol>
      </li>
    </ul>
  </li>
  <li>Azja
    <ul>
      <li>Japonia
        <ol>
          <li>Tokio</li>
          <li>Osaka</li>
        </ol>
      </li>
    </ul>
  </li>
</ul>
```

```
</ol>  
</li>  
<li>Chiny</li>  
</ul>  
</li>  
<li>Afryka  
<ul>  
<li>Kenia</li>  
<li>Czad</li>  
</ul>  
</li>  
</ul>
```

Wynik interpretacji kodu został pokazany na rysunku 2.8.

### Kontynenty - państwa - miasta

- Europa
  - Francja
  - Grecja
    - 1. Ateny
    - 2. Saloniki
    - 3. Patras
- Azja
  - Japonia
    - 1. Tokio
    - 2. Osaka
  - Chiny
- Afryka
  - Kenia
  - Czad

**Rysunek 2.8.** Lista zagnieżdziona

### Listy definicji

Listy definicji służą do tworzenia słowników, wyjaśnień i opisów różnych terminów. Składają się z dwóch elementów: nazwy terminu i jego definicji. Tworzy się je przy użyciu znacznika `<dl>`. Wewnątrz niego umieszcza się nazwę terminu ujętą w znacznik `<dt>` oraz definicję ujętą w znacznik `<dd>`.

## Przykład 2.24

```
<dl>
  <dt>kontynent</dt>
  <dd>olbrzymi obszar lądu otoczony ze wszystkich stron morzami i oceanami,
  a z innymi kontynentami połączony najwyżej wąskimi przesmykami</dd>
  <dt>państwo</dt>
  <dd>forma organizacji społeczeństwa mająca monopol na stanowienie i wykonywanie prawa na określonym terytorium</dd>
  <dt>miasto</dt>
  <dd>historycznie ukształtowana jednostka osadnicza charakteryzująca się dużą intensywnością zabudowy</dd>
</dl>
```

Wynik interpretacji kodu został pokazany na rysunku 2.9.

Definicje	
kontynent	olbrzymi obszar lądu otoczony ze wszystkich stron morzami i oceanami, a z innymi kontynentami połączony najwyżej wąskimi przesmykami
państwo	forma organizacji społeczeństwa mająca monopol na stanowienie i wykonywanie prawa na określonym terytorium
miasto	historycznie ukształtowana jednostka osadnicza charakteryzująca się dużą intensywnością zabudowy

**Rysunek 2.9.** Lista definicji

## Ćwiczenie 2.5

Zaprojektowaną w ćwiczeniu 2.3 stronę internetową zmodyfikuj, tworząc dla drugiego bloku `<section>` listę taką jak ta pokazana na rysunku 2.10. Użyty w tekście znak `->` zastąp punktorem listy.

### Czwarta rewolucja przemysłowa to:

- nowa jakość komunikacji, w której zarówno świat cyfrowy, jak i rzeczywisty są połączone ze sobą, dzięki czemu maszyny, produkty w różnych fazach przetwarzania, systemy oraz ludzie – mając indywidualny adres IP – wymieniają cyfrową informację poprzez protokół internetowy;
- inteligentne sensory z wbudowanymi systemami indywidualnej identyfikacji, przetwarzania danych i komunikacji;
- przetwarzanie danych w chmurze lub we mgle, z dynamiką reakcji na poziomie milisekund;
- analityka dużych zbiorów danych dotyczących wszystkich aspektów rozwoju produktów i produkcji;
- techniki symulacji funkcjonowania obiektów rzeczywistych w ich wirtualnych odwzorowaniach w oparciu o dane dostarczane i przetwarzane w czasie rzeczywistym, pozwalające na testowanie i optymalizowanie konfiguracji procesów produkcyjnych przed wprowadzeniem fizycznych zmian;
- bezpośrednią komunikację między urządzeniami;
- zaawansowane interfejsy człowiek – maszyna;
- rozwiązań cybersecurity, zapewniające bezpieczną, pewną komunikację i identyfikację oraz dostęp zarządczy do systemów i urządzeń;
- nowa generacja robotów, charakteryzującą się aktywną interakcją z otoczeniem i z innymi robotami oraz adaptacją do zmieniających się warunków i wymagań;
- systemy rzeczywistości rozszerzonej, wspomagające projektowanie i serwisowanie urządzeń;
- technologie wytwórczości przyrostowego, np. 3-D printing – zarówno do prototypowania, jak i realizacji indywidualnych zamówień.

Przedstawiona wyżej短短ka informacja na temat czwartej rewolucji przemysłowej pochodzi ze stron:

[Wikipedia](#)

[Rewolucja przemysłowa](#)

[Copyright 2018](#)

**Rysunek 2.10.** Tekst sformatowany z użyciem listy nieuporządkowanej

### Ćwiczenie 2.6

Zmień wygląd strony internetowej utworzonej w ćwiczeniu 2.4 zgodnie z rysunkiem 2.11. W pierwszym bloku <section> przy opisywaniu państw użyj listy definicji.

#### Zwiedzamy Afrykę

Afryka to drugi pod względem wielkości kontynent na Ziemi. Zajmuje 30,37 mln km<sup>2</sup>, czyli ponad 20,3% ogólnej powierzchni lądowej świata. Przechodzą przez nią południk 0°, obydwa zwrotniki i równik.

#### Jedziemy do Azji

Azja razem z Europą tworzy Eurazję, największy kontynent na Ziemi. Z powodów historycznych i kulturowych sama Azja bywa również nazywana kontynentem. Nazywana jest często kontynentem wielkich kontrastów geograficznych.

#### Płyniemy do Australii

Nazwa "Australija" pochodzi od łacińskiego określenia "Terra Australis", czyli "Ziemia Południowa". W dawnej Europie wierzono, że na półkuli południowej znajduje się naprawdę duży kontynent. Ostatecznie nazwa ta przypadła kontynentowi o wiele mniejszymu niż zakładano.

#### Teraz odwiedzamy Amerykę Południową

Ameryka Południowa to kontynent o powierzchni 17,8 mln km<sup>2</sup>, leżący na półkuli zachodniej oraz w większej części na półkuli południowej, a w mniejszej – na półkuli północnej. Niektóre uważa się również za subkontynent Ameryki.

#### Podążamy do Ameryki Północnej

Ameryka Północna to kontynent o powierzchni 24 242 000 km<sup>2</sup> (co stanowi 16,3% całkowitej powierzchni lądów na kuli ziemskiej), położony na półkulach: północnej i zachodniej. Do Ameryki Północnej należy Ameryka Środkowa.

#### Wracamy do Europy

Europa to kontynent o powierzchni 10,2 mln km<sup>2</sup>, co stanowi ok. 2% całej powierzchni kuli ziemskiej. Europa zajmuje 1/5 Eurazji i jako jej największy półwysep jest najbardziej wysunięta na zachód.

**Rysunek 2.11.** Tekst sformatowany z użyciem listy definicji

## Rozwiązańe

Rozwiązaniem ćwiczenia 2.6 mógłby być następujący kod:

```
<div>
    <header>
        <h1>Podróż dookoła świata</h1>
        <h3>Wybieramy się w podróż dookoła świata</h3>
        <p>Przewodnikiem będzie Adam Bilik</p>
    </header>
    <nav>NAWIGACJA</nav>
    <section>
        <h3>Podróż dookoła świata</h3>
        <dl>
            <dt>Zwiedzamy Afrykę</dt>
            <dd>Afryka to drugi pod względem wielkości kontynent na Ziemi. Zajmuje 30,37 mln km2, czyli ponad 20,3% ogólnej powierzchni lądowej świata. Przechodzą przez niego południk 0°, obydwa zwrotniki i równik.</dd>
            <dt>Jedziemy do Azji</dt>
            <dd>Azja razem z Europą tworzy Eurazję, największy kontynent na Ziemi. Z powodów historycznych i kulturowych sama Azja bywa również nazywana kontynentem. Nazywana jest często kontynentem wielkich kontrastów geograficznych.</dd>
            <dt>Płyniemy do Australii</dt>
            <dd>Nazwa "Australia" pochodzi od łacińskiego określenia "Terra Australis", czyli "Ziemia Południowa". W dawnej Europie wierzono, że na półkuli południowej znajduje się naprawdę duży kontynent. Ostatecznie nazwa ta przypadła kontynentowi o wiele mniejszemu niż zakładano.</dd>
            <dt>Teraz odwiedzamy Amerykę Południową</dt>
            <dd>Ameryka Południowa to kontynent o powierzchni 17,8 mln km2, leżący na półkuli zachodniej oraz w większej części na półkuli południowej, a w mniejszej – na półkuli północnej. Niekiedy uważana jest również za subkontynent Ameryki.</dd>
            <dt>Podążamy do Ameryki Północnej</dt>
            <dd>Ameryka Północna to kontynent o powierzchni 24 242 000 km2 (co stanowi 16,3% całkowitej powierzchni lądów na kuli ziemskiej), położony na półkuli północnej i zachodniej. Do Ameryki Północnej należy Ameryka Środkowa.</dd>
            <dt>Wracamy do Europy</dt>
        </dl>
    </section>
</div>
```

```

<dd>Europa to kontynent o powierzchni 10,2 mln km2, co stanowi ok. 2% całej powierzchni kuli ziemskiej. Europa zajmuje 1/5 Eurazji i jako jej największy półwysep jest najbardziej wysunięta na zachód.</dd>
</dl>
</section>
<section>
<h3>Informacje dodatkowe</h3>
<p>Podróż dookoła świata pozwoli Ci zwiedzić większość kontynentów, zawitać do wielu państw i zobaczyć dziesiątki miast. Umożliwi dotarcie do miejsc, które ze względu na duże odległości są rzadko odwiedzane.</p>
</section>
<footer>
<p>Przedstawiona wyżej krótka informacja na temat kontynentów pochodzi ze strony:</p>
<p>Wikipedia</p>
<p>Copyright 2018</p>
</footer>
</div>

```

## 2.2.5. Tabele

Tabele służą do tworzenia zestawień w postaci tabelarycznej. Do tego celu używa się znaczników `<table>`, `<tr>`, `<th>`, `<td>` i `<caption>`. Natomiast formatowanie tabel powinno się odbywać za pomocą stylów CSS, w związku z czym w HTML5 większość atrybutów związanych z formatowaniem tabel jest traktowana jako przestarzała.

Tabele składają się z wierszy. Każdy wiersz ma określoną liczbę komórek, dlatego tworząc tabelę w języku HTML, należy nie tylko zdefiniować ją za pomocą znacznika `<table>`, ale także zdefiniować każdy jej wiersz przy użyciu znacznika `<tr>` oraz każdą komórkę w wierszu za pomocą znacznika `<td>`. Nagłówek tabeli jest definiowany z zastosowaniem znacznika `<th>`. Domyślnie nagłówki tabeli są pogrubione i wyśrodkowane.

### Przykład 2.25

```

<table>
<tr><th>Nazwisko</th><th>Imię</th><th>Wiek</th></tr>
<tr><td>Korsak</td><td>Tadeusz</td><td>36</td></tr>
<tr><td>Polak</td><td>Anna</td><td>27</td></tr>
<tr><td>Makowski</td><td>Paweł</td><td>23</td></tr>
<tr><td>Zaleska</td><td>Monika</td><td>24</td></tr>
</table>

```

W wyniku zastosowanych poleceń powstała tabela składająca się z wiersza nagłówka oraz z trzech kolumn i czterech wierszy.

Komórki tabeli (znacznik `<td>`) są traktowane jak kontenery danych, w których można umieścić dowolne elementy, np. dane tekstowe, obrazy, listy, inne tabele.

Domyślnie tabela jest wyświetlaną bez obramowań. Rodzaj i styl obramowania są ustawiane za pomocą właściwości obramowań w arkuszach stylów CSS. Obramowanie należy zdefiniować zarówno dla tabeli, jak i dla jej komórek.

## Krawędzie tabeli

Mimo że arkusze stylów zostaną opisane w rozdziale 3., niżej znajdują się przykłady ze zdefiniowanymi właściwościami tabel.

### Przykład 2.26

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Tabela</title>
    <meta charset="UTF-8">
    <style>
        table, th, td {
            border: 1px solid black;
        }
    </style>
</head>
<body>
    <table>
        <tr><th>Nazwisko</th><th>Imię</th><th>Wiek</th></tr>
        <tr><td>Korsak</td><td>Tadeusz</td><td>36</td></tr>
        <tr><td>Polak</td><td>Anna</td><td>27</td></tr>
        <tr><td>Makowski</td><td>Paweł</td><td>23</td></tr>
        <tr><td>Zaleska</td><td>Monika</td><td>24</td></tr>
    </table>
</body>
</html>
```

Zapis w definicji stylów atrybutu `border: 1px solid black` dla tabeli, nagłówka i komórki stworzy obramowania wokół tabeli i poszczególnych komórek ciągłą czarną linią o szerokości 1 px.

### Przykład 2.27

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Tabela</title>
    <meta charset="UTF-8">
    <style>
        table, th, td {
            border: 1px solid black;
            border-collapse: collapse;
        }
    </style>
</head>
<body>
    <table>
        <tr><th>Nazwisko</th><th>Imię</th><th>Wiek</th></tr>
        <tr><td>Korsak</td><td>Tadeusz</td><td>36</td></tr>
        <tr><td>Polak</td><td>Anna</td><td>27</td></tr>
        <tr><td>Makowski</td><td>Paweł</td><td>23</td></tr>
        <tr><td>Zaleska</td><td>Monika</td><td>24</td></tr>
    </table>
</body>
</html>
```

Zapis w definicji stylów atrybutów `border: 1px solid black; border-collapse: collapse;` spowoduje, że obramowanie wokół komórek tabeli będzie pojedyncze.

### Przykład 2.28

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Tabela</title>
    <meta charset="UTF-8">
    <style>
        table, th, td {
            border: 1px solid black;
```

```
border-collapse: collapse;  
}  
  
th, td {  
    padding: 10px;  
}  
  
</style>  
  
</head>  
  
<body>  
  
<table>  
    <tr><th>Nazwisko</th><th>Imię</th><th>Wiek</th></tr>  
    <tr><td>Korsak</td><td>Tadeusz</td><td>36</td></tr>  
    <tr><td>Polak</td><td>Anna</td><td>27</td></tr>  
    <tr><td>Makowski</td><td>Paweł</td><td>23</td></tr>  
    <tr><td>Zaleska</td><td>Monika</td><td>24</td></tr>  
  
</table>  
  
</body>  
  
</html>
```

Dodanie w definicji stylów atrybutu `padding: 10px;` dla nagłówka i komórki spowoduje ustawienie odstępu 10 px między zawartością komórki a jej obramowaniem.

## Łączenie komórek

W języku HTML oprócz tabel, które mają taką samą liczbę komórek w każdym wierszu, można tworzyć tabele z różną liczbą komórek w wierszu. Efekt ten uzyskujemy poprzez łączenie komórek z sąsiednich kolumn. Aby połączyć kilka komórek w wierszu, należy w znaczniku `<td>` lub `<th>`, opisującym komórkę tabeli, zdefiniować atrybut `colspan="n"`, gdzie `n` określa, ile sąsiednich komórek zostanie połączonych.

### Przykład 2.29

```
<!DOCTYPE html>  
  
<html lang="pl-PL">  
  
<head>  
  
    <title>Tabela</title>  
    <meta charset="UTF-8">  
  
<style>  
  
table, th, td {  
  
    border: 1px solid black;
```

```

border-collapse: collapse;
}

th, td {
    padding: 5px;
    text-align: left;
}

</style>

</head>

<body>

<table>

<tr><th colspan="2">Nazwisko i imię</th><th>Wiek</th></tr>
<tr><td>Korsak</td><td>Tadeusz</td><td>36</td></tr>
<tr><td>Polak</td><td>Anna</td><td>27</td></tr>
<tr><td>Makowski</td><td>Paweł</td><td>23</td></tr>
<tr><td>Zaleska</td><td>Monika</td><td>24</td></tr>

</table>

</body>
</html>

```

Użycie w definicji tabeli atrybutu `<tr><th colspan="2">Nazwisko i imię</th><th>Wiek</th></tr>` spowoduje połączenie komórek z dwóch kolumn w wskazanym wierszu.

Łączenie komórek tabeli może się odbywać również w kolumnach. Łączone są komórki tej samej kolumny znajdujące się w sąsiednich wierszach. Do takiego łączenia komórek służy definiowany w znaczniku `<td>` atrybut `rowspan="n"`.

### Przykład 2.30

```

<!DOCTYPE html>

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

    <title>Tabela</title>
    <meta charset="UTF-8">
    <style>

        table, th, td {

            border: 1px solid black;
            border-collapse: collapse;
        }
    </style>

```

```

    th, td {
        padding: 5px;
        text-align: left;
    }

```

</style>

</head>

<body>

<table>

Nazwisko	Imię	Miejscowość
Korsak	Tadeusz	Kraków
Polak	Anna	
Makowski	Paweł	Gdańsk
Zaleska	Monika	Poznań

</table>

</body>

</html>

Użycie w definicji tabeli atrybutu `<td rowspan="2">Kraków</td>` spowoduje połączenie komórek z dwóch wierszy we wskazanej kolumnie.

## Znacznik <caption>

Znacznik `<caption>` pozwala dodać do tabeli tytuł. Znacznik ten musi być umieszczony bezpośrednio po znaczniku `<table>`. Przeglądarki internetowe tytuł podany w znaczniku `<caption>` umieszczają jako tekst wycentrowany nad tabelą.

### Przykład 2.31

```

<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Tabela</title>
    <meta charset="UTF-8">
    <style>
        table, th, td {
            border: 1px solid black;
        }
    </style>

```

```

</head>

<body>

    <table>

        <caption>Dane osobowe</caption>

        <tr><th>Nazwisko</th><th>Imię</th><th>Wiek</th></tr>
        <tr><td>Korsak</td><td>Tadeusz</td><td>36</td></tr>
        <tr><td>Polak</td><td>Anna</td><td>27</td></tr>
        <tr><td>Makowski</td><td>Paweł</td><td>23</td></tr>
        <tr><td>Zaleska</td><td>Monika</td><td>24</td></tr>

    </table>

</body>

</html>

```

## Ćwiczenie 2.7

Utworzoną w ćwiczeniu 2.5 stronę internetową zmodyfikuj, dodając kolejny blok `<section>`, umieszczony między pierwszym a drugim blokiem. W utworzonym bloku `<section>` umieść tabelę z danymi jak na rysunku 2.12.

### Czwarta rewolucja przemysłowa

Czwarta rewolucja przemysłowa to zanikanie bariery między ludźmi i maszynami. Powstają inteligentne fabryki z cyberfizycznymi systemami produkcji w otoczeniu: Internetu ludzi, Internetu rzeczy, Internetu usług, Internetu danych.

Numer rewolucji	Początek	Określenie	Wynalazek	Rok	Charakterystyka
Pierwsza	koniec XVIII wieku	Wiek pary	mechaniczne krosno tkackie	1784	Produkcja mechaniczna wspomagana siłą pary i wody
Druga	początek XX wieku	Wiek elektryczności	linia produkcyjna	1870	Produkcja masowa z zastosowaniem energii elektrycznej
Trzecia	lata 70. XX wieku	Wiek komputerów	programowalny układ logiczny	1969	Automatyzacja produkcji przemysłowej
Czwarta	początek XXI wieku	Wiek zanikania bariery ludzie – maszyny	internet	1991	Inteligentne fabryki z cyberfizycznymi systemami produkcji

### Czwarta rewolucja przemysłowa to:

- nowa jakość komunikacji, w której zarówno świat cyfrowy, jak i rzeczywisty są połączone ze sobą, dzięki czemu maszyny, produkty w różnych fazach przetwarzania, systemy oraz ludzie – mając indywidualny adres IP – wymieniają cyfrową informację poprzez protokół internetowy;

**Rysunek 2.12.** Fragment układu strony po dodaniu tabeli

### Rozwiążanie

Rozwiązaniem ćwiczenia 2.7 może być następujący kod:

```

<section>

    <h2>Rewolucje przemysłowe – zestawienie</h2>

    <table>

```

```

<tr><th>Numer rewolucji</th><th>Początek</th><th>Określenie
</th><th>Wynalazek</th><th>Rok</th><th>Charakterystyka</th></tr>
<tr><td>Pierwsza</td><td>koniec XVIII wieku</td><td>Wiek pary
</td><td>mechaniczne krosno tkackie</td><td>1784</td><td>Produkcja
mechaniczna wspomagana siłą pary i wody</td></tr>
<tr><td>Druga</td><td>początek XX wieku</td><td>Wiek elektryczności
</td><td>linia produkcyjna</td><td>1870</td><td>Produkcja masowa z
zastosowaniem energii elektrycznej</td></tr>
<tr><td>Trzecia</td><td>lata 70. XX wieku</td><td>Wiek komputerów
</td><td>programowalny układ logiczny</td><td>1969</td><td>Automatyzacja
produkcyjnej przemysłowej</td></tr>
<tr><td>Czwarta</td><td>początek XXI wieku</td><td>Wiek zanikania bariery
ludzie - maszyny</td><td>internet</td><td>1991</td><td>Inteligentne fabryki
z cyberfizycznymi systemami produkcji</td></tr>
</table>
</section>

```



## 2.3. Obrazy i odsyłacze

Obecnie każda strona internetowa zawiera elementy graficzne. Język HTML dopuszcza umieszczanie na stronie elementów graficznych zapisanych w różnych formatach. Najczęściej są one zapisywane w formacie GIF, JPEG lub PNG.

### 2.3.1. Formaty obrazów

#### Format GIF

GIF to format 8-bitowy z kompresją bezstratną, pozwalający na zapisanie maksymalnie 256 kolorów. Najlepiej nadaje się do zapisywania elementów graficznych zawierających duże obszary o tym samym kolorze i niewielkiej liczbie szczegółów. Dlatego jest wykorzystywany przy projektowaniu nagłówków, przycisków, pasków i prostej grafiki. Główną zaletą tego formatu jest niewielki rozmiar plików. GIF jest również bardzo przydatny w projektowaniu stron internetowych, ponieważ umożliwia tworzenie grafik transparentnych (przezroczystych). Potrzebne jest to przy tworzeniu grafiki, której tło należy ukryć (np. przycisku o owalnym kształcie). W tym przypadku wystarczy uczynić transparentnym tło grafiki. Kolejna właściwość formatu GIF to możliwość stosowania przeplotu. Oznacza ona, że obraz jest wczytywany fragmentami. Początkowo wczytywana jest co czwarta linia obrazu, co powoduje, że od razu widoczny jest jego zarys. Kolejne przejścia to odczyt kolejnych linii — obraz stopniowo nabiera właściwego wyglądu. Daje to wrażenie szybszego wczytywania strony.

W formacie GIF można również zastosować mechanizm redukcji palety kolorów. Polega to na zmniejszeniu liczby kolorów, które zapisujemy w pliku. Decyzja o liczbie kolorów w palecie należy do użytkownika i jest zależna od liczby kolorów wykorzystywanych w tworzonym obrazie.

## Format JPG

W formacie JPG najpierw tworzony jest obraz czarno-biały, a informacje o kolorach są zapisywane osobno. Na powstały obrazie jest przeprowadzana analiza dotycząca barw, a kolory, które są niewidoczne dla oka, zostają usunięte. W formacie JPG można regulować stopień kompresji, co powoduje zmniejszenie rozmiaru pliku, ale może prowadzić do pogorszenia jakości obrazu. Ponieważ w tym formacie zapisywana jest pełna, 24-bitowa paleta kolorów, JPG nadaje się doskonale do zapisywania fotografii.

## Format PNG

Format PNG pozwala na zapisanie 48-bitowej palety kolorów i 16-bitowej palety odciem szarości. Umożliwia tworzenie grafik transparentnych i ma efektywny i bezstratny mechanizm kompresji.

### 2.3.2. Wstawianie obrazu do dokumentu

Do umieszczania obrazów na stronie internetowej służy znacznik `<img>`. Znacznik ten jest pusty, zawiera tylko atrybuty. Nie ma również znacznika zamykającego.

#### Atrybuty obrazu

Atrybut `src="plik"` określa nazwę i położenie pliku z grafiką.

```

```

Jeżeli obrazek znajduje się w tej samej lokalizacji co strona, wystarczy podać nazwę pliku; jeżeli obrazek znajduje się w innej lokalizacji, należy podać pełny adres URL.

```

```

Jeśli obrazek jest przechowywany w specjalnym podfolderze z obrazami, znajdującym się w katalogu głównym strony, to adres pliku może zawierać tylko ścieżkę względową w podanej postaci:

```

```

Jeśli obrazek jest przechowywany w podfolderze z obrazami w bieżącym folderze, to adres pliku ma postać:

```

```

Jeśli obrazek znajduje się w podfolderze z obrazami, znajdującym się w folderze o jeden poziom powyżej bieżącego folderu, to adres pliku ma postać:

```

```

Najlepszą metodą jest używanie względnych ścieżek do plików, ponieważ w takim przypadku pliki nie będą powiązane z bieżącym adresem URL i odnośniki do różnych lokalizacji będą działać poprawnie niezależnie od serwera, na którym umieszczona zostanie strona.

Jeśli obrazek z jakiegoś powodu nie jest wyświetlany na stronie (brak obrazka w lokalizacji, przeglądarka nie obsługuje grafiki, wyświetlanie obrazów zostało wyłączone), zamiast grafiki mogą być wyświetlone teksty zastępcze. Do projektowania tekstów zastępczych służy atrybut `alt="tekst zastępczy"`. Atrybut `alt` powinien być zawsze umieszczany w znaczniku `<img>`. Bez niego walidacja strony internetowej wykryje błędy. Poza tym tekst zastępczy jest czytany przez specjalne programy dla osób z dysfunkcjami jako normalny tekst na stronie. Również wyszukiwarki i ich roboty posługują się opisami zawartymi w tym atrybucie do analizowania zawartości strony.

```

```

### Przykład 2.32

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Obrazy</title>
    <meta charset="UTF-8">
</head>
<body>
    <h2>Obrazy mogą poprawić wygląd każdej strony internetowej</h2>
    
</body>
</html>
```

Wynik działania kodu, gdy obrazek nie został znaleziony, pokazuje rysunek 2.13.

**Obrazy mogą poprawić wygląd każdej strony internetowej**

zdjęcie Antwerpia

**Rysunek 2.13.** Brak obrazka — pojawił się tekst zastępczy

Wynik działania kodu, gdy obrazek istnieje, został pokazany na rysunku 2.14.

**Obrazy mogą poprawić wygląd każdej strony internetowej**



**Rysunek 2.14.** Obraz wstawiony na stronę

Stosując dla znacznika `<img>` atrybuty `height="x"` i `width="y"`, można określić wysokość i szerokość obrazka. Atrybuty te określają szerokość i wysokość obrazka w pikselach. Do określenia wysokości i szerokości obrazka powinno się stosować arkusze stylów, ale alternatywnie mogą być używane atrybuty `height` i `width` znacznika `<img>`. Oba rozwiązania są poprawne w HTML5. Określenie wysokości i szerokości wstawianego obrazka jest wymagane. Brak tych atrybutów może spowodować miganie strony podczas ładowania obrazu.

Oto wysokość i szerokość obrazka zdefiniowane za pomocą atrybutów znacznika `<img>`:

```

```

A to wysokość i szerokość obrazka zdefiniowane za pomocą stylów:

```

```

### Przykład 2.33

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
  <title>Obrazy</title>
  <meta charset="UTF-8">
</head>
<body>
  <h2>Obrazy mogą poprawić wygląd każdej strony internetowej</h2>
  
</body>
</html>
```

Wynik działania kodu został pokazany na rysunku 2.15.



**Rysunek 2.15.** Ustawiona wielkość wyświetlanego obrazka

W przypadku obrazów o dużych rozmiarach istnieje możliwość utworzenia ich miniaturow. Pozwoli to na szybkie wyświetlenie zawartości, szczególnie gdy nie korzystamy z szybkich łączów internetowych.

```
<a href="obrazek.gif"></a>
```

Znacznik `<a>` tworzy łącze do adresu URL podanego jako wartość atrybutu `href`. Wewnątrz znacznika `<a>` umieszczony został znacznik `<img>`, który wyświetli miniaturę obrazka na stronie. Kliknięcie widocznej na stronie miniatury obrazka spowoduje przejście do strony o podanym adresie URL.

## Znaczniki grupujące

Język HTML5 wprowadza nowe znaczniki semantyczne usprawniające obsługę grafiki.

### Znacznik `<figure>`

Znacznik `<figure>` może służyć do oznaczenia rysunku, zdjęcia lub diagramu. Znacznikiem tym otaczamy dane multimedialne, które są istotne dla treści zawartych np. w sekcji lub artykule, ale które stanowią samodzielna całość. Element umieszczony w znaczniku `<figure>` powinien zawierać znacznik `<figcaption>`, mieszczący podpis pod rysunkiem, oraz znacznik `<img>`, zawierający element grafiki, aby zaznaczyć związek między rysunkiem i podpisem. Tak oznaczony element powinien nieść ze sobą jakąś wartość semantyczną (może zostać zastosowany w innym kontekście, np. może zostać użyty do spisu ilustracji na kolejnej podstronie).

### Znacznik `<figcaption>`

Znacznikiem `<figcaption>` powinien być otoczony podpis pod rysunkiem.

## Przykład 2.34

```
<figure>
```

```
  
```

```
<figcaption>Zdjęcie 2. Miasto Antwerpia</figcaption>
</figure>
```

Wynik działania kodu został pokazany na rysunku 2.16.

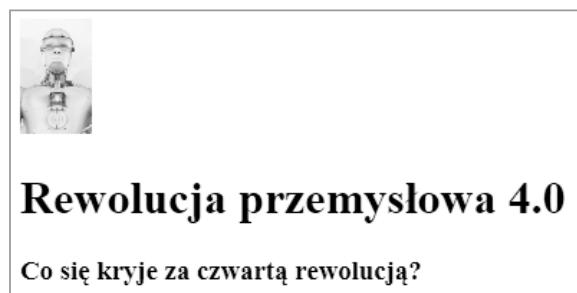
**Rysunek 2.16.**

Obrazek z podpisem,  
otoczony znacznikiem  
<figure>



**Ćwiczenie 2.8**

Przygotuj obrazek, który mógłby być logo strony na temat technologii 4.0. Umieść przygotowane logo w sekcji header tworzonej strony internetowej o rewolucji przemysłowej 4.0. Określ wymiary obrazka oraz umieść tekst zastępczy. Strona internetowa powinna wyglądać podobnie jak ta pokazana na rysunku 2.17.



**Rysunek 2.17.** Logo umieszczone w nagłówku strony

*Rozwiążanie*

Fragment kodu:

```
<header>
  
  <h1>Rewolucja przemysłowa 4.0</h1>
  <h3>Co się kryje za czwartą rewolucją?</h3>
  <p>Autor: Paweł Nadrożny</p>
  <nav>NAWIGACJA</nav>
</header>
```

Takie umieszczenie logo nie jest najlepszym rozwiązaniem. Obraz powinien zostać otoczony tekstem znajdującym się w nagłówku. Efekt taki uzyskamy, używając właściwości float w stylach zdefiniowanych za pomocą kaskadowych arkuszy stylów. Sposób działania kaskadowych arkuszy stylów omówiono w rozdziale 3.

### Przykład 2.35

```
<header>
  
  <h1>Rewolucja przemysłowa 4.0</h1>
  <h3>Co się kryje za czwartą rewolucją?</h3>
  <p>Autor: Paweł Nadrożny</p>
  <nav>NAWIGACJA</nav>
</header>
```

Wynik działania kodu został pokazany na rysunku 2.18.



**Rysunek 2.18.** Obraz na stronie otoczony tekstem

### Ćwiczenie 2.9

Przygotuj kilka obrazków przedstawiających ciekawe miejsca z różnych kontynentów. Umieść przygotowane obrazki na stronie utworzonej w ćwiczeniu 2.6. Określ wymiary obrazków oraz umieść dla każdego tekstu zastępczy. Strona internetowa powinna wyglądać podobnie jak ta pokazana na rysunku 2.19.

#### Rozwiązanie

Fragment kodu:

```
<header>
  
  <h1>Podróż dookoła świata</h1>
  <h3>Wybieramy się w podróż dookoła świata</h3>
  <p>Przewodnikiem będzie Adam Bilik</p>
  <nav>NAWIGACJA</nav>
</header>
```

 **Podróż dookoła świata**

Wybieramy się w podróż dookoła świata

Przewodnikiem będzie Adam Bilik

**NAWIGACJA**

**Zwiedzamy kontynenty**

**Zwiedzamy Afrykę**



Afryka to drugi pod względem wielkości kontynent na Ziemi. Zajmuje 30,37 mln km<sup>2</sup>, czyli ponad 20,3% ogólnej powierzchni lądowej świata. Przechodzą przez niego południk 0°, obydwa zwrotniki i równik.

**Jedziemy do Azji**



Azja razem z Europą tworzy Eurazję, największy kontynent na Ziemi. Z powodów historycznych i kulturowych sama Azja bywa również nazywana kontynentem. Nazywana jest często kontynentem wielkich kontrastów geograficznych.

**Rysunek 2.19.** Fragment strony internetowej

```
<section>
  <h3>Zwiedzamy kontynenty</h3>
  <article>
    <h4>Zwiedzamy Afrykę</h4>
    
    <p>Afryka to drugi pod względem wielkości kontynent na Ziemi. Zajmuje 30,37 mln km2, czyli ponad 20,3% ogólnej powierzchni lądowej świata. Przechodzą przez niego południk 0°, obydwa zwrotniki i równik.</p>
  </article>
  <article>
    <h4>Jedziemy do Azji</h4>
    
    <p>Azja razem z Europą tworzy Eurazję, największy kontynent na Ziemi. Z powodów historycznych i kulturowych sama Azja bywa również nazywana kontynentem. Nazywana jest często kontynentem wielkich kontrastów geograficznych.</p>
  </article>
</section>
```

### 2.3.3. Odsyłacze (hiperłącza)

Częstym elementem stron internetowych są odsyłacze (hiperłącza) prowadzące do innych stron, innych plików lub usług. Mogą one przenosić użytkownika do dowolnej innej strony WWW, ale mogą również prowadzić do elementu na tej samej stronie lub do dowolnego dokumentu bieżącej witryny.

Definiowany odsyłacz ma postać:

```
<a href="adres URL">Tekst połączenia</a>
```

Atrybut `href="adres URL"` określa, do jakiego dokumentu lub do jakiej strony prowadzi łącze. Tekst odsyłacza będzie widoczny na stronie ze zdefiniowanym połączeniem. Kliknięcie odsyłacza przeniesie użytkownika do wybranej strony lub wybranego dokumentu.

Przykłady odsyłaczy:

- do innych dokumentów — `<a href="plik.html">Dokument</a>`,
- do pliku z grafiką — `<a href="grafika.jpg">Rysunek</a>`,
- do innej strony internetowej — `<a href="http://men.gov.pl">MEN</a>`,
- do zasobu dostępnego przez FTP — `<a href="ftp://ftp.tpnet.pl">TPNET</a>`.

Odsyłacz umieszczony na stronie ma domyślnie zdefiniowane kolory:

- nieodwiedzone łącze — kolor niebieski, podkreślenie,
- odwiedzone łącze — kolor fioletowy, podkreślenie,
- aktywne łącze — kolor czerwony, podkreślenie.

Domyślne kolory mogą być zmieniane w deklaracji arkuszy stylów.

#### Ćwiczenie 2.10

Dla strony utworzonej w ćwiczeniu 2.8 utwórz w stopce dwa odsyłacze do stron z informacją o czwartej rewolucji przemysłowej (mogą to być dowolne strony).

#### Rozwiążanie

Fragment kodu umieszczonego w stopce:

```
<footer id="stopka">
    <p>Przedstawiona wyżej krótka informacja na temat czwartej rewolucji przemysłowej pochodzi ze stron:</p>
    <p><a href="https://pl.wikipedia.org/wiki/Czwarta_revolucja_przemysłowa">Wikipedia</a></p>
    <p><a href="https://www.controlengineering.pl/czwarta-revolucja-przemyslowa-i-przemysl-40-co-oznacza-te-pojecia/">Rewolucja przemysłowa</a></p>
```

```
<p>Copyright 2018</p>
</footer>
```

## Odsyłacz graficzny

Na stronie internetowej oprócz odsyłaczy tekstowych można umieszczać odsyłacze graficzne. W tym przypadku kliknięcie elementu graficznego przeniesie użytkownika do wybranej strony internetowej.

```
<a href="adres.htm"></a>
```

### Ćwiczenie 2.11

Przygotuj mały obrazek w postaci ikony. Dla strony utworzonej w ćwiczeniu 2.9 utwórz w stopce odsyłacz graficzny zawierający przygotowany obrazek. Odsyłacz powinien przenosić do strony z informacją o kontynentach (może to być dowolna strona).

#### Rozwiążanie

Fragment kodu umieszczonego w stopce:

```
<footer id="stopka">
    <p>Przedstawiona wyżej krótka informacja na temat kontynentów pochodzi ze strony: <a href="https://pl.wikipedia.org/wiki/Kontynent">
        
    </a></p>
    <p>Copyright 2018</p>
</footer>
```

## Zakładki

Odsyłacze pozwalają również na przenoszenie się do określonego miejsca w obrębie tej samej strony. Zastosowanie tego mechanizmu ułatwia nawigowanie zwłaszcza w długim dokumencie, niemieszczącym się w całości na ekranie. Najpierw należy utworzyć zakładkę, a następnie trzeba zdefiniować odsyłacz do niej. Po kliknięciu odsyłacza zostaniemy przeniesieni na stronę do miejsca, gdzie zlokalizowana została zakładka.

Odnośnik jest definiowany z użyciem znacznika `<a>`:

```
<a name="odnośnik">Tekst odnośnika</a>
```

gdzie *odnośnik* określa wybieraną część strony.

Teraz należy zdefiniować odsyłacz do istniejącego odnośnika:

```
<a href="#odnośnik">Tekst odnośnika</a>
```

## Przykład 2.36

```
<section id="sekcja1">
    <p><a href="#zakladka">Przejjdź do Azji</a></p>
    <h2>Europa</h2>
    <p>Europa to kontynent o powierzchni 10,2 mln km2, co stanowi ok. 2% całej powierzchni kuli ziemskiej. Europa zajmuje 1/5 Eurazji i jako jej największy półwysep jest najbardziej wysunięta na zachód.</p>
    <h2>Afryka</h2>
    <p>Afryka to drugi pod względem wielkości kontynent na Ziemi. Zajmuje 30,37 mln km2, czyli ponad 20,3% ogólnej powierzchni lądowej świata. Przechodzą przez niego południk 0°, obydwa zwrotniki i równik.</p>
    <h2 id="zakladka">Azja</h2>
    <p>Azja razem z Europą tworzy Eurazję, największy kontynent na Ziemi. Z powodów historycznych i kulturowych sama Azja bywa również nazywana kontynentem. Nazywana jest często kontynentem wielkich kontrastów geograficznych.</p>
    <h2>Australia</h2>
    <p>Nazwa "Australia" pochodzi od łacińskiego określenia "Terra Australis", czyli "Ziemia Południowa". W dawnej Europie wierzono, że na półkuli południowej znajduje się naprawdę duży kontynent. Ostatecznie nazwa ta przypadła kontynentowi o wiele mniejszymu niż zakładano.</p>
</section>
```

## Mapy odsyłaczy

Dowolna ilustracja umieszczona na stronie internetowej może być odsyłaczem do innej strony WWW, może też stanowić odsyłacz do wielu różnych stron internetowych. W tym celu ilustrację można podzielić na kilka obszarów o określonym kształcie i do każdego z nich przypisać adres internetowy. Powstanie tzw. mapy odsyłaczy. Kliknięcie takiego obszaru spowoduje przeniesienie do strony, której adres został z nim powiązany.

Definiowanie każdego obszaru osobno jest czasochłonne, ale istnieje możliwość zbudowania za pomocą znacznika `<map>` mapy odsyłaczy. Wraz ze znacznikiem `<map>` należy wykorzystać znacznik `<area>`.

Znacznik `<map>` ma następujące atrybuty:

- `name=" "` — definiuje nazwę mapy;
- `href=" "` — definiuje połączenie;
- `coords=" "` — definiuje współrzędne końcowych punktów aktywnego obszaru;
- `shape=" "` — określa kształt aktywnego obszaru: `circle` — koło, `rect` — kwadrat, `polygon` — wielokąt;

- title="" — odpowiednik atrybutu alt dla wstawianej grafiki (wyświetla tekst zastępczy).

Jeżeli dla wstawianego obrazu będzie deklarowana mapa odsyłaczy, to do znacznika <img> należy dodać atrybut usemap="#nazwamapy".

### Przykład 2.37

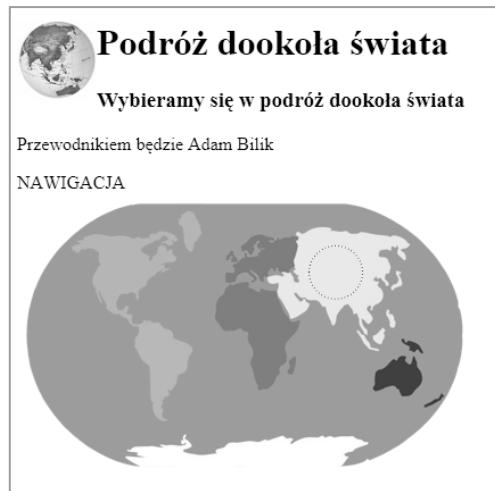
```


<map name="nazwa_mapy">

    <area shape="rect" coords="0,0,99,49" href="strona1.html" alt="strona1">
    <area shape="circle" coords="157,100,40" href="strona2.html" alt="strona 2">
</map>
```

### Ćwiczenie 2.12

Przygotuj obrazek przedstawiający mapę świata. Dla tworzonej strony na temat podróży dookoła świata zdefiniuj po sekcji header nową sekcję (<section id="sekcja0">). W powstałej sekcji umieść przygotowany obrazek. Utwórz mapę odsyłaczy, w której obrazek zostanie podzielony na oddzielne obszary odpowiadające kontynentom. Dla każdego obszaru określ kształt i do każdego z nich przypisz adres internetowy odnoszący się do wybranego kontynentu. Po wybraniu każdego z obszarów odsyłacz powinien przenosić do strony z informacją o kontynencie. Wygląd fragmentu strony został przedstawiony na rysunku 2.20. Widoczne na rysunku koło pokazuje obszar, którego kliknięcie spowoduje przejście do strony opisującej dany kontynent.



**Rysunek 2.20.**

Mapa odsyłaczy

### Rozwiążanie

Fragment kodu, za pomocą którego można zdefiniować mapę odsyłaczy do pokazanego obrazka:

```
<section id="sekcja0">

    <h2>Mapa świata</h2>

    
```

```

<map name="mapa_swiata">
    <area shape="rect" coords="75,45,100,80" href="strona1.html" alt="Ameryka
Północna">
    <area shape="circle" coords="220,120,18" href="afryka.html" alt="Afryka">
    <area shape="circle" coords="280,70,24" href="azja.html" alt="Azja">
</map>
</section>

```



## 2.4. Formularze

Dzięki formularzom, które mogą być umieszczane bezpośrednio na stronie internetowej, można zbierać informacje od użytkowników witryny. Internauci wypełniają formularz bezpośrednio na stronie — zebrane w ten sposób informacje mogą być przesłane np. pocztą elektroniczną (formularz pocztowy) i po zainstalowaniu specjalnego oprogramowania lub skryptów wykonywanych bezpośrednio w przeglądarce zostać przetworzone na potrzeby właściciela witryny. Formularze mogą służyć do tworzenia ankiet, książek gości, zamawiania towarów czy przesyłania dodatkowych informacji.

Formularze mogą zawierać: pola tekstowe, przyciski opcji, pola wyboru, listy rozwijane. W języku HTML istnieje grupa znaczników służących do projektowania formularzy.

Działanie formularza polega na podaniu (na stronie internetowej) informacji przez użytkownika i po naciśnięciu przycisku wysłaniu ich na serwer. Na serwerze dostarczone dane są przetwarzane przez odpowiednie aplikacje, a gdy jest to konieczne, po nawiązaniu połączenia z bazą są zapisywane w bazie danych lub z niej odczytywane. Po odczytaniu z bazy dane mogą zostać przesłane ponownie do użytkownika strony internetowej.

### 2.4.1. Znacznik <form>

Każdy formularz zaczyna się od znacznika otwierającego `<form>` i kończy się znacznikiem zamkającym `</form>`. Pomiędzy tymi znacznikami znajdują się wszystkie inne polecenia dotyczące formularza. Podstawowe elementy formularza to pola tekstowe i listy oraz przycisk powodujący wysłanie danych.

#### Przykład 2.38

```

<form action="mailto:nauka@gmail.com" method="post">
    <!--treść formularza-->
</form>

```

lub

```

<form action="mailto:nauka@gmail.com?subject=temat" method="post">
    <!--treść formularza-->

```

```
</form>
```

albo

```
<form action="http://localhost/skrypt_form.php" method="post">
```

Znacznik `<form>` powinien posiadać następujące atrybuty:

- `action="adres"`. To najważniejszy atrybut, ponieważ ustala adres, na który będzie wysłany formularz. Jako wartość tego atrybutu można również podać adres poczty albo adres skryptu PHP lub CGI, który zostanie wywołany po zatwierdzeniu formularza. Dane wprowadzone do formularza zostaną przekazane do wybranego skryptu. Dodanie parametru `?subject=temat` umożliwia podanie tytułu e-maila.
- `method="post"` określa sposób (metodę) przekazania informacji do wskazanego skryptu lub na wskazany adres. Może przyjąć jedną z dwóch wartości: `post` lub `get`. Wartością domyślną jest `get`, ale zaleca się stosowanie metody `post`.
- `enctype=" "` ustala sposób kodowania przesyłanych danych. Wybierany jest tylko wtedy, gdy formularz przekazywany jest metodą `post`. Dla metody `get` zawsze ma ustawioną wartość `"application/x-www-form-urlencoded"`. Domyślnie dla metody `post` też przyjmuje wartość `"application/x-www-form-urlencoded"`. Przy przesyłaniu pliku na serwer należy ustawić dla tego atrybutu wartość `"multipart/form-data"`. Przy wysyłaniu zwykłego formularza atrybut ten można pominąć, natomiast przy wysyłaniu pliku należy nadać atrybutowi wartość `"multipart/form-data"`. W formularzach pocztowych należy ustawić dla tego atrybutu wartość `"text/plain"`.

## 2.4.2. Pole INPUT

Pole INPUT jest najważniejszym elementem formularza. Może być wyświetlane w różny sposób w zależności od podanego typu w atrybucie `type`. Jest definiowane przy użyciu znacznika `<input>`.

### Przykład 2.39

```
<form action="http://localhost/skrypt_form.php" method="post">
<p>Podaj swój adres e-mail:</p>
<input type="text" name="adres" value="">
<input type="submit" value="Wyślij">
</form>
```

Znacznik `<input>` ma następujące atrybuty:

- `name` — określa nazwę danego pola oraz nazwę zmiennej, do której zostanie podstawiona wprowadzona wartość. Każde pole formularza powinno posiadać ten atrybut. Należy zwrócić uwagę na to, aby wartość tego atrybutu była różna dla wszystkich pól formularza.

- `value` — określa domyślną wartość pola.
- `size` — określa liczbę znaków mieszczących się w widocznej części pola.
- `maxlength` — określa maksymalną liczbę znaków, jaką można wpisać do pola.
- `type` — określa typ pola.

## Typy pól

Oto przykładowe wartości, jakie może przyjmować atrybut `type` dla elementu `input`:

- `type="text"` — pole tekstowe,
- `type="checkbox"` — pole zaznaczenia opcji,
- `type="radio"` — pole wyboru,
- `type="password"` — pole wprowadzania hasła,
- `type="file"` — dołączanie do formularza pliku,
- `type="submit"` — przycisk,
- `type="reset"` — przycisk,
- `type="hidden"` — pole niewidoczne na stronie,
- `type="button"` — przycisk dowolnego przeznaczenia.

### Atrybut `type="text"`

Pole z tak zdefiniowanym atrybutem `type` służy do wprowadzania tekstu jednowierszowego.

### Przykład 2.40

```
<form action="http://localhost/skrypt_form.php" method="post">
  <h4>Formularz kontaktowy</h4>
  <p>Podaj swoje nazwisko i imię:</p>
  <input type="text" name="nazwisko" value="" size="30"></p>
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 2.21.

**Formularz kontaktowy**

Podaj swoje nazwisko i imię:

**Rysunek 2.21.** Formularz zawierający pole tekstowe

Jeżeli nie zostanie zdefiniowany typ pola, domyślnie będzie ono polem tekstowym. Jeżeli nie podamy wartości atrybutu `size`, to wartość domyślna będzie wynosiła 20 znaków.

## Atrybut type="checkbox"

Pole z tak zdefiniowanym atrybutem type służy do zaznaczenia opcji wyboru (można zaznaczyć lub usunąć zaznaczenie).

### Przykład 2.41

```
<form action="http://localhost/skrypt_form.php" method="post">
    <h4>Formularz kontaktowy</h4>
    <p>Podaj swoje nazwisko i imię: <input type="text" name="nazwisko" value="" size="30"></p>
    <p><input type="checkbox" name="opcje"> Zgadzam się na przetwarzanie moich danych osobowych</p>
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 2.22.

**Formularz kontaktowy**

Podaj swoje nazwisko i imię:

Zgadzam się na przetwarzanie moich danych osobowych

**Rysunek 2.22.** Formularz zawierający pole opcji

## Atrybut type="radio"

Pole z tak zdefiniowanym atrybutem type jest polem wyboru (z grupy opcji można zaznaczyć tylko jedną opcję). Kliknięcie wartości powoduje zaznaczenie opcji, ale równocześnie usuwa zaznaczenie innej opcji.

### Przykład 2.42

```
<form action="http://localhost/skrypt_form.php" method="post">
    <h3>Wykształcenie:</h3>
    <p><input type="radio" value="Podstawowe" name="grupa_opcji" checked>Podstawowe<br>
        <input type="radio" value="Średnie" name="grupa_opcji">Średnie<br>
        <input type="radio" value="Wyższe" name="grupa_opcji">Wyższe</p>
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 2.23.

### Wykształcenie:

- Podstawowe
- Średnie
- Wyższe

**Rysunek 2.23.**

Formularz zawierający grupę opcji

Atrybut `checked` podany przy jednej z wartości z listy spowoduje, że pole to będzie domyślnie wybrane. Dla pól typu `radio` wszystkie utworzone pola muszą mieć taką samą nazwę (atrybut `name=""`). Atrybut `value=""` jest w tym przypadku konieczny i musi być inny dla każdego pola.

### Atrybut `type="password"`

Pole z tak zdefiniowanym atrybutem `type` jest polem wprowadzania hasła (wprowadzane w polu znaki nie są widoczne). Po wpisaniu hasła pojawią się kropki zamiast wprowadzonej wartości. Należy jednak pamiętać, że mimo to dane wysyłane na serwer nie są szyfrowane.

### Przykład 2.43

```
<form action="http://localhost/skrypt_form.php" method="post">
    <p>Podaj hasło: <input type="password" name="haslo" size="10"
    maxlength="14"></p>
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 2.24.

**Rysunek 2.24.** Formularz zawierający pole wprowadzania hasła

### Atrybut `type="file"`

Pole z tak zdefiniowanym atrybutem `type` pozwala dołączyć do formularza plik, który po zatwierdzeniu formularza zostanie przesłany na serwer. Przy tworzeniu pola typu `file` musi powstać pole, w którym użytkownik może wpisać ścieżkę dostępu i nazwę pliku, oraz przycisk [Przeglądaj](#), za pomocą którego użytkownik może wyszukać plik. Formularz zawierający pole tego typu musi być przekazywany metodą `post` i należy ustawić dla niego sposób kodowania danych na `enctype="multipart/form-data"`.

### Przykład 2.44

```
<form enctype="multipart/form-data" action="http://localhost/skrypt_form.php"
method="post">
    <p>Wyślij ten plik <input type="file" name="plik" size="30"></p>
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 2.25.

**Rysunek 2.25.** Formularz do dodawania plików

### Atrybut type="submit"

Pole z tak zdefiniowanym atrybutem `type` tworzy przycisk, który uruchamia wykonanie akcji zdefiniowanej w atrybucie `action`. Wartość przypisana do tego pola jest wyświetiana jako napis na przycisku.

### Atrybut type="reset"

Pole z tak zdefiniowanym atrybutem `type` powoduje utworzenie przycisku, którego kliknięcie usuwa zawartość wszystkich pól formularza.

### Przykład 2.45

```
<form action="http://localhost/skrypt_form.php" method="post">
    <h4>Formularz kontaktowy</h4>
    <p>Podaj swoje nazwisko i imię:</p>
    <p><input type="text" name="nazwisko" value="" size="30"></p>
    <p>Wykształcenie:</p>
    <p>
        <input type="radio" value="Podstawowe" name="grupa_opcji" checked>
        Podstawowe<br>
        <input type="radio" value="Średnie" name="grupa_opcji"> Średnie<br>
        <input type="radio" value="Wyzsze" name="grupa_opcji"> Wyzsze<br>
    </p>
    <p><input type="checkbox" name="opcje"> Zgadzam się na przetwarzanie moich
    danych osobowych</p>
    <p><input type="submit" value="Wyślij" style="margin-right: 10px;"> <input type="reset"
    value="Wyczysć"></p>
</form>
```

Wynik działania kodu zdefiniowanego w przykładzie został pokazany na rysunku 2.26.

**Formularz kontaktowy**

Podaj swoje nazwisko i imię:

Wykształcenie:

Podstawowe  
 Średnie  
 Wyzsze

Zgadzam się na przetwarzanie moich danych osobowych

**Wyślij** **Wyczysć**

**Rysunek 2.26.** Formularz zawierający pola typu "submit" i "reset"

## Atrybut type="hidden"

Pole z tak zdefiniowanym atrybutem type tworzy pole tekstowe niewidoczne na stronie. Pole jest niewidoczne i niedostępne dla użytkownika, ale jego wartość jest przesyłana wraz z formularzem. Nie ma znaczenia, w którym miejscu formularza zostaną umieszczone ukryte pola. Mogą one służyć do zapisywania informacji z formularzy wcześniej wypełnionych przez użytkownika.

## Przykład 2.46

```
<input type="hidden" name="Data urodzenia" value="3 marca 1980">
```

## Atrybut type="button"

Pole z tak zdefiniowanym atrybutem type powoduje utworzenie przycisku dowolnego przeznaczenia.

## Przykład 2.47

```
<input type="button" value="Przycisk">
```

Utworzony w ten sposób przycisk po kliknięciu nie spowoduje żadnego działania. W celu jego wykorzystania należy zdefiniować skrypt realizujący obsługę zdarzeń.

## Atrybut type="color"

Pole z tak zdefiniowanym atrybutem type służy do ustawiania w polu input koloru.

```
<input type="color" value="#F0F0F0">
```

## Przykład 2.48

```
<form action="http://localhost/skrypt_form.php"
method="post">
<p>Wybierz kolor z podanej palety:</p>
<input type="color" name="kolor"
value="#ff0000">
<input type="submit" value="Wyślij">
</form>
```

Wynik wykonania kodu został pokazany na rysunku 2.27.



**Rysunek 2.27.**  
Wybór koloru z palety barw

## Atrybut type="email"

Pole z tak zdefiniowanym atrybutem type służy do wprowadzania adresu e-mail.

## Przykład 2.49

```
<form action="http://localhost/skrypt_form.php" method="post">
```

```
<p>Wpisz adres e-mail:</p>
<input type="email" name="email">
<input type="submit" value="Wyślij">
</form>
```

### Atrybut type="number"

Pole z atrybutem type ustawionym na wartość number służy do wprowadzania liczb całkowitych. Za pomocą dodatkowych atrybutów można ustawić zakres liczb możliwych do wprowadzenia.

### Przykład 2.50

```
<form action="http://localhost/skrypt_form.php" method="post">
<p>Wprowadź liczby z przedziału od 10 do 100:</p>
<input type="number" name="liczba" min="10" max="100" step="5" value="20">
<input type="submit" value="Wyślij">
</form>
```

W podanym przykładzie zostanie wyświetcone pole numeryczne do wprowadzania liczb. Można wprowadzić liczby z zakresu od 10 do 100 z krokiem równym 5. Domyślnie zostanie wyświetlona wartość 20.

### Atrybut type="date"

Pole z atrybutem type ustawionym na wartość date służy do wprowadzania dat. Po ustawieniu type="date" w polu tekstowym powinien się pojawić wzorzec określający sposób wpisania daty.

### Przykład 2.51

```
<form action="http://localhost/skrypt_form.php" method="post">
<p>Podaj datę urodzin:</p>
<input type="date" name="data">
<input type="submit" value="Wyślij">
</form>
```

Wynik wykonania kodu został pokazany na rysunku 2.28.

Podaj datę urodzin:



**Rysunek 2.28.** Pole formularza z ustawionym type="date" dla znacznika <input>

## 2.4.3. Inne elementy formularza

### Pole BUTTON

Pole **BUTTON** służy do tworzenia w formularzu przycisków i jest definiowane za pomocą znacznika `<button>`. Ma podobne działanie jak znacznik `<input>`, który dla atrybutu `type` przyjmuje jedną z następujących wartości: `button`, `reset` lub `submit`.

```
<button type="submit" name="przycisk1">Wyślij</button>
<button type="reset" name="przycisk2">Usuń</button>
<button type="button" name="przycisk3">Powrót</button>
```

Różnica polega na tym, że przyciski tworzone za pomocą znacznika `<button>` mogą zawierać elementy ustawiające wygląd przycisku. Znacznik `<button>` może mieć atrybut `style`, powodujący zmianę wyglądu tekstu na przycisku.

### Przykład 2.52

```
<button type="submit" name="przycisk" style="font: 12pt Arial; background: red">Przycisk</button>
```

Aby umieścić na przycisku obraz, należy wykorzystać znacznik `<img>`.

### Przykład 2.53

```
<button type="submit" name="obrazek">
    
</button>
```

### Etykieta

Opis pola, który wyświetla się na formularzu, nie jest związany z tym polem. Używając znacznika `<label>`, można związać tekst etykiety z polem, którego dotyczy opis.

Definicję etykiety można zapisać następująco:

```
<label><input type="checkbox" name="nazwa" value="wartość">Etykieta pola
</label>
```

lub

```
<input type="checkbox" name="nazwa" value="wartość" id="pole">
<label for="pole">Etykieta pola</label>
```

Po zdefiniowaniu takiego polecenia wyświetli się pole oraz skojarzona z nim etykieta. Efekt jest taki, że kliknięcie zarówno w obrębie pola, jak i etykiety powoduje zaznaczenie opcji lub usunięcie zaznaczenia. Tekst etykiety może zostać zdefiniowany przed definicją pola `input` lub po tej definicji. Etykiety można przypisywać do wszystkich typów pól formularza.

## Przykład 2.54

```
<form action="mailto:nauka@gmail.com" method="post">
    <h2>Formularz kontaktowy:</h2>
    <p>Podaj swoje nazwisko i imię:</p>
    <input name="adres" value="" size="30">
    <h3>Wykształcenie:</h3>
    <label><input type="radio" value="Podstawowe" name="grupa_opcji" checked> Podstawowe</label><br>
    <label><input type="radio" value="Średnie" name="grupa_opcji"> Średnie</label><br>
    <label><input type="radio" value="Wyższe" name="grupa_opcji"> Wyższe</label><br>
    <label><input type="checkbox" name="opcje"> Zgadzam się na przetwarzanie moich danych osobowych </label>
    <input type="submit" value="Wyślij">&ampnbsp &ampnbsp
    <input type="reset" value="Wyczyść">
</form>
```

Niektóre z pól zostały otoczone znacznikiem `<label>`. Wynik działania tak zapisanego kodu widać dopiero podczas obsługi formularza. Kliknięcie zarówno opcji wyboru, jak i tekstu ją opisującego powoduje zaznaczenie wybranej opcji.

## Grupowanie elementów formularza

Jeżeli projektowany formularz zawiera wiele powiązanych ze sobą elementów, można te elementy pogrupować za pomocą znacznika `<fieldset>`. Powinien się on znaleźć za znacznikiem `<form>`, ale przed znacznikami definiującymi poszczególne elementy. Stosując znacznik `<legend>`, można dodać tytuł dla pogrupowanych elementów.

Sposób grupowania elementów:

```
<fieldset>
    <legend>Tytuł</legend>
    Definiowanie elementów grupy
</fieldset>
```

## Przykład 2.55

```
<form action="http://localhost/skrypt_form.php" method="post">
    <h3>Dane osobowe:</h3>
    <p>Podaj swoje nazwisko i imię:</p>
    <input name="dane" value="" size="30">
```

```
<fieldset>
  <legend><h4>Wykształcenie:</h4></legend>
  <label><input type="radio" value="Podstawowe" name="grupa_opcji" checked>
    Podstawowe</label><br>
  <label><input type="radio" value="Średnie" name="grupa_opcji">
    Średnie</label><br>
  <label><input type="radio" value="Wyższe" name="grupa_opcji">
    Wyższe</label><br>
</fieldset>
<label> Pracuję: <input type="checkbox" name="opcje"></label>
<input type="submit" value="Wyślij" style="margin-right: 10px;">
<input type="reset" value="Wyczyść">
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 2.29.

**Dane osobowe:**

Podaj swoje nazwisko i imię:

**Wykształcenie:**

Podstawowe  
 Średnie  
 Wyższe

Pracuję:

**Rysunek 2.29.** Grupowanie elementów formularza

Do opisania grupy elementów zastosowany został znacznik `<legend>`, w którym zdefiniowano nagłówek grupy (*Wykształcenie*). Znacznik `<legend>` może zostać pominięty.

## Pole SELECT

Pole `SELECT` służy do wyświetlania w formularzu listy wartości i zachęca do wybrania jednej lub kilku z nich. Jest definiowane przy użyciu znacznika `<select>`, wewnątrz

którego mogą wystąpić znaczniki `<option>`. Znacznik `<select>` definiuje listę, a poszczególne wartości tej listy są definiowane za pomocą znacznika `<option>`.

Definicja listy jest następująca:

```
<select name="lista">
<option>Pierwszy element listy</option>
<option>Drugi element listy</option>
<option>Trzeci element listy</option>
</select>
```

Dla znacznika `<select>` można używać atrybutów:

- `size="n"` —  $n$  określa liczbę pozycji widocznych w polu SELECT. Jeżeli atrybut nie zostanie podany, to pole wyboru będzie działać z `size="0"` i użytkownik nie będzie mógł wybrać żadnej wartości z listy.
- `multiple` — można na liście zaznaczyć więcej niż jedną pozycję.

Dla znacznika `<option>` można używać atrybutów:

- `selected` — pozycja, dla której został zdefiniowany ten atrybut, będzie automatycznie zaznaczona na liście. Jeżeli atrybut ten nie zostanie użyty, domyślnie będzie zaznaczany pierwszy element na liście.
- `value="wartość"` — określa wartość danej pozycji przekazywaną przez wysyłany formularz.

## Przykład 2.56

```
<form action="http://localhost/skrypt_form.php" method="post">
<h3>Formularz wyboru języka:</h3>
<select name="języki" size="5" multiple>
<option>Język angielski</option>
<option value="Język niemiecki" selected="selected">Język niemiecki
</option>
<option value="Język francuski">Język francuski</option>
<option value="Język włoski" selected="selected">Język włoski</option>
<option value="Język rosyjski">Język rosyjski</option>
<option value="Język hiszpański">Język hiszpański</option>
</select>
</form>
```

Wynik zdefiniowanej listy został pokazany na rysunku 2.30.

### Formularz wyboru języka:



**Rysunek 2.30.** Formularz z polem listy wartości

Zaznaczenie więcej niż jednej pozycji jest możliwe po wcisnięciu klawisza *Ctrl*.

## Listy zagnieżdżone

Stosując znaczniki `<select>` i `<option>`, definiujemy listę wartości. Zdefiniowanej liście można nadać strukturę wielopoziomową, tworząc podmenu dla każdej wartości listy. Do tego służy znacznik `<optgroup>` z atrybutem `label="tytuł grupy"`, gdzie *tytuł grupy* to nazwa podmenu.

```
<optgroup label="tytuł">
    <!--lista wartości-->
</optgroup>
```

Dla każdego podmenu należy utworzyć przy użyciu znacznika `<option>` oddzielną listę wartości i przed nim wstawić znacznik `<optgroup>`.

## Przykład 2.57

```
<form action="http://localhost/skrypt_form.php" method="post">
    <h3>Formularz wyboru języka:</h3>
    <select name="języki" size="10" multiple>
        <optgroup label="Języki słowiańskie">
            <option selected="selected" value="polski">polski</option>
            <option value="rosyjski">rosyjski</option>
            <option value="ukraiński">ukraiński</option>
        </optgroup>
        <optgroup label="Języki germanckie">
            <option value="angielski">angielski</option>
            <option selected="selected" value="niemiecki">niemiecki</option>
        </optgroup>
        <optgroup label="Języki romańskie">
            <option value="francuski">francuski</option>
        </optgroup>
    </select>
</form>
```

```

<option value="włoski">włoski</option>
<option value="hiszpański">hiszpański</option>
</optgroup>
<optgroup label="Języki bałtyckie">
    <option value="litewski">litewski</option>
    <option value="łotewski">łotewski</option>
</optgroup>
</select>
</form>

```

Wynik interpretacji kodu został pokazany na rysunku 2.31.



**Rysunek 2.31.** Formularz z polem listy zagnieżdżonej

## Pole TEXTAREA

Do wprowadzenia w formularzu większej ilości tekstu można użyć pola TEXTAREA, które definiujemy za pomocą znacznika `<textarea>`. Znacznik `<textarea>` może zawierać następujące atrybuty:

- `name="nazwa"` — nazwa przypisana do pola,
- `rows="n"` — liczba wierszy w polu tekstowym,
- `cols="n"` — liczba kolumn (znaków) w polu tekstowym.

Na przykład:

```
<textarea name="nazwa" rows="n" cols="m"></textarea>
```

Tekst umieszczony pomiędzy znacznikami `<textarea>` i `</textarea>` będzie wyświetlany jako wartość domyślna widoczna w polu bezpośrednio po wczytaniu zawartości strony do przeglądarki.

**Przykład 2.58**

```
<form action="http://localhost/skrypt_form.php" method="post">
<h3>Życiorys:</h3>
<textarea name="blok" rows="8" cols="40">Wprowadź swój tekst</textarea>
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 2.32.

**Rysunek 2.32.** Formularz z polem do wprowadzania bloku tekstu

Rozmiar obszaru tekstowego dla znacznika `<textarea>` można też definiować za pomocą arkuszy stylów.

**Przykład 2.59**

```
<form action="http://localhost/skrypt_form.php" method="post">
<h3>Życiorys:</h3>
<textarea name="blok" style="width:250px; height:100px">Wprowadź swój tekst
</textarea>
</form>
```

**Nieaktywne pola formularza**

Cały formularz można uczynić nieaktywnym poprzez zdefiniowanie atrybutu `disabled`. Można również dodać atrybut `disabled` do dowolnego elementu formularza, sprawiając, że tylko ten element stanie się nieaktywny. Najlepsze efekty daje związanie atrybutu `disabled` ze znacznikiem `<input>`.

Przykład: `<input disabled name="nazwisko" value="Podaj nazwisko" size="25">`.

Inną metodą może być uniemożliwienie wprowadzenia zmian w polu przez użytkownika. Do tego służy atrybut `readonly`, którego użycie spowoduje zablokowanie pola.

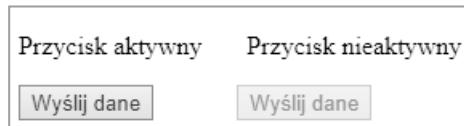
Przykład: `<input readonly name="nazwisko" value="Podaj nazwisko" size="25">`.

Atrybut `readonly` może być używany dla pól tekstowych, pól haseł, pól wyboru i bloków tekstu.

### Przykład 2.60

```
<form action="mailto:nauka@gmail.com" method="post">
    <p>Przycisk aktywny    &nbsp; &nbsp; &nbsp; Przycisk nieaktywny</p>
    <input type="submit" value="Wyślij dane" name="przycisk1" /> &nbsp; &nbsp;
    &nbsp; &nbsp;
    <input type="submit" value="Wyślij dane" name="przycisk2"
    disabled="disabled">
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 2.33.



**Rysunek 2.33.** Zdefiniowane przyciski: aktywny i nieaktywny

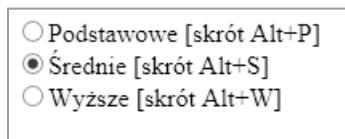
### Klawisze skrótów

Do definiowanych w formularzu elementów można przypisywać klawisze skrótów. Zastosowanie kombinacji klawiszy spowoduje przejście do wybranego elementu. Do definiowania klawiszy skrótów służy atrybut `accesskey` umieszczony w definicji elementu, którego będzie dotyczył.

### Przykład 2.61

```
<input type="radio" value="Podstawowe" name="grupa_opcji"
accesskey="P">Podstawowe [skrót Alt+P]<br>
<input type="radio" value="Średnie" name="grupa_opcji" accesskey="S">Średnie
[skrót Alt+S]<br>
<input type="radio" value="Wyższe" name="grupa_opcji" accesskey="W">Wyższe
[skrót Alt+W]<br>
```

Wynik interpretacji kodu został pokazany na rysunku 2.34.



**Rysunek 2.34.** Formularz z przypisanymi do elementów klawiszami skrótów

## Pole DATALIST

Pole DATALIST może służyć jako element autouzupełniania wartości wprowadzanych w znaczniku `<input>`. Za pomocą znacznika `<datalist>` można zdefiniować predefiniowaną listę wartości. Po wybraniu pola tekstowego pojawi się lista rozwijana z wcześniejszymi zdefiniowanymi wartościami.

Znacznik `<datalist>` musi być poprzedzony znacznikiem `<input>`. Wewnątrz znacznika `<datalist>` muszą wystąpić znaczniki `<option>`. Znacznik `<datalist>` definiuje listę, a poszczególne wartości tej listy są definiowane za pomocą znacznika `<option>`. Pełna definicja pola DATALIST ma postać:

```
<input list="nazwa listy">
<datalist id="nazwa listy">
<option value="wartość1"></option>
<option value="wartość2"></option>
<option value="wartość3"></option>
</datalist>
```

W tej konstrukcji atrybut `list` znacznika `<input>` musi odpowiadać atrybutowi `id` znacznika `<datalist>`.

### Przykład 2.62

```
<form action="http://localhost/skrypt_form.php" method="post">
<h3>Dane osobowe:</h3>
<p>Podaj swoje nazwisko i imię:</p>
<input name="dane" value="" size="30">
<h4>Wykształcenie:</h4>
<input list="wyksztalcenie" name="wyksz">
<datalist id="wyksztalcenie">
<option value="podstawowe"></option>
<option value="średnie"></option>
<option value="wyższe"></option>
</datalist>
<input type="submit" value="Wyślij">
</form>
```

Wynik wykonania kodu został pokazany na rysunku 2.35.

**Dane osobowe:**

Podaj swoje nazwisko i imię:

**Wykształcenie:**

podstawowe	wyślij
średnie	
wyższe	

**Rysunek 2.35.**  
Formularz z listą do autouzupełniania

## Atrybut autofocus

Do automatycznego ustawienia kurSORA w wybranym polu formularza służy atrybut `autofocus`. Ustawienie tego atrybutu spowoduje, po uruchomieniu strony, automatyczne umieszczenie kurSORA w oznaczonym polu formularza.

### Przykład 2.63

```
<input name="adres" type="text" autofocus>
```

## Atrybut autocomplete

Do wyłączania autouzupełniania w wybranych polach lub w całym formularzu służy atrybut `autocomplete`, przyjmujący wartość `on` lub `off`. Przeglądarki internetowe często automatycznie proponują uzupełnienie wybranego pola wartościами, które użytkownik wpisywał do tego pola wcześniej. Jeżeli użytkownik korzysta z ogólnie dostępnego komputera, to kolejny użytkownik może zobaczyć wcześniej wpisane dane. Aby temu zapobiec, atrybut `autocomplete` powinien być zawsze ustawiany na wartość `off`.

### Przykład 2.64

```
<input type="text" autocomplete="off">
```

## Atrybut required

Atrybut `required` jest elementem walidacji formularzy. Służy do sprawdzenia, czy pole zostało wypełnione. Jeżeli formularz nie ma wypełnionego pola, nie zostanie wysłany.

### Przykład 2.65

```
<input type="text" required>
```

## Atrybut pattern

Kolejnym atrybutem przeznaczonym do walidacji formularzy jest `pattern`. Pozwala zdefiniować wyrażenie regularne i sprawdza, czy wartość wprowadzona do pola jest zgodna z zadeklarowanym wyrażeniem.

### Przykład 2.66

```
<input type="text" name="kod" pattern="[0-9]{2}-[0-9]{3}">
```

W podanym przykładzie, aby wprowadzić kod pocztowy, należy wpisać dwie cyfry ( $[0-9]\{2\}$ ), następnie znak minus i na końcu trzy cyfry ( $[0-9]\{3\}$ ).

## Podsumowanie

Z formularzami na stronach internetowych spotykamy się bardzo często. Przy budowaniu formularzy konieczna jest znajomość tworzenia i działania omówionych elementów. Należy pamiętać, że aby w pełni wykorzystać formularze, nie wystarczy znajomość języka

HTML. Niezbędna jest jeszcze znajomość języka przetwarzania po stronie serwera (ang. *server-side*) i języka po stronie klienta (ang. *client-side*). Formularz projektujemy w języku HTML. Kod przetwarzający dane, które zostały wprowadzone do formularza, piszemy w języku przetwarzania po stronie serwera (np. PHP, Perl), a obsługę zdarzeń związanych z formularzem tworzymy w języku po stronie klienta (np. JavaScript, VBScript).

### Ćwiczenie 2.13

Z pomocą poznanych narzędzi języka HTML zaprojektuj podstronę służącą do rejestracji użytkowników korzystających z zasobów oferowanych przez odwiedzaną przez nich stronę internetową. Strona powinna zostać zbudowana za pomocą znaczników blokowych, a formularz powinien zostać zdefiniowany przy użyciu dostępnych elementów znacznika `<form>`. Przykładowy wygląd formularza został pokazany na rysunku 2.36.

**Formularz rejestracyjny:**

Podaj swoje nazwisko:

Podaj swoje imię:

Wybierz płeć:  
 Kobieca  Mężczyzna

Podaj wiek:

Podaj adres e-mail:

Wybierz wykształcenie:  
 Podstawowe  
 Średnie  
 Wyższe

Wybierz języki:  
 Język angielski  
 Język niemiecki  
 Język francuski  
 Język włoski  
 Język rosyjski  
 Język hiszpański

Wybierz zainteresowania:  
 Motoryzacja  
 Sport  
 Muzyka  
 Taniec  
 Gry komputerowe  
 Inne (podaj jakie):

Wpisz swoje uwagi:

Zgadzam się na przetwarzanie moich danych osobowych

**Rysunek 2.36.** Formularz rejestracyjny

## Rozwiążanie

Rozwiązaniem ćwiczenia 2.13 może być następujący kod:

```
<!DOCTYPE HTML>
<html lang="pl-PL">
<head>
    <title>Formularz rejestracyjny</title>
    <meta charset="UTF-8">
</head>
<body>
    <form action="http://localhost/skrypt_form.php" method="post">
        <div>
            <h1>Formularz rejestracyjny:</h1>
            <table>
                <tr>
                    <td>Podaj swoje nazwisko:</td>
                    <td><input type="text" name="nazw" value="" size="30"></td>
                </tr>
                <tr>
                    <td>Podaj swoje imię:</td>
                    <td><input type="text" name="im" value="" size="30"></td>
                </tr>
            </table>
            <p>Wybierz płeć:<br></p>
            <input type="radio" value="kobieta" name="plec">Kobieta
            <input type="radio" value="mężczyzna" name="plec">Mężczyzna
            <p>Podaj wiek:</p>
            <input list="wiek" name="wiek">
            <datalist id="wiek">
                <option value="16-24">
                <option value="25-39">
                <option value="40-60">
                <option value="powyżej 60">
            </datalist>
            <p>Podaj adres e-mail:</p>
```

```
<input type="email" name="adr" value="" size="30">  
<p>Wybierz wykształcenie:</p>  
<input type="radio" value="podstawowe" name="wykszt">Podstawowe<br>  
<input type="radio" value="średnie" name="wykszt">Średnie<br>  
<input type="radio" value="wyższe" name="wykszt">Wyższe  
<p>Wybierz języki:</p>  
<select name="języki" size="6" multiple>  
    <option value="Język angielski">Język angielski</option>  
    <option value="Język niemiecki">Język niemiecki</option>  
    <option value="Język francuski">Język francuski</option>  
    <option value="Język włoski">Język włoski</option>  
    <option value="Język rosyjski">Język rosyjski</option>  
    <option value="Język hiszpański">Język hiszpański</option>  
</select>  
<p>Wybierz zainteresowania:</p>  
<input type="checkbox" name="hobby" value="moto">Motoryzacja<br>  
<input type="checkbox" name="hobby" value="sport">Sport<br>  
<input type="checkbox" name="hobby" value="muza">Muzyka<br>  
<input type="checkbox" name="hobby" value="tan">Taniec<br>  
<input type="checkbox" name="hobby" value="gry">Gry komputerowe<br>  
<input type="checkbox" name="hobby" value="inna">Inne (podaj jakie):  
<input name="hobby">  
<p>Wpisz swoje uwagi:</p>  
<textarea name="uwagi" cols="37" rows="7"></textarea>  
<p><input type="checkbox" name="opcje"> Zgadzam się na przetwarzanie moich  
danych osobowych</p>  
<p>  
    <input type="submit" value="Wyślij" name="wyslij">  
    <input type="reset" value="Wyczyść" name="zeruj">  
</p>  
</div>  
</form>  
</body>  
</html>
```

## Ćwiczenie 2.14

Wykorzystując możliwości definiowania różnych elementów formularza, zaprojektuj ankietę na temat zainteresowań osób odwiedzających Twoją stronę. Ankieta powinna zawierać m.in. podstawowe dane osobowe oraz pytania do ankietowanych. Zastanów się, jakie informacje chciałbyś uzyskać od odwiedzających witrynę.

## 2.5. Multimedia

W większości przeglądarek internetowych można bez problemu wyświetlać tekst i grafikę zapisaną w formatach: GIF, JPG lub PNG. Również pliki z innymi rozszerzeniami, np. zawierające dźwięk, animację lub wideo, w większości przeglądarek działają prawidłowo.

Pliki multimedialne mogą mieć różne rozszerzenia. W tabeli 2.3 podano te najczęściej spotykane.

**Tabela 2.3.** Formaty plików multimedialnych

Format	Rozszerzenie	Typ pliku
WAV	.wav	format plików dźwiękowych typu <i>Wave Form Audio Format</i>
MIDI	.mid, .midi	format plików dźwiękowych MIDI
AVI	.avi	format plików wideo typu AVI
MP3	.mp3	format kompresji plików audio MP3
MPEG	.mpg, .mpeg	format plików wideo typu MPEG
QuickTime	.mov	format związany z technologią QuickTime
WMV	.wmv	format plików wideo typu WMV
Ogg	.ogg	format kontenera strumieni danych (dźwięk, obraz, napis)
WebM	.webm	format plików wideo przeznaczony do zastosowań internetowych
MPEG-4, MP4	.mp4	format plików wideo

### 2.5.1. Odsyłacze do plików

Jedną z metod prezentowania plików multimedialnych na stronie jest zdefiniowanie odsyłacza do tych plików. Metoda ta działa w każdej przeglądarce i sprawdza się przy każdym formacie plików.

### Przykład 2.67

```
<a href="muzyka.mp3">Moja muzyka</a>
```

lub

```
<a href="film.avi"></a>
```

Po kliknięciu odsyłacza do pliku zostanie automatycznie otwarte okno z programem obsługującym ten typ plików i nastąpi odtworzenie pliku. Ten sposób odtwarzania plików spowoduje, że plik nie będzie otwierany na stronie, lecz w osobnym programie, w związku z czym na komputerze użytkownika musi być zainstalowany odpowiedni program, który odczyta ten plik.

### Przykład 2.68

```
<!DOCTYPE HTML>

<html lang="pl-PL">
  <head>
    <title>Muzyka</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <p>Dźwięk i film:</p>
    <a href="rod.wma">Moja muzyka</a><br><br><br>
    <a href="film1.avi"></a>
  </body>
</html>
```

W HTML5 do osadzania plików multimedialnych na stronie internetowej mogą zostać wykorzystane nowe znaczniki, `<audio>` i `<video>`, które równocześnie dostarczają prostych odtwarzaczy dźwięku i wideo. W przypadku takiego odtwarzania plików multimedialnych nie są już potrzebne dodatkowe wtyczki. Bardziej skomplikowane projekty nadal wymagają użycia dodatkowych narzędzi w zależności od przeglądarki internetowej oraz dostępnych na stronie formatów plików.

## 2.5.2. Pliki wideo

### Znacznik `<video>`

Aby zamieścić na stronie internetowej film, należy dla znacznika `<video>` w atrybutie `src` podać ścieżkę do pliku z filmem oraz ustalić wysokość i szerokość wyświetlanego odtwarzacza.

### Przykład 2.69

```
<video height="400" width="500" controls src="ruch.ogg">
    Ten typ video nie jest obsługiwany!
</video>
```

Na stronie internetowej zarezerwowany zostanie obszar o podanych wymiarach do wyświetlenia filmu. Wymiary obrazu powinny zostać podane, w przeciwnym razie może nastąpić migotanie strony podczas ładowania obrazu. Wymiary powinny odpowiadać rozdzielczości filmu. Atrybut `controls` dodaje do obrazu panel sterujący zawierający takie elementy jak *odtwórz*, *przesuń*, *głośność*, *powiększ*, *pełny ekran* (rysunek 2.37).



**Rysunek 2.37.** Film umieszczony na stronie internetowej za pomocą znacznika `<video>`

W HTML5 obsługiwane są trzy formaty plików wideo. Są to formaty: MP4, WebM i Ogg. Jednak nie wszystkie przeglądarki internetowe je obsługują. Dlatego warto przygotować kilka plików wideo w różnych formatach, a następnie zadeklarować je w kodzie jako alternatywę. Niezbędny do tego jest znacznik `<source>`.

### Przykład 2.70

```
<video height="400" width="500" controls>
    <source src="ruch.ogg" type="video/ogg">
    <source src="ruch.mp4" type="video/mp4">
    Ten typ video nie jest obsługiwany!
</video>
```

## 2.5.3. Pliki audio

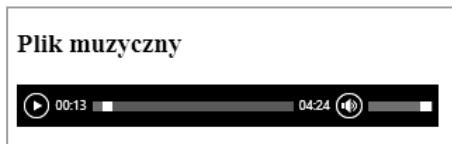
### Znacznik `<audio>`

Do umieszczania na stronie internetowej elementów audio należy użyć znacznika `<audio>`.

## Przykład 2.71

```
<audio controls>
  <source src="muza.ogg" type="audio/ogg">
  <source src="muza.wav" type="audio/wav">
  Ten typ audio nie jest obsługiwany!
</audio>
```

Atrybut `controls` dodaje panel sterujący zawierający takie elementy jak *odtwarz*, *przesuń*, *wycisz* (rysunek 2.38). W znaczniku `<source>` można podać alternatywne pliki audio, które zostaną przez przeglądarkę wybrane, jeżeli nie rozpozna ona formatu pierwszego pliku audio.



**Rysunek 2.38.** Panel sterowania dla plików audio

Nie każdy format plików audio jest obsługiwany przez wszystkie przeglądarki, dlatego w przykładzie podane zostały dwie wersje pliku muzycznego. W HTML5 obsługiwane są trzy formaty plików audio: MP3, Wav i Ogg. Nie wszystkie przeglądarki internetowe obsługują te formaty. Dlatego warto przygotować kilka wersji pliku w różnych formatach, po czym zadeklarować je w kodzie zgodnie z przykładem 2.71.

### 2.5.4. Znacznik `<canvas>`

Znacznik `<canvas>` służy do tworzenia grafiki rastrowej na stronach WWW. Dzięki niemu można rysować grafikę w czasie rzeczywistym. Znacznik `<canvas>` definiuje na ekranie prostokątny obszar (plotno), w którym tworzony jest rysunek.

Działanie tego znacznika oparte jest na renderingu trybu natychmiastowego (ang. *immediate mode rendering*). Rendering trybu natychmiastowego daje maksymalną kontrolę nad każdym pikselem obrazu, ale wymaga jawnego pisania kodu JavaScript rysującego grafikę.

Znacznik `<canvas>` daje możliwość:

- rysowania prostokątów i ścieżek (linii, łuków, krzywych Béziera lub krzywych drugiego stopnia),
- wypełniania figur kolorem (z kanałem alfa), wzorem lub gradientem (liniowym bądź radialnym),
- rysowania napisów (tekstu),
- obsługi przezroczystości,
- przekształceń (przesuwania, skalowania, obracania, przekrzywiania itp.),

- osadzania obrazów rastrowych (PNG, JPEG, GIF),
- cieniowania.

Korzystanie ze znacznika `<canvas>` jest dość pracochłonne (np. nie ma możliwości rysowania bardziej skomplikowanych figur geometrycznych, łatwego rysowania wykresów itp.).

```
<canvas id="kanwa" width="200" height="200"></canvas>
```

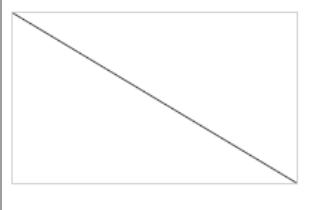
Użycie znacznika `<canvas>` spowoduje wydzielenie na stronie internetowej prostokątnego obszaru roboczego (*plotna*). Domyślnie ten obszar nie ma obramowania i nie zawiera żadnej treści. Wymagane jest dodanie atrybutów `width`, `height` oraz `id`. W skrypcie musi wystąpić odwołanie do podanego `id`. Jeżeli do wydzielonego obszaru chcemy dodać obramowanie, należy to zrobić za pomocą arkuszy stylów.

### Przykład 2.72

```
<h3>Rysuje niebieską linię</h3>
<canvas id="kanwa" width="200" height="120" style="border: 1px solid #aec5ce;">
  Brak obsługi canvas
</canvas>
<!--kod JavaScript-->
<script>
  var x = document.getElementById("kanwa");
  var xtx = x.getContext('2d');
  xtx.moveTo(0, 0);
  xtx.lineTo(200, 120);
  xtx.strokeStyle = "blue";
  xtx.stroke();
</script>
```

W wyniku wykonania kodu narysowana została linia (rysunek 2.39). Do tego celu użyte zostały funkcje: `moveTo` (początek linii) i `lineTo` (koniec linii). Dla obydwu funkcji dwa pierwsze parametry określają współrzędne punktu początkowego, dwa kolejne — punktu końcowego. Atrybut `strokeStyle` definiuje kolor pędzla rysującego. Obramowanie kanwy zostało zdefiniowane za pomocą stylów.

**Rysuje niebieską linię**



**Rysunek 2.39.**

Linia umieszczona w obszarze kanwy

### Przykład 2.73

```
<h3>Rysuje zielony kwadrat</h3>

<canvas id="kanwa" width="200" height="150" style="border: 1px solid #93c6f2;">
    Brak obsługi canvas
</canvas>

<script>

    var k = document.getElementById("kanwa");

    var kxt = k.getContext('2d');

    kxt.fillStyle = "green";

    kxt.fillRect(50, 50, 50, 50);

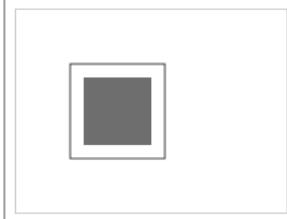
    kxt.strokeStyle = "black";

    kxt.strokeRect(40, 40, 70, 70);

</script>
```

W wyniku wykonania kodu narysowany został prostokąt (rysunek 2.40). Do tego celu użyte zostały funkcje: `fillRect` (wypełnienie kolorem) i `strokeRect` (rysowanie krawędzi). Dla obydwu funkcji dwa pierwsze parametry określają współrzędne punktu początkowego, dwa kolejne — szerokość i wysokość. Atrybuty `fillStyle` i `strokeStyle` definiują kolor pędzla rysującego. Obramowanie kanwy zostało zdefiniowane za pomocą stylów.

**Rysuje zielony kwadrat**



**Rysunek 2.40.**

Dwa kwadraty umieszczone w obszarze kanwy

## 2.5.5. Wtyczki

Z pomocą aplikacji pomocniczych (wtyczek) można rozszerzać funkcjonalność przeglądarki internetowej. Wtyczki mogą być umieszczane na stronie internetowej przy użyciu dwóch znaczników: `<embed>` i `<object>`. Mogą być one używane do skanowania w poszukiwaniu wirusów, wyświetlania map, dołączania czytników PDF czy odtwarzania animacji lub plików multimedialnych.

### Znacznik `<embed>`

Znacznik `<embed>` definiuje kontener dla aplikacji zewnętrznej lub wtyczki. Może zostać wykorzystany do umieszczenia na stronie plików multimedialnych, grafiki, animacji, ale również do dołączenia pliku HTML, np. `<embed src="kod.html">`.

W przeciwnieństwie do innych znaczników atrybuty stosowane przez znacznik `<embed>` zależą od typu wtyczki używanej do otwierania pliku. Znacznik `<embed>` nie ma znacznika zamykającego, nie może również posiadać atrybutu tekstu alternatywnego.

Jedynym wymaganym atrybutem jest `src`. Jako wartość tego atrybutu powinna zostać podana ścieżka dostępu do pliku, który będzie odtwarzany, np. `<embed src="plik">`.

Zwykle podawane są również atrybuty `width` i `height`, określające obszar wyświetlanego wtyczki. Jeżeli pominie się te atrybuty, przeglądarki w różny sposób interpretują ich brak. Czasami pojawiają się komunikaty o błędach, niekiedy przeglądarka nadaje obrazom preferowane rozmiary, czasami rozmiary zostaną nadane w sposób przypadkowy.

```
<embed src="muzyka.mp3" width="25" height="30">
```

Atrybut `type` określa rodzaj treści prezentowanej w kontenerze.

```
<embed src="napis.swf" type="application/vnd.adobe.flash-movie">
```

Atrybut `name` definiuje nazwę osadzonego obiektu. Nie jest on konieczny, ale może być przydatny podczas korzystania ze skryptów (np. JavaScript).

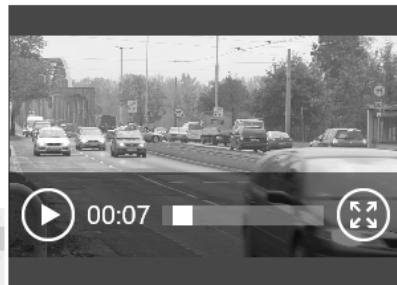
Znacznik `<embed>` może sprawiać kłopoty nawet w przeglądarkach, które obsługują pliki multimedialne. Należy pamiętać o podawaniu rozmiarów umieszczanych obiektów, ponieważ może się okazać, że użytkownik ma problemy z obsługą obiektów multimedialnych.

## Przykład 2.74

```
<!DOCTYPE HTML>
<html lang="pl-PL">
<head>
  <title>Muzyka</title>
  <meta charset="UTF-8">
</head>
<body>
  <h3>Dźwięk i film:</h3>
  <embed src="muzyka.mp3" width="170" height="70">
  <embed src="ruch.mp4" width="350" height="250">
</body>
</html>
```

Poniżej pokazane zostały rezultaty uruchomienia podanego w przykładzie kodu w różnych przeglądarkach (rysunki 2.41–2.43).

Dźwięk i film:



**Rysunek 2.41.** Odtwarzanie plików w kontenerze utworzonym za pomocą znacznika <embed>

Dźwięk i film:



**Rysunek 2.42.** Odtwarzanie plików w kontenerze. Zmiana panelu sterowania odtwarzaniem

Dźwięk i film:



**Rysunek 2.43.** Odtwarzanie plików w kontenerze. Panel sterowania jest elementem przeglądarki

## Znacznik <object>

Znacznik `<object>` definiuje obiekt, który został osadzony w dokumencie HTML. Służy do osadzania wtyczek takich jak aplety Java, czytniki PDF, odtwarzacze Flash. Za jego pomocą można umieszczać na stronie:

grafiki, np.

```
<object data="obrazek.jpg"></object>
```

animacje, np.

```
<object width="100" height="100" data="ruch.swf"></object>
```

a także dołączać pliki HTML, np.

```
<object width="200" height="300" data="strona.html"></object>
```

Dla znacznika `<object>` wymagane jest zdefiniowanie atrybutu `data`. Atrybut ten określa adres URL zasobu, który będzie użyty przez znacznik `<object>`. Kolejny atrybut, `type`, określa typ danych dla atrybutu `data`. Atrybuty `width` i `height` określają rozmiar kontenera, w którym będzie wyświetlany obiekt. Atrybut `name` definiuje zaś nazwę obiektu.

W HTML5 obiekty mogą występować w formularzu i mogą być przesyłane za pomocą formularza. Nowy atrybut, `form`, określa formularz, do którego należy obiekt.



# 2.6. Inne elementy strony internetowej

## 2.6.1. Zdarzenia

Korzystając ze strony internetowej, użytkownik wykonuje różne czynności. Do elementów, które zostały zaprojektowane na stronie internetowej, mogą być przypisane zdarzenia. Jeżeli użytkownik wykona jedną z czynności, która jest zdefiniowana jako zdarzenie, to element strony może zareagować na tę czynność. Aby było to możliwe, należy mieć skrypt, który realizuje określone działania, i należy powiązać go ze zdarzeniem, które może wystąpić. Skrypt możemy napisać w jednym z języków skryptowych (JavaScript, VBScript) lub możemy skorzystać z gotowych skryptów dostępnych w internecie.

W języku HTML zdefiniowano grupę zdarzeń wewnętrznych, które uruchamiają skrypty. Łączenie skryptów ze zdarzeniami występuje najczęściej w formularzach, przy wprowadzaniu i wysyłaniu danych.

**Zdarzenia wewnętrzne to:**

- `onclick` — występuje, gdy nastąpiło kliknięcie myszą;
- `ondblclick` — występuje, gdy nastąpiło dwukrotne kliknięcie myszą;
- `onmousedown` — występuje, gdy przycisk myszy został wciśnięty;
- `onmouseup` — występuje, gdy przycisk myszy został zwolniony;

- onmouseover — występuje, gdy kurSOR myszy został umieszczony na elemencie;
- onmousemove — występuje, gdy kurSOR myszy został przesunięty wewnątrz elementu;
- onmouseout — występuje, gdy kurSOR myszy został przesunięty poza element;
- onkeypress — występuje, gdy klawisz klawiatury został naciśnięty i zwolniony;
- onkeydown — występuje, gdy klawisz klawiatury został naciśnięty;
- onkeyup — występuje, gdy klawisz klawiatury został zwolniony;
- onfocus — występuje, gdy wybrany został element formularza;
- onblur — występuje, gdy kurSOR opuszcza element formularza;
- onselect — występuje, gdy w polu tekstowym został zaznaczony tekst (dotyczy tylko pól tekstowych);
- onchange — występuje, gdy element formularza zmienia wartość;
- onsubmit — występuje, gdy nastąpiło wysłanie formularza;
- onreset — występuje, gdy nastąpiło wyczyszczenie formularza;
- onload — występuje, gdy dokument został wczytany;
- onunload — występuje, gdy dokument został usunięty.

Dla większości zdarzeń brak zdefiniowanej obsługi jest równoznaczny z ustawieniem domyślnym.

## 2.6.2. Skrypty

Skrypt to niewielki program umieszczony bezpośrednio na stronie internetowej. Umożliwia wykonanie działań dynamicznych. Skrypty mogą być automatycznie wykonywane przez przeglądarkę podczas ładowania strony lub uruchamiane, gdy zaistnieje zdarzenie powiązane ze skryptem.

Wywołanie skryptu odbywa się poprzez znacznik `<script>`. Dla skryptów automatycznie uruchamianych ma on postać:

```
<script type="text/język">
```

Atrybut `type="text/język"` określa używany język skryptowy.

### Przykład 2.75 (dla języka JavaScript)

```
<script type="text/javascript">
treść...
</script>
```

### Przykład 2.76 (dla języka VBScript)

```
<script type="text/vbscript">
treść...
</script>
```

Atrybut `type` w HTML5 jest atrybutem opcjonalnym. Nowe atrybuty HTML5 to m.in. `async` — dla skryptów wykonywanych asynchronicznie i `src` — do określenia adresu URL zewnętrznego pliku skryptu.

## 2.7. Pytania i zadania

### 2.7.1. Pytania

1. Co należy zrobić, aby w dokumencie HTML były dostępne polskie litery?
2. Co to jest znacznik HTML? Podaj przykłady znaczników, które powinny znaleźć się na stronie.
3. Co to są atrybuty HTML? Podaj przykłady atrybutów.
4. W jaki sposób w dokumencie HTML umieszczany jest komentarz?
5. Za pomocą jakiego znacznika w dokumencie HTML tworzone są listy wypunktowane?
6. Gdzie w dokumencie HTML mają zastosowanie listy definicji?
7. Które formaty plików graficznych są najczęściej stosowane na stronach internetowych?
8. Do czego służą formularze umieszczane na stronach internetowych?
9. Do czego służy atrybut znacznika `<form>` zapisany w postaci: `method="post"`?
10. Jeżeli w znaczniku `<input>` nie wystąpi atrybut `type`, to jaki typ domyślnie przyjmie pole `INPUT`?
11. Jaka jest różnica między przyciskiem tworzonym za pomocą znacznika `<button>` a przyciskiem tworzonym przy użyciu znacznika `<input>`, który dla atrybutu `type` przyjmie wartość `button`?
12. Do czego służy używane w formularzu pole `SELECT`?
13. Jakie elementy walidacji formularzy pojawiły się w języku HTML5?
14. Jakie znaczniki w HTML5 są wykorzystywane do osadzania plików multimedialnych na stronie internetowej?
15. Do czego w HTML5 służy znacznik `<canvas>`? Opisz jego działanie.

### Pytania egzaminacyjne

#### Zadanie 1.

W języku HTML, aby zdefiniować słowa kluczowe strony, należy użyć zapisu:

- A. `<meta keywords="psy, koty, gryzonie">`.
- B. `<meta name="keywords"="psy, koty, gryzonie">`.
- C. `<meta name="keywords" content="psy, koty, gryzonie">`.
- D. `<meta name="description" content="psy, koty, gryzonie">`.

**Zadanie 2.**

W języku HTML, aby zdefiniować poziomą linię, należy użyć znacznika:

- A.** <line>.
- B.** <br>.
- C.** <hl>.
- D.** <hr>.

**Zadanie 3.**

W języku HTML zapisano kod wstawiający grafikę na stronę internetową.

```

```

Jeżeli *rysunek.png* nie zostanie odnaleziony, przeglądarka:

- A.** nie wyświetli strony internetowej.
- B.** w miejscu grafiki wypisze tekst *pejzaż*.
- C.** w miejscu grafiki wypisze tekst *rysunek.png*.
- D.** w miejscu grafiki wypisze błąd wyświetlania strony.

**Zadanie 4.**

Który z odsyłaczy posiada poprawną konstrukcję?

- A.** <a href='mailto:adres'>tekst</a>.
- B.** <a href='http://adres'>tekst</a>.
- C.** <a href="http://adres">tekst<\a>.
- D.** <a href="mailto:adres">tekst</a>.

**Zadanie 5.**

Fragment kodu napisany w języku HTML zamieszczony w ramce przedstawia listę:

- A.** wypunktowaną.
- B.** numerowaną.
- C.** odnośników.
- D.** skrótów.

```
<ol>
<li>Pierwszy</li>
<li>Drugi</li>
<li>Trzeci</li>
</ol>
```

**Zadanie 6.**

W jaki sposób należy odwołać się do pliku *default.css*, jeśli *index.html* znajduje się w katalogu *Strona*, a plik *default.css* w podkatalogu *Style* z katalogu *Strona*?

- A.** <link rel="stylesheet" type="text/css" href=".style/default.css" />.
- B.** <link rel="stylesheet" type="text/css" href="c:/style/default.css" />.

- C.** <link rel="stylesheet" type="text/css" href="..\style\default.css" />.
- D.** <link rel="stylesheet" type="text/css" href="c:\style/default.css" />.

### Zadanie 7.

Zamieszczony w ramce kod wyświetla tabelę składającą się z:

- A.** dwóch wierszy i dwóch kolumn.
- B.** dwóch wierszy i jednej kolumny.
- C.** jednego wiersza i dwóch kolumn.
- D.** jednego wiersza i jednej kolumny.

```
<table border="1">
<tr>
<td> 1 </td>
    <td> 2 </td>
</tr>
</table>
```

### Zadanie 8.

Aby poprawnie zdefiniować hierarchiczną strukturę tekstu witryny internetowej, należy zastosować:

- A.** znacznik <div>.
- B.** znaczniki <frame> i <table>.
- C.** znacznik <p> z formatowaniem.
- D.** znaczniki <h1>, <h2> oraz <p>.

### Zadanie 9.

Który z zapisów znacznika <meta> jest prawidłowy ze względu na wykorzystane atrybuty?

- A.** <meta background=blue>.
- B.** <meta name="!DOCTYPE">.
- C.** <meta title="Strona dla hobbystów">.
- D.** <meta name="description" content="Masz jakieś hobby? To jest strona dla Ciebie!">.

### Zadanie 10.

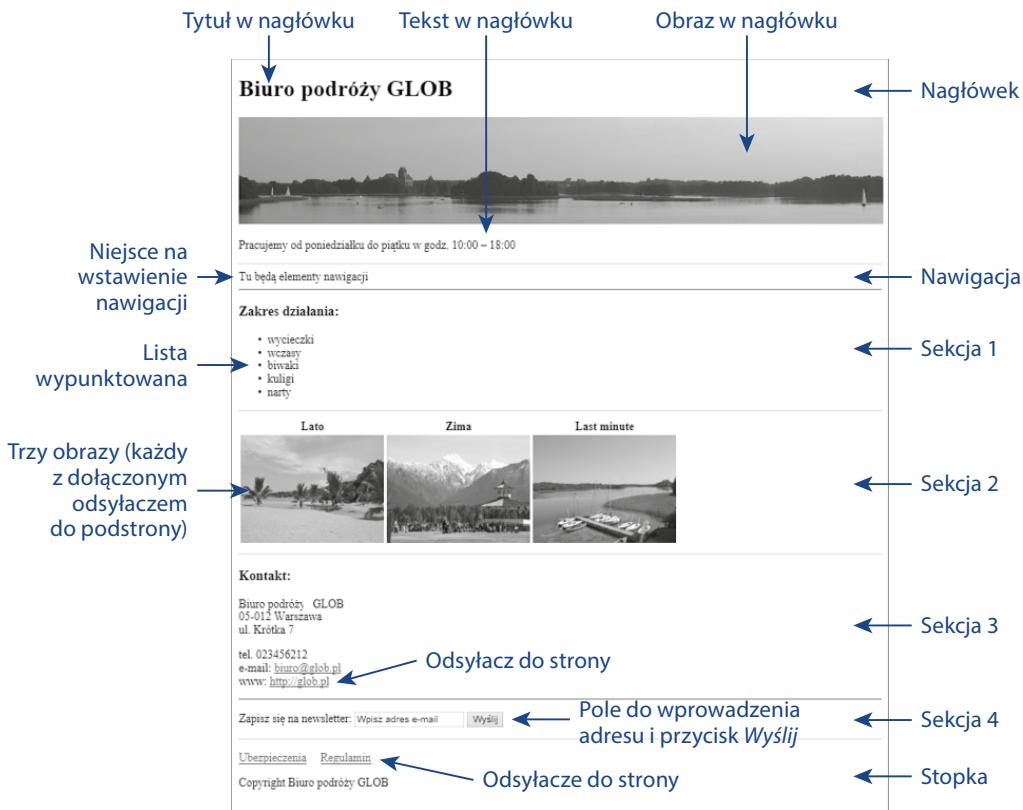
Który z wymienionych znaczników należy do części <head> dokumentu HTML?

- A.** <img>.
- B.** <title>.
- C.** <span>.
- D.** <section>.

## 2.7.2. Zadania

### Zadanie 1.

Na rysunku 2.44 przedstawiony został projekt witryny internetowej dla biura podróży GLOB. Utwórz witrynę internetową na podstawie podanego projektu.



Rysunek 2.44. Schemat układu strony internetowej

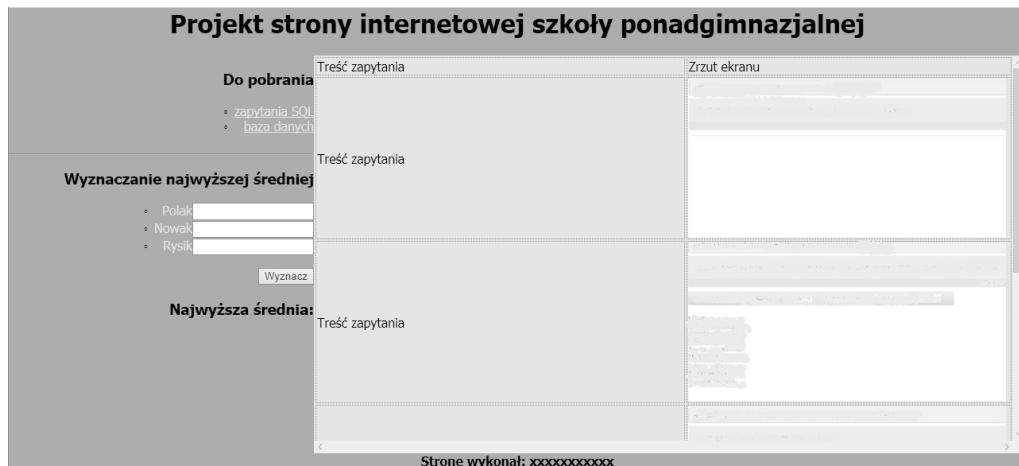
### Zadanie 2.

W grupie osób biorących udział w projekcie należy za pomocą języka HTML utworzyć kilka stron internetowych, które będą podstronami witryny na temat konkursów przedmiotowych odbywających się na terenie szkoły. Każda z osób biorących udział w projekcie przygotuje jedną stronę. Na stronie powinny się znaleźć informacje o wybranym konkursie (np. jego termin, tematyka, warunki udziału), zdjęcia, odnośniki do informacji dostępnych w internecie. Strona powinna zostać zbudowana za pomocą znaczników tworzących różnego rodzaju listy, tabele oraz formularze umożliwiające np. rejestrację uczestników konkursów.

## Zadanie egzaminacyjne — cz. I

### Witryna internetowa

Stwórz prostą witrynę składającą się z jednej strony internetowej, o nazwie *index.html*. Plik zapisz w podfolderze *www*. Wygląd witryny powinien być zgodny z rysunkiem 2.45.



**Rysunek 2.45.** Witryna internetowa

Cechy witryny:

1. Zastosowany właściwy standard kodowania polskich znaków.
2. Tytuł strony: *Szkoła ponadgimnazjalna*.
3. Strona podzielona za pomocą znaczników sekcji na baner, panele lewy i prawy oraz stopkę, tak aby po uruchomieniu strony w przeglądarce jej wygląd był zgodny z rysunkiem.
4. Zawartość banera: nagłówek pierwszego stopnia o treści: *Projekt strony internetowej szkoły ponadgimnazjalnej*.
5. Zawartość panelu lewego:
  - a) nagłówek trzeciego stopnia o treści: *Do pobrania*,
  - b) poniżej w postaci listy punktowanej dwa odnośniki:  
» *zapytania SQL*,  
» *baza danych*,
  - c) kliknięcie odnośnika o treści: *zapytania SQL* powoduje pobranie/wyświetlenie pliku *zapytania.txt* z podfolderu *baza*,
  - d) kliknięcie odnośnika o treści: *baza danych* powoduje pobranie/wyświetlenie pliku *szkola\_nowa.sql* z podfolderu *baza*,
  - e) poniżej linia pozioma przez całą szerokość panelu lewego,

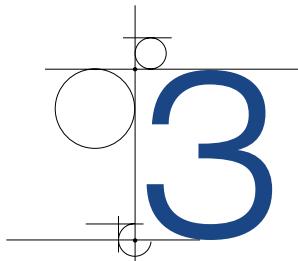
- f)** poniżej napis w nagłówku trzeciego stopnia: *Wyznaczenie najwyższej średniej*,
- g)** poniżej lista punktowana z nazwiskami: Polak, Nowak, Rysik, obok każdego nazwiska pole edycyjne typu tekstowego przeznaczone na wpisanie średniej ocen ucznia,
- h)** poniżej przycisk z etykietą *Wyznacz*,
- i)** poniżej napis w nagłówku trzeciego stopnia *Najwyższa średnia*.

**6.** Zawartość panelu prawego:

- a)** tabela o rozmiarach: 5 wierszy i 2 kolumny,
- b)** szerokość prawej kolumny tabeli: 405 px,
- c)** komórki wiersza nagłówkowego zawierają kolejno napisy: *Treść zapytania* i *Zrzut ekranu*,
- d)** w komórkach tabeli w kolejnych wierszach: treści zapytań z pliku *zapytania.txt* i obok — odpowiadające im zrzuty ekranu w formacie JPG zgodnie z rysunkiem,
- e)** obrazy powinny posiadać tekst alternatywny zawierający nazwę pliku.

**7.** Zawartość stopki: nagłówek czwartego stopnia o treści: *stronę wykonał*:

....., w miejsce kropki wpisany Twój numer PESEL.



# Kaskadowe arkusze stylów (CSS)

W początkowej fazie rozwoju język HTML był rzeczywiście strukturalnym językiem znaczników. W kolejnych jego wersjach dodano elementy opisujące wygląd strony (np. zmianę koloru czy wielkości tekstu). Z czasem pomieszanie warstwy strukturalnej dokumentu z warstwą prezentacyjną zaczęło stwarzać coraz większe problemy.

Rozwiązywanie tych problemów stało się możliwe po oddzieleniu warstwy strukturalnej i treści od warstwy prezentacyjnej. Za warstwy: strukturalną, semantyczną i treści wciąż odpowiedzialny jest język HTML, a za warstwę prezentacyjną — język CSS (ang. *Cascading Style Sheets*).

Język CSS określa układ graficzny dokumentu HTML: parametry czcionki, wysokość i szerokość obrazków, ich położenie, rodzaj tła itp. Wszystkie polecenia dotyczące formatowania powinny zostać umieszczone w oddzielnym pliku (arkuszu) i powiązane z elementami zdefiniowanymi w kodzie HTML.

Cechy języka CSS:

- Oddziela strukturę informacyjną witryny od jej warstwy prezentacyjnej.
- Daje większe możliwości formatowania tekstu.
- Pozwala zapisać wszystkie informacje dotyczące wyglądu dokumentu w oddzielnym pliku tekstowym dołączonym później do dokumentów HTML.
- Umożliwia formatowanie wielu dokumentów przy użyciu jednego arkusza stylów.
- Umożliwia stosowanie różnych układów graficznych w zależności od typu urządzenia, na którym zamierzamy wyświetlać projektowaną stronę.

Formatowanie z wykorzystaniem CSS stało się podstawą tworzenia stron internetowych. Należy pamiętać również o tym, że w języku HTML występują przestarzałe znaczniki i atrybuty formatowania (elementy zdeprecjonowane), które będą stopniowo wycofywane z przeglądarek internetowych i zastępowane nowymi metodami formatowania dostępnymi w języku CSS.

Język CSS oferuje nowe możliwości formatowania, które były niedostępne w HTML, np. dotyczące formatowania tekstu, formatowania tła, definiowania obramowania, dodatkowe właściwości definiowania list, pozycjonowania, zmianę wyglądu odsyłaczy, stosowanie filtrów.

## 3.1. Wstawianie stylów

Kaskadowe arkusze stylów definiują jedynie sposób formatowania elementów dokumentu HTML, ale ich nie tworzą. Elementy te muszą zostać zdefiniowane za pomocą znaczników w dokumencie HTML.

W podstawowej składni kaskadowych arkuszy stylów jest kilka stałych elementów:

```
selektor {właściwość: wartość;}
```

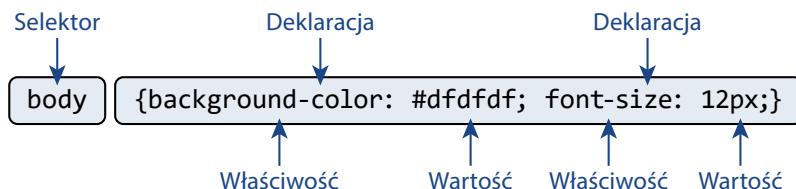
- **selektor** — określa, do jakich znaczników języka HTML odnosi się dalsza część definicji stylu. Selektorem może być dowolny znacznik HTML, np. *p*, *li*, *body*.
- **właściwość** — określa, jaki atrybut znacznika będzie ustawiany, np. wielkość czcionki dla akapitu, tło dla znacznika *body*.
- **wartość** — określa konkretną wartość atrybutu, np. *12pt*, *blue*.

Dla jednego selektora można zdefiniować kilka atrybutów. W definicji stylu należy je oddzielić średnikami.

### Przykład 3.1

```
body {background-color: #fdfdfd; font-size: 12px;}
```

Przykład pokazuje ustawienie koloru tła strony i wielkość czcionki na stronie (rysunek 3.1).



**Rysunek 3.1.** Składnia CSS

Reguły obowiązujące w składni kaskadowych arkuszy stylów:

- Selektor wskazuje na element HTML, dla którego chcemy ustawić styl.
- Blok deklaracji zawiera jedną lub kilka deklaracji rozdzielonych średnikami.
- Każda deklaracja zawiera nazwę właściwości oraz po dwukropku wartość przypisaną tej właściwości.
- Każda deklaracja kończy się średnikiem.
- Blok deklaracji jest otoczony nawiasami klamrowymi.

W dokumencie HTML istnieje kilka sposobów na dołączanie stylów CSS:

- styl lokalny (w linii),
- wewnętrzny arkusz stylów,
- zewnętrzny arkusz stylów.

Sposób wstawiania stylów zależy od konkretnej sytuacji. Podczas projektowania witryny najczęściej stosuje się różne sposoby osadzania arkuszy stylów.

### 3.1.1. Styl lokalny (w linii)

Korzystając ze stylu lokalnego, można zdefiniować formatowanie pojedynczego elementu strony. Taki styl jest definiowany w tej samej linii, w której znajduje się element formatowany, dlatego nazywany jest on również stylem w linii (ang. *inline*). Umieszczany jest bezpośrednio w dokumencie HTML. Ten sposób definiowania stylów miesza warstwę struktury i treści z warstwą prezentacji. Nie spełnia wymogów kaskadowych arkuszy stylów i powinien być stosowany bardzo oszczędnie.

#### Przykład 3.2

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Kaskadowe arkusze stylów</title>
    <meta charset="UTF-8">
</head>
<body style="background-color: #fdfdf;">
    <p style="color: #911e5d;">Podróż dookoła świata</p>
</body>
</html>
```

#### Znacznik `<span>` — rozciąganie stylu

Znacznik `<span>` służy do grupowania kilku elementów strony (np. słów, obrazków, akapitów) w celu nadania im określonego stylu. Ten znacznik sam w sobie nie ma narzędziowych żadnych właściwości, nie wywiera żadnego wpływu na zgrupowane elementy, dlatego elementy te będzie określał wyłącznie styl. Styl dla znacznika `<span>` ustala się przy użyciu stylu CSS.

```
<span style="właściwość: wartość;"> ... </span>
```

Znacznik `<span>` zwykle jest wykorzystywany wtedy, gdy trzeba inaczej sformatować kilka znaków w obszarze o określonym sposobie formatowania.

### Przykład 3.3

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Kaskadowe arkusze stylów</title>
    <meta charset="UTF-8">
</head>
<body style="background-color: #fdfdfd;">
    <h3>Wybieramy się w <span style="color: red;">podróż</span> dookoła świata.
</h3>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 3.2.

Wybieramy się w podróż dookoła świata.

**Rysunek 3.2.** Zastosowanie znacznika `<span>` do fragmentu tekstu

### Znaczniki wydzielonych bloków

Używając znaczników blokowych, elementy strony można grupować w bloki i formatać za pomocą stylów. Zdefiniowane bloki mogą zawierać m.in. akapity, listy oraz inne bloki. Znaczniki te umożliwiają wydzielanie na stronie większych, logicznych fragmentów w celu nadania im specyficznego formatowania z użyciem stylów.

Metoda definiowania stylu dla bloków jest podobna jak w przypadku znacznika `<span>`. Ustala się go za pomocą stylu lokalnego, np.:

```
<div style="właściwość: wartość;"> ... </div>
```

### Przykład 3.4

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Bloki</title>
    <meta charset="UTF-8">
</head>
<body style="background-color: #DFDFDF;">
    <div style="width: 500px; height: 50px; background-color: yellow;">
```

```
<h3>Zwiedzanie <span style="color: red">Europy</span> rozpoczniemy od <span  
style="color: red">Madrytu</span>.</h3>  
</div>  
</body>  
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 3.3.

Zwiedzanie Europy rozpoczniemy od Madrytu.

**Rysunek 3.3.** Definiowanie stylu dla bloku `<div>`

## 3.1.2. Wewnętrzny arkusz stylów

Wewnętrzny arkusz stylów jest umieszczany w dokumencie HTML dzięki zastosowaniu znacznika `<style>`, który zawsze występuje w części nagłówkowej dokumentu (`<head>`). Tę metodę wstawiania do dokumentu arkusza stylów stosuje się, gdy elementy formatowane pojawiają się na stronie wielokrotnie i wszystkie powinny mieć takie same atrybuty formatowania. Przyjmijmy, że chcemy, żeby wszystkie teksty zdefiniowane jako akapity miały kolor niebieski. Po wpisaniu odpowiedniej deklaracji stylów wszystkie akapity będą napisane czcionką niebieską bez definiowania jej koloru przy każdym elemencie. Zmiana koloru czcionki w deklaracji stylów zmieni kolor czcionki wszystkich akapitów.

### Przykład 3.5

```
<!DOCTYPE html>  
<html lang="pl-PL">  
<head>  
    <title>Kaskadowe arkusze stylów</title>  
    <meta charset="UTF-8">  
    <style>  
        p {color: blue;}  
    </style>  
</head>  
<body>  
    <p>Zwiedzanie Europy rozpoczniemy od Madrytu.</p>  
    <p>Zwiedzanie Azji rozpoczniemy od Pekinu.</p>  
    <p>Najciekawszym krajem w Afryce jest Kenia.</p>  
</body>  
</html>
```

Wynik wykonania kodu prezentuje rysunek 3.4.

Zwiedzanie Europy rozpoczniemy od Madrytu.  
Zwiedzanie Azji rozpoczniemy od Pekinu.  
Najciekawszym krajem w Afryce jest Kenia.

**Rysunek 3.4.** Definiowanie wewnętrznego arkusza stylów dla znacznika <p>

### Przykład 3.6

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Wewnętrzny arkusz stylów</title>
    <meta charset="UTF-8">
    <style>
        body {background-color: #c6bac0;}
        h2 {font-size: 20pt;}
        h3 {color: green; font-size: 14pt;}
        p {color: blue;}
    </style>
</head>
<body>
    <h2>Wybieramy się w podróż dookoła świata</h2>
    <h3>Europa</h3>
    <p>Zwiedzanie Europy rozpoczniemy od Madrytu.</p>
    <h3>Azja</h3>
    <p>Zwiedzanie Azji rozpoczniemy od Pekinu.</p>
    <h3>Afryka</h3>
    <p>Najciekawszym krajem w Afryce jest Kenia.</p>
</body>
</html>
```

Wynik wykonania kodu został pokazany na rysunku 3.5.

## Wybieramy się w podróż dookoła świata

### Europa

Zwiedzanie Europy rozpoczęmy od Madrytu.

### Azja

Zwiedzanie Azji rozpoczęmy od Pekinu.

### Afryka

Najciekawszym krajem w Afryce jest Kenia.

**Rysunek 3.5.** Definiowanie wewnętrznego arkusza stylów dla różnych znaczników

### Ćwiczenie 3.1

Przygotuj teksty i obrazy opisujące wybraną dyscyplinę sportową. Za pomocą poznanych znaczników języka HTML zaprojektuj szablon strony internetowej na ten temat. Zaproponuj teksty, które znajdą się w nagłówku i stopce strony. Umieść przygotowane wcześniej teksty w odpowiednich blokach i podziel je na akapity. Używając stylu lokalnego, zdefiniuj rodzaj, kolor i rozmiar czcionki dla tekstów wyświetlanych na stronie. To samo wykonaj, używając wewnętrznego arkusza stylów. Kiedy warto zastosować styl lokalny, a kiedyewnętrzny arkusz stylów?

### 3.1.3. Zewnętrzny arkusz stylów

Największą z zalet stosowania CSS jest możliwość wstawiania zewnętrznych arkuszy stylów. Polega ona na utworzeniu pliku tekstuowego z rozszerzeniem `.css`, który będzie zawierał definicję wszystkich stylów używanych w projektowanej witrynie. W dokumencie HTML powinien znaleźć się odnośnik do tzw. zewnętrznego arkusza stylów, czyli do pliku z rozszerzeniem `.css`.

Odnośnik umieszczony w dokumencie HTML powinien mieć postać:

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

Wartością atrybutu `href` powinna być ścieżka dostępu do pliku zawierającego zdefiniowane style. Aby utrzymać porządek w strukturze plików strony, warto umieszczać pliki stylów w odrębnych folderach, np. `css`.

Odnośnik do arkusza stylów powinien zostać umieszczony w części nagłówkowej dokumentu (`<head>`).

### Przykład 3.7

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

```
<head>
    <title>Kaskadowe arkusze stylów</title>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css" href="arkusz.css">
</head>
<body>
    <p>Wybieramy się w podróż dookoła świata</p>
    <p>Zwiedzanie Europy rozpoczniemy od Madrytu.</p>
</body>
</html>
```

Zaletą tak zdefiniowanych arkuszy stylów jest to, że do wielu dokumentów HTML można dołączyć ten sam plik arkusza stylów, czyli za pomocą jednego pliku CSS można formatować w taki sam sposób wiele stron WWW (np. kilka stron internetowych tej samej witryny).

Można również w jednym dokumencie HTML umieszczać dowolną liczbę zewnętrznych arkuszy stylów. W dokumencie tym za każdym razem musi wystąpić osobny element `<link rel="stylesheet">` odwołujący się do arkusza stylów. Jeżeli pojawi się konflikt dotyczący sposobu formatowania tego samego elementu w różnych arkuszach stylów, wiążące będą deklaracje z arkusza dołączonego najpóźniej.

### Ćwiczenie 3.2

Utwórz plik z rozszerzeniem `.css`. Zdefiniuj w nim style tekstów wyświetlanego na stronie. Dołącz go do dokumentu HTML opisującego stronę internetową z ćwiczenia 3.1.

#### 3.1.4. Import arkusza stylów

Do zewnętrznego lub wewnętrznego arkusza stylów można zaimportować zewnętrzny arkusz stylów. Plik z importowanym arkuszem stylów może się znajdować w dowolnym miejscu; należy tylko prawidłowo zdefiniować ścieżkę dostępu do niego. Importowanie arkusza stylów odbywa się podobnie jak dołączanie zewnętrznego arkusza. Priorytet dołączonego arkusza jest taki sam jak priorytet arkusza, do którego został zaimportowany.

Polecenie importowania arkusza ma postać:

```
<style>
    @import url("adres importowanego arkusza stylów.css");
</style>
```

Adres arkusza stylów powinien zostać umieszczony w nawiasach okrągłych. Przed poleceniem `import` musi zostać umieszczony znak `@`, a na końcu adresu arkusza stylów — średnik.

Oprócz importowania arkusza można między znacznikami `<style>` i `</style>` umieszczać inne deklaracje stylów. Polecenia importowania arkusza muszą występować na początku arkusza stylów, przed właściwymi regułami CSS.

### 3.1.5. Kaskadowość arkuszy stylów

Zdarza się, że w dokumencie są umieszczone odwołania do kilku zewnętrznych arkuszy stylów lub na stronie używane są: zewnętrzny arkusz stylów, deklaracja stylów w nagłówku strony oraz style lokalne. Wtedy może się pojawić konflikt dotyczący sposobu formatowania tego samego elementu w różnych arkuszach stylów. Zawsze pierwszeństwo mają style, które są umieszczone bliżej formatowanego elementu. Kaskadowość arkuszy stylów polega na ścisłym określeniu priorytetów stylów i przestrzeganiu zasad formatowania zgodnie z priorytetami. Ważność stylów została ustalona w następujący sposób:

1. Styl lokalny.
2. Rozciąganie stylu.
3. Wydzielone bloki.
4. Wewnętrzny arkusz stylów.
5. Zewnętrzny arkusz stylów.
6. Import stylów do zewnętrznego arkusza.
7. Atrybuty definiowane w dokumencie HTML.

Styl z numerem 1 ma najwyższy priorytet, a styl z numerem 7 — najniższy. Styl z najwyższym priorytetem ma pierwszeństwo w formatowaniu elementów strony. Style o niższym priorytecie formatują element tylko wtedy, gdy style o wyższym priorytecie nie dotyczą tego elementu. Należy zauważyć, że atrybuty definiowane dla elementu w dokumencie HTML mają najniższy priorytet, a więc style CSS definiowane w dowolny sposób zawsze są ważniejsze od tych atrybutów.

Postępując zgodnie z zasadami kaskadowości stylów, w dokumencie HTML polecenie dołączenia zewnętrznego arkusza stylów trzeba umieścić przed definicją wewnętrznego arkusza stylów.

Zasady kaskadowości można zmieniać. Służy do tego polecenie `!important` umieszczone w deklaracji stylu po wartości, której dotyczy. Takie umiejscowienie tego polecenia spowoduje, że właściwość zdefiniowana dla elementu będzie miała pierwszeństwo, nawet jeżeli ma niższy priorytet. Polecenie `!important` dotyczy tylko właściwości, dla której zostało użyte; pozostałe właściwości elementu będą obsługiwane zgodnie z ogólnymi zasadami kaskadowości stylów.

#### Przykład 3.8

Użycie polecenia `!important` w definicji stylu:

```
p {font-size: 20pt !important; color: green;}
```

Fragment kodu:

```
<p style="font-size: 12pt; color: red">Kaskadowe arkusze stylów</p>
```

### Przykład 3.9

```
<!DOCTYPE html>

<html lang="pl-PL">

<head>
    <title>Arkusz stylów</title>
    <meta charset="UTF-8">
    <style>
        body {background-color: #c0c0c0;}
        p {font-size: 18pt !important; color: green;}
    </style>
</head>
<body>
    <p style="font-size: 12pt; color: #a21b5f;">Wybieramy się w podróż dookoła świata.</p>
    <p style="font-size: 14pt;">Będziemy odwiedzali po kolej wszystkie kontynenty.</p>
</body>
</html>
```

Na rysunku 3.6 pokazany został efekt działania polecenia `!important`. Dla pierwszego znacznika `<p>` zastosowany został kolor czcionki zdefiniowany za pomocą stylu lokalnego (bo ma wyższy priorytet od stylu wewnętrznego), a rozmiar czcionki pochodzi z wewnętrznego arkusza stylów (mimo że ma on niższy priorytet, to polecenie `!important` wymusza pierwszeństwo zastosowania stylu czcionki z wewnętrznego arkusza stylów).

Wybieramy się w podróż dookoła świata.

Będziemy odwiedzali po kolej wszystkie kontynenty.

**Rysunek 3.6.** Efekt działania polecenia `!important`

### Ćwiczenie 3.3

W zewnętrznym arkuszu stylów został zdefiniowany selektor `h2`:

```
h2 {color: blue; text-align: center; font-size: 12pt;}
```

W wewnętrznym arkuszu stylów również został zdefiniowany selektor h2:

```
h2 {text-align: right; font-size: 24pt;}
```

Jeżeli strona ma powiązanie z zewnętrznym arkuszem stylów, to jakie właściwości będzie miał znacznik <h2>? Sprawdź swoją odpowiedź, dołączając tak zdefiniowane style do swojej strony internetowej.

### 3.1.6. Dziedziczenie

W języku HTML często występuje zagnieżdzanie jednego elementu wewnętrz drugiego. Jeżeli dla elementu nadzewnętrznego w arkuszach stylów zostały zdefiniowane właściwości, to w większości przypadków elementowi podzewnętrznemu zostaną przypisane te same właściwości, nawet jeżeli nie zostały dla niego zdefiniowane. Mechanizm ten nazywamy **dziedziczeniem**.

#### Przykład 3.10

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
  <title>Dziedziczenie</title>
  <meta charset="UTF-8">
  <style>
    body {background-color: #c0c0c0; }
    div {font-size: 16pt; color: green; }
  </style>
</head>
<body>
  <div>
    <p>Wybieramy się w podróż dookoła świata.</p>
    <p>Będziemy odwiedzali po kolej wszystkie <span>kontynenty</span>.</p>
  </div>
</body>
</html>
```

W podanym przykładzie dla tekstu w bloku <div> został zdefiniowany kolor zielony. Wewnątrz bloku wystąpiły dwa akapity i znacznik <span>, którym nie nadano żadnego stylu, więc dziedziczą go po bloku <div> (rysunek 3.7).

Wybieramy się w podróż dookoła świata.

Będziemy odwiedzali po kolej wszystkie kontynenty.

**Rysunek 3.7.** Dziedziczenie stylu dla tekstu w akapitach

Na rysunku 3.8 widzimy efekt zdefiniowania atrybutu koloru dla fragmentu tekstu wewnętrz drugiego akapitu. Ten fragment nie dziedziczył stylu po bloku <div>.

Wybieramy się w podróż dookoła świata.

Będziemy odwiedzali po kolej wszystkie kontynenty.

**Rysunek 3.8.** Brak dziedziczenia stylu dla tekstu „kontynenty”



## 3.2. Składnia języka CSS

Arkusz stylów jest zwykłym plikiem tekstowym i można go utworzyć za pomocą dowolnego edytora tekstu. Można również skorzystać ze specjalnego edytora CSS.

Edytory CSS przyspieszają i ułatwiają pracę, ponieważ automatyzują niektóre działania. Na przykład można automatycznie zdefiniować wielkość i kolor czcionki, kolor tła czy kolor odsyłaczy.

Plik będący zewnętrznym arkuszem stylów musi zostać zapisany z rozszerzeniem `.css`.

Jeżeli do arkusza stylów będą importowane inne arkusze stylów, to polecenia importowania `@import` muszą się znaleźć na początku tego arkusza w postaci:

```
@import url(ścieżka/plik.css);
```

W języku CSS parametry formatowania definiuje się za pomocą **reguł stylów**. Każda reguła odnosi się do określonego elementu.

Podstawowa składnia wygląda następująco:

```
selektor {właściwość: wartość;}
```

Powyższa składnia dotyczy stylów wewnętrznych, definiowanych w części nagłówkowej dokumentu HTML, oraz stylów zewnętrznych. Natomiast dla stylów lokalnych składnia ma postać:

```
<značnik style="właściwość: wartość;">
```

Do dokumentu HTML można dołączyć arkusz stylów zapisany w pliku `arkusz.css` z podaną przykładową zawartością.

## Przykład 3.11

```
body {background-color: #DFDFDF; }
p {font-color: blue; font-size: 5; }
```

Jeżeli dokument HTML i plik z arkuszem stylów zostaną umieszczone w odpowiednich lokalizacjach i do dokumentu HTML zostanie dołączony zewnętrzny plik CSS przy użyciu elementu <link>, to po otwarciu dokumentu zdefiniowane w arkuszu stylów ustawienia dotyczące tła strony oraz wyglądu tekstu będą widoczne na tej stronie.

W pliku CSS można umieszczać komentarze. Komentarze zaczynają się znakami /\*, a kończą się znakami \*/.

## Przykład 3.12

```
/*To jest komentarz*/
body {background-color: #DFDFDF; /*tutaj też został dodany komentarz*/
p {color: blue; font-size: 5; font-family: courier new; }
```

### 3.2.1. Wartości i jednostki

Każda właściwość używana do definiowania selektora zawiera zbiór dopuszczalnych wartości oraz dopuszczalne rodzaje wartości.

Mogą to być:

- **Liczby** — całkowite i rzeczywiste. Liczba całkowita składa się z cyfr od 0 do 9. Liczba rzeczywista może zawierać część całkowitą i część ułamkową. Część ułamkowa zapisana jest po kropce.
- **Procenty**. Wartość procentowa jest zawsze określana względem innej wartości. Na przykład może zostać użyta do określenia wielkości bloków. Podana w procentach wielkość bloku będzie się zmieniała wraz ze zmianą rozmiaru bloku, który go zawiera (rozmiar ten zwykle zależy od rozmiaru okna przeglądarki).
- **Słowa kluczowe**. Nie zawsze wartości wyrażone za pomocą słów kluczowych są poprawnie interpretowane przez przeglądarki.
- **Jednostki długości**. Mogą być zapisywane jako wartości bezwzględne i wartości względne.
  - » **Wartości bezwzględne:** in (cal, 1 in = 2,54 cm), cm (centymetr), mm (milimetr), pt (punkty, 1 cal to 72 punkty), pc (pica, 1 pc = 12 pt), px ( piksele, 1 px = 0,75 pt). Jednostki bezwzględne są rzadko stosowane. Należy ich używać wtedy, gdy chcemy, aby wybrany element miał taki sam rozmiar niezależnie od rozdzielczości monitora.
  - » **Wartości względne:** em (1 em jest wielkością rozmiaru czcionki, jaką posiada dany element; wartość 0,1 em jest najmniejszą wartością, jaką przeglądarki są w stanie rozpoznać), ex (obliczana na podstawie wysokości małej litery x w użytej

czcionce). Jednostki względne są stosowane, gdy odwołujemy się do rozmiaru innego elementu.

- **Jednostki czasu.** Mogą być liczone w sekundach (s) lub milisekundach (ms).
- **Ciągi znaków.** Muszą być umieszczone pomiędzy pojedynczym lub podwójnym cudzysłowem, np. "ciąg znaków".
- **URL** — np. url(zdjecia/obraz.jpg). Błądem jest wstawienie spacji przed nawiasem otwierającym.

W języku HTML przyjmuje się, że wartości liczbowe bez podanych jednostek mają jednostkę px. Natomiast w języku CSS brak jednostki jest traktowany jako błąd. Niektóre przeglądarki ignorują ten błąd i przyjmują jednostkę px. Gdy podana wartość jest równa 0, jednostka nie ma znaczenia.

### 3.2.2. Kolory

W języku HTML zostało zdefiniowanych 16 podstawowych kolorów. W wersji CSS 2.2 oprócz 16 podstawowych kolorów zdefiniowano dodatkowy kolor podstawowy, orange (#FFA500) — pomarańczowy. W CSS3 zostało zdefiniowanych dodatkowo 130 kolorów rozszerzonych.

Niektóre polecenia arkuszy stylów zawierają definicję kolorów, np.:

```
body {background-color: #c0c0c0;}  
h3 {color: #f0f0f0;}  
<p style= "background-color: yellow";>Podróż dookoła świata</p>  
h2 {color: green; background-color: #c0c0c0;}
```

Wartość dla atrybutu color lub background-color może być podawana na trzy sposoby. Można stosować słowną angielską nazwę koloru (blue, red), wprowadzić odpowiednik koloru zapisany w kodzie heksadecymalnym (#ffcc00, #aa2237) lub w kodzie RGB (rgb(120, 12, 270), rgb(39, 21, 110)).

Znak # przed kodem koloru oznacza, że kolor jest definiowany właśnie w kodzie heksadecymalnym. Kod musi się składać z sześciu lub trzech cyfr, nawet jeśli są to same zera. Im większa jest wartość liczbową kodu, tym jaśniejszy kolor.

W modelu RGB kod koloru zawiera trzy składowe zapisane w postaci liczb oddzielonych przecinkiem. Napis rgb przed kodem koloru oznacza definiowanie koloru w kodzie RGB.

Poza kolorami zapisywanyimi w systemie szesnastkowym można stosować nazwy kolorów. Lista dostępnych nazw kolorów jest ograniczona.

Paleta podstawowych kolorów zgodna ze specyfikacją języków HTML i XHTML została podana w tabeli 3.1.

**Tabela 3.1.** Paleta kolorów

Kolor	Nazwa	Kod HEX	Kod RGB
czarny	black	#000000	rgb(0,0,0)
srebrny	silver	#c0c0c0	rgb(192,192,192)
szary	gray	#808080	rgb(128,128,128)
biały	white	#ffffff	rgb(255,255,255)
kasztanowy	maroon	#800000	rgb(128,0,0)
czerwony	red	#ff0000	rgb(255,0,0)
purpurowy	purple	#800080	rgb(128,0,128)
fuksja	fuchsia	#ff00ff	rgb(255,0,255)
zielony	green	#008000	rgb(0,128,0)
limonkowy	lime	#00ffff	rgb(0,255,0)
oliwkowy	olive	#808000	rgb(128,128,0)
żółty	yellow	#ffff00	rgb(255,255,0)
granatowy	navy	#000080	rgb(0,0,128)
niebieski	blue	#0000ff	rgb(0,0,255)
morski	teal	#008080	rgb(0,128,128)
akwamaryna	aqua	#00ffff	rgb(0,255,255)

Więcej informacji o barwach można znaleźć m.in. na stronie [www.barwy.net](http://www.barwy.net).

### Ćwiczenie 3.4

Przeanalizuj sposób kodowania kolorów i odpowiedz na pytanie: Ile maksymalnie kolorów można uzyskać, stosując do definiowania koloru kod heksadecymalny?

Przykłady podawania wartości kolorów w CSS3:

1. *Predefiniowane* — podawanie kolorów w postaci ich nazw.

```
p {background-color: orange;}
```

2. *#rrggbb* — podawanie wartości każdej składowej kolorów podstawowych (rr — czerwony, gg — zielony, bb — niebieski). Wszystkie liczby są dwucyfrowe i zapisane w systemie szesnastkowym, np. #ff00cc.

```
p {background-color: #ff0000;}
```

3. *#rgb* — jako składowe kolorów trzeba podawać liczby jednocyfrowe w systemie szesnastkowym, np. #f0c.

```
p {background-color: #f00;}
```

- 4.** `rgb(R, G, B)` — można podać oddzielnie każdą składową koloru w systemie dziesiętnym.

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

- 5.** `rgb(R%, G%, B%)` — można podać oddzielnie każdą składową koloru w postaci wartości procentowej.

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

- 6.** *28 kolorów systemowych* — kolory pochodzą z systemu operacyjnego (kolor pulpitu, okien, menu). Gdy użytkownik zmieni kolory w swoim systemie, zmieniają się również kolory zdefiniowane tą metodą. Sama deklaracja wygląda tak samo jak w punkcie 1.

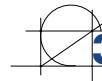
- 7.** `hsl(H, S%, L%)` — odmienny od RGB sposób opisu kolorów. *H (Hue)* określa barwę, *S (Saturation)* — nasycenie, *L (Lightness)* — jasność. Kolor powstaje przez wybranie określonej barwy (*H*), a następnie sterowanie nasyceniem (*S*) i jasnością (*L*) aż do uzyskania pożądanego efektu.

```
p {background-color: hsl(120,65%,75%);}
```

- 8.** `rgba(R, G, B, A)`, `rgba(R%, G%, B%, A%)`, `hsla(H, S%, L%, A)` — uzyskiwanie koloru z przezroczystością. *A (alpha)* określa wartość przezroczystości z zakresu od 0 (całkowita przezroczystość) do 1 (brak przezroczystości).

```
p {background-color: rgba(255,0,0,0.5);}
```

```
p {background-color: hsla(120,65%,75%,0.3);}
```



## 3.3. Selektory

Selektorem może być dowolny element języka HTML, dla którego chcemy zdefiniować parametry formatowania. W zależności od tego, w jaki sposób odwołujemy się w definicji reguły do formatowanych elementów, wyróżniamy następujące rodzaje selektorów:

- selektory elementów,
- selektory atrybutów,
- selektory specjalne,
- selektory pseudoklas,
- selektory pseudoelementów.

### 3.3.1. Selektory elementów

Selektory elementów wybierają elementy na podstawie ich nazwy. Składnia dla selektorów elementów to podana już podstawowa składnia języka CSS:

```
selektor {właściwość: wartość;}
```

## Selektor typu elementu

Jest to podstawowy typ selektora. Służy do definiowania formatowania znaczników występujących na stronie (np.: p, h3, table, img).

### Przykład 3.13

```
p {color: blue; font-size: 4pt;}  
h3 {color: green; font-size: 24pt;}
```

## Selektor uniwersalny

Selektor uniwersalny (inaczej: ogólny) to selektor pasujący do wszystkich znaczników. Jest oznaczany gwiazdką \*. Składnia tego selektora to:

```
* {właściwość: wartość;}
```

Zamiast grupować elementy, np.:

```
p, h1, h2, h3, h4, table {font-family: courier new; color: green;}
```

można użyć selektora uniwersalnego:

```
* {font-family: courier new; color: green;}
```

## Selektor potomka

Przy użyciu selektora potomka można formatować elementy, które są zawarte wewnątrz innych znaczników, czyli leżą niżej w hierarchii drzewa dokumentu. Składnia selektora potomka to:

```
selektor selektor {właściwość: wartość;}
```

Jeżeli w dokumencie HTML wewnątrz znacznika znajduje się inny znacznik, np. wewnątrz znacznika <div> został umieszczony znacznik <span> w podanej postaci:

```
<div>  
    Wybieramy się w podróż dookoła świata.  
    <p>Będziemy odwiedzali po kolej wszystkie <span>kontynenty</span>.</p>  
</div>
```

to definicja stylu dla selektora potomka będzie wyglądać następująco:

```
div span {font-size: 14pt; color: blue;}
```

W tym przypadku nie ma znaczenia, czy znacznik <span> znajduje się bezpośrednio w znaczniku <div>, czy w kolejnych znacznikach zawartych w jego wnętrzu.

### Przykład 3.14

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Selektor potomka</title>
    <meta charset="UTF-8">
    <style>
        body {background-color: grey;}
        div span {font-size: 14pt; color: blue;}
    </style>
</head>
<body>
    <div>
        Wybieramy się w podróż dookoła <span>świata</span>.
        <p>Będziemy odwiedzali po kolej wszystkie <span>kontynenty</span>. </p>
    </div>
    <p>Zwiedzanie Europy rozpoczniemy od <span>Madrytu</span>. </p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 3.9.

Wybieramy się w podróż dookoła świata.  
Będziemy odwiedzali po kolej wszystkie kontynenty.  
Zwiedzanie Europy rozpoczniemy od Madrytu.

**Rysunek 3.9.** Działanie selektora potomka

Jak widać, potomek nie musi być umieszczony bezpośrednio w znaczniku, którego jest potomkiem. W powyższym przykładzie w znaczniku `<div>` znajduje się znacznik `<p>`, a dopiero w nim znacznik `<span>`, którego dotyczy definiowany styl.

### Selektor dziecka

Selektor dziecka służy do definiowania formatowania elementów, które znajdują się o jeden rząd niżej w hierarchii drzewa dokumentu. Ma on postać:

`rodzic > dziecko {właściwość: wartość;}`

gdzie symbol `>` oznacza bezpośredni związek między elementami.

Znacznik będący dzieckiem musi wystąpić bezpośrednio wewnątrz znacznika nadzędnego.

### Przykład 3.15

```
<!DOCTYPE html>

<html lang="pl-PL" >

<head>

    <title>Selektor dziecka</title>

    <meta charset="UTF-8">

    <style>
        body {background-color: #c0c0c0;}
        div > span {font-size: 14pt; color: green;}
    </style>

</head>

<body>

    <div>

        Wybieramy się w podróż dookoła świata.

        <p>Będziemy odwiedzali po kolej wszystkie kontynenty.</p>

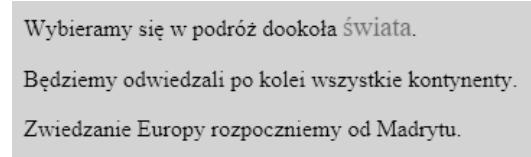
    </div>

    <p>Zwiedzanie Europy rozpoczniemy od Madrytu.</p>

</body>

</html>
```

Wynik interpretacji kodu został zilustrowany na rysunku 3.10.



The screenshot shows a single 

element containing two 

elements. The first 

element contains the text "Wybieramy się w podróż dookoła świata." with a green "ś". The second 

element contains the text "Będziemy odwiedzali po kolej wszystkie kontynenty." with a green "B". Below these, a third 

element contains the text "Zwiedzanie Europy rozpoczniemy od Madrytu." with a green "Z". All other text and punctuation are black.

**Rysunek 3.10.** Działanie selektora dziecka

Kolor *zielony* i wielkość liter *14pt* zostały nadane tylko tekstowi otoczonemu znacznikiem , który został umieszczony bezpośrednio w bloku 

. Znacznik  nieumieszczony bezpośrednio w znaczniku 

nie został objęty zdefiniowanym stylem.

125

## Selektor sąsiadującego brata

Dla elementów znajdujących się w tym samym rzędzie hierarchii można zdefiniować selektor sąsiadującego brata. Umożliwia on nadanie drugiemu z sąsiadujących elementów zdefiniowanych atrybutów formatowania. Selektor sąsiadującego brata przyjmuje postać:

```
brat1 + brat2 {właściwość: wartość;}
```

gdzie symbol + oznacza następowanie po sobie elementów.

### Przykład 3.16

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Selektor sąsiadującego brata</title>
    <meta charset="UTF-8">
    <style>
        body {background-color: #c0c0c0;}
        div + p {font-size: 16pt; color: red;}
    </style>
</head>
<body>
    <div>
        Wybieramy się w podróż dookoła <span>świata</span>.
        <p>Będziemy odwiedzali po kolej wszystkie <span>kontynenty</span>.</p>
    </div>
    <p>Zwiedzanie Europy rozpoczniemy od <span>Madrytu</span>.</p>
    <p>Zwiedzanie Azji rozpoczniemy od Pekinu.</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 3.11.

#### Rysunek 3.11.

Działanie selektora  
sąsiadującego brata

Wybieramy się w podróż dookoła świata.  
Będziemy odwiedzali po kolej wszystkie kontynenty.  
Zwiedzanie Europy rozpoczniemy od Madrytu.  
Zwiedzanie Azji rozpoczniemy od Pekinu.

W powyższym przykładzie tylko znacznik `<div>` oraz przedostatni znacznik `<p>` znajdują się w tym samym rzędzie hierarchii. Przedostatni znacznik `<p>` występuje

bezpośrednio po znaczniku `<div>` i tylko dla tekstu otoczonego tym znacznikiem `<p>` jest realizowane formatowanie zdefiniowane dla selektora sąsiadującego brata.

## Selektor braci

Dla elementów znajdujących się w tym samym rzędzie hierarchii można zdefiniować selektor braci. Umożliwia on nadanie drugiemu i wszystkim kolejnym sąsiadującym elementom zdefiniowanych atrybutów formatowania. Selektor braci przyjmuje postać:

```
brat1 ~ brat2 {właściwość: wartość;}
```

### Przykład 3.17

```
<!DOCTYPE html>

<html lang="pl-PL">
<head>
    <title>Selektor braci</title>
    <meta charset="UTF-8">
    <style>
        body {background-color: #c0c0c0;}
        div ~ p {font-size: 16pt; color: red;}
    </style>
</head>
<body>
    <div>
        Wybieramy się w podróż dookoła świata</span>.
        <p>Będziemy odwiedzali po kolej wszystkie kontynenty</span>.</p>
    </div>
    <p>Zwiedzanie Europy rozpoczniemy od Madrytu.</p>
    <p>Zwiedzanie Azji rozpoczniemy od Pekinu.</p>
</body>
</html>
```

Wynik wykonania kodu został zilustrowany na rysunku 3.12.

#### Rysunek 3.12.

Działanie selektora braci

Wybieramy się w podróż dookoła świata.

Będziemy odwiedzali po kolej wszystkie kontynenty.

Zwiedzanie Europy rozpoczniemy od Madrytu.

Zwiedzanie Azji rozpoczniemy od Pekinu.

W powyższym przykładzie znacznik `<div>` oraz dwa ostatnie znaczniki `<p>` znajdują się w tym samym rzędzie hierarchii. Dlatego dla tekstu otoczonego tymi znacznikami `<p>` jest realizowane formatowanie zdefiniowane dla selektora braci.

### Ćwiczenie 3.5

Zastanów się, jak — wykorzystując arkusze stylów oraz selektory potomka, dziecka i braci — należy zdefiniować style, które pozwolą na wyświetlenie tekstów sformatowanych tak jak na rysunku 3.13. Przygotuj stronę, na której uzyskasz podobny rezultat.

**Ten tekst jest selektorem potomka. Zdefiniuj to w arkuszu stylów.**

**Ten tekst jest selektorem dziecka. Zdefiniuj to w arkuszu stylów.**

**Ten tekst jest selektorem braci. Zdefiniuj to w arkuszu stylów.**

**Rysunek 3.13.** Wynik działania różnych selektorów

### 3.3.2. Selektory atrybutów

W języku CSS można formatować znaczniki na podstawie atrybutów, które one mają. Na przykład jeżeli dla znacznika `<p>` (akapit) zostanie zdefiniowany atrybut `id`, to inaczej będzie wyświetlany akapit z `id="lewy"`, a inaczej akapit z `id="prawy"`.

Postać selektora atrybutów:

```
selektor[atribut="wartość atrybutu"] {właściwość: wartość;}
```

#### Przykład 3.18

```
p[id="zolty"] {font-size: 16pt; color: yellow;}
```

#### Prosty selektor atrybutu

Prosty selektor atrybutu jest wykorzystywany dla elementów, którym został nadany określony atrybut, ale jego wartość nie ma znaczenia. Ma on postać:

```
selektor[atribut] {właściwość: wartość;}
```

lub

```
[atribut] {właściwość: wartość;}
```

#### Przykład 3.19

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Selektor atrybutu</title>
```

```

<meta charset="UTF-8">
<style>
body {background-color: #c0c0c0;}
p[id] {font-size: 14pt; color: green;}
</style>
</head>
<body>


Wybieramy się w podróż dookoła świata.



Będziemy odwiedzali po kolej wszystkie kontynenty.



Zwiedzanie Europy rozpoczniemy od Madrytu.


</body>
</html>

```

Wynik interpretacji kodu został pokazany na rysunku 3.14.

Wybieramy się w podróż dookoła świata.

Będziemy odwiedzali po kolej wszystkie kontynenty.

Zwiedzanie Europy rozpoczniemy od Madrytu.

**Rysunek 3.14.** Działanie prostego selektora atrybutu

W powyższym przykładzie dla akapitów pierwszego i drugiego został zdefiniowany atrybut `id` o różnych wartościach. Niezależnie od wartości tego atrybutu akapity zostały sformatowane zgodnie z definicją stylu dla znacznika `<p>`. Ponieważ trzeci akapit nie ma zdefiniowanego atrybutu `id`, nie został sformatowany.

## Selektor atrybutu o określonej wartości

Ten rodzaj selektora określa formatowanie, gdy atrybut ma określoną wartość. Ma on postać:

```
selektor[atribut="wartość"] {właściwość: wartość;
```

lub

```
[atribut="wartość"] {właściwość: wartość;
```

### Przykład 3.20

```

<!DOCTYPE html>
<html lang="pl-PL">
<head>

```

```
<title>Selektor atrybutu</title>
<meta charset="UTF-8">
<style>
body {background-color: #c0c0c0;}
p[id="center"] {font-size: 14pt; color: blue;}
</style>
</head>
<body>


Wybieramy się w podróż dookoła świata.



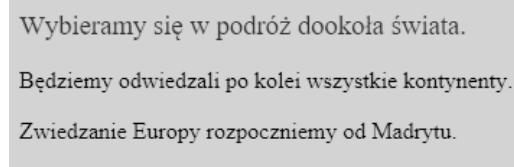
Będziemy odwiedzali po kolej wszystkie kontynenty.



Zwiedzanie Europy rozpoczniemy od Madrytu.


</body>
</html>
```

Wynik wykonania kodu jest widoczny na rysunku 3.15.



Wybieramy się w podróż dookoła świata.  
Będziemy odwiedzali po kolej wszystkie kontynenty.  
Zwiedzanie Europy rozpoczniemy od Madrytu.

#### Rysunek 3.15. Działanie selektora atrybutu o określonej wartości

W tym przykładzie tylko pierwszy akapit został sformatowany zgodnie z definicją stylów, ponieważ tylko tu został zdefiniowany atrybut `id` z wartością `center`.

### Selektor atrybutu zawierającego określony wyraz

Ten rodzaj selektora może zostać wykorzystany, gdy wartość atrybutu składa się z kilku wyrazów. Wystarczy, że w wartości atrybutu wystąpi podany wyraz, aby dany element został odpowiednio sformatowany. Postać selektora to:

```
selektor[atribut~="wyraz"] {właściwość: wartość;}
```

lub

```
[atribut~="wyraz"] {właściwość: wartość;}
```

#### Przykład 3.21

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
<title>Selektor atrybutu</title>
```

```

<meta charset="UTF-8">
<style>
body {background-color: #c0c0c0;}
p[title~="jest"] {font-size: 14pt; color: green;}
</style>
</head>
<body>


Wybieramy się w podróż dookoła świata.



Będziemy odwiedzali po kolej wszystkie kontynenty.



Zwiedzanie Europy rozpoczniemy od Madrytu.


</body>
</html>

```

Wynik interpretacji kodu został pokazany na rysunku 3.16.

Wybieramy się w podróż dookoła świata.

Będziemy odwiedzali po kolej wszystkie kontynenty.

Zwiedzanie Europy rozpoczniemy od Madrytu.

**Rysunek 3.16.** Działanie selektora atrybutu zawierającego określony wyraz

### Ćwiczenie 3.6

Wykorzystując arkusze stylów oraz selektory atrybutów, zdefiniuj style, które pozwalą na wyświetlenie tekstów umieszczanych na stronie i zapisanych w akapicie w różnych kolorach w zależności od wyrównania tych tekstów na stronie (np. dla tekstów wyrownanych do lewej będzie to kolor niebieski, dla tekstów wyśrodkowanych — kolor zielony, a dla tekstów wyrównanych do prawej — kolor czerwony). Natomiast dla tekstów umieszczonych w znaczniku `<h2>` kolory będą różne w zależności od zastosowanej wielkości czcionki (np. czcionka o rozmiarze 16pt — kolor żółty, czcionka o rozmiarze 20pt — kolor granatowy).

## 3.3.3. Selektory specjalne

### Selektor klasy

Selektory pozwalają na nadanie określonych atrybutów takim samym elementom występującym na stronie. Ale czasami chcemy zastosować pewien styl formatowania do jednej grupy elementów, a do drugiej inny. Możemy wtedy dla każdej grupy utworzyć klasę o dowolnej nazwie i zdefiniować styl dla selektora klasy.

Selektor klasy ma postać:

```
selektor.nazwa_klasy {właściwość: wartość;}
```

gdzie *nazwa\_klasy* to dowolna nazwa.

Nazwa klasy to pojedynczy wyraz, który musi spełniać kilka warunków:

- Musi zaczynać się od kropki.
- Mogą być użyte litery, cyfry, znak podkreślenia, łącznik.
- Po kropce musi zostać wpisana litera.
- Rozróżniane są małe i wielkie litery.
- Stosowanie polskich znaków nie jest zalecane.
- Nie może zawierać spacji.

### Przykład 3.22

```
p.tekst1 {color: red; font-size: 24pt;}  
p.tekst2 {color: green; font-size: 20pt;}  
p.tekst3 {font-family: courier; color: blue; font-size: 16pt;}
```

Odwołanie do klasy w dokumencie HTML ma postać:

```
<znacznik class="nazwa_klasy"> ... </znacznik>
```

W odwołaniu nazwa klasy jest wartością atrybutu `class`. W nazwie klasy podowanej jako wartość atrybutu nie wstawia się kropki.

### Przykład 3.23

```
<!DOCTYPE html>  
<html lang="pl-PL">  
<head>  
    <title>Selektor klasy</title>  
    <meta charset="UTF-8">  
    <style>  
        body {background-color: #c0c0c0;}  
        p.tekst1 {color: red; font-size: 20pt;}  
        p.tekst2 {color: green; font-size: 16pt;}  
        p.tekst3 {font-family: courier; color: blue; font-size: 14pt;}  
    </style>  
</head>  
<body>  
    <p class="tekst1">Wybieramy się w podróż dookoła świata.</p>
```

```

<p class="tekst2">Będziemy odwiedzali po kolej wszystkie kontynenty.</p>
<p class="tekst3">Zwiedzanie Europy rozpoczniemy od Madrytu.</p>
<p class="tekst2">Zwiedzanie Azji rozpoczniemy od Pekinu.</p>
</body>
</html>

```

Wynik interpretacji kodu został zilustrowany na rysunku 3.17.

Wybieramy się w podróż dookoła świata.

Będziemy odwiedzali po kolej wszystkie kontynenty.

Zwiedzanie Europy rozpoczniemy od Madrytu.

Zwiedzanie Azji rozpoczniemy od Pekinu.

**Rysunek 3.17.** Działanie selektora klasy

Możliwość tworzenia klas i definiowania dla nich odpowiednich stylów jest jedną z podstawowych właściwości kaskadowych arkuszy stylów. Wykorzystanie tego narzędzia znacznie przyspiesza pracę podczas definiowania stylów dla witryny internetowej. Już podczas projektowania witryny powinno się określić, jakie grupy elementów będą w niej występowały, i dla wszystkich powtarzalnych elementów utworzyć klasy, a następnie umieścić deklaracje tych klas w zewnętrznym arkuszu stylów. Przy ewentualnej modyfikacji wyglądu witryny wystarczy zmodyfikować deklaracje odpowiednich klas w arkuszu stylów.

## Uniwersalny selektor klasy

Zdarza się, że definiując klasę, nie określamy, jakich znaczników będzie ona dotyczyła. Tak zdefiniowana klasa może zostać użyta do sformatowania dowolnego elementu bez względu na typ znacznika. Ma ona postać:

```
.nazwa_klasy {właściwość: wartość;}
```

Wszystkie elementy, którym nadamy klasę o podanej nazwie, zostaną sformatowane zgodnie z definicją stylu podaną dla tej klasy.

### Przykład 3.24

```

<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Selektor klasy</title>
    <meta charset="UTF-8">

```

```

<style>
  body {background-color: #c0c0c0;}
  .nad {color: blue; font-size: 16pt;}
</style>

</head>
<body>
  <p class="nad">Wybieramy się w podróż dookoła świata.</p>
  <h2 class="nad">Będziemy odwiedzali po kolej wszystkie kontynenty.</h2>
  Zwiedzanie<span class="nad"> Europy</span> rozpoczęliśmy od Madrytu.
</body>
</html>

```

Wynik interpretacji kodu został przedstawiony na rysunku 3.18.

Wybieramy się w podróż dookoła świata.

**Będziemy odwiedzali po kolej wszystkie kontynenty.**

Zwiedzanie Europy rozpoczęliśmy od Madrytu.

**Rysunek 3.18.** Efekt zastosowania klasy do formatowania różnych rodzajów znaczników

Deklaracja klasy jest przydatna zwłaszcza wtedy, gdy w różnych miejscach witryny mają występować elementy o takich samych parametrach formatowania, lecz nie możemy (lub nie chcemy) przy ich definiowaniu posługiwać się selektorem typu.

## Selektor identyfikatora

Selektor identyfikatora jest stosowany, gdy chcemy nadać określone atrybuty formatowania elementowi, który ma przypisany jednoznaczny identyfikator, czyli występuje raz — w odróżnieniu od klasy.

Selektor identyfikatora ma postać:

`selektor#identyfikator {właściwość: wartość;}`

lub

`#identyfikator {właściwość: wartość;}`

Deklaracja identyfikatora musi się rozpoczynać znakiem #.

## Przykład 3.25

`h3#nag13 {color: red;}`

W ten sposób znacznikowi `<h3>` o id równym `nagl3` nadano kolor czerwony. Odwołanie w kodzie do identyfikatora ma postać:

```
<h3 id="nagl3">Podróż dookoła świata</h3>
```

### Przykład 3.26

```
<!DOCTYPE html>

<html lang="pl-PL">
<head>
    <title>Selektor identyfikatora</title>
    <meta charset="UTF-8">
    <style>
        body {background-color: #c0c0c0;}
        #tekst {font-size: 16pt; color: green;}
    </style>
</head>
<body>
    <p id="tekst">Wybieramy się w podróż dookoła świata.</p>
    <p>Będziemy odwiedzali po kolej wszystkie kontynenty.</p>
    <p>Zwiedzanie Europy rozpoczniemy od Madrytu.</p>
</body>
</html>
```

Wynik wykonania kodu został pokazany na rysunku 3.19.

```
Wybieramy się w podróż dookoła świata.  
Będziemy odwiedzali po kolej wszystkie kontynenty.  
Zwiedzanie Europy rozpoczniemy od Madrytu.
```

**Rysunek 3.19.** Efekt zastosowania identyfikatora

## 3.3.4. Pseudoklasy

W języku CSS style są dodawane do elementów lub grup elementów na podstawie ich nazw, atrybutów lub zawartości.

Jest też inna możliwość dodawania stylu. Element nabywa styl lub traci go w związku z działaniem użytkownika lub zmienia się w zależności od umiejscowienia. Przykładem takiego zachowania są linki (*odsyłacze*), które zmieniają swój wygląd po najechaniu

na nie kursorem myszy, ich kliknięciu lub otwarciu linku. Element otrzymuje lub traci pseudoklasę podczas interakcji z użytkownikiem. Do takiej dynamicznej zmiany stylu elementu służą **pseudoklasy**.

Odsyłacze (linki) mają właściwości, które określają działanie użytkownika. Są to:

- `:link` — link nieaktywny, nie został przez użytkownika odwiedzony;
- `:visited` — link odwiedzony, strona była otwierana;
- `:hover` — link gotowy do kliknięcia, kursor myszy ustawiony nad linkiem;
- `:active` — link odwiedzany, strona jest obecnie wczytana.

### Odsyłacz podstawowy (pseudoklasa `:link`)

Odsyłacz podstawowy nadaje atrybuty formatowania wszystkim jeszcze nieodwiedzonym odnośnikom.

```
a:link {właściwość: wartość;}
```

Po zadeklarowaniu klasy można ustawić atrybuty formatowania dla wybranych odsyłaczy, które nie były jeszcze odwiedzane.

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

Przy takim zapisie atrybuty formatowania zostaną nadane tylko tym nieodwiedzonym odsyłaczom, które należą do podanej klasy.

### Przykład 3.27

```
a:link {color: green; background: yellow;}  
a.nowa:link {color: orange;}
```

### Odsyłacz odwiedzony (pseudoklasa `:visited`)

Odsyłacz odwiedzony nadaje atrybuty formatowania odsyłaczowi, który był otwierany, a informacja o tym została umieszczona w pamięci przeglądarki.

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

Z deklaracją klasy ustawiane są atrybuty formatowania dla odsyłaczy wybranej klasy, które były otwierane.

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

### Przykład 3.28

```
a:visited {color: red;}  
a.po_nowa:visited {color: red;}
```

## Wskazanie kursorem myszy (pseudoklasa :hover)

Nadaje atrybuty formatowania odsyłaczowi, gdy myszka znajduje się nad nim, ale nie aktywuje go.

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

Z deklaracją klasy ustawiane są atrybuty formatowania dla odsyłaczy wybranej klasy, które są w danej chwili wskazane myszą.

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

### Przykład 3.29

```
a:hover {color: yellow;}  
a.po_nowa:hover {color: yellow;}
```

#### UWAGA

Pseudoklasa :hover może być używana do definiowania stylów elementów innych niż odsyłacze.

### Przykład 3.30

```
<!DOCTYPE html>  
<html lang="pl-PL">  
<head>  
    <title>Pseudoklasa :hover</title>  
    <meta charset="UTF-8">  
    <style>  
        body {background-color: #c0c0c0;}  
        .blok {height: 25px; width: 100px; background: aqua; color: black;}  
        .blok:hover {background: yellow; color: red;}  
    </style>  
</head>  
<body>  
    <div class="blok">Europa</div>  
</body>  
</html>
```

Przykład pokazuje zastosowanie pseudoklasy :hover dla znacznika <div>.

## Odsyłacz aktywny (pseudoklasa :active)

Odsyłacz aktywny nadaje atrybuty formatowania odsyłaczowi, który w danej chwili jest aktywny, np. gdy użytkownik naciśnie i przytrzyma lewy przycisk myszy nad obszarem wybranego elementu, aż do momentu, gdy przycisk ten zostanie puszczyony.

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

Z deklaracją klasy ustawiane są atrybuty formatowania dla odsyłaczy wybranej klasy, które są w danej chwili aktywne.

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

### Przykład 3.31

```
a:active {color: blue;}  
a.po_nowa:active {color: blue;}
```

Kolejność deklarowania w arkuszu stylów pseudoklas przypisanych do odsyłaczy ma znaczenie dla ich prawidłowego działania.

Poprawna kolejność deklarowania pseudoklas:

```
a:link {właściwość: wartość;}  
a:visited {właściwość: wartość;}  
a:hover {właściwość: wartość;}  
a:active {właściwość: wartość;}
```

### Przykład 3.32

```
<!DOCTYPE html>  
<html lang="pl-PL">  
<head>  
    <title>Pseudoklasa odsyłacz</title>  
    <meta charset="UTF-8">  
    <style>  
        body {background-color: #c0c0c0;}  
        a:link {color: green;}  
        a:visited {color: red;}  
        a:hover {color: yellow;}  
        a:active {color: blue;}  
    </style>  
</head>  
<body>
```

```
<p>Link do strony wydawnictwa: <a href="http://helion.pl/">HELION</a></p>
</body>
</html>
```

Podany kod działa następująco: odnośnik znajdujący się pod tekstem HELION po otwarciu strony jest wyświetlany w kolorze zielonym (pseudoklasa `:link`), po najechaniu myszą na tekst jego kolor zmienia się na żółty (pseudoklasa `:hover`), po wciśnięciu i przytrzymaniu klawisza myszy kolor tekstu zmienia się na niebieski (pseudoklasa `:active`), a po odwiedzeniu strony kolor tekstu zmieni się na czerwony (pseudoklasa `:visited`).

### Pseudoklasa `:focus`

Nadaje atrybuty formatowania odsyłaczowi (wcześniej wybranemu) lub polu formularza, na którym został ustawiony kurSOR. Na przykład wtedy, gdy odsyłacz został wybrany za pośrednictwem klawisza `Tab` lub w polu formularza znalazła się kurSOR.

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

### Ćwiczenie 3.7

Wykorzystując arkusze stylów, zdefiniuj pseudoklasy, które umożliwiają dynamiczną zmianę stylu linków znajdujących się na stronie internetowej zaprojektowanej w poprzednich ćwiczeniach. Link podstawowy powinien mieć kolor czerwony, odwiedzany — kolor zielony, wskazany przez kurSOR myszy — kolor pomarańczowy, a link aktywny — kolor żółty.

## 3.3.5. Selektory pseudoelementów

W języku HTML nie ma mechanizmów dostępu do takich elementów strony jak pierwsza litera lub pierwsza linia akapitu bez otaczania ich znacznikami. Do formatowania tych specjalnych elementów można wykorzystać **pseudoelementy** dostępne w języku CSS.

### Pierwsza linia (`:first-line`)

Pseudoelement `:first-line` nadaje określone formatowanie wszystkim pierwszym liniom znacznika, do którego odnosi się selektor.

```
selektor:first-line {właściwość: wartość;}
```

Selektorem może być dowolny znacznik języka HTML.

### Przykład 3.33

```
p:first-line {color: red; font-size: 16pt;}
```

## Pierwsza litera (:first-letter)

Pseudoelement :first-letter nadaje odrębne formatowanie pierwszej literze np. akapitu. Służy głównie do poprawienia wyglądu tekstu poprzez zaprojektowanie ozdobnej pierwszej litery (inicjału).

```
selektor:first-letter {właściwość: wartość;}
```

### Przykład 3.34

```
p:first-letter {font-family: arial; font-size: 24pt; color: blue;}
```

## 3.3.6. Grupowanie selektorów

Jeżeli w arkuszu stylów zostały zdefiniowane elementy o tym samym stylu, to można te elementy połączyć w grupy i nadać im styl poprzez wspólną deklarację.

### Przykład 3.35

```
p {color: green;}  
h2 {color: green;}  
h3 {color: green;}
```

Ten ciąg deklaracji można zastąpić jedną wspólną definicją stylu.

### Przykład 3.36

```
p, h2, h3 {color: green;}
```

Kolejne selektory przypisane do stylu powinny zostać oddzielone przecinkami.

## 3.4. Właściwości elementów

W kaskadowych arkuszach stylów formatowanie elementów strony jest realizowane przez ustawianie właściwości tych elementów.

### 3.4.1. Czcionki

Czcionki są najistotniejszym elementem strony, który podlega formatowaniu. Za pomocą właściwości czcionki można określić rodzinę czcionki (font-family), jej styl (font-style), rozmiar (font-size) i pogrubienie (font-weight).

#### Rodzaj czcionki

Ze względu na to, że nie wszystkie czcionki są dostępne na komputerze użytkownika, przy wyborze rodzaju czcionki określa się jej preferowany typ. Można również zdefiniować kilka rodzajów czcionek. Jeżeli na komputerze użytkownika nie ma pierwszego rodzaju czcionki, to z listy zostanie wybrany kolejny.

Rodzaj czcionki ustala się za pomocą atrybutu `font-family`.

```
selektor {font-family: rodzaj1, rodzaj2, rodzaj3,...}
```

Definiując rodzaj czcionki, możemy podać kilka nazw własnych (np. *Arial*) oraz nazwy rodzin (np. *serif*). Podczas renderowania strony przeglądarka spróbuje użyć pierwszej z czcionek. Jeżeli nie zostanie ona znaleziona na komputerze użytkownika, zostanie podjęta próba z kolejnymi czcionkami. Przy wprowadzaniu kilku rodzajów czcionek należy ich nazwy oddzielić przecinkiem i spacją. Jeżeli nazwa czcionki składa się z kilku wyrazów, całą nazwę ujmujemy w apostrofy, np. 'Times New Roman'.

Zamiast nazwy czcionki można podać nazwę rodziny czcionek. Rozwiążanie to warto stosować równolegle z nazwami własnymi. Dzięki temu w przypadku gdy przeglądarka nie znajdzie żadnej z zadeklarowanych czcionek, zostanie podjęta próba dopasowania innej czcionki z danej rodziny.

Nazwy rodzin czcionek to:

- `serif` — czcionki szeryfowe, mające wykończenie liter i cyfr. Do tej grupy należą m.in.: Times New Roman, Georgia, Bodoni.
- `sans-serif` — czcionki bezszeryfowe. Końcówki znaków są proste; np.: Arial, Verdana, Futura.
- `monospace` — czcionki monotypiczne. Znaki mają stałą szerokość; np. Courier, Courier New.
- `cursive` — czcionki mające cechy czcionki pochyłej. Znaki wyglądają jak pisane ręcznie; np. Comic Sans, Florence.
- `fantasy` — czcionki dekoracyjne; np. Impact, OldTown.

### Przykład 3.37

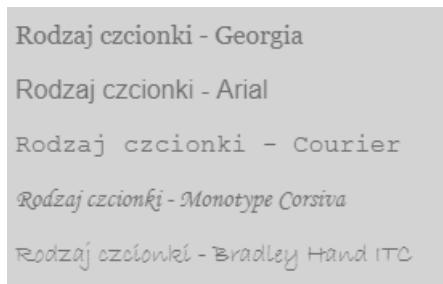
```
<p style="font-family: Georgia, Courier, Arial">Arkusze stylów</p>
<p style="font-family: Georgia, Courier, serif">Arkusze stylów</p>
```

### Przykład 3.38

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Rodziny czcionek</title>
    <meta charset="UTF-8">
    <style>
        body {background-color: #c0c0c0;}
        p {font-size: 18pt; color: green;}
    </style>
```

```
</head>  
  
<body>  
    <p style="font-family: Georgia;">Rodzaj czcionki - Georgia</p>  
    <p style="font-family: Arial;">Rodzaj czcionki - Arial</p>  
    <p style="font-family: Courier;">Rodzaj czcionki - Courier</p>  
    <p style="font-family: 'Monotype Corsiva';">Rodzaj czcionki - Monotype  
Corsiva</p>  
    <p style="font-family: 'Bradley Hand ITC';">Rodzaj czcionki - Bradley Hand  
ITC</p>  
</body>  
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 3.20.



**Rysunek 3.20.** Zastosowanie różnych rodzajów czcionek

## Rozmiar czcionki

Wielkość czcionki jest definiowana w atrybucie `font-size`.

```
selektor {font-size: rozmiar;}
```

Wielkość czcionki można ustawać na cztery sposoby:

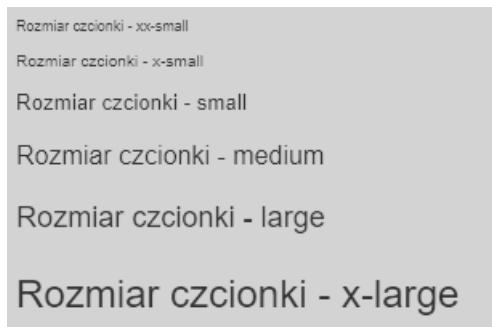
**1. Według słów kluczowych**, które pozwalają określić rozmiar czcionki względem rozmiaru podstawowego. W języku CSS występuje siedem słów kluczowych:

- `xx-small` — czcionka najmniejsza,
- `x-small` — czcionka mniejsza,
- `small` — czcionka mała,
- `medium` — czcionka średnia (jest to podstawowy rozmiar czcionki),
- `large` — czcionka duża,
- `x-large` — czcionka większa,
- `xx-large` — czcionka największa.

### Przykład 3.39

```
<!DOCTYPE html>
<html lang="pl-PL" >
<head>
    <title>Rozmiar czcionki</title>
    <meta charset="UTF-8">
    <style>
        body {background-color: #c0c0c0;}
        p {color: blue; font-family: Arial;}
    </style>
</head>
<body>
    <p style="font-size: xx-small;">Rozmiar czcionki - xx-small</p>
    <p style="font-size: x-small;">Rozmiar czcionki - x-small</p>
    <p style="font-size: small;">Rozmiar czcionki - small</p>
    <p style="font-size: medium;">Rozmiar czcionki - medium</p>
    <p style="font-size: large;">Rozmiar czcionki - large</p>
    <p style="font-size: x-large;">Rozmiar czcionki - x-large</p>
</body>
</html>
```

Wynik interpretacji kodu został zilustrowany na rysunku 3.21.



**Rysunek 3.21.** Zastosowanie różnych rozmiarów czcionki

**2. Za pomocą wartości względnych** — wielkość czcionki jest ustalana względem wielkości czcionki elementu nadzawanego:

- **smaller** — czcionka mniejsza od bieżącej,
- **larger** — czcionka większa od bieżącej.

### Przykład 3.40

```
<p style="font-size: smaller;">Wybieramy się w podróż dookoła świata.</p>
<p style="font-size: larger;">Będziemy odwiedzali po kolej wszystkie kontynenty.</p>
```

**3.** Poprzez wielkość podaną w jednostkach miary. Jednostki, które mogą zostać użyte do określenia wielkości czcionki, to:

- *piksele* (px),
- *punkty* (pt),
- *centymetry* (cm),
- *cale* (in),
- *milimetry* (mm),
- *pica* (pc),
- *em* (em).

### Przykład 3.41

```
<p style="font-size: 12pt;">Wybieramy się w podróż dookoła świata.</p>
<p style="font-size: 1cm;">Będziemy odwiedzali po kolej wszystkie kontynenty.
</p>
```

**4.** Poprzez wielkość podaną w procentach. Każda przeglądarka ma zdefiniowany podstawowy rozmiar tekstu. Wartości procentowe określają wielkość czcionki w stosunku do rozmiaru podstawowego.

### Przykład 3.42

```
<p style="font-size: 120%;">Wybieramy się w podróż dookoła świata.</p>
<p style="font-size: 75%;">Będziemy odwiedzali po kolej wszystkie kontynenty.
</p>
```

**5.** Poprzez zastosowanie responsywnego rozmiaru czcionki. Rozmiar tekstu można ustawić za pomocą jednostki *vw* (ang. *viewport width*). W ten sposób rozmiar tekstu widocznego na stronie będzie związany z rozmiarem okna przeglądarki. Jednostka *vw* oznacza 1% aktualnego rozmiaru szerokości okna, w którym wyświetlana jest zawartość strony.

### Przykład 3.43

```
<p style="font-size: 16;">Wybieramy się w podróż dookoła świata.</p>
<p style="font-size: 10vw;">Będziemy odwiedzali po kolej wszystkie kontynenty.</p>
```

Jeżeli rozmiar czcionki nie zostanie określony, to na stronie zastosowany zostanie najprawdopodobniej domyślny rozmiar zwykłego tekstu, czyli 16 pikseli.

## Styl czcionki

Styl czcionki jest definiowany za pomocą atrybutu `font-style`.

```
selektor {font-style: styl;}
```

Dostępne są trzy style:

- `normal` — czcionka podstawowa,
- `italic` — czcionka pochylona,
- `oblique` — czcionka pochylona (podobna do poprzedniej, może być wygenerowana przez pochylenie zwykłej czcionki; jeżeli styl czcionki `italic` jest niedostępny, automatycznie wybierany jest styl `oblique`).

### Przykład 3.44

```
<!DOCTYPE html>

<html lang="pl-PL">
<head>
    <title>Styl czcionki</title>
    <meta charset="UTF-8">
    <style>
        body {background-color: #c0c0c0;}
        p {font-size: 14pt; color: blue; font-family: Arial;}
    </style>
</head>
<body>
    <p>Styl czcionki - normal</p>
    <p style="font-style: italic;">Styl czcionki - italic</p>
    <p style="font-style: oblique;">Styl czcionki - oblique</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 3.22.

Styl czcionki – normal  
Styl czcionki – italic  
Styl czcionki – oblique

**Rysunek 3.22.** Zastosowane różne style czcionki

## Wariant czcionki

Atrybut `font-variant` pozwala wybrać między czcionką normalną a kapitalikami (wyświetlanie wielkich liter o rozmiarze czcionki przeznaczonym dla małych liter).

```
selektor {font-variant: wartość;
```

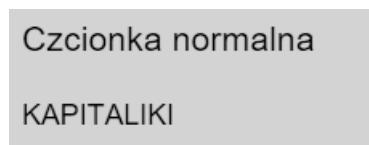
Atrybut `font-variant` może przyjmować wartości:

- `normal` — czcionka normalna,
- `small-caps` — kapitaliki.

### Przykład 3.45

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Kapitaliki</title>
    <meta charset="UTF-8">
    <style>
        body {background-color: #c0c0c0;}
        p {font-size: 16pt; font-family: Arial;}
    </style>
</head>
<body>
    <p>Czcionka normalna</p>
    <p style="font-variant: small-caps;">kapitaliki</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 3.23.



**Rysunek 3.23.** Czcionka normalna i kapitaliki

## Waga czcionki

Za pomocą atrybutu `font-weight` można określić wagę czcionki:

```
selektor {font-weight: wartość;
```

Waga czcionki określana jest następującymi wartościami:

- `normal` — czcionka normalna,
- `bold` — czcionka pogrubiona,
- `100, 200, 300, 400` (równoważne z `normal`), `500, 600, 700` (równoważne z `bold`), `800, 900` — określają wagę czcionki. Nie każda czcionka ma wszystkie dziewięć stopni grubości.

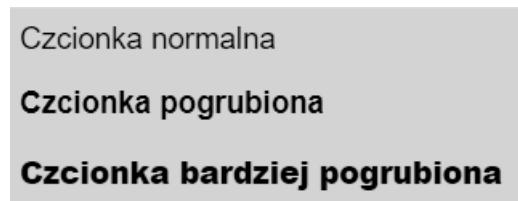
Można również używać wartości względnych:

- `lighter` — czcionka mniej wytłuszczonej od przypisanej domyślnie,
- `bolder` — czcionka bardziej wytłuszczonej od przypisanej domyślnie.

### Przykład 3.46

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Kapitaliki</title>
    <meta charset="UTF-8">
    <style>
        body {background-color: #c0c0c0;}
        p {font-size: 14pt; font-family: Arial;}
    </style>
</head>
<body>
    <p style="font-weight: normal;">Czcionka normalna</p>
    <p style="font-weight: bold;">Czcionka pogrubiona</p>
    <p style="font-weight: 900;">Czcionka bardziej pogrubiona</p>
</body>
</html>
```

Wynik interpretacji kodu został przedstawiony na rysunku 3.24.



**Rysunek 3.24.** Czcionka normalna i czcionka pogrubiona

## Atrybuty czcionki

Podczas definiowania formatu czcionki można użyć następujących atrybutów:

- font-style,
- font-variant,
- font-weight,
- font-size,
- font-family.

Często czcionce przypisywanych jest wiele atrybutów równocześnie. W takim przypadku można użyć wspólnego polecenia zawierającego wszystkie atrybuty dotyczące czcionek:

```
selektor {font: wartości atrybutów;}
```

Jako wartości atrybutów powinny zostać podane konkretne wartości, które będą definiowane dla czcionek. Należy oddzielić je od siebie spacjami. Wartości atrybutów powinny zostać wpisane w podanej wyżej kolejności. Zmiana kolejności może spowodować złą interpretację atrybutu lub jego pominięcie. Niektóre z wartości atrybutów mogą zostać pominięte, ale wartości dla atrybutów font-size i font-family są wymagane.

### Przykład 3.47

```
<p style="font: italic bold 12pt Georgia">
```

### Ćwiczenie 3.8

Wykorzystując arkusze stylów, zdefiniuj style określające właściwości czcionki (rodzaj, rozmiar, styl) dla znaczników: `<h2>`, `<h3>`, `<p>` na Twojej stronie internetowej.

## 3.4.2. Tekst

Język CSS pozwala na dowolne formatowanie tekstu poprzez dodanie do niego stylu. Umożliwia to nie tylko szybką zmianę wyglądu tekstu na stronie, ale także swobodne manipulowanie tekstem i grafiką umieszczaną w tekście.

### Wcięcie tekstu

Przy użyciu atrybutu `text-indent` można definiować wcięcie pierwszego wiersza akapitu.

```
selektor {text-indent: wartość;}
```

Jako wartość należy podać wielkość wcięcia.

### Przykład 3.48

```
p {text-indent: 20px;}
```

Wszystkie akapyt oznaczone znacznikiem `<p>` będą miały w pierwszym wierszu wcięcie równe 20px.

## Wyrównanie tekstu

Wyrównanie tekstu definiujemy za pomocą atrybutu `text-align`. Odpowiada on atrybutowi `align` w języku HTML.

```
selektor {text-align: wartość;}
```

Wyrównanie określa się za pomocą wartości:

- `left` — wyrównanie tekstu do lewego marginesu,
- `right` — wyrównanie tekstu do prawego marginesu,
- `center` — wyśrodkowanie tekstu,
- `justify` — rozłożenie tekstu równomiernie między marginesami.

### Przykład 3.49

```
p {text-align: justify;}  
h2 {text-align: right;}  
h3 {text-align: center;}
```

## Odstęp między wierszami tekstu

Do określenia wielkości odstępu między wierszami tekstu zapisanego w bloku służy atrybut `line-height`.

```
selektor {line-height: odstęp;}
```

Jako wartość atrybutu należy podać konkretną wielkość odstępu. Może to być:

- *liczba* — określa odstęp jako wielokrotność obecnego rozmiaru czcionki;
- *wysokość* — jest podawana z jednostką miary: px, pt, cm itp.; określa stałą wysokość odstępu;
- *wartość procentowa* — określa odstęp jako wartość procentową obecnego rozmiaru czcionki.

### Przykład 3.50

```
p {line-height: 0.8;}
```

## Ozdabianie tekstu

Do dodawania efektów, takich jak podkreślenie, przekreślenie tekstu, umieszczenie linii nad tekstem, używany jest atrybut `text-decoration`.

```
selektor {text-decoration: wartość;}
```

Rodzaj efektu określa się za pomocą wartości:

- `none` — bez zmian,
- `underline` — tekst podkreślony,

- `overline` — linia umieszczona nad tekstem,
- `line-through` — tekst przekreślony,
- `blink` — migotanie tekstu (nie wszystkie przeglądarki interpretują ten efekt).

### Przykład 3.51

```
p {text-decoration: underline;}  
h2 {text-decoration: overline;}  
h3 {text-decoration: line-through;}
```

Podane wartości można łączyć w jednej deklaracji.

### Przykład 3.52

```
p {text-decoration: underline line-through overline;}
```

### Odstęp między literami

Do definiowania odstępu między literami tekstu stosuje się atrybut `letter-spacing`.

```
selektor {letter-spacing: wartość;}
```

### Przykład 3.53

```
p {letter-spacing: 3px;}
```

### Odstęp między wyrazami

Do definiowania odstępu między wszystkimi wyrazami tekstu stosuje się atrybut `word-spacing`.

```
selektor {word-spacing: wartość;}
```

### Przykład 3.54

```
p {word-spacing: 12px;}
```

### Transformacja tekstu

Do ustawiania i zmiany wielkości liter tekstu służy atrybut `text-transform`. Atrybut ten kontroluje wielkość liter w tekście i dokonuje ich transformacji.

```
selektor {text-transform: wartość;}
```

Atrybut może przyjmować wartości:

- `capitalize` — zamienia na wielkie litery wszystkie pierwsze litery wyrazów w tekście, np. słowa „jan kowalski” zamieni na „Jan Kowalski”.
- `uppercase` — zamienia wszystkie litery tekstu na wielkie, np. słowa „jan kowalski” zamieni na „JAN KOWALSKI”.

- lowercase — zamienia wszystkie litery tekstu na małe, np. słowa „JAN KOWALSKI” zamieni na „jan kowalski”.
- none — bez transformacji. Tekst jest wyświetlany na stronie tak, jak został umieszczony w kodzie.

### Przykład 3.55

```
h3 {text-transform: capitalize;}  
h2 {text-transform: uppercase;}
```

## Białe znaki

Przy użyciu atrybutu `white-space` można sterować wyświetaniem na stronie spacji oraz innych białych znaków. Wstawienie w kodzie kilku spacji obok siebie spowoduje ich zamianę na pojedynczą spację. Również wstawienie spacji na początku tekstu jest niemożliwe. Atrybut `white-space` pozwoli pozbyć się tych niedogodności.

```
selektor {white-space: wartość;}
```

Atrybut może przyjmować wartości:

- normal — zamienia kilka sąsiednich spacji na jedną. Wiersze tekstu są przełamywane automatycznie.
- pre — umożliwia wyświetlanie kilku spacji obok siebie. Wiersze tekstu są przełamywane w miejscu wystąpienia znaku nowej linii (`Enter`).
- nowrap — zamienia kilka sąsiednich spacji na jedną. Wiersze tekstu są przełamywane w miejscu wystąpienia znacznika `<br>`.
- pre-wrap — umożliwia wyświetlanie kilku spacji obok siebie. Wiersze tekstu są przełamywane w miejscu wystąpienia znaku nowej linii (`Enter`) oraz automatycznie na końcu szerokości strony.
- pre-line — zamienia kilka sąsiednich spacji na jedną. Wiersze tekstu są przełamywane w miejscu wystąpienia znaku nowej linii (`Enter`) oraz automatycznie na końcu szerokości strony.

Nie wszystkie przeglądarki prawidłowo interpretują ten atrybut.

### Przykład 3.56

```
<!DOCTYPE html>  
<html lang="pl-PL" >  
<head>  
  <title>Białe znaki</title>  
  <meta charset="UTF-8">  
  <style>  
    body {background-color: #c0c0c0; }
```

```
p {white-space: pre-line;}  
</style>  
</head>  
<body>  
<p>  
Wybieramy się w podróż dookoła świata.  
Będziemy odwiedzali po kolej wszytke kontynenty.  
Zwiedzanie Europy rozpoczniemy od Madrytu. Zwiedzanie Azji rozpoczniemy od Pekinu. Najciekawszym krajem w Afryce jest Kenia. Australię planujemy zwiedzać na końcu.  
</p>  
</body>  
</html>
```

Wynik interpretacji kodu jest widoczny na rysunku 3.25.

Wybieramy się w podróż dookoła świata.  
Będziemy odwiedzali po kolej wszytke kontynenty.  
Zwiedzanie Europy rozpoczniemy od Madrytu. Zwiedzanie Azji rozpoczniemy od Pekinu. Najciekawszym krajem w Afryce jest Kenia. Australię planujemy zwiedzać na końcu.

**Rysunek 3.25.** Tekst został przełamany w miejscu wystąpienia znaku Enter. Nie ma powtarzających się spacji

### 3.4.3. Tło i kolor

W języku CSS kolor tekstu oraz właściwości tła można definiować dla różnych elementów z użyciem stylów.

#### Kolor

Korzystając z atrybutu `color`, można opisać pierwszoplanowy kolor wybranego elementu.

```
selektor {color: wartość;}
```

#### Przykład 3.57

```
h3 {color: #f0f0f0;}  
body {color: green;}
```

#### Kolor tła

Kolor tła dla elementu jest definiowany za pomocą atrybutu `background-color`.

```
selektor {background-color: wartość;}
```

### Przykład 3.58

```
body {background-color: #c0c0c0; }
p {background-color: lightblue; }
<p style="background-color: yellow;">Tło koloru żółtego</p>
h2 {color: green; background-color: #c0c0c0; }
```

### Grafika jako tło

Do umieszczenia obrazka jako tła służy atrybut `background-image`.

```
selektor {background-image: wartość; }
```

Wartość atrybutu jest nazwą pliku, w którym została zapisana grafika.

### Przykład 3.59

```
body {background-color: #f0f0f0; background-image: url("tlo.jpg"); }
```

W podanym przykładzie obrazek został zlokalizowany w tym samym folderze co arkusz stylów. Gdy odwołujemy się do obrazka znajdującego się w innej lokalizacji, musimy podać pełną ścieżkę do pliku.

### Powtarzanie grafiki w tle

Przy powtarzaniu obrazka w tle można określić sposób jego powtarzania, stosując atrybut `background-repeat`.

```
selektor {background-repeat: wartość; }
```

Jako wartość powtarzania można wybrać:

- `repeat` — powtarzanie tła w obu kierunkach,
- `repeat-x` — powtarzanie tła tylko w kierunku poziomym,
- `repeat-y` — powtarzanie tła tylko w kierunku pionowym,
- `no-repeat` — brak powtarzania tła.

### Przykład 3.60

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Grafika w tle</title>
    <meta charset="UTF-8">
    <style>
        body {
            background-color: #cad8f3;
```

```

background-image: url("obraz.png");
background-repeat: repeat-y;
}

p {
    font-size: 16pt;
    font-family: Arial;
}

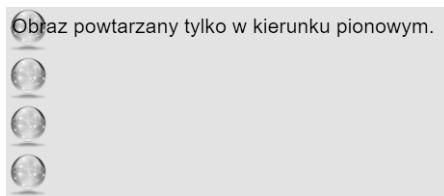
</style>

</head>

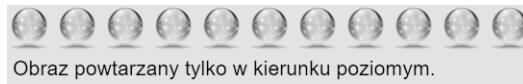
<body>
    <p>Obraz powtarzany tylko w kierunku pionowym.</p>
</body>
</html>

```

Wynik powtarzania tła tylko w kierunku pionowym został pokazany na rysunku 3.26. Natomiast wynik powtarzania tła tylko w kierunku poziomym — na rysunku 3.27.



**Rysunek 3.26.** Wstawiony obraz powtarzany w pionie



**Rysunek 3.27.** Wstawiony obraz powtarzany w poziomie

## Blokada grafiki w tle

Jeżeli treść strony jest przewijana za pomocą suwaka, to wstawiona jako tło strony grafika przesuwa się razem z tekstem. Aby grafika w trakcie takich działań była nieruchoma, należy użyć atrybutu `background-attachment`.

```
selektor {background-attachment: blokada;}
```

Jako wartość blokady można wybrać:

- `scroll` — przewijanie tła,
- `fixed` — tło nieruchome względem okna przeglądarki,
- `local` — tło nieruchome względem elementu, dla którego zostało zdefiniowane.

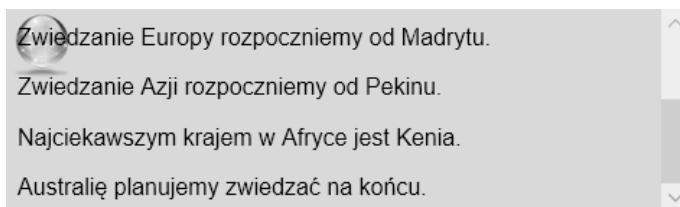
## Przykład 3.61

```
body {background-color: #ffccff; background-image: url("obraz.png");  
background-attachment: fixed;}
```

## Przykład 3.62

```
<!DOCTYPE html>  
  
<html lang="pl-PL">  
  
<head>  
  
    <title>Pozycja tła</title>  
  
    <meta charset="UTF-8">  
  
    <style>  
  
        body {  
  
            background-color: #c0cfc0;  
  
            background-image: url("obraz.png");  
  
            background-repeat: no-repeat;  
  
            background-attachment: fixed;  
  
        }  
  
        p {  
  
            font-size: 12pt;  
  
            font-family: Arial;  
  
            white-space: pre-wrap;  
  
        }  
  
    </style>  
  
</head>  
  
<body>  
  
    <p>Obraz został zablokowany.</p>  
  
    <p>Wybieramy się w podróż dookoła świata.</p>  
  
    <p>Będziemy odwiedzali po kolej wszytkie kontynenty.</p>  
  
    <p>Zwiedzanie Europy rozpoczniemy od Madrytu.</p>  
  
    <p>Zwiedzanie Azji rozpoczniemy od Pekinu.</p>  
  
    <p>Najciekawszym krajem w Afryce jest Kenia.</p>  
  
    <p>Australię planujemy zwiedzać na końcu.</p>  
  
</body>  
</html>
```

Wynik wykonania kodu został zilustrowany na rysunku 3.28.



**Rysunek 3.28.** Grafika tła unieruchomiona. Suwak okna przesunięty w dół

### Ćwiczenie 3.9

Dla utworzonej strony internetowej umieść grafikę jako tło strony lub jako tło umieszczonego na niej elementu. Sprawdź działanie atrybutu `background-attachment` dla wartości: `scroll`, `fixed` i `local`.

### Pozycja tła

Grafika wstawiona na stronę zostanie domyślnie umieszczona w lewym górnym rogu ekranu. Do zmiany standardowych ustawień i pozycjonowania grafiki w dowolnym miejscu na stronie służy atrybut `background-position`.

```
selektor {background-position: pozycja;}
```

Wartość atrybutu może być ustalana w różny sposób. Można określić położenie grafiki na stronie, podając jedną wartość, można to zrobić za pomocą dwóch parametrów, można też zapisać jej położenie jako wartość procentową szerokości ekranu.

Niżej zostały podane wartości parametru `pozycja`.

#### Jedna wartość:

- `center` — grafika zostanie umieszczona na środku strony;
- `left` — grafika zostanie umieszczona z lewej strony;
- `right` — grafika zostanie umieszczona z prawej strony;
- `top` — grafika zostanie umieszczona na górze strony;
- `bottom` — grafika zostanie umieszczona na dole strony;
- jednostka długości — grafika zostanie umieszczona w określonej odległości od lewego marginesu.

#### Dwie wartości:

- `left top` — grafika zostanie umieszczona w lewym górnym rogu strony;
- `left bottom` — grafika zostanie umieszczona w lewym dolnym rogu strony;
- `right top` — grafika zostanie umieszczona w prawym górnym rogu strony;
- `right bottom` — grafika zostanie umieszczona w prawym dolnym rogu strony.

Definiowanie tych parametrów ma sens, jeżeli ograniczymy powtarzanie grafiki na stronie (mogą być użyte parametry powtarzania: `no-repeat`, `repeat-x`, `repeat-y`).

### Przykład 3.63

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Pozycja tła</title>
    <meta charset="UTF-8">
    <style>
        body {
            background-color: #f5b270;
            background-image: url("drzewo1.png");
            background-repeat: no-repeat;
            background-position: right top;
        }
        p {
            font-size: 12pt;
            font-family: Arial;
        }
    </style>
</head>
<body>
    <p>
        Drzewo – wieloletnia roślina o zdrewniałym jednym pędzie głównym (pniu) albo zdrewniałych kilku pędach głównych i gałęziach tworzących koronę.

        Do drzew zaliczają się największe rośliny lądowe. Od innych roślin drzewiastych (krzewów i krzewinek) różnią się posiadaniem pnia lub pni rozgałęziających się dopiero od pewnej wysokości.

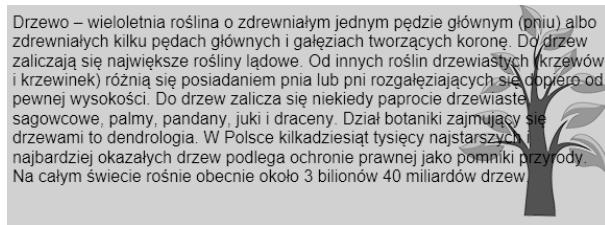
        Do drzew zalicza się niekiedy paprocie drzewiaste, sagowcowe, palmy, pandany, juki i draceny.

        Dział botaniki zajmujący się drzewami to dendrologia.

        W Polsce kilkadziesiąt tysięcy najstarszych i najbardziej okazałych drzew podlega ochronie prawnej jako pomniki przyrody.

        Na całym świecie rośnie obecnie około 3 bilionów 40 miliardów drzew.
    </p>
</body>
</html>
```

Wynik wykonania kodu został pokazany na rysunku 3.29.



**Rysunek 3.29.** Grafika została umieszczona w prawym górnym rogu strony

## Usunięcie tła

Jeżeli na stronie (lub na kilku stronach) nie chcemy stosować wcześniej zdefiniowanego tła dla stron lub elementów, to najprostszym rozwiązaniem jest niedołączanie do tych stron zewnętrznego arkusza stylów. Takie rozwiązanie nie zawsze jest dobre, ponieważ inne deklaracje znajdujące się w zewnętrznym arkuszu stylów najczęściej są wykorzystywane na stronie. Lepszym rozwiązaniem jest zachowanie odnośnika do zewnętrznego arkusza stylów, a dodatkowo w wewnętrznym arkuszu stylów lub w stylu `inline` usunięcie tła dla danej strony. Można do tego użyć polecenia:

```
selektor {background: none;}
```

Polecenie to usuwa wszystkie atrybuty dotyczące tła.

Podobną metodą można usunąć tylko wybrane atrybuty tła. Należy wtedy w poleceniu użyć nazwy atrybutu, który zostanie usunięty.

## Atrybuty tła

Możliwe jest również zdefiniowanie wszystkich właściwości tła przy użyciu jednego atrybutu `background`.

```
selektor {background: wartości_atrybutów;}
```

Jako wartości atrybutów trzeba podać konkretne wartości oddzielone spacjami. W ten sposób można zdefiniować następujące atrybuty tła:

- kolor,
- tło obrazkowe,
- powtarzanie,
- blokadę tła,
- pozycję.

Nie trzeba podawać pełnej listy atrybutów. Opuszczonemu atrybutowi zostanie przypisana wartość domyślna. Nie wszystkie przeglądarki akceptują tę skróconą formę deklaracji atrybutów tła i wtedy trzeba podawać ich pełną deklarację.



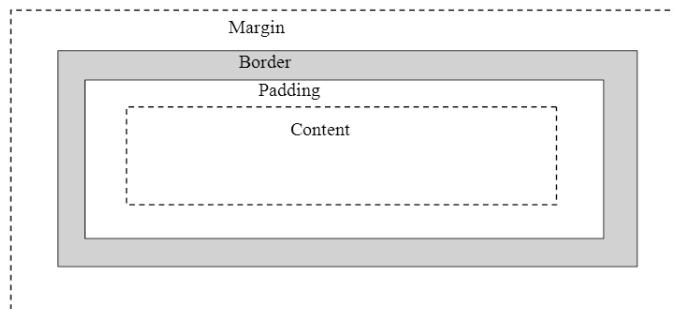
## 3.5. Model blokowy CSS

Wszystkie elementy projektowane w języku HTML można traktować jak prostokątne bloki odpowiednio sformatowane i rozmieszczone na stronie internetowej. Każdy z takich bloków ma marginesy, obramowania, wypełnienie i zawartość.

Model blokowy został zamieszczony na rysunku 3.30.

**Rysunek 3.30.**

Model blokowy CSS



- **Margin** — margines wokół ramki (margines zewnętrzny). Jest to pusty obszar wokół ramki, nie ma koloru tła i jest przezroczysty.
- **Border** — obramowanie wokół zawartości elementu; ma styl i kolor.
- **Padding** — odstęp między obramowaniem i zawartością elementu (margines wewnętrzny).
- **Content** — zawartość elementu, np. teksty lub obrazy.

Każdy element w modelu blokowym jest opisywany przez trzy atrybuty: *padding*, *margin* i *border*. Podane atrybuty tworzą układ graficzny strony internetowej.

### 3.5.1. Obramowanie (border)

Obramowanie to ramka (ang. *border*) narysowana wokół elementu. Obramowanie może być wykorzystywane do dekoracji elementu lub do oddzielenia go od innych elementów. Można je definiować dla jednej lub dla wielu krawędzi bloku.

#### Szerokość obramowania

Do ustawienia szerokości obramowania służy atrybut *border-width*. Przy jego użyciu można definiować obramowanie jednakowe dla wszystkich krawędzi lub dla każdej z osobna.

```
selektor {border-width: wartości;}
```

Możliwe ustawienia atrybutu:

- *jedna wartość* — jednakowa szerokość dla wszystkich krawędzi;
- *dwie wartości* — taka sama szerokość dla krawędzi poziomych i taka sama szerokość dla krawędzi pionowych;

- *trzy wartości* — pierwsza wartość określa szerokość krawędzi górnej, druga — jednocześnie szerokość dwóch krawędzi pionowych, trzecia — szerokość krawędzi dolnej;
- *cztery wartości* — każda wartość określą osobno szerokość kolejnych krawędzi (górnej, prawej, dolnej, lewej).

Szerokość krawędzi może być definiowana w dowolnych jednostkach lub za pomocą trzech wartości:

- `thin` — cienkie obramowanie,
- `medium` — średnie obramowanie,
- `thick` — grube obramowanie.

Możliwe jest również definiowanie szerokości pojedynczych krawędzi. Służą do tego atrybuty przypisane do poszczególnych krawędzi.

Atrybut `border-top-width` definiuje szerokość górnej krawędzi.

```
selektor {border-top-width: szerokość;}
```

Atrybut `border-bottom-width` definiuje szerokość dolnej krawędzi.

```
selektor {border-bottom-width: szerokość;}
```

Atrybut `border-left-width` definiuje szerokość lewej krawędzi.

```
selektor {border-left-width: szerokość;}
```

Atrybut `border-right-width` definiuje szerokość prawej krawędzi.

```
selektor {border-right-width: szerokość;}
```

### Przykład 3.64

```
p {border-width: 1px;}  
div {border-bottom-width: 20px;}  
h3 {border-width: thick;}
```

## Styl obramowania

Do definiowania stylu obramowania służy atrybut `border-style`. Za jego pomocą można definiować styl obramowania jednakowy dla wszystkich krawędzi lub dla każdej inny.

```
selektor {border-style: wartości;}
```

Podobnie jak w przypadku atrybutu `border-width`, możliwe są następujące ustawienia atrybutu:

- *jedna wartość* — jednakowy styl dla wszystkich krawędzi;
- *dwie wartości* — taki sam styl dla krawędzi poziomych i taki sam styl dla krawędzi pionowych;

- *trzy wartości* — pierwsza wartość określa styl krawędzi górnej, druga — jednocześnie styl dwóch krawędzi pionowych, trzecia — styl krawędzi dolnej;
- *cztery wartości* — każda wartość określa osobno styl kolejnych krawędzi (górną, prawą, dolną, lewą).

Możliwe jest, jak w przypadku atrybutu `border-width`, definiowanie stylu pojedynczych krawędzi. Służą do tego atrybuty przypisane do poszczególnych krawędzi.

Atrybut `border-top-style` definiuje styl górnej krawędzi.

```
selektor {border-top-style: styl;}
```

Atrybut `border-bottom-style` definiuje styl dolnej krawędzi.

```
selektor {border-bottom-style: styl;}
```

Atrybut `border-left-style` definiuje styl lewej krawędzi.

```
selektor {border-left-style: styl;}
```

Atrybut `border-right-style` definiuje styl prawej krawędzi.

```
selektor {border-right-style: styl;}
```

Wartości, które mogą być przypisywane do stylów obramowania, to:

- `none` — brak obramowania,
- `hidden` — obramowanie ukryte,
- `dotted` — linia kropkowana,
- `dashed` — linia kreskowana,
- `solid` — linia ciągła,
- `double` — linia ciągła podwójna,
- `groove` — „rowek”,
- `ridge` — „grzbiet”,
- `inset` — „ramka”,
- `outset` — „przycisk”.

### UWAGA

Przy projektowaniu ramek obowiązkowe jest podanie wartości dla stylu i dla szerokości.

## Kolor obramowania

Do definiowania koloru obramowania służy atrybut `border-color`. Jako wartości koloru można używać kodów kolorów zapisanych heksadecymalnie (#ff00ff), kodów `rgb(r,g,b)` lub ich nazw.

```
selektor {border-color: kolor;}
```

Podobnie jak w przypadku innych atrybutów obramowania, możliwe jest podanie jednej, dwóch, trzech lub czterech wartości koloru.

Możliwe jest też definiowanie koloru pojedynczych krawędzi. Służą do tego atrybuty:

- border-top-color,
  - border-bottom-color,
  - border-left-color,
  - border-right-color.

### Przykład 3.65

```
<!DOCTYPE html>

<html lang="pl-PL">

<head>
    <title>Obramowanie</title>
    <meta charset="UTF-8" />
    <style>
        body {
            background-color: #f2f2f2;
        }
        p {
            font-size: 14pt;
            font-family: Arial;
            border-width: 1px;
            border-style: dashed;
            border-color: red;
        }
        h1 {
            border-width: 20px;
            border-style: outset;
            border-color: blue;
        }
        h2 {
            border-width: thick;
            border-style: dotted;
            border-color: gold;
        }
    </style>
</head>
<body>
    <h1>Obramowanie</h1>
    <h2>Obramowanie</h2>
    <p>Obramowanie</p>
</body>
</html>
```

```


Wynik wykonania kodu został pokazany na rysunku 3.31.



The screenshot shows a simple web page structure. At the top is a black header bar containing the text "Wybieramy się w podróż dookoła świata." in white. Below this is a white main content area enclosed in a black rectangular border. Inside this border, there is a black header with the text "Będziemy odwiedzali po kolej wszytne kontynenty." in white. Below this header is a black section containing three lines of text: "Zwiedzanie Europy rozpoczniemy od Madrytu.", "Zwiedzanie Azji rozpoczniemy od Pekinu.", and "Najciekawszym krajem w Afryce jest Kenia." All text is in white.



Rysunek 3.31. Definiowanie koloru obramowania (border)



## Atrybuty obramowania



Podobnie jak w przypadku tła, używając atrybutu border, można zdefiniować równocześnie kilka właściwości obramowania.



```
selektor {border: wartości_atrybutów;}
```



Wartości atrybutów dotyczą szerokości (width), stylu (style) i koloru (color). Mogą być wpisane w dowolnej kolejności i muszą być oddzielone znakiem spacji.



163


```

**Przykład 3.66**

```
p {border: double 1px red; }
h2 {border: 20px outset blue; }
h3 {border: thick dotted gold; }
```

**3.5.2. Marginesy zewnętrzne (margin)**

Margines zewnętrzny określa przestrzeń wokół definiowanego elementu. Przestrzeń ta oddziela element od innych elementów strony.

Dla każdego elementu można zdefiniować jego odległość od góry, dołu, lewej i prawej strony. Służy do tego atrybut `margin`.

```
selektor {margin-pozycja: rozmiar; }
```

Jako *pozycja* może wystąpić jedno z określeń dotyczących położenia marginesu:

- `top` — margines górny,
- `bottom` — margines dolny,
- `left` — margines lewy,
- `right` — margines prawy.

Jako *rozmiar* podawana jest wartość w jednostkach długości (np. `px`, `cm`). Można wpisać jako rozmiar `auto`. Spowoduje to nadanie przez przeglądarkę automatycznej wartości dla marginesu.

**Przykład 3.67**

```
p {margin-top: 1cm; }
h2 {margin-bottom: 2cm; }
h3 {margin-left: 20%; }
```

Za pomocą atrybutu `margin` można też ustawać marginesy dla wszystkich czterech boków elementu.

```
selektor {margin: wartości; }
```

Jako wartości atrybutu można podać:

- *jedną wartość* — wszystkie marginesy jednakowe;
- *dwie wartości* — pierwsza oznacza marginesy górny i dolny, druga — marginesy lewy i prawy;
- *trzy wartości* — pierwsza oznacza margines górny, druga — marginesy lewy i prawy, trzecia — margines dolny;
- *cztery wartości* — każda wartość określa osobno marginesy górny, prawy, dolny, lewy.

Jeśli przy takim zapisie chcemy pominąć jeden z marginesów, wystarczy ustawić jego wartość na 0.

### Przykład 3.68

```
p {margin: 1cm;}
h2 {margin: 20px 10px;}
h3 {margin: 2cm 15px 1.5cm 12px;}
```

## 3.5.3. Marginesy wewnętrzne (padding)

Margines wewnętrzny to przestrzeń między zawartością elementu a obramowaniem.

Podobnie jak dla marginesów zewnętrznych, można zdefiniować tę odległość od góry, dołu, lewej i prawej strony. Do definiowania marginesów wewnętrznych służy atrybut padding.

```
selektor {padding-pozycja: rozmiar;
```

Jako *pozycja* mogą wystąpić określenia dotyczące położenia marginesów:

- `top` — margines górny,
- `bottom` — margines dolny,
- `left` — margines lewy,
- `right` — margines prawy.

### Przykład 3.69

```
p {padding-top: 25px;}
h3 {padding-left: 2cm;}
```

Można również ustawać marginesy wewnętrzne dla wszystkich czterech boków elementu na takich samych zasadach jak dla marginesu zewnętrznego.

```
selektor {padding: wartości;
```

Wartości atrybutu:

- *jedna wartość* — wszystkie marginesy jednakowe;
- *dwie wartości* — pierwsza oznacza marginesy górnny i dolny, druga — marginesy lewy i prawy;
- *trzy wartości* — pierwsza oznacza margines górnny, druga — marginesy lewy i prawy, trzecia — margines dolny;
- *cztery wartości* — każda wartość określa osobno marginesy górnny, prawy, dolny, lewy.

Aby pominąć jeden z marginesów wewnętrznych, należy wpisać wartość 0.

Różnice związane z definiowaniem marginesów zewnętrznych i wewnętrznych pokazuje rysunek 3.32.

## Ćwiczenie 3.10

Wykorzystując poznane narzędzia, zaprojektuj stronę, która będzie wyglądała podobnie jak ta na rysunku 3.32.



**Rysunek 3.32.** Definiowanie marginesów wewnętrznych i zewnętrznych

### 3.5.4. Obrys

Obramowanie wokół elementu może być tworzone również poprzez zdefiniowanie obrysu. Obrys różni się od obramowania tym, że:

- nie zajmuje miejsca, dzięki czemu nie wpływa na rozmiar i położenie elementu;
- jest tworzony na wierzchu elementu;
- jest definiowany dla wszystkich krawędzi równocześnie.

Obrys można definiować przy użyciu atrybutu `outline`.

```
selektor {outline: wartości;}
```

W ten sposób wybiera się kolor (ang. `color`), styl (ang. `style`) i grubość (ang. `width`) obrysu. Obowiązkowym atrybutem jest styl obrysu. Pozostałe dwa atrybuty występują opcjonalnie. Sposób definiowania obrysu jest taki sam jak przy definiowaniu obramowania.

### Przykład 3.70

```
p {outline: solid 3px red;}
h2 {outline-style: dashed;}
h3 {outline-width: 10px; outline-color: blue;}
```

### 3.5.5. Rozmiary elementów

W modelu blokowym można ustalać dokładne rozmiary różnych elementów, np. akapitu, tabeli, obrazka, bloku `div`.

## Szerokość

Do definiowania szerokości elementu służy atrybut `width`.

```
selektor {width: szerokość;}
```

### Przykład 3.71

```
h3 {width: 20%;}
```

## Szerokość minimalna

Za pomocą atrybutu `min-width` można określić minimalną szerokość elementu. Pozwala to ustawić rozmiar elementu tak, aby nigdy nie był mniejszy od podanej wartości.

```
selektor {min-width: szerokość;}
```

### Przykład 3.72

```
img {min-width: 200px;}
```

## Szerokość maksymalna

Atrybut `max-width` pozwala określić maksymalną szerokość elementu. Element może być węższy od podanej wartości, ale nie może być od niej szerszy.

```
selektor {max-width: szerokość;}
```

### Przykład 3.73

```
img {max-width: 300px;}
```

## Wysokość

Atrybut `height` ustala wysokość elementu.

```
selektor {height: wysokość;}
```

### Przykład 3.74

```
h3 {height: 3cm;}
```

## Wysokość minimalna

Za pomocą atrybutu `min-height` można określić minimalną wysokość elementu. Oznacza to, że wysokość elementu nigdy nie będzie mniejsza od podanej wartości.

```
selektor {min-height: wysokość;}
```

### Przykład 3.75

```
img {min-height: 300px;}
```

## Wysokość maksymalna

Atrybut `max-height` pozwala określić maksymalną wysokość elementu. Element może być niższy od podanej wartości, ale nie może być od niej wyższy.

```
selektor {max-height: wysokość;} 
```

### Przykład 3.76

```
img {max-height: 5cm;} 
```

### Przykład 3.77

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Bejrut</title>
    <meta charset="UTF-8">
    <style>
        img.roz1 {height: auto; margin-right: 20px;}
        img.roz2 {height: 120px; margin-right: 20px;}
        p.blok {height: 100px; width: 130px;}
    </style>
</head>
<body>
    
    
    
    <p class="blok">
        Bejrut to stolica i największe miasto Libanu, leżące w środkowej części kraju.
    </p>
    <p>
        Liczy 2 mln 60 tys. mieszkańców, większość pochodzenia arabskiego.
        Znajduje się tu wiele zabytków i dobytku historycznego kraju.
        Bejrut stanowi siedzibę libańskiego rządu i odgrywa kluczową rolę w gospodarce kraju.
        Miasto jest również ważnym ośrodkiem życia kulturalnego.
    </p>
</body>
</html>
```

Wynik wykonania kodu został pokazany na rysunku 3.33.



**Rysunek 3.33.** Określenie rozmiarów elementów strony

## Przepelnienie

Jeżeli zawartość elementu nie mieści się w rozmiarach podanych za pomocą atrybutów `width` i `height`, możliwe jest ukrycie niemieszczącej się zawartości, powiększenie rozmiarów elementu i pokazanie całej zawartości lub wyświetlenie suwaków, które pozwolą (poprzez przewijanie) na obejrzenie całej zawartości elementu. Do określania sposobu pokazywania zawartości elementu w takiej sytuacji służy atrybut `overflow`.

```
selektor {overflow: sposob;}
```

Wartościami parametru `sposób` mogą być:

- `visible` — pokazywana jest cała zawartość elementu, niezależnie od zdefiniowanego rozmiaru;
- `hidden` — treść niemieszcząca się w zdefiniowanym rozmiarze elementu zostanie ukryta;
- `scroll` — powoduje przepelenie i dodanie paska przewijania, który pozwoli na przewijanie zawartości wewnątrz bloku i obejrzenie całej zawartości elementu; pasek przewijania zostanie dodany zarówno w poziomie, jak i w pionie;
- `auto` — jeżeli jest to konieczne, suwaki zostaną wyświetcone.

Atrybutu `overflow` można używać tylko dla elementów blokowych ze zdefiniowaną wysokością. Domyślną wartością jest `visible`, co oznacza, że element nie jest obcinany, nawet jeżeli nie mieści się w zdefiniowanych rozmiarach.

### Przykład 3.78

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Przepelenienie</title>
    <meta charset="UTF-8">
    <style>
        div {width: 250px; height: 200px; overflow: scroll;}
    </style>
</head>
<body>
    <div>
        
    </div>
</body>
</html>
```

Wynik wykonania kodu został pokazany na rysunku 3.34.



**Rysunek 3.34.** Suwaki pozwalają obejrzeć całe zdjęcie

Wartość `auto` działa podobnie jak wartość `scroll`, ale paski przewijania są dodawane tylko, gdy to jest konieczne.

Do określenia sposobu wyświetlania treści tylko w poziomie lub pionie służą atrybuty `overflow-x` i `overflow-y`. Atrybut `overflow-x` określa, co należy zrobić z prawą/lewą krawędzią treści, a atrybut `overflow-y` — co zrobić z górną/dolną krawędzią.



## 3.6. Inne elementy

### 3.6.1. Listy

W języku HTML można definiować listy punktowane (`<ul>`) i numerowane (`<ol>`). Jeżeli nie odpowiada nam sposób, w jaki te listy są wyświetlane przez przeglądarki, w języku CSS możemy zdefiniować inny sposób wyświetlania list.

Możliwe są: zmiana rodzaju punktora, zdefiniowanie własnego punktora oraz określenie położenia punktora i jego odległości od tekstu.

#### Styl listy

Większość przeglądarek w listach punktowanych wyświetla okrągłe punktory, a w listach numerowanych cyfry.

Za pomocą atrybutu `list-style-type` można zmienić punktor lub cyfrę wykazu na liście.

```
selektor {list-style-type: typ;}
```

Selektorami mogą być znaczniki definiujące listy: `<ul>`, `<ol>` oraz `<li>`.

Parametr `typ` określa wygląd punktora (*markera*).

Wybrane typy punktorów:

- `disc` — koło,
- `circle` — okrąg,
- `square` — kwadrat,
- `decimal` — liczby arabskie,
- `lower-roman` — małe cyfry rzymskie,
- `upper-roman` — duże cyfry rzymskie,
- `lower-alpha` — małe litery,
- `upper-alpha` — duże litery,
- `none` — brak punktora.

Na rysunku 3.35 zostały przedstawione różne typy punktorów.



**Rysunek 3.35.**

Definiowanie punktorów listy

## Ćwiczenie 3.11

Zaprojektuj na stronie internetowej listę punktowaną, listę numerowaną oraz listę z podpunktami. Zastosuj różne typy punktorów. Sprawdź ich wygląd. Stronę przetestuj w kilku przeglądarkach internetowych.

## Obraz jako punktor

Punktorem listy może być obrazek zapisany w pliku. Do ustawienia obrazka jako markera służy atrybut `list-style-image`.

```
selektor {list-style-image: url(ściezka dostępu);}
```

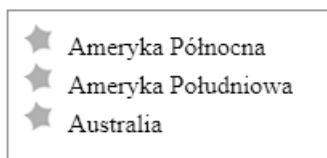
Ścieżka dostępu określa położenie pliku z obrazkiem, który ma być punktorem.

## Przykład 3.79

```
<!DOCTYPE html>

<html lang="pl-PL">
<head>
    <title>Punktor graficzny</title>
    <meta charset="UTF-8">
    <style>
        ul {list-style-image: url(punktor.png); }
    </style>
</head>
<body>
    <ul>
        <li>Ameryka Północna</li>
        <li>Ameryka Południowa</li>
        <li>Australia</li>
    </ul>
</body>
</html>
```

Wynik wykonania kodu zilustrowano na rysunku 3.36.



**Rysunek 3.36.** Obraz jako punktor

## Pozycja punktora

Oprócz typu punktora można ustalić położenie punktora względem tekstu. Służy do tego atrybut `list-style-position`.

```
selektor {list-style-position: pozycja;}
```

Parametr *pozycja* określa zachowanie się tekstu, który nie zmieści się w jednej linii, względem punktora. Możliwe są dwie wersje:

- `outside` — punktor pojawia się na zewnątrz zawartości (rysunek 3.37);
- `inside` — punktor pojawia się wewnątrz zawartości (rysunek 3.38).

★ Afryka to drugi pod względem wielkości kontynent na Ziemi. Zajmuje 30,37 mln km<sup>2</sup>, czyli ponad 20,3% ogólnej powierzchni lądowej świata. Przechodzą przez niego południk 0°, obydwa zwrotniki i równik.

★ Azja razem z Europą tworzy Eurazję, największy kontynent na Ziemi. Z powodów historycznych i kulturowych sama Azja bywa również nazywana kontynentem. Nazywana jest często kontynentem wielkich kontrastów geograficznych.

★ Nazwa „Australia” pochodzi od łacińskiego określenia „Terra Australis”, czyli „Ziemia Południowa”. W dawnej Europie wierzono, że na półkuli południowej znajduje się naprawdę duży kontynent. Ostatecznie nazwa ta przypadła kontynentowi o wiele mniejszemu niż zakładano.

**Rysunek 3.37.** Punktor na zewnątrz zawartości

★ Afryka to drugi pod względem wielkości kontynent na Ziemi. Zajmuje 30,37 mln km<sup>2</sup>, czyli ponad 20,3% ogólnej powierzchni lądowej świata. Przechodzą przez niego południk 0°, obydwa zwrotniki i równik.

★ Azja razem z Europą tworzy Eurazję, największy kontynent na Ziemi. Z powodów historycznych i kulturowych sama Azja bywa również nazywana kontynentem. Nazywana jest często kontynentem wielkich kontrastów geograficznych.

★ Nazwa „Australia” pochodzi od łacińskiego określenia „Terra Australis”, czyli „Ziemia Południowa”. W dawnej Europie wierzono, że na półkuli południowej znajduje się naprawdę duży kontynent. Ostatecznie nazwa ta przypadła kontynentowi o wiele mniejszemu niż zakładano.

**Rysunek 3.38.** Punktor wewnątrz zawartości

## Atrybuty listy

Używając atrybutu `list-style`, można równocześnie zdefiniować właściwości listy: `list-style-type`, `list-style-image` oraz `list-style-position`. Wartości atrybutów muszą być oddzielone spacjami.

```
selektor {list-style: wartości_atrybutów;}
```

## Przykład 3.80

```
<ul style="list-style: square inside;">
  <li>Europa</li>
  <li>Azja</li>
  <li>Afryka</li>
</ul>
```

## 3.6.2. Tabele

W języku HTML tabele są stosowane do przedstawiania danych. Jeżeli chcemy za pomocą tabel zaprezentować dane, to użycie stylów CSS znacznie poprawi ich wygląd.

### Podpis

W celu określenia położenia podpisu dodanego do tabeli używamy atrybutu `caption-side`.

```
selektor {caption-side: ustawienie;}
```

Parametr `ustawienie` może przyjmować wartości:

- `top` — podpis zostanie wyświetlony nad tabelą;
- `bottom` — podpis zostanie wyświetlony pod tabelą;
- `left` — podpis zostanie wyświetlony z lewej strony tabeli;
- `right` — podpis zostanie wyświetlony z prawej strony tabeli.

Nie wszystkie przeglądarki prawidłowo interpretują ten atrybut.

### Obramowanie

Do definiowania krawędzi tabeli używa się atrybutu `border`.

```
selektor {border: wartości;}
```

Dla tego atrybutu można ustawić grubość, rodzaj oraz kolor obramowania.

### Przykład 3.81

```
table, tr, td {border: 2px solid red;}
```

Atrybut `border-collapse` określa sposób obramowania komórek tabeli. Możliwe jest obramowanie sąsiednich komórek za pomocą jednej linii lub obramowanie każdej komórki osobno.

```
selektor {border-collapse: model;}
```

Wartości parametru `model` to:

- `collapse` — pojedyncze obramowanie;
- `separate` — komórki tabeli będą od siebie odseparowane.

### Przykład 3.82

```
table {border-collapse: separate;}
```

Można także określić sposób obramowania dla pustych komórek. Służy do tego atrybut `empty-cells`.

```
selektor {empty-cells: wartość;}
```

Atrybut może przyjmować wartości:

- `show` — obramowanie wokół pustych komórek powinno być wyświetlane;
- `hide` — obramowanie wokół pustych komórek powinno zostać ukryte.

Do tabel oraz do ich komórek można stosować wszystkie atrybuty definiowane dla obramowań w modelu blokowym.

## Rozmieszczenie

Do rozplanowania tabeli na stronie można użyć atrybutu `table-layout`.

```
selektor {table-layout: rozmieszczenie;}
```

Atrybut może przyjmować wartości:

- `auto` — szerokość tabeli jest ustawiana automatycznie i zależy od zawartości komórek;
- `fixed` — szerokość tabeli wynika z zadeklarowanej szerokości poszczególnych komórek oraz szerokości całej tabeli.

Różne przeglądarki różnie interpretują to polecenie, jeśli nie zostanie podana ogólna szerokość tabeli lub jeśli szerokość tabeli jest większa niż suma szerokości jej komórek.

### Przykład 3.83

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Obramowanie tabel</title>
    <meta charset="UTF-8">
    <style>
        table, tr, td {
            border: 1px solid blue;
            border-collapse: collapse;
            table-layout: fixed;
        }
        th {
            border: 1px solid green;
            border-collapse: collapse;
        }
    </style>
```

```

</head>
<body>
  <table>
    <tr><th>Europa</th><th>Azja</th><th>Afryka</th></tr>
    <tr><td>Francja</td><td>Indie</td><td>Egipt</td></tr>
    <tr><td>Hiszpania</td><td>Chiny</td><td>Nigeria</td></tr>
    <tr><td>Włochy</td><td>Japonia</td><td>Kongo</td></tr>
  </table>
</body>
</html>

```

Wynik wykonania kodu został pokazany na rysunku 3.39.

Europa	Azja	Afryka
Francja	Indie	Egipt
Hiszpania	Chiny	Nigeria
Włochy	Japonia	Kongo

**Rysunek 3.39.** Wyświetlenie tabeli ze zdefiniowanymi obramowaniami

## Odstępy między komórkami

Do definiowania odstępów między komórkami służy atrybut `border-spacing`.

```
selektor {border-spacing: odstęp;}
```

Odstęp to zapisana w jednostkach długość odległość między komórkami tabeli. Można podać jedną wartość (i wtedy dotyczy ona wszystkich odstępów) lub dwie wartości rozdzielone spacją (i wtedy definiują one poziomy oraz pionowy odstęp między komórkami).

## Szerokość i wysokość tabeli

Za pomocą właściwości `width` i `height` można zdefiniować szerokość i wysokość tabeli.

### Przykład 3.84

```
table {width: 100%;}
th {height: 50px;}
```

### Przykład 3.85

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
```

```
<title>Obramowanie tabel</title>
<meta charset="UTF-8">
<style>
table {
    width: 100%
}

table, tr, td {
    border: 1px solid blue;
    border-collapse: collapse;
    table-layout: fixed;
}

th {
    border: 1px solid green;
    border-collapse: collapse;
    height: 50px;
}

</style>

</head>
<body>
<table>

<tr><th>Europa</th><th>Azja</th><th>Afryka</th></tr>
<tr><td>Francja</td><td>Indie</td><td>Egipt</td></tr>
<tr><td>Hiszpania</td><td>Chiny</td><td>Nigeria</td></tr>
<tr><td>Włochy</td><td>Japonia</td><td>Kongo</td></tr>

</table>
</body>
</html>
```

Wynik wykonania kodu przedstawiono na rysunku 3.40.

Europa	Azja	Afryka
Francja	Indie	Egipt
Hiszpania	Chiny	Nigeria
Włochy	Japonia	Kongo

**Rysunek 3.40.** Tabela ze zdefiniowanymi: wysokością i szerokością

### 3.6.3. Pozycjonowanie

Pozycjonowanie pozwala zdefiniować położenie elementów na stronie internetowej. Elementy można rozmieszczać nie tylko względem brzegów strony, ale również względem jej poszczególnych elementów. Można też umieszczać elementy tak, że jeden będzie przykrywał inny.

Do pozycjonowania elementów służy atrybut `position`.

```
selektor {position: rodzaj; parametry;} 
```

Parametr *rodzaj* określa sposób pozycjonowania elementu i może przyjmować wartości:

- `static` — przywraca normalne pozycjonowanie elementu. Jest to wartość przydatna wtedy, gdy wcześniej (np. w arkuszach stylów) została podana deklaracja pozycjonowania tego typu elementów. Użycie w definicji stylu `inline` dla wybranego elementu deklaracji `position: static` zniesie tę deklarację dla danego elementu.
- `relative` (inaczej: pozycjonowanie względne) — pozwala przesunąć wybrany element w inne miejsce w stosunku do położenia pierwotnego. Poprzez ustawienie parametrów można określić sposób przesunięcia. Element zostanie umieszczony „na sztywno” w miejscu określonym przez parametry.
- `absolute` — pozwala przesunąć wybrany element w inne miejsce względem brzegów strony lub bloku (jeżeli element został umieszczony w bloku). Poprzez ustawienie parametrów można określić sposób przesunięcia. Element zostanie umieszczony „na sztywno” w miejscu określonym przez parametry.
- `fixed` — działa podobnie jak pozycjonowanie absolutne. Różnica polega na tym, że pozycjonowanie `fixed` ustala położenie elementu zawsze względem krawędzi okna przeglądarki. Tak pozycjonowany element jest nieruchomy przy przewijaniu strony (jest przez cały czas widoczny w tym samym miejscu).

Jako *parametr* należy podawać:

- `left: wartość;` — przesunięcie o określoną wartość w stosunku do lewej krawędzi położenia pierwotnego,
- `top: wartość;` — przesunięcie o określoną wartość w stosunku do górnej krawędzi położenia pierwotnego,
- `right: wartość;` — przesunięcie o określoną wartość w stosunku do prawej krawędzi położenia pierwotnego,
- `bottom: wartość;` — przesunięcie o określoną wartość w stosunku do dolnej krawędzi położenia pierwotnego.

Można łączyć ze sobą różne parametry. Trzeba jednak pamiętać o tym, że parametr `left` ma pierwszeństwo przed `right`, a parametr `top` ma pierwszeństwo przed `bottom`.

Wartość parametru *parametr* oznacza określoną odległość od danej krawędzi; odległość ta powinna być podawana w jednostkach długości. Wpisanie wartości ujemnej spowoduje przesunięcie w przeciwną stronę. Można użyć słowa `auto`, aby ustawić wartość domyślną.

## Przykład 3.86

```

h1 {
    position: absolute;
    top: 100px;
    left: 200px;
}

img {
    position: relative;
    left: 350px;
    bottom: 150px;
}

div {
    position: fixed;
    top: 20px;
    right: 10px;
    border: dotted 1px;
    padding: 5px;
    background-color: yellow;
}

```

## Przezroczystość

Przy użyciu atrybutu `opacity` można zdefiniować przezroczystość dla elementów wstawianych na stronę.

```
selektor { opacity: nieprzezroczystość; }
```

Parametr `nieprzezroczystość` to liczba z przedziału od `0,0` do `1,0`. `0` oznacza całkowitą przezroczystość, `1` — brak przezroczystości.

Zdefiniowany poziom przezroczystości jest nadawany wszystkim elementom znajdującym się wewnątrz określonego elementu. Elementom wewnętrznym nie można zmniejszyć przezroczystości (np. nie mogą one stać się zupełnie nieprzezroczyste), ale można im nadać większą przezroczystość w stosunku do elementu zewnętrznego.

## Ćwiczenie 3.12

Wykorzystując poznane narzędzia do pozycjonowania elementów, na utworzonej stronie, w rogach rozmieść cztery zdjęcia zgodnie z rysunkiem 3.41. Zaprojektuj stronę tak, aby lista widoczna z prawej była elementem nieruchomym w czasie przewijania zawartości strony. Wstawione elementy odpowiednio sformatuj.

**Rysunek 3.41.**

Strona zawiera pozycjonowanie obrazów

Zwiedzamy Afrykę  
Afryka to drugi pod względem wielkości kontynent na Ziemi. Zajmuje 30 37 mln km<sup>2</sup>, czyli ponad 20% powierzchni lądowej świata. Przechodzą przez obydwa półkuli.

Jedziemy do Azji  
Azja to kontynent leżący Eurazję, największy kontynent na Ziemi. W historycznych i kulturalnych sama Azja była również nazywana kontynentem. Nazywana jest często kontynentem wielkich kontrastów geograficznych.

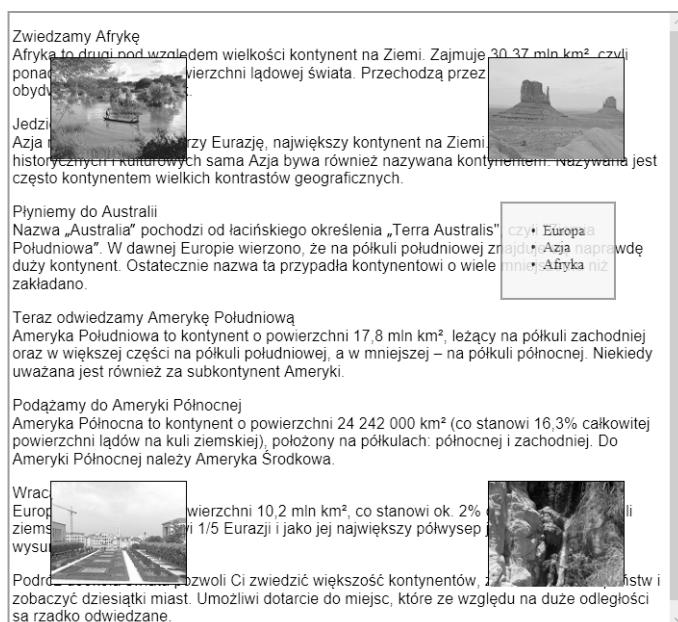
Plynimy do Australii  
Nazwa „Australia” pochodzi od łacińskiego określenia „Terra Australis” (czyli „Europa Południowa”). W dawnej Europie wierzano, że na półkuli południowej znajduje się duży kontynent. Ostatecznie nazwa ta przypadła kontynentowi o wiele mniejszym rozmiarach.

Teraz odwiedzamy Amerykę Południową  
Ameryka Południowa to kontynent o powierzchni 17,8 mln km<sup>2</sup>, leżący na półkuli zachodniej oraz w większej części na półkuli południowej, a w mniejszej – na półkuli północnej. Niekiedy uważana jest również za subkontynent Ameryki.

Podążamy do Ameryki Północnej  
Ameryka Północna to kontynent o powierzchni 24 242 000 km<sup>2</sup> (co stanowi 16,3% całkowitej powierzchni lądu na kuli ziemskiej), położony na półkulach: północnej i zachodniej. Do Ameryki Północnej należy Ameryka Środkowa.

Wracamy do Europy  
Europa to kontynent o powierzchni 10,2 mln km<sup>2</sup>, co stanowi ok. 2% całkowitej powierzchni lądu na kuli ziemskiej, i jest jednym z pięciu kontynentów, które zajmują 1/5 Eurazji i jako jej największy półwysep tworzą ją. Wysunięty na północny zachód kontynent Europy to Skandynawia.

Podróżujemy po Europie  
Podróżując po Europie, możesz zwiedzić wiele miast. Wysunięty na północny zachód kontynent Europy to Skandynawia. Wyspy, na których znajdują się największe miasta Europy, są rzadko odwiedzane.


**Rozwiązań**

Rozwiązaniem ćwiczenia 3.12 może być następujący kod:

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Pozycjonowanie</title>
    <meta charset="UTF-8">
    <style>
        p {
            font-size: 13pt;
            font-family: Arial;
        }
        div {
            width: 150px;
            height: 113px;
            border: solid 1px black;
        }
        #lista {

```

```
position: fixed;  
top: 210px;  
right: 60px;  
border: double 2px;  
padding: 5px;  
background-color: #f0f0f0;  
width: 3cm;  
height: 2.5cm;  
opacity: 0.8  
}  
  
#blok1 {  
    position: absolute;  
    top: 50px;  
    left: 50px;  
}  
  
#blok2 {  
    position: absolute;  
    top: 50px;  
    right: 50px;  
}  
  
#blok3 {  
    position: absolute;  
    bottom: 50px;  
    right: 50px;  
}  
  
#blok4 {  
    position: absolute;  
    bottom: 50px;  
    left: 50px;  
}  
  
</style>  
</head>  
<body>  
    <div id="lista">  
        <ul>
```

```
<li>Europa</li>
<li>Azja</li>
<li>Afryka</li>
</ul>
</div>
<div id="blok1">

</div>
<div id="blok2">

</div>
<div id="blok3">

</div>
<div id="blok4">

</div>
<p>
Zwiedzamy Afrykę<br>
Afryka to drugi pod względem wielkości kontynent na Ziemi. Zajmuje 30,37 mln km2, czyli ponad 20,3% ogólnej powierzchni lądowej świata. Przechodzą przez niego południk 0°, obydwa zwrotniki i równik.
</p>
<p>
Jedziemy do Azji<br>
Azja razem z Europą tworzy Eurazję, największy kontynent na Ziemi. Z powodów historycznych i kulturowych sama Azja bywa również nazywana kontynentem. Nazywana jest często kontynentem wielkich kontrastów geograficznych.
</p>
<p>
Płyniemy do Australii<br>
Nazwa Australia pochodzi od łacińskiego określenia "Terra Australis", czyli "Ziemia Południowa". W dawnej Europie wierzano, że na półkuli południowej znajduje się naprawdę duży kontynent. Ostatecznie nazwa ta przypadła kontynentowi o wiele mniejszemu niż zakładano.
</p>
```

<p>

Teraz odwiedzamy Amerykę Południową<br>

Ameryka Południowa to kontynent o powierzchni 17,8 mln km<sup>2</sup> leżący na półkuli zachodniej oraz w większej części na półkuli południowej, a w mniejszej – na półkuli północnej. Niekiedy uważana jest również za subkontynent Ameryki.

</p>

<p>Podążamy do Ameryki Północnej<br>

Ameryka Północna to kontynent o powierzchni 24 242 000 km<sup>2</sup> (co stanowi 16,3% całkowitej powierzchni lądów na kuli ziemskiej), położony na półkulach: północnej i zachodniej. Do Ameryki Północnej należy Ameryka Środkowa.

</p>

<p>Wracamy do Europy<br>

Europa to kontynent o powierzchni 10,2 mln km<sup>2</sup>, co stanowi ok. 2% całej powierzchni kuli ziemskiej. Europa stanowi 1/5 Eurazji i jako jej największy półwysep jest najbardziej wysunięta na zachód.

</p>

<p>

Podróż dookoła świata pozwoli Ci zwiedzić większość kontynentów, zawitać do wielu państw i zobaczyć dziesiątki miast. Umożliwi dotarcie do miejsc, które ze względu na duże odległości są rzadko odwiedzane.

</p>

</body>

</html>

### Ćwiczenie 3.13

Wykorzystując poznane narzędzia do pozycjonowania elementów, ułóż zdjęcia na stronie podobnie jak na rysunku 3.42.

#### Rysunek 3.42.

Przykładowe położenie elementów na stronie uzyskane w wyniku pozycjonowania



### Rozwiążanie

Rozwiązaniem ćwiczenia 3.13 może być następujący kod:

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Pozycjonowanie relative</title>
    <meta charset="UTF-8">
    <style>
        #blok1 {position: relative; left: 40px; }
        #blok2 {position: relative; left: 80px; top: -80px; }
        #blok3 {position: relative; left: 120px; top: -160px; }
        #blok4 {position: relative; left: 160px; top: -240px; }
    </style>
</head>
<body>
    <div id="blok1">
        
    </div>
    <div id="blok2">
        
    </div>
    <div id="blok3">
        
    </div>
    <div id="blok4">
        
    </div>
</body>
</html>
```

### Nakładanie elementów (warstwy)

Gdy kilka elementów nachodzi na siebie, to można określić, w jaki sposób te elementy będą nakładane. Służy do tego atrybut `position`. Nakładającym się elementom można nadać numer (`z-index`) i zastosować regułę, że elementy z wyższym numerem będą nakładane na elementy z niższym numerem.

```
selektor {position: rodzaj; parametry; z-index: numer;} 
```

Wyrażenie `position: rodzaj; parametry;` określa parametry pozycjonowania.

Parametr `numer` mówi o kolejności nakładania pozycjonowanych elementów. Element, któremu nadamy numer 1, znajdzie się na samym spodzie i będzie zakryty przez inne elementy.

### Przykład 3.87

```
#blok3 {position: relative; left: 120px; top: -160px; z-index: 4;} 
```

Poprawnie interpretowane są również wartości ujemne.

### Przykład 3.88

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Pozycjonowanie relative</title>
    <meta charset="UTF-8">
    <style>
        #blok1 {position: relative; left: 40px; z-index: 1;}
        #blok2 {position: relative; left: 80px; top: -80px; z-index: 2;}
        #blok3 {position: relative; left: 120px; top: -160px; z-index: 4;}
        #blok4 {position: relative; left: 160px; top: -240px; z-index: 3;}
    </style>
</head>
<body>
    <div id="blok1">
        
    </div>
    <div id="blok2">
        
    </div>
    <div id="blok3">
        
    </div>
    <div id="blok4">
        
    </div>
</body>
</html> 
```

Wynik wykonania kodu został zilustrowany na rysunku 3.43.

### Rysunek 3.43.

Zmiana kolejności elementów



## Obcinanie elementu

Do wycięcia z elementu jego fragmentu w kształcie prostokąta służy atrybut `position: absolute;` `clip: rect(góra, prawo, dół, lewo);` z odpowiednimi parametrami.

```
selektor {position: absolute; clip: rect(góra, prawo, dół, lewo);}
```

Wartości: `góra`, `prawo`, `dół`, `lewo` oznaczają współrzędne wyciętego prostokąta. Wartości te są określane względem lewego górnego rogu elementu. Powinny być podawane w jednostkach długości. Można używać określenia `auto`, co oznacza pominięcie cięcia z danej strony.

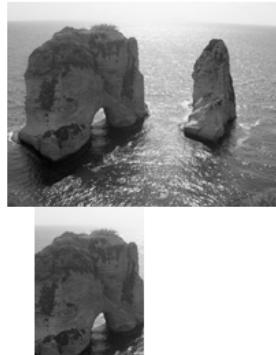
### Przykład 3.89

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Obcinanie elementu</title>
    <meta charset="UTF-8">
    <style>
        #blok {
            position: absolute;
            clip: rect(0px, 100px, 110px, 20px);
        }
    </style>
</head>
<body>
    
    <div id="blok"></div>
</body>
</html>
```

Wynik wykonania kodu został pokazany na rysunku 3.44.

#### Rysunek 3.44.

Obcinanie elementu



## Wyrównanie w pionie

Atrybut `vertical-align` pozwala na zdefiniowanie wyrównania elementu w stosunku do innych elementów strony.

```
selektor {vertical-align: sposób;}
```

Jako wartość parametru *sposób* należy podać:

- `baseline` — wyrównuje linię bazową elementu do linii bazowej elementu nadzewnętrznego;
- `middle` — ustawia element na środku wysokości elementów sąsiednich;
- `text-top` — wyrównuje górną krawędź elementu do górnej krawędzi tekstu elementu nadzewnętrznego;
- `text-bottom` — wyrównuje podstawę elementu do podstawy tekstu elementu nadzewnętrznego;
- `super` — ustawia element jako indeks górnny;
- `sub` — ustawia element jako indeks dolny;
- `top` — wyrównuje górną krawędź elementu do górnej krawędzi najwyższego elementu w linii;
- `bottom` — wyrównuje podstawę elementu do podstawy elementu położonego najniżej w linii.

### Przykład 3.90

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
  <title>Wyrównanie w pionie</title>
  <meta charset="UTF-8">
```

```
<style>
body {background-color: #d0d0d0;}
span {vertical-align: 4mm;}
</style>
</head>
<body>
<h3>Atrybut <span>vertical-align</span> definiuje wyrównanie elementu.</h3>
</body>
</html>
```

Wynik wykonania kodu został pokazany na rysunku 3.45.

Atrybut **vertical-align** definiuje wyrównanie elementu.

**Rysunek 3.45.** Wyrównanie elementu w stosunku do innych elementów

## Ustawienie w poziomie

Do ustawiania elementu względem elementów, które z nim sąsiadują w poziomie, służy atrybut float. Potocznie mówi się, że element „pływą” po lewej lub prawej stronie okna przeglądarki lub bloku.

```
selektor {float: sposób;
```

Parametr *sposób* może przyjmować wartości:

- left — element zostanie ustawiony po lewej stronie względem sąsiednich elementów;
- right — element zostanie ustawiony po prawej stronie względem sąsiednich elementów;
- none — element nie będzie sąsiadował z innymi elementami.

Najczęściej atrybut jest wykorzystywany do określenia sposobu oblewania tekstem elementu płynącego (float).

### Przykład 3.91

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
<title>Pływanie elementów</title>
<meta charset="UTF-8">
```

```

<style>
#blok1 {float:left;}
#blok2 {float:right;}
</style>

</head>

<body>



<p>Afryka to drugi pod względem wielkości kontynent na Ziemi. Zajmuje 30,37 mln km2, czyli ponad 20,3% ogólnej powierzchni lądowej świata. Przechodzą przez niego południk 0°, obydwa zwrotniki i równik. Większość granic Afryki to granice morskie. Afrykę opływają dwa oceany: Atlantycki i Indyjski. Jednym morzem Atlantyku jest opływające od północy Morze Śródziemne, które oddziela Afrykę od Europy.</p>



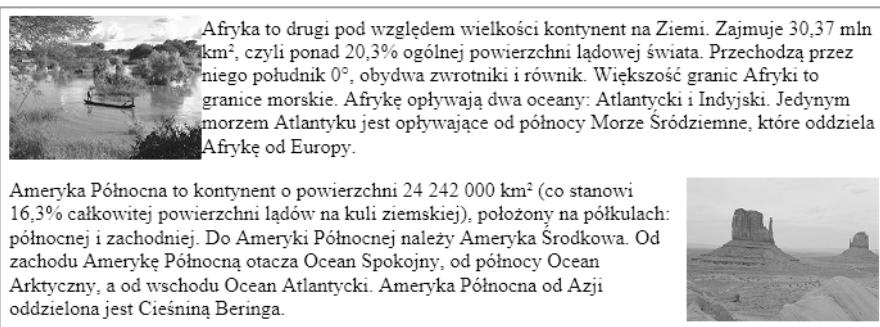
<p>Ameryka Północna to kontynent o powierzchni 24 242 000 km2 (co stanowi 16,3% całkowitej powierzchni lądów na kuli ziemskiej), położony na półkulach: północnej i zachodniej. Do Ameryki Północnej należy Ameryka Środkowa. Od zachodu Amerykę Północną otacza Ocean Spokojny, od północy Ocean Arktyczny, a od wschodu Ocean Atlantycki. Ameryka Północna od Azji oddzielona jest Cieśniną Beringa.</p>

</body>

</html>

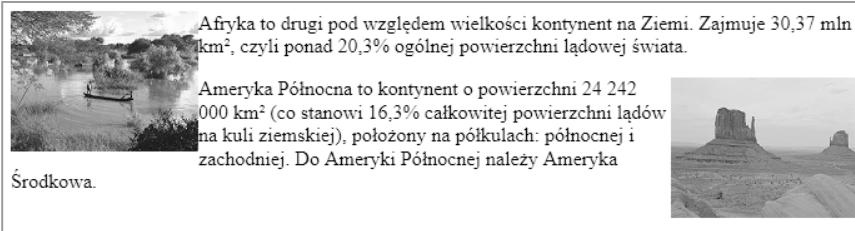
```

Wynik wykonania kodu zilustrowano na rysunku 3.46.



**Rysunek 3.46.** Zastosowanie atrybutu float do ustawienia tekstu wokół zdjęcia

Jeżeli tekst umieszczony z prawej lub lewej strony zdjęcia będzie zbyt krótki, to zdjęcie nie zostaną ułożone jedno pod drugim, tak jak na rysunku wyżej, lecz znajdą się obok siebie (rysunek 3.47). Dodatkowo ułożenie zdjęć będzie zależało od rozdzielczości ekranu i rozmiaru okna przeglądarki.

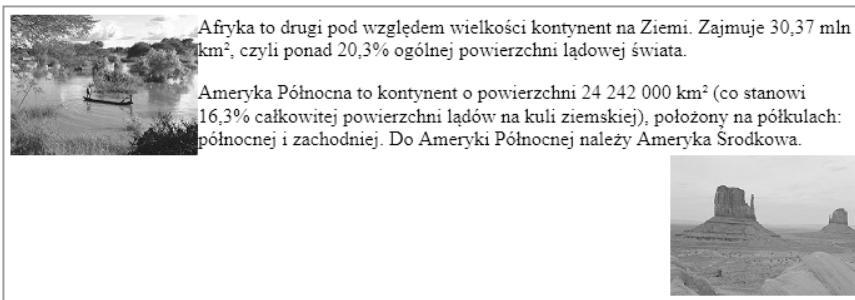
**Rysunek 3.47.** Błędy przy rozmieszczaniu tekstu wokół zdjęcia

Aby temu zapobiec, należy użyć atrybutu `clear`, który ustala pozycję kolejnych elementów względem elementu pływającego (rysunek 3.48).

```
selektor {clear: sposób;}
```

Wartość atrybutu `clear` określa sposób zachowania się kolejnych elementów względem elementu pływającego:

- `left` — lewy bok następnego elementu nie przylega do elementu pływającego;
- `right` — prawy bok następnego elementu nie przylega do elementu pływającego;
- `both` — żaden bok następnego elementu nie przylega do elementu pływającego;
- `none` — brak ograniczeń.

**Rysunek 3.48.** Efekt zastosowania atrybutu `clear`

### Ćwiczenie 3.14

Popraw podany w przykładzie 3.91 kod tak, aby drugie zdjęcie było wyświetlane poniżej pierwszego zdjęcia niezależnie od długości tekstu występującego obok zdjęcia oraz bez względu na wielkość okna przeglądarki i rozdzielczość ekranu (rysunek 3.48).

### Kolumny na stronie

Atrybut `float` może być wykorzystany do tworzenia kolumn w dokumencie. W tym celu musimy zawartość każdej kolumny zawrzeć w znaczniku `<div>`. Następnie, np. przy trzech kolumnach, należy określić szerokość każdego znacznika `<div>`. Po zdefiniowaniu dla każdego znacznika `<div>` atrybutu `float: left` uzyskamy „pływanie” kolejnych kolumn z lewej strony.

## Przykład 3.92

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Kolumny na stronie</title>
    <meta charset="UTF-8">
    <style>
        div {width:32%; float:left; margin: 5px;}
        .blok {float:left; margin: 10px;}
    </style>
</head>
<body>
    <div>
        
        <p>Wartość określa sposób pozycjonowania elementu i może być: static - przywraca normalne pozycjonowanie elementu. Przydatne wtedy, gdy wcześniej, np. w arkuszach stylów, została podana deklaracja pozycjonowania tego typu elementów. Użycie w definicji stylu inline dla wybranego elementu deklaracji position: static zniesie tę deklarację dla danego elementu.</p>
    </div>
    <div>
        
        <p>Wartość określa sposób pozycjonowania elementu i może być: static - przywraca normalne pozycjonowanie elementu. Przydatne wtedy, gdy wcześniej, np. w arkuszach stylów, została podana deklaracja pozycjonowania tego typu elementów. Użycie w definicji stylu inline dla wybranego elementu deklaracji position: static zniesie tę deklarację dla danego elementu.</p>
    </div>
    <div>
        
        <p>Wartość określa sposób pozycjonowania elementu i może być: static - przywraca normalne pozycjonowanie elementu. Przydatne wtedy, gdy wcześniej, np. w arkuszach stylów, została podana deklaracja pozycjonowania tego typu elementów. Użycie w definicji stylu inline dla wybranego elementu deklaracji position: static zniesie tę deklarację dla danego elementu.</p>
    </div>
</body>
</html>
```

Na rysunku 3.49 przedstawiono wynik wykonania kodu.

 Wartość określa sposób pozycjonowania elementu i może być: static - przywraca normalne pozycjonowanie elementu. Przydatne wtedy, gdy wcześniej, np. w arkuszach stylów, została podana deklaracja pozycjonowania tego typu elementów. Użycie w definicji stylu inline dla wybranego elementu deklaracji position: static zniесie tę deklarację dla danego elementu.	 Wartość określa sposób pozycjonowania elementu i może być: static - przywraca normalne pozycjonowanie elementu. Przydatne wtedy, gdy wcześniej, np. w arkuszach stylów, została podana deklaracja pozycjonowania tego typu elementów. Użycie w definicji stylu inline dla wybranego elementu deklaracji position: static zniесie tę deklarację dla danego elementu.	 Wartość określa sposób pozycjonowania elementu i może być: static - przywraca normalne pozycjonowanie elementu. Przydatne wtedy, gdy wcześniej, np. w arkuszach stylów, została podana deklaracja pozycjonowania tego typu elementów. Użycie w definicji stylu inline dla wybranego elementu deklaracji position: static zniесie tę deklarację dla danego elementu.
--	--	--

**Rysunek 3.49.** Dokument z trzema kolumnami

### Ćwiczenie 3.15

Przygotuj krótkie teksty z informacją o zimowej olimpiadzie w Pjongczang oraz obrazy pokazujące zmagania sportowców. Za pomocą poznanych znaczników języka HTML zaprojektuj szablon strony internetowej na ten temat. Zaproponuj teksty, które znajdują się w nagłówku i stopce strony. Umieść przygotowane wcześniej teksty i obrazy w odpowiednich blokach. Wykorzystując style CSS i poznane narzędzia, zaprojektuj stronę internetową.

### Ćwiczenie 3.16

Utwórz na stronie galerię zdjęć na temat samochodów przyszłości. Rozplanuj położenie obrazków tak, aby obok siebie wyświetlane były po trzy zdjęcia. Układ zdjęć nie powinien zostać zaburzony przy zmianie wielkości okna przeglądarki. Określ minimalny rozmiar każdego zdjęcia.

## 3.6.4. Sposoby wyświetlania elementów

Wszystkie definiowane elementy są wyświetlane na stronie w sposób domyślny. Jeżeli chcemy zrezygnować z takiego pokazywania elementu, możemy zdefiniować atrybut display. Za jego pomocą można określić, jak dany element będzie wyświetlany na stronie.

```
selektor {display: sposób;}
```

Parametr atrybutu określa sposób pokazywania elementu na stronie i może przyjmować wartości:

- **block** — element wyświetlany jako element blokowy z odstępami od góry i od dołu;
- **inline** — element wyświetlany w linii z innymi elementami;
- **list-item** — element wyświetlany jako element wykazu listy `<li>`;
- **none** — element nie będzie wyświetlany;

- `inline-block` — element wyświetlany podobnie jak element zastępowany; mimo że element ma charakter blokowy (odstęp, obramowanie, np. `<div>`), jest wyświetlany jako liniowy;
- `run-in` — jeżeli po elemencie występuje element będący blokiem, to ten pierwszy element zostanie wyświetlony w jednej linii z blokiem, w przeciwnym razie element zostanie wyświetlony w bloku.

### UWAGA

Element zastępowany to element, dla którego istnieje tylko wymiar wewnętrzny, np.: `<img>` (zawartość elementu jest zastępowana obrazem określonym przez atrybut `src`), `<input>`, `<textarea>`, `<select>`, `<object>`.

Atrybut `display` jest bardzo przydatny, jeżeli chcemy zrezygnować z domyślnego sposobu wyświetlania. Jeżeli np. element jest wyświetlany w bloku (czyli dodawane są linie odstępu między sąsiednimi elementami), to po zastosowaniu `display: inline` elementy zostaną ustalone w jednej linii.

### Element blokowy

Elementy blokowe zawsze są wyświetlane od nowej linii i zajmują całą dostępną na stronie szerokość. Elementami blokowymi są m.in.:

- `<div>`,
- `<h1> ... <h6>`,
- `<p>`,
- `<form>`,
- `<header>`,
- `<footer>`,
- `<section>`.

### Element wbudowany (element InLine)

Element wbudowany (*InLine*) jest wyświetlany w miejscu wstawienia (nie rozpoczyna się od nowej linii) i zajmuje na szerokość tyle miejsca, ile potrzeba na wyświetlenie jego zawartości. Elementami wbudowanymi są m.in.:

- `<span>`,
- `<a>`,
- `<img>`.

Atrybut `display` może zostać wykorzystany do zamiany sposobu wyświetlania elementów strony. I tak element *InLine* może zostać zmieniony na element blokowy, a element blokowy na element *InLine*.

Przykładem jest tworzenie za pomocą elementu <li> menu poziomego.

Domyślne wyświetlanie elementu <li> zostało pokazane na rysunku 3.50.

### Rysunek 3.50.

Domyślne  
wyświetlanie elementów



Podaną listę utworzoną za pomocą elementu blokowego przekształcimy na element *InLine*, aby uzyskać menu poziome.

### Przykład 3.93

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Elementy wbudowane</title>
    <meta charset="UTF-8">
    <style>
        body {background-color: #d9d9d9;}
        li {display: inline;}
    </style>
</head>
<body>
    <h3>Lista kontynentów:</h3>
    <ul>
        <li><a href="strona_E.html">Europa</a></li>
        <li><a href="strona_Az.html">Azja</a></li>
        <li><a href="strona_Af.html">Afryka</a></li>
    </ul>
</body>
</html>
```

Rysunek 3.51 przedstawia wynik wykonania kodu.

### Lista kontynentów:

Europa Azja Afryka

### Rysunek 3.51.

Zastosowanie atrybutu  
display: InLine

W podobny sposób element typu *InLine*, np. element <a>, można przekształcić w element blokowy.

### Przykład 3.94

```
<!DOCTYPE html>

<html lang="pl-PL">
<head>
    <title>Zmiana sposobu wyświetlania elementu</title>
    <meta charset="UTF-8">
    <style>
        a {display: block;}
    </style>
</head>
<body>
    <h3>Lista kontynentów:</h3>
    <a href="strona_E.html">Europa</a>
    <a href="strona_Az.html">Azja</a>
    <a href="strona_Af.html">Afryka</a>
</body>
</html>
```

### Wartość inline-block

Elementy, którym został nadany atrybut *display* z wartością *inline-block*, są elementami wbudowanymi, ale mogą posiadać szerokość (*width*) i wysokość (*height*).

Do ustawiania elementu względem innych elementów, które z nim sąsiadują w poziomie, służy atrybut *float*. Atrybut *clear* pozwala ustalić pozycję kolejnych elementów względem elementu pływającego.

### Przykład 3.95

```
<!DOCTYPE html>

<html lang="pl-PL">
<head>
    <title>Pływanie elementów</title>
    <meta charset="UTF-8">
    <style>
        .blok_p {
            float: left;
        }
    </style>
</head>
<body>
</body>
</html>
```

```
width: 150px;  
height: 120px;  
margin: 10px;  
border: 3px solid #d9d9d9;  
}  
.blok_s {  
clear: left;  
border: 3px solid green;  
}  
</style>  
</head>  
<body>  
<h2>Zdjęcia z podróży</h2>  
<div class="blok_p"></div>  
<div class="blok_p"></div>  
<div class="blok_p"></div>  
<div class="blok_p"></div>  
<div class="blok_p"></div>  
<div class="blok_p"></div>  
<div class="blok_s"></div>  
</body>  
</html>
```

Podobny efekt można uzyskać za pomocą wartości inline-block atrybutu display i wtedy nie ma potrzeby użycia atrybutu clear.

### Przykład 3.96

```
<!DOCTYPE html>  
<html lang="pl-PL">  
<head>  
<title>Działanie wartości inline-block</title>  
<meta charset="UTF-8">  
<style>  
.blok_p {  
display: inline-block;  
width: 150px;
```

```

height: 120px;
margin: 10px;
border: 3px solid #d9d9d9;
}

.blok_s {
    border: 3px solid green;
}

</style>

</head>

<body>

<h2>Zdjęcia z podróży</h2>

<div class="blok_p"></div>
<div class="blok_p"></div>
<div class="blok_p"></div>
<div class="blok_p"></div>
<div class="blok_p"></div>
<div class="blok_p"></div>
<div class="blok_s"></div>

</body>
</html>

```

## Ukrywanie elementów

Ukrywanie elementów strony można uzyskać po ustawieniu atrybutu `display` na `none`. Innym atrybutem, za pomocą którego można sterować wyświetlaniem elementów na stronie, jest atrybut `visibility`.

```
selektor {visibility: typ;}
```

Jako wartość parametru `typ` należy podać:

- `visible` — element będzie widoczny na ekranie.
- `hidden` — element będzie ukryty, niewidoczny na ekranie. W jego miejscu pojawi się pusta przestrzeń.
- `collapse` — zastosowana dla tabeli ukrywa całą zawartość wiersza lub kolumny. Dla innych elementów działa jak `hidden`.

Podobny efekt jak w przypadku `visibility: hidden` można uzyskać, stosując atrybut `display: none`. Różnica polega na tym, że jeśli użyjemy `display: none`, element zostanie całkowicie usunięty ze strony.



## 3.7. Menu w języku CSS

Zaprojektowane na stronie internetowej menu umożliwia nawigację w jej obrębie. Paski nawigacyjne buduje się najczęściej na podstawie standardowej listy nieuporządkowanej. Lista jest zbiorem występujących po sobie pozycji, podobnie budowane jest menu. Za pomocą języka CSS można utworzoną listę zamienić w atrakcyjne menu w postaci pasków nawigacyjnych.

### Lista menu

Budowanie menu zaczynamy od utworzenia listy w języku HTML. Lista taka powinna zawierać linki do odpowiednich miejsc wywoływanych przez menu.

#### Przykład 3.97

```
<ul>
  <li><a href="start.html">Start</a></li>
  <li><a href="nowe.html">Aktualności</a></li>
  <li><a href="szkolenia.html">Szkolenia</a></li>
  <li><a href="uslugi.html">Usługi</a></li>
</ul>
```

Po zdefiniowaniu zawartości menu można przystąpić do jego formatowania z użyciem stylów CSS. Dołączenie deklaracji stylów pozwoli na stworzenie elastycznego i funkcjonalnego menu nawigacyjnego. Sam wygląd menu zostanie zmieniony tylko przez dołączenie arkusza stylów z odpowiednimi deklaracjami.

Dobrym rozwiązaniem jest zdefiniowanie dla znacznika `<ul>` identyfikatora lub klasy, aby w wyniku formatowania tworzonego menu nie doprowadzić do przeformatowania zwykłych list, które mogą się znaleźć na stronie. Pozwoli to również na umieszczanie na stronie kilku różnie wyglądających zestawów menu.

Zanim przystąpimy do definiowania menu, na przygotowanej liście trzeba usunąć punktory oraz wyzerować marginesy. Należy również zadeklarować wyświetlanie listy w bloku oraz określić szerokość tego bloku.

#### Przykład 3.98

```
ul {display: block; list-style-type: none; margin: 0; padding: 0;}
```

Tak zapisana deklaracja w arkuszu stylów jest podstawą do tworzenia pionowych i poziomych pasków nawigacyjnych.

### 3.7.1. Menu pionowe

Aby utworzyć pionowe menu nawigacyjne, należy do podanych deklaracji dla list dołączyć deklarację dotyczącą znacznika `<a>` występującego w zdefiniowanej liście oraz znacznika `<li>`.

## Przykład 3.99

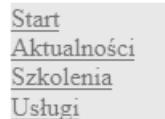
```
li a {display: block; width:100;}
```

Atrybut `display: block` wyświetli linki jako elementy blokowe, a `width:100` ustawia szerokość menu.

## Przykład 3.100

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Menu pionowe</title>
    <meta charset="UTF-8">
    <style>
        ul {
            list-style-type: none;
            margin: 0;
            padding: 0;
            width: 100px;
        }
        li a {
            display: block;
            background-color: #dddddd;
        }
    </style>
</head>
<body>
    <ul>
        <li><a href="strona_n.html">Start</a></li>
        <li><a href="strona_a.html">Aktualności</a></li>
        <li><a href="strona_k.html">Szkolenia</a></li>
        <li><a href="strona_m.html">Usługi</a></li>
    </ul>
</body>
</html>
```

W wyniku wykonania kodu powstanie menu pionowe (rysunek 3.52).



**Rysunek 3.52.**  
Podstawowe  
menu pionowe

Utworzoną listę można za pomocą kaskadowych arkuszy stylów zmienić w atrakcyjne menu.

### Przykład 3.101

```
ul a:link, ul a:visited {  
    display: block;  
    width: 168px;  
    text-decoration: none;  
    background-color: #c0c0c0;  
    color: black;  
    padding: 4px;  
    border: 2px outset #c0c0c0;  
}  
  
ul a:hover {  
    border-style: inset;  
    color: blue;  
    padding: 6px 3px 3px 6px;  
}
```

W tej części definicji stylów został zdefiniowany wygląd przycisków menu.

Każdy z elementów listy zostanie wyświetlony w bloku, w związku z tym nie tylko tekst, ale cały blok stanie się odnośnikiem.

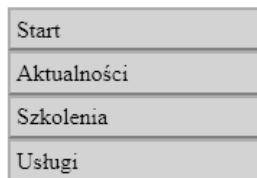
Deklaracje `a:link` oraz `a:visited` określają wygląd przycisków przed odwiedzeniem strony lub po jej odwiedzeniu (zniesione podkreślenie, ustalony kolor tła i kolor tekstu, margines wewnętrzny 4px i ramka 2px z ustawionym stylem ramki).

Deklaracja `a:hover` określa wygląd przycisku, gdy znajdzie się nad nim kurSOR myszy (zmiana stylu obramowania oraz przesunięcie marginesu wewnętrznego).

Przykładowe menu pionowe zostało pokazane na rysunku 3.53.

#### Rysunek 3.53.

Wygląd menu zdefiniowany za pomocą arkuszy stylów



Rozwiązaniem przykładu pokazanego na rysunku 3.53 może być kod z przykładu 3.102.

### Przykład 3.102

```
<!DOCTYPE html>

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

    <title>Menu pionowe</title>
    <meta charset="UTF-8">
    <style>

        ul, ul li {
            display: block;
            list-style: none;
            margin: 0;
            padding: 0;
        }

        ul {
            width: 200px;
        }

        ul a:link, ul a:visited {
            display: block;
            width: 166px;
            text-decoration: none;
            background-color: #c0c0c0;
            color: black;
            padding: 4px;
            border: 2px outset #c0c0c0;
        }

        ul a:hover {
            border-style: inset;
            color: blue;
            padding: 6px 3px 3px 6px;
        }

    </style>
</head>
<body>
    <ul>
        <li><a href="start.html">Start</a></li>
```

```
<li><a href="nowe.html">Aktualności</a></li>
<li><a href="szkolenia.html">Szkolenia</a></li>
<li><a href="uslugi.html">Usługi</a></li>
</ul>
</body>
</html>
```

### Ćwiczenie 3.17

Wykorzystując poznane narzędzia, utwórz na stronie internetowej menu pionowe. Zmieniając deklaracje stylów w arkuszu stylów, nadaj elementom menu atrakcyjny wygląd.

## 3.7.2. Menu poziome

Przy tworzeniu menu poziomego również korzystamy ze zdefiniowanej wcześniej listy nieuporządkowanej.

Domyślnie lista jest tworzona w układzie pionowym. Ułożenie pozycji listy obok siebie można uzyskać na dwa sposoby:

- wyświetlając kolejne elementy listy w linii (atribut `display: inline;`);
- ustawiając płynwanie dla każdej pozycji listy (atribut `float`).

Podobnie jak w przypadku menu pionowego, w przygotowanym menu należy usunąć punktory oraz wyzerować marginesy.

### Przykład 3.103

```
ul, li {display: block; list-style: none; margin: 0; padding: 0;}
```

### Wyświetlanie kolejnych elementów listy w linii

Tak jak przy menu pionowym, utworzoną listę można sformatować za pomocą kaskadowych arkuszy stylów. Wyświetlanie kolejnych elementów listy w linii zostało zrealizowane za pomocą atrybutu `display: inline`.

### Przykład 3.104

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Menu poziome_1</title>
    <style>
        ul, ul li {
```

```
display: block;
list-style: none;
margin: 0;
padding: 0;
}

ul li {
    display: inline;
    white-space: nowrap;
}

ul a:link, ul a:visited {
    text-decoration: none;
    background-color: #c0c0c0;
    color: black;
    padding: 4px;
    border: 2px outset #c0c0c0;
}

ul a:hover {
    border-style: inset;
    color: blue;
    padding: 6px 3px 3px 6px;
}

</style>

</head>

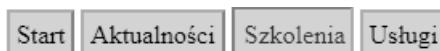
<body>

<ul>
    <li><a href="start.html">Start</a></li>
    <li><a href="nowe.html">Aktualności</a></li>
    <li><a href="szkolenia.html">Szkolenia</a></li>
    <li><a href="uslugi.html">Usługi</a></li>
</ul>

</body>
</html>
```

Dodane polecenie `white-space: nowrap;` blokuje zawijanie tekstu w każdej pozycji menu. Jeśli wszystkie pozycje menu nie zmieszcza się w jednej linii, podział nastąpi między elementami listy, a nie wewnątrz elementu. Elementy zmieniają swoje położenie

w odpowiedzi na `hover`. Po najechaniu na wybrany element menu zmienia on swoje położenie (rysunek 3.54).



**Rysunek 3.54.** Menu poziome — kolejne elementy listy w linii

Metoda ta daje poprawne efekty i nie wpływa na elementy znajdujące się poniżej menu. Nie można jednak za jej pomocą zdefiniować jednakowej szerokości pozycji menu.

## Float dla każdej pozycji listy

Wyświetlanie kolejnych elementów listy w linii zostało zrealizowane za pomocą atrybutu `float`.

### Przykład 3.105

```
<!DOCTYPE html>

<html lang="pl-PL">
<head>
    <title>Menu poziome</title>
    <meta charset="UTF-8">
    <style>
        ul li {
            display: inline;
            white-space: nowrap;
        }
        ul, li {
            float: left;
        }
        ul a:link, ul a:visited {
            text-decoration: none;
            background-color: #c0c0c0;
            color: black;
            padding: 4px;
            border: 2px outset #c0c0c0;
        }
        ul a:hover {
            border-style: inset;
        }
    </style>
</head>
<body>
    <ul>
        <li><a href="#">Start</a></li>
        <li><a href="#">Aktualności</a></li>
        <li><a href="#">Szkolenia</a></li>
        <li><a href="#">Usługi</a></li>
    </ul>
</body>
</html>
```

```

    color: blue;
    padding: 6px 3px 3px 6px;
}
</style>
</head>
<body>
<ul>
    <li><a href="start.html">Start</a></li>
    <li><a href="nowe.html">Aktualności</a></li>
    <li><a href="szkolenia.html">Szkolenia</a></li>
    <li><a href="uslugi.html">Usługi</a></li>
</ul>
</body>
</html>
```

Efekt zastosowania tej metody ilustruje rysunek 3.55.



**Rysunek 3.55.** Menu poziome z zastosowaniem atrybutu float

Wygląd menu można poprawić, umieszczając elementy w blokach. Jeżeli elementy zostaną umieszczone w bloku, będzie można zdefiniować dla nich stałą szerokość.

### Przykład 3.106

```

<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Menu poziome_1</title>
    <meta charset="UTF-8">
    <style>
        ul, ul li {
            display: block;
            list-style: none;
            margin: 0;
            padding: 0;
        }
        ul li {
```

```
float: left;
margin-right: 10px;
}

ul a:link, ul a:visited {
    display: block;
    position: relative;
    text-decoration: none;
    background-color: #c0c0c0;
    color: black;
    padding: 4px;
    border: 2px outset #c0c0c0;
}

ul a:hover {
    border-style: inset;
    color: blue;
    padding: 6px 3px 3px 6px;
}

</style>
</head>
<body>
<ul>
    <li><a href="start.html">Start</a></li>
    <li><a href="nowe.html">Aktualności</a></li>
    <li><a href="szkolenia.html">Szkolenia</a></li>
    <li><a href="uslugi.html">Usługi</a></li>
</ul>
</body>
</html>
```

Po umieszczeniu elementów menu w blokach wynik w przeglądarce jest prawidłowy (rysunek 3.56).



**Rysunek 3.56.** Elementy menu umieszczone w blokach

Dodatkowo można określić stałą szerokość bloków i wyśrodkowanie tekstu menu.

### Przykład 3.107

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Menu poziome_2</title>
    <meta charset="UTF-8">
    <style>
        ul, ul li {
            display: block;
            list-style: none;
            margin: 0;
            padding: 0;
        }
        ul li {
            float: left;
            margin-right: 5px;
        }
        ul a:link, ul a:visited {
            display: block;
            width: 100px;
            text-align: center;
            position: relative;
            text-decoration: none;
            background-color: #c0c0c0;
            color: black;
            padding: 4px;
            border: 2px outset #c0c0c0;
        }
        ul a:hover {
            border-style: inset;
            color: blue;
            padding: 6px 3px 3px 6px;
        }
    </style>
</head>
```

```

<body>
  <ul>
    <li><a href="start.html">Start</a></li>
    <li><a href="nowe.html">Aktualności</a></li>
    <li><a href="szkolenia.html">Szkolenia</a></li>
    <li><a href="uslugi.html">Usługi</a></li>
  </ul>
</body>
</html>

```

W wyniku wykonania kodu zostanie utworzone menu poziome jak na rysunku 3.57.



**Rysunek 3.57.** Elementy menu z określona stałą szerokością

Umieszczenie elementów w blokach daje możliwość zdefiniowania jednakowej szerokości wszystkich pozycji menu oraz ustalenia odległości między pozycjami. Mogą się natomiast pojawić problemy z prawidłowym wyświetleniem w niektórych przeglądarkach. Trzeba również pamiętać o przypisaniu właściwości `clear` następnemu elementowi strony w celu uniknięcia „przyklejenia” tego elementu do menu.

### Ćwiczenie 3.18

Zdefiniuj menu poziome, które będzie zawierało poziomy pasek nawigacyjny w wybranym kolorze tła. Gdy wskaźnik myszy zostanie ustawiony nad linkiem, kolor tła wybranego linku ulegnie zmianie (rysunek 3.58).



**Rysunek 3.58.** Przykład menu poziomego

### Rozwiązanie

Rozwiązaniem ćwiczenia 3.18 może być następujący kod:

```

<!DOCTYPE html>
<html lang="pl-PL">
<head>
  <title>Menu poziome</title>
  <meta charset="UTF-8">

```

```
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #33415d;
}
li {
    float: left;
}
li a {
    display: block;
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}
li a:hover {
    background-color: #0d6ba1;
}
</style>
</head>
<body>
<ul>
    <li><a href="start.html">Start</a></li>
    <li><a href="nowe.html">Aktualności</a></li>
    <li><a href="szkolenia.html">Szkolenia</a></li>
    <li><a href="uslugi.html">Usługi</a></li>
</ul>
</body>
</html>
```

### 3.7.3. Menu zagnieżdżone

Podobnie jak w przypadku menu pionowego lub poziomego, budując menu zagnieżdżone, należy zacząć od utworzenia listy. Lista powinna zawierać linki do odpowiednich miejsc wywoływanych przez menu.

#### Przykład 3.108

```
<ul id="blok">
    <li><a href="start.html">Start</a></li>
    <li><a href="aktualnosci.html">Aktualności</a></li>
    <li><a href="szkolenia.html">Szkolenia</a>
        <ul>
            <li><a href="kurs1.html">HTML</a></li>
            <li><a href="kurs2.html">CSS</a></li>
            <li><a href="kurs3.html">JavaScript</a></li>
        </ul>
    </li>
    <li><a href="uslugi.html">Usługi</a></li>
</ul>
```

Po zdefiniowaniu zawartości menu można przystąpić do tworzenia menu zagnieżdżonego. Ponieważ zaproponowana lista ma podlistę tylko na drugiej pozycji, nasze menu będzie się rozwijało jedynie na tej właśnie pozycji.

Tak jak w przypadku menu pionowego i poziomego, w przygotowanym menu trzeba usunąć punktory oraz wyzerować marginesy.

#### Przykład 3.109

```
ul, ul li {margin: 0; padding: 0;}
#blok li {list-style: none;}
```

W definicji stylów został zdefiniowany identyfikator `#blok`, który w treści dokumentu HTML został przypisany do znacznika listy `<ul>`.

Następnym etapem jest zdefiniowanie wyglądu menu.

#### Przykład 3.110

```
#blok, #blok ul {
    width: 140px;
}
#blok li {
    padding-bottom: 1px;
```

```
position: relative;  
}  
  
#blok ul {  
    position: absolute;  
    top: 0;  
    left: 160px;  
    padding-left: 1px;  
    visibility: hidden;  
}  
  
#blok li:hover ul {  
    visibility: visible;  
    width: auto;  
    height: auto;  
}
```

Listom została nadana szerokość 140px, elementy listy zostały rozdzielone przerwą (padding-bottom: 1px). Zdefiniowano położenie względne dla listy głównej w stosunku do strony internetowej oraz położenie podlisty w stosunku do listy głównej.

Dla wewnętrznej listy (#blok ul) zdefiniowano pozycjonowanie absolutne w stosunku do listy głównej oraz jej ukrycie (visibility: hidden;).

Gdy ustawimy kurSOR myszy na elemencie menu, wyświetli się podmenu.

Pozostało jeszcze zdefiniowanie, jak ma wyglądać element listy, gdy ustawimy nad nim kurSOR myszy. Aby element menu głównego był nadal podświetlony, gdy zostanie wybrany element podmenu, należy odpowiednio wykorzystać zasady dziedziczenia.

### Przykład 3.111

```
#blok a, #blok li:hover li a {  
    display: block;  
    font: 12px/30px verdana, sans-serif;  
    text-decoration: none;  
    padding: 0 10px;  
    width: 140px;  
    background-color: #c0c0cf;  
}  
  
#blok li:hover a, #blok li:hover li:hover a {  
    background-color: #333033;  
    color: white;  
}
```

Pierwsza linia definiuje wygląd każdego elementu menu, druga — wygląd elementu wskazanego kursorem myszy. Jeżeli będzie to element podmenu, element z menu głównego też pozostanie podświetlony.

Efekt pracy pokazano w przykładzie poniżej.

### Przykład 3.112

```
<!DOCTYPE html>

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

<title>Menu zagnieżdżone</title>

<meta charset="UTF-8">

<style>

ul, ul li {
    margin: 0;
    padding: 0;
}

#blok li {
    list-style: none;
}

#blok, #blok ul {
    width: 140px;
}

#blok ul {
    position: absolute;
    top: 0;
    left: 160px;
    padding-left: 1px;
    visibility: hidden;
}

#blok li {
    padding-bottom: 1px;
    position: relative;
}

#blok li:hover ul {
    visibility: visible;
```

```

width: auto;
height: auto;
}

#blok a, #blok li:hover li a {
display: block;
font: 12px/30px verdana, sans-serif;
text-decoration: none;
padding: 0 10px;
width: 140px;
color: black;
background-color: #c0c0cf;
}

#blok li:hover a, #blok li:hover li:hover a {
background-color: #5f699c;
color: white;
}

</style>

</head>
<body>

<ul id="blok">
<li><a href="aktualnosci.html">Aktualności</a></li>
<li><a href="szkolenia.html">Szkolenia</a>
<ul>
<li><a href="kurs1.html">HTML</a></li>
<li><a href="kurs2.html">CSS</a></li>
<li><a href="kurs3.html">JavaScript</a></li>
</ul>
</li>
<li><a href="uslugi.html">Usługi</a></li>
</ul>
</body>
</html>

```

Wynik wykonania kodu został pokazany na rysunku 3.59.

Aktualności	
Szkolenia	HTML
Usługi	CSS
	JavaScript

**Rysunek 3.59.**  
Menu zagnieżdżone

## Ćwiczenie 3.19

Korzystając z powyższego kodu, zdefiniuj menu zagnieżdżone dla pozostałych dwóch opcji menu głównego. Samodzielnie zaprojektuj dla tworzonej strony internetowej wygląd menu z zastosowaniem arkusza stylów i poleceń języka CSS.



## 3.8. Zasady projektowania szablonu strony internetowej

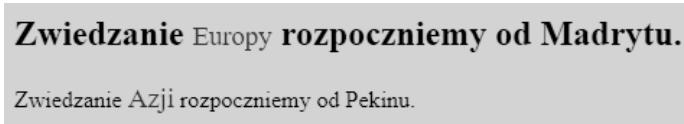
### 3.8.1. Definiowanie reguł

Reguły definiowane w arkuszu CSS dla znacznika dotyczą wszystkich użytych w dokumencie znaczników tego typu. Oznacza to, że jeżeli zdefiniujemy w arkuszu stylów wygląd znacznika `<span>`, to w każdym miejscu na stronie tekst otoczony znacznikiem `<span>` będzie wyglądał tak samo, niezależnie od tego, czy znajdzie się w nagłówku strony, w treści dokumentu, czy w menu.

#### Przykład 3.113

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Szablon strony - definiowanie reguł</title>
    <meta charset="UTF-8">
    <style>
        body {
            background-color: #c0c0c0;
        }
        span {
            font-size: 14pt;
            color: blue;
            font-weight: normal;
        }
    </style>
</head>
<body>
    <h2>Zwiedzanie <span>Europy</span> rozpoczniemy od Madrytu.</h2>
    <p>Zwiedzanie <span>Azji</span> rozpoczniemy od Pekinu.</p>
</body>
</html>
```

W podanym przykładzie znacznikiem `<span>` został otoczony fragment tekstu umieszczony w nagłówku `<h2>` oraz fragment akapitu. Wynik wykonania kodu widać na rysunku 3.60. Teksty sformatowane z użyciem znacznika `<span>` wyglądają tak samo.



**Rysunek 3.60.** Formatowanie tekstów otoczonych znacznikiem `<span>`

Aby odróżnić przy formatowaniu podobne elementy umieszczone w różnych miejscach dokumentu, możemy zastosować jedną z trzech technik:

- reguły kontekstowe,
- identyfikatory,
- klasy.

## Reguły kontekstowe

Reguły kontekstowe pozwalają na zdefiniowanie w arkuszu stylów wyglądu każdego elementu w zależności od jego położenia.

W podanym przykładzie teksty sformatowane z użyciem znacznika `<span>` wyglądają tak samo, mimo że znajdują się w różnych miejscach i byłoby wskazane, aby wyglądały inaczej. Jeżeli w trakcie definiowania reguł określmy kontekst znacznika `<span>`, będzie możliwe sformatowanie w inny sposób obu fragmentów tekstu.

### Przykład 3.114

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Szablon strony - reguły kontekstowe</title>
    <meta charset="UTF-8">
    <style>
        body {
            background-color: #c0c0c0;
        }
        h2 span {
            font-size: 14pt;
            color: blue;
            font-weight: normal;
        }
    </style>
</head>
<body>
    <h2>Zwiedzanie</h2>
    <span>Europe rozpoczniemy od Madrytu.</span>
    <span>Azji rozpoczniemy od Pekinu.</span>
</body>
</html>
```

```

        }

p span {
    font-size: 14pt;
    color: red;
    font-weight: normal;
}


```

</style>

</head>

<body>

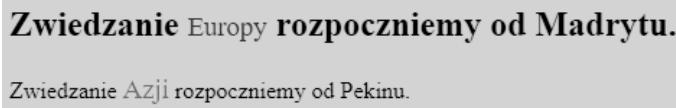
<h2>Zwiedzanie <span>Europy</span> rozpoczniemy od Madrytu.</h2>

<p>Zwiedzanie <span>Azji</span> rozpoczniemy od Pekinu.</p>

</body>

</html>

W podanym przykładzie znacznik `<span>` został zdefiniowany w kontekście nagłówka `<h2>` oraz akapitu `<p>`. Efekt jest taki, że każdy fragment otoczony znacznikiem `<span>` wygląda inaczej (rysunek 3.61).



**Rysunek 3.61.** Zastosowanie reguł kontekstowych

## Identyfikatory

Nie zawsze możliwe jest definiowanie wyglądu elementów w zależności od kontekstu. Na przykład próba zdefiniowania różnego wyglądu dla dwóch akapitów znajdujących się w tym samym obszarze dokumentu przy użyciu reguł kontekstowych nie powiedzie się, ponieważ obydwa akapity występują w tym samym kontekście. W takiej sytuacji można zastosować identyfikatory. Identyfikator jednoznacznie określa wybrany element. Dzięki niemu każdy, nawet najmniejszy element strony może wyglądać inaczej.

### Przykład 3.115

```

<!DOCTYPE html>
<html lang="pl-PL">
<head>

<title>Szablon strony - identyfikatory</title>
<meta charset="UTF-8">
<style>

```

```
body {  
    background-color: #c0c0c0;  
}  
  
p#ak1 {  
    font-size: 14pt;  
    color: blue;  
}  
  
p#ak2 {  
    font-size: 12pt;  
    color: green;  
}  
  
</style>  
</head>  
<body>  
    <p id="ak1">Najciekawszym krajem w Afryce jest Kenia.</p>  
    <p id="ak2">Australię planujemy zwiedzać na końcu.</p>  
</body>  
</html>
```

Każdy akapit będzie wyświetlony w inny sposób, ponieważ zostały im przypisane identyfikatory, a w arkuszu stylów dla każdego identyfikatora zdefiniowano inny sposób wyświetlania (rysunek 3.62).

Najciekawszym krajem w Afryce jest Kenia.  
Australię planujemy zwiedzać na końcu.

**Rysunek 3.62.** Zastosowanie identyfikatorów

## Klasy

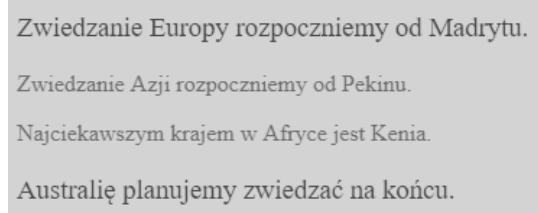
Czasami na stronie znajduje się grupa elementów, które powinny wyglądać tak samo. W takiej sytuacji nie trzeba każdemu elementowi przypisywać identyfikatora. Można te elementy połączyć w grupy i dla każdej grupy zdefiniować klasę, a następnie, używając pojedynczych reguł w CSS, za jej pomocą określić wygląd całej grupy elementów.

### Przykład 3.116

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

```
<head>
    <title>Klasy</title>
    <meta charset="UTF-8">
    <style>
        body {
            background-color: #c0c0c0;
        }
        p.t1 {
            font-size: 14pt;
            color: #5e1138;
        }
        p.t2 {
            font-size: 12pt;
            color: #186d28;
        }
    </style>
</head>
<body>
    <p class="t1">Zwiedzanie Europy rozpoczniemy od Madrytu.</p>
    <p class="t2">Zwiedzanie Azji rozpoczniemy od Pekinu.</p>
    <p class="t2">Najciekawszym krajem w Afryce jest Kenia.</p>
    <p class="t1">Australię planujemy zwiedzać na końcu.</p>
</body>
</html>
```

Wynik zdefiniowania dwóch klas dla akapitu przedstawiono na rysunku 3.63.



Zwiedzanie Europy rozpoczniemy od Madrytu.  
Zwiedzanie Azji rozpoczniemy od Pekinu.  
Najciekawszym krajem w Afryce jest Kenia.  
Australię planujemy zwiedzać na końcu.

**Rysunek 3.63.** Zastosowanie różnych klas do formatowania akapitów

Jeżeli wybrany element jest definiowany za pomocą kilku reguł, to obowiązują następujące priorytety:

- 1.** Identyfikator.
- 2.** Klasa.
- 3.** Znacznik kontekstowy.
- 4.** Znacznik.

## Znacznik \*

W modelu blokowym CSS wszystkie elementy można traktować jak prostokątne bloki. Dla każdego z tych bloków można definiować marginesy zewnętrzne i wewnętrzne oraz obramowania. Jeżeli dla elementu umieszczonego na stronie te parametry nie zostaną zdefiniowane, to przyjmą wartości domyślne. Różne przeglądarki w odmienny sposób interpretują wartości domyślne elementu, co może prowadzić do nieprawidłowego wyświetlania zawartości strony. Aby temu zapobiec, należy jako jedną z pierwszych deklaracji strony umieścić deklarację zerowania marginesów zewnętrznych i wewnętrznych dla wszystkich elementów występujących na stronie.

Specjalny znacznik, który możemy do tego wykorzystać, to selektor uniwersalny \*. Zdefiniowanie dla niego reguły oznacza, że powstała reguła odnosi się do wszystkich znaczników na stronie.

### Przykład 3.117

```
* {margin: 0px; padding: 0px;}
```

## Bloki danych

Do rozmieszczania elementów na stronie stosuje się najczęściej omówione już elementy grupujące: <div>, <header>, <nav>, <section> i <footer>. Są to elementy, które grupują elementy strony, ale nie w celu ich formatowania, tylko w celu ich odpowiedniego rozmieszczenia. Znaczniki te są odpowiednikami ramek, w których zgrupowane elementy można dowolnie przemieszczać na stronie. Znacznikom grupującym można przypisywać identyfikatory lub klasy. Poprzez deklarację reguły dla identyfikatorów i klas każdy znacznik może być formatowany. Wewnątrz znacznika grupującego mogą być umieszczone inne elementy, np.: obrazy, teksty lub kolejne znaczniki.

Elementy <div>, <header>, <nav>, <section> i <footer> tworzą bloki, w których są umieszczone inne elementy, może więc się zdarzyć, że element umieszczony wewnątrz znacznika grupującego jest szerszy niż wielkość tego znacznika i nie mieści się w nim. W takim przypadku zbyt szerokie elementy mogą zachodzić na sąsiednie bloki, zasłaniając ich zawartość, lub blok może zostać przeniesiony niżej. Takie zachowanie bloków niszczy układ całej strony.

Aby zachować układ strony, można dla podanych znaczników umieszczanych na stronie zdefiniować atrybut overflow jako hidden. Atrybut overflow definiuje sposób wyświetlania, gdy zawartość nie mieści się w zdefiniowanym rozmiarze bloku; wartość hidden wymusi obcięcie niemieszczącego się fragmentu. Fragment zawartości przestanie być widoczny na stronie, ale zostanie zachowany układ strony.

**Przykład 3.118**

```
div {overflow: hidden;}
```

**WSKAZÓWKA**

Na początku pracy nad nową stroną wskazane jest zdefiniowanie dla wszystkich znaczników grupujących atrybutu `overflow: hidden;`

**Kolory**

Dobrym zwyczajem jest również zdefiniowanie koloru tła i koloru tekstu. Domyślny kolor tła to biały, a tekstu — czarny. Nie można jednak zakładać, że są to kolory obowiązujące u wszystkich użytkowników naszej strony. Jeżeli użytkownik zmieni kolory domyślne w systemie operacyjnym swojego komputera, to wyświetlając zaprojektowaną przez nas stronę, może uzyskać nieoczekiwane efekty kolorystyczne.

**Przykład 3.119**

```
body {background-color: #fff; color: #000;}
```

**Wymiary strony**

Przystępując do projektowania szablonu strony o stałej szerokości, należy ustalić tę szerokość. Zwykle przyjmuje się, że strona internetowa jest przygotowana dla ekranu o rozdzielczości 1024 pikseli. Oczywiście można wybrać dowolną szerokość, ale trzymając się standardów obowiązujących w internecie, uzyskamy lepsze wyniki wyświetlania strony w dowolnej przeglądarce.

Przyjmując, że na stronie pojawia się suwak do przewijania treści, przy zerowych marginach strony szerokość szablonu powinna wynosić 1000 px. Jeżeli dla projektowanego szablonu strony zostanie zdefiniowany zewnętrzny element, np. `<div>`, w którym będzie umieszczona cała zawartość strony, i nadamy mu identyfikator `ramka`, to właśnie ten element powinien mieć szerokość 1000 px.

**Przykład 3.120**

```
#ramka {
    width: 1000px;
    margin-right: auto;
    margin-left: auto;
}
```

Aby strona była zawsze widoczna na środku przeglądarki, niezależnie od rozmiaru jej okna, należy ustawić wartości marginów prawego i lewego na `auto`. Oznacza to, że

obydwa marginesy będą miały taką samą wartość i że będzie ona automatycznie dostosowywana do szerokości okna przeglądarki.

## 3.8.2. Szablon strony

Pamiętajmy, że używając znaczników `<div>`, `<header>`, `<nav>`, `<section>` i `<footer>`, możemy grupować w bloki elementy strony i formatować je za pomocą stylów. Bloki te mogą zawierać akapity, listy, obrazy oraz inne bloki. Umożliwiają one wydzielanie na stronie większych, logicznych fragmentów w celu nadania im specyficznego formatowania za pomocą stylów CSS. Z tych powodów wymienione bloki świetnie nadają się do budowy szablonu strony internetowej. Ponieważ tak zbudowana strona może zawierać wiele różnych znaczników, w celu ich odróżnienia warto każdemu z nich przypisać identyfikator (atrybut `id`). Jeżeli definiujemy znaczniki, które mogą się powtórzyć na stronie, należy przypisać je do klasy (atrybut `class`).

W praktyce najczęściej mamy do czynienia z szablonami typu:

- nagłówek, jedna kolumna, stopka;
- nagłówek, dwie kolumny, stopka;
- nagłówek, trzy kolumny, stopka.

Jeżeli zajrzymy na najpopularniejsze portale internetowe, najczęściej spotkamy wersję z dwiema lub trzema kolumnami. Witryny tematyczne często mają układ dwukolumnowy: pierwsza kolumna zawiera menu tematyczne, druga — treść. Mogą wystąpić różne modyfikacje tych układów, ale podstawowa struktura w postaci kolumn zostaje zachowana. Można planować układ z większą liczbą kolumn, ale przy wielu kolumnach strona staje się nieczytelna.

Szablon oparty na stałej liczbie kolumn jest najczęściej stosowaną formą układu strony. Jest on niezależny od rozdzielczości ekranu, a elementy rozplanowane na stronie zawsze pozostają w miejscach, w których zostały zaplanowane.

### Szablon z trzema kolumnami

Na nasze potrzeby przygotujemy szablon strony z trzema kolumnami.

W pierwszym etapie musimy w języku HTML zdefiniować szablon strony przy użyciu znaczników blokowych i semantycznych.

#### Przykład 3.121

```
<body>
  <header id="naglowek">
    Tutaj znajdzie się zawartość nagłówka strony
    <nav>Tutaj będzie NAWIGACJA</nav>
  </header>
```

```
<div id="ramka">  
  <section class="kolumny" id="kol1">  
    <h3>Kolumna 1</h3>  
    Tutaj znajdzie się zawartość kolumny1  
    <article class="artyk" id="ar1_1">  
      <h4>Artykuł 1_1</h4>  
      Artykuł w kolumnie1  
    </article>  
    <article class="artyk" id="ar2_1">  
      <h4>Artykuł 2_1</h4>  
      Artykuł w kolumnie1  
    </article>  
  </section>  
  <section class="kolumny" id="kol2">  
    <h3>Kolumna 2</h3>  
    Tutaj znajdzie się zawartość kolumny2  
    <article class="artyk" id="ar1_2">  
      <h4>Artykuł 1_2</h4>  
      Artykuł w kolumnie2  
    </article>  
    <article class="artyk" id="ar2_2">  
      <h4>Artykuł 2_2</h4>  
      Artykuł w kolumnie2  
    </article>  
    <article class="artyk" id="ar3_2">  
      <h4>Artykuł 3_2</h4>  
      Artykuł w kolumnie2  
    </article>  
  </section>  
  <section class="kolumny" id="kol3">  
    <h3>Kolumna 3</h3>  
    Tutaj znajdzie się zawartość kolumny3  
    <article class="artyk" id="ar1_3">  
      <h4>Artykuł 1_3</h4>  
      Artykuł w kolumnie3
```

```

</article>
</section>
<footer id="stopka">
    Tutaj znajdzie się zawartość stopki strony
</footer>
</div>
</body>

```

Tak przygotowany szablon stanowi strukturę strony, którą za pomocą stylów CSS można w dowolny sposób formatować. Projektując kolejne arkusze CSS dla tego samego kodu HTML, możemy zmieniać wygląd strony. Natomiast wypełniając go odpowiednią treścią, możemy tworzyć kolejne strony oparte na tym samym szablonie.

W przygotowanym szablonie wszystkie bloki zostały ułożone jeden pod drugim. Aby kolumny szablonu ułożyć pionowo obok siebie, należy użyć atrybutu `float`.

Atrybut ten był już wykorzystywany do oblewania ilustracji tekstem. Jeżeli dla kilku sąsiednich elementów nadamy atrybutowi `float` wartość `left`, to elementy te ustawią się obok siebie, pod warunkiem że pozwoli na to szerokość bloku, w którym się znajdują.

W związku z tym wystarczy dla kolumn zdefiniować regułę zawierającą atrybut `float: left`. Musimy pamiętać, aby suma szerokości wszystkich kolumn nie przekroczyła całkowitej szerokości strony.

### Przykład 3.122

```

#kol1 {float: left; width: 30%;}
#kol2 {float: left; width: 30%;}
#kol3 {float: left; width: 30%;}

```

Lub

### Przykład 3.123

```
.kolumny {float: left; width: 30%;}
```

Należy pamiętać również o tym, aby stopce nadać odpowiednie atrybuty uniemożliwiające jej wyświetlenie z lewej strony kolumn. Posłuży do tego atrybut `clear` z wartością `both`.

### Przykład 3.124

```
#stopka {clear: both; width: 100%;}
```

Jeśli do projektowania szablonu strony wykorzystamy wcześniej poznane reguły projektowania, to kod szablonu strony może być podobny do podanego w poniższym przykładzie.

### Przykład 3.125

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
<meta charset="UTF-8">
<title>Trzy kolumny</title>
<style>
* {
    margin: 0px;
    padding: 0px;
}
section {
    overflow: hidden;
}
#ramka {
    width: 1000px;
    margin-right: auto;
    margin-left: auto;
}
#naglowek {
    width: 100%;
    background-color: #FCF;
}
#stopka {
    width: 100%;
    background-color: #93F;
    clear: both;
}
.kolumny {
    width: 30%;
    margin-top: 10px;
    margin-right: 2%;
    margin-bottom: 2%;
    margin-left: 10px;
}
#koll {
```

```
float: left;  
background-color: #3FF;  
}  
  
#kol2 {  
    float: left;  
    background-color: #CF3;  
}  
  
#kol3 {  
    background-color: #F93;  
}  
}  
</style>  
</head>  
<body>  
    <header id="naglowek">  
        Tutaj znajdzie się zawartość nagłówka strony  
        <nav>Tutaj będzie NAWIGACJA</nav>  
    </header>  
    <div id="ramka">  
        <section class="kolumny" id="kol1">  
            <h3>Kolumna 1</h3>  
            Tutaj znajdzie się zawartość kolumny1  
            <article class="artyk" id="arl_1">  
                <h4>Artykuł 1_1</h4>  
                Artykuł w kolumnie1  
            </article>  
            <article class="artyk" id="ar2_1">  
                <h4>Artykuł 2_1</h4>  
                Artykuł w kolumnie1  
            </article>  
        </section>  
        <section class="kolumny" id="kol2">  
            <h3>Kolumna 2</h3>  
            Tutaj znajdzie się zawartość kolumny2  
            <article class="artyk" id="arl_2">  
                <h4>Artykuł 1_2</h4>  
                Artykuł w kolumnie2  
            </article>  
        </section>  
    </div>
```

```
</article>

<article class="artyk" id="ar2_2">
    <h4>Artykuł 2_2</h4>
    Artykuł2 w kolumnie2
</article>

<article class="artyk" id="ar3_2">
    <h4>Artykuł 3_2</h4>
    Artykuł3 w kolumnie2
</article>

</section>

<section class="kolumny" id="kol3">
    <h3>Kolumna 3</h3>
    Tutaj znajdzie się zawartość kolumny3
    <article class="artyk" id="ar1_3">
        <h4>Artykuł 1_3</h4>
        Artykuł1 w kolumnie3
    </article>
</section>

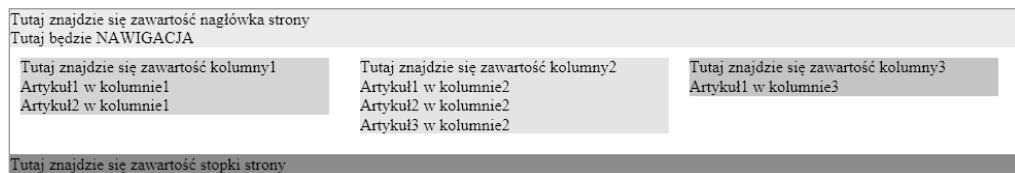
<footer id="stopka">
    Tutaj znajdzie się zawartość stopki strony
</footer>

</div>

</body>

</html>
```

Wynik wykonania kodu pokazuje rysunek 3.64.



**Rysunek 3.64.** Szablon strony z trzema kolumnami

Po określeniu przeznaczenia strony szablon może zostać zindywidualizowany, co pokazuje rysunek 3.65.

**Rewolucja przemysłowa 4.0**  
Autor: Paweł Nadrożny

**Start** **Kursy** **Artykuły** **Kontakt**

**Pierwsza rewolucja przemysłowa**

**Druga rewolucja przemysłowa**

**Trzecia rewolucja przemysłowa**

**Czwarta rewolucja przemysłowa**

**Rewolucja przemysłowa**

**Pierwsza rewolucja przemysłowa**  
Pierwsza rewolucja przemysłowa to wiek pary.  
Produkcja mechaniczna jest wspomagana siłą pary i wody.

**Druga rewolucja przemysłowa**  
Druga rewolucja przemysłowa to wiek elektryczności.  
Rozpoczyna się produkcja masowa z zastosowaniem energii elektrycznej.

**Trzecia rewolucja przemysłowa**  
Trzecia rewolucja przemysłowa to wiek komputerów. Postępuje automatyzacja produkcji przemysłowej z wykorzystaniem sterowników programowalnych.

**Czwarta rewolucja przemysłowa**  
Czwarta rewolucja przemysłowa to zanikanie bariery między ludźmi i maszynami.  
Powstają inteligentne fabryki z cyberfizycznymi systemami produkcji w otoczeniu: Internetu ludzi, Internetu rzeczy, Internetu usług, Internetu danych.

**Informacje dodatkowe**

Pracę wykonał: Jan Kowalski

**Rysunek 3.65.** Zmodyfikowany szablon strony z trzema kolumnami

Realizacją podanego szablonu strony może być przedstawiony niżej kod.

### Przykład 3.126

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <meta charset="utf-8">
    <title>Szablon strony</title>
    <style>
        * {
            margin: 0px;
            padding: 0px;
        }
        section {
            overflow: hidden;
            padding: 25px;
        }
        #ramka {
            width: 999px;

```

```
margin-right: auto;  
margin-left: auto;  
}  
  
#naglowek {  
width: 100%;  
background-color: #fcf;  
font-size: 12pt;  
color: #5e1138;  
font-family: Arial;  
font-style: italic;  
text-align: center;  
font-weight: bold;  
}  
  
#stopka {  
width: 100%;  
background-color: #c0c0c0;  
color: blue;  
font-family: Arial;  
font-style: italic;  
padding: 5px;  
text-align: center;  
clear: both;  
}  
  
.kolumny {  
margin-top: 10px;  
margin-right: 2%;  
margin-bottom: 2%;  
margin-left: 10px;  
background-color: #c0c0c0;  
}  
  
.art {  
background-color: #9ca0c9;  
padding: 5px;  
margin: 4px;  
}
```

```
#kol1 {  
    float: left;  
    width: 180px;  
}  
  
#kol2 {  
    float: left;  
    width: 490px;  
}  
  
#kol3 {  
    width: 100px;  
}  
  
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
    background-color: #33415d;  
}  
  
li {  
    float: left;  
}  
  
li a {  
    display: block;  
    color: white;  
    text-align: center;  
    padding: 14px 16px;  
    text-decoration: none;  
}  
  
li a:hover {  
    background-color: #0d6ba1;  
}  
  
</style>  
</head>  
<body>  
    <div id="ramka">
```

```
<header id="naglowek">  
    <h1>Rewolucja przemysłowa 4.0</h1>  
    <p>Autor: Paweł Nadrożny</p>  
    <nav>  
        <ul>  
            <li><a href="start.html">Start</a></li>  
            <li><a href="nowe.html">Kursy</a></li>  
            <li><a href="serwis.html">Artykuły</a></li>  
            <li><a href="uslugi.html">Kontakt</a></li>  
        </ul>  
    </nav>  
</header>  
  
<section class="kolumny" id="kol1">  
    <h4>Pierwsza rewolucja przemysłowa</h4><br>  
    <h4>Druga rewolucja przemysłowa</h4><br>  
    <h4>Trzecia rewolucja przemysłowa</h4><br>  
    <h4>Czwarta rewolucja przemysłowa</h4>  
</section>  
  
<section class="kolumny" id="kol2">  
    <h3>Rewolucja przemysłowa</h3>  
    <article class="art">  
        <h4>Pierwsza rewolucja przemysłowa</h4>  
        <p>Pierwsza rewolucja przemysłowa to wiek pary.<br> Produkcja mechaniczna jest wspomagana siłą pary i wody.</p>  
    </article>  
    <article class="art">  
        <h4>Druga rewolucja przemysłowa</h4>  
        <p>Druga rewolucja przemysłowa to wiek elektryczności.</p>  
        <p>Rozpoczyna się produkcja masowa z zastosowaniem energii elektrycznej.</p>  
    </article>  
    <article class="art">  
        <h4>Trzecia rewolucja przemysłowa</h4>  
        <p>Trzecia rewolucja przemysłowa to wiek komputerów. Postępuje automatyzacja produkcji przemysłowej z wykorzystaniem sterowników programowalnych.</p>  
    </article>  
</section>
```

```
</article>

<article class="art">
    <h4>Czwarta rewolucja przemysłowa</h4>
    <p>Czwarta rewolucja przemysłowa to zanikanie bariery między ludźmi i maszynami.</p>
    <p>Powstają intelligentne fabryki z cyberfizycznymi systemami produkcji w otoczeniu: Internetu ludzi, Internetu rzeczy, Internetu usług, Internetu danych.</p>
</article>
</section>

<section class="kolumny" id="kol3">
    <h4>Informacje dodatkowe</h4>
</section>

<footer id="stopka">
    <p>Pracę wykonał: Jan Kowalski</p>
</footer>
</div>
</body>
</html>
```

### Ćwiczenie 3.20

Wykorzystując poznane narzędzia, przygotuj własny szablon strony internetowej. Utwórz menu. Zmieniając deklaracje stylów w arkuszu stylów, nadaj elementom strony atrakcyjny wygląd.

## 3.8.3. Płynny szablon strony

Płynny szablon pozwala, aby przy zmianie szerokości okna przeglądarki zmieniała się szerokość kolumn strony. Najczęściej zmiana szerokości dotyczy tylko wybranych kolumn, np. kolumna z menu pozostaje niezmieniona, płynnie zmienia się natomiast szerokość kolumny z treścią.

Na nasze potrzeby przygotujemy szablon strony z dwiema kolumnami.

### Przykład 3.127

```
<body>
    <div id="ramka">
        <header id="naglowek">
            Tutaj znajdzie się zawartość nagłówka strony
        </header>
        <content id="content">
            <h1>Strona internetowa</h1>
            <p>Witaj na naszej stronie!</p>
            <ul>
                <li>Link 1</li>
                <li>Link 2</li>
                <li>Link 3</li>
            </ul>
        </content>
    </div>
</body>
```

```
</header>

<section class="kolumny" id="kol1">
    <h2>Kolumna 1</h2>
    Tutaj znajdzie się zawartość kolumny1
</section>

<section class="kolumny" id="kol2">
    <h2>Kolumna 2</h2>
    Tutaj znajdzie się zawartość kolumny2
</section>

<footer id="stopka">
    Tutaj znajdzie się zawartość stopki strony
</footer>
</div>
</body>
```

Dla tak przygotowanego szablonu można, przy wykorzystaniu reguł CSS, określić sposób wyświetlania kolumn.

### Przykład 3.128

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <meta charset="UTF-8">
    <title>Płynny szablon</title>
    <style>
        * {
            margin: 0px;
            padding: 0px;
        }
        section {
            overflow: hidden;
            padding: 25px;
        }
        #ramka {

```

```
margin-right: auto;  
margin-left: auto;  
min-width: 400px;  
max-width: 900px;  
}  
  
#naglowek {  
background-color: #FCF;  
font-size: 16pt;  
color: #5e1138;  
font-family: Arial;  
font-style: italic;  
text-align: center;  
font-weight: bold;  
}  
  
#stopka {  
background-color: #c0c0c0;  
font-size: 14pt;  
color: blue;  
font-family: Arial;  
font-style: italic;  
clear: both;  
}  
  
.kolumny {  
margin-top: 10px;  
margin-right: 2%;  
margin-bottom: 2%;  
margin-left: 10px;  
background-color: #c0c0c0;  
}  
  
#koll1 {  
float: left;  
width: 100px;  
}  
  
</style>  
</head>
```

```
<body>  
  <div id="ramka">  
    <header id="naglowek">Nagłówek mojej strony</header>  
    <section class="kolumny" id="kol1">  
      <h3>Menu</h3>  
    </section>  
    <section class="kolumny" id="kol2">  
      <h3>Treść</h3>  
    </section>  
    <footer id="stopka">Prace wykonał: Jan Kowalski</footer>  
  </div>  
</body>  
</html>
```

Jeżeli nie zostaną określone szerokości kolumn ani szerokość szablonu, to szablon zajmie całe okno przeglądarki. Zmiana szerokości okna przeglądarki spowoduje automatyczną modyfikację zawartości strony i jej dostosowanie do aktualnych rozmiarów okna.

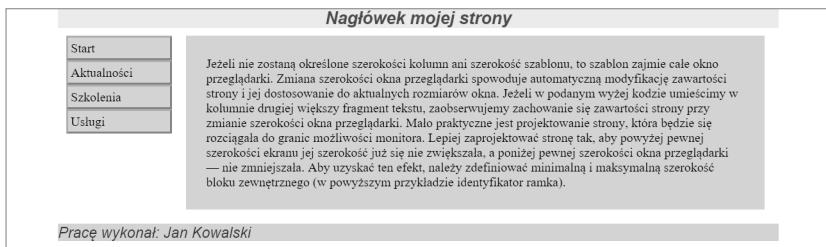
Jeżeli w podanym wyżej kodzie umieścimy w kolumnie drugiej większy fragment tekstu, zaobserwujemy zachowanie się zawartości strony przy zmianie szerokości okna przeglądarki.

Mało praktyczne jest projektowanie strony, która będzie się rozciągała do granic możliwości monitora. Lepiej zaprojektować stronę tak, aby powyżej pewnej szerokości ekranu jej szerokość już się nie zwiększała, a poniżej pewnej szerokości okna przeglądarki — nie zmniejszała. Aby uzyskać ten efekt, należy zdefiniować minimalną i maksymalną szerokość bloku zewnętrznego (w powyższym przykładzie identyfikator `ramka`).

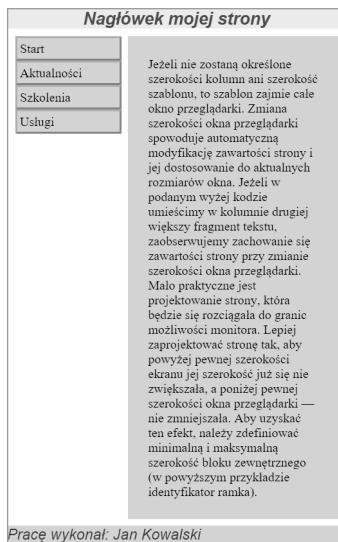
### Przykład 3.129

```
#ramka {  
  margin-right: auto;  
  margin-left: auto;  
  min-width: 400px;  
  max-width: 900px;  
}
```

Po ustawieniu tych parametrów zawartość strony będzie skalowana w zakresie podanych wartości. Dalsze zwiększanie szerokości strony spowoduje pokazanie tła strony (rysunek 3.66), a zmniejszanie szerokości poniżej zadeklarowanej spowoduje ukrycie części zawartości i wyświetlenie suwaków, za pomocą których będzie można obejrzeć ukryte fragmenty strony (rysunek 3.67).



**Rysunek 3.66.** Płynny szablon strony. Okno przeglądarki zmaksymalizowane



**Rysunek 3.67.** Płynny szablon strony. Okno przeglądarki zminimalizowane

Jeżeli planujemy umieszczenie menu w pierwszej kolumnie, to kolumna ta powinna mieć stałą szerokość, a dynamicznie zmieniać się powinna tylko szerokość drugiej kolumny.

### Przykład 3.130

```
#kol1 {float: left; width: 100px;}
```

Jeśli taką deklarację wprowadzimy dla pierwszej kolumny, to dla drugiej kolumny nie musimy deklarować szerokości, ponieważ automatycznie wypełni ona pozostałą część strony. Należy pamiętać o jeszcze jednej zależności, mianowicie w przypadku układów o płynnej szerokości nie ma potrzeby ustawiania atrybutu `float` dla kolumny położonej z prawej strony. Dlatego w kodzie dokumentu trzeba usunąć ten parametr.

### Ćwiczenie 3.21

Na podstawie szablonu przygotowanego w ćwiczeniu 3.20 utwórz stronę internetową, na której zaprezentujesz osiągnięcia piłkarzy nożnych.



## 3.9. Responsywny układ strony

Responsywny układ strony to taki układ, który zmienia swoje atrybuty zdefiniowane za pomocą stylów CSS w zależności od różnych zewnętrznych czynników. Jest wiele czynników, pod których wpływem elementy strony mogą otrzymywać nowe atrybuty. Jednak najczęściej to szerokość okna przeglądarki internetowej wymusza konieczność definiowania dodatkowych reguł CSS. Zadziałają one, gdy dokument HTML zostanie wyświetlony w oknie przeglądarki internetowej o określonej szerokości (komputer, tablet, telefon).

Responsywny projekt strony internetowej pozwala za pomocą kodu HTML i stylów CSS zmienić rozmiar zawartości, przesunąć określona zawartość lub ukryć ją, tak aby strona dobrze wyglądała na dowolnym urządzeniu.

W HTML5 za pomocą znacznika <meta> możliwe jest dołączenie metatagu viewport i przejęcie kontroli nad widocznym dla użytkownika obszarem wyświetlania strony internetowej. Do strony internetowej należy dołączyć następujący zapis znacznika <meta>:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Spowoduje on poprawne wyświetlanie strony na urządzeniach mobilnych.

Zdefiniowany w znaczniku <meta> element "viewport" udostępnia przeglądarce internetowej polecenia dotyczące kontrolowania wymiarów strony i skalowania.

Składnia width=device-width określa szerokość strony, która powinna odpowiadać szerokości ekranu urządzenia, gdzie będzie wyświetlana strona.

Część initial-scale=1.0 ustawia początkowy poziom powiększenia w momencie, gdy strona jest otwierana przez przeglądarkę.

W specyfikacji CSS3 rozbudowane zostały *media queries* (zapytania o media), co pozwala na rozpoznawanie rozdzielczości urządzeń, na których wyświetlana jest strona internetowa. W zależności od wykrytych rozdzielczości wczytywane są różne arkusze stylów. Wybrane elementy strony są przenoszone w inne miejsce lub są ukrywane. W wyniku tego strony o tej samej strukturze HTML są wyświetlane w różny sposób.

Dodając regułę @media, spowodujemy włączanie określonych atrybutów stylów CSS tylko wtedy, gdy spełnione zostaną wymagane warunki.

### Przykład 3.131

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Strona responsywna</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

<style>
body {
background-color: #c0c0c0;
}

@media only screen and (max-width: 500px) {
body {
background-color: lightblue;
}
}

</style>

</head>

<body>

<p>Zmiana koloru tła, gdy szerokość strony mniejsza od 500 pikseli.</p>

</body>

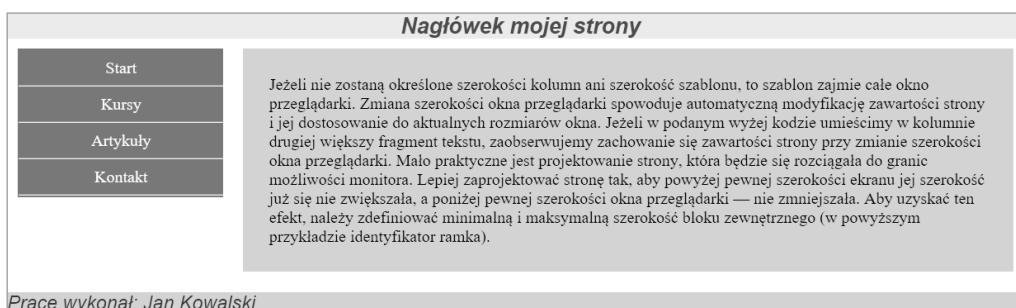
</html>

```

W podanym przykładzie za pomocą reguły `@media` zdefiniowany został warunek narzucony na szerokość strony równy `500px`. Efekt zmiany koloru tła strony można zaobserwować po zmniejszeniu rozmiaru okna przeglądarki.

Aby tworzone strony internetowe dobrze wyglądały na małym ekranie, można definiować tzw. punkty przerwania. Po ich zdefiniowaniu wybrane części projektowanej strony będą się zachowywały inaczej po obu stronach punktów przerwania.

Strona, która będzie wyświetlana na urządzeniach mobilnych, powinna być zbudowana na bazie jednej kolumny. Zapewni to poprawne skalowanie strony i dopasowanie szerokości kolumny do ekranu urządzenia wyświetlającego. Natomiast oryginalna strona w najprostszym przypadku może zawierać dwie kolumny, np. kolumnę z menu oraz kolumnę informacyjną (rysunek 3.68). Można się jeszcze zastanowić, które elementy strony mogą zostać ukryte dla urządzeń mobilnych.



**Rysunek 3.68.** Wygląd oryginalnej strony z dwiema kolumnami

### Przykład 3.132

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Strona responsywna</title>
    <style>
        * {
            margin: 0px;
            padding: 0px;
        }
        section {
            overflow: hidden;
        }
        #ramka {
            margin-right: auto;
            margin-left: auto;
            min-width: 300px;
            max-width: 1000px;
        }
        #naglowek {
            background-color: #FCF;
            font-size: 16pt;
            color: #5e1138;
            font-family: Arial;
            font-style: italic;
            text-align: center;
            font-weight: bold;
        }
        #stopka {
            background-color: #c0c0c0;
            font-size: 14pt;
            color: blue;
            font-family: Arial;
        }
    </style>

```

```
font-style: Italic;  
clear: both;  
}  
  
.kolumny {  
margin-top: 10px;  
margin-right: 2%;  
margin-bottom: 2%;  
margin-left: 10px;  
background-color: #c0c0c0;  
}  
  
#kol1 {  
float: left;  
width: 20%;  
}  
  
#kol2 {  
padding: 25px;  
}  
  
ul {  
list-style-type: none;  
margin: 0;  
padding: 0;  
width: 100%;  
background-color: #555;  
border: 1px solid #555;  
}  
  
li a {  
display: block;  
color: #fff;  
padding: 8px 16px;  
text-decoration: none;  
}  
  
li {  
text-align: center;  
border-bottom: 1px solid #f1f1f1;  
}
```

```
li a.active {  
    background-color: #4caf50;  
    color: white;  
}  
  
li a:hover:not(.active) {  
    background-color: #f1f1f1;  
    color: black;  
}  
  
@media only screen and (max-width:620px) {  
    #kol1 {  
        width: 100%;  
    }  
}
```

</style>

</head>

<body>

<div id="ramka">

<header id="naglowek">Nagłówek mojej strony</header>

<section class="kolumny" id="kol1">

<h3>Menu</h3>

<ul>

<li><a href="start.html">Start</a></li>

<li><a href="nowe.html">Kursy</a></li>

<li><a href="serwis.html">Artykuły</a></li>

<li><a href="uslugi.html">Kontakt</a></li>

</ul>

</section>

<section class="kolumny" id="kol2">

<h3>Opis</h3>

Jeżeli nie zostaną określone szerokości kolumn ani szerokość szablonu, to szablon zajmie całe okno przeglądarki. Zmiana szerokości okna przeglądarki spowoduje automatyczną modyfikację zawartości strony i jej dostosowanie do aktualnych rozmiarów okna.

Jeżeli w podanym wyżej kodzie umieścimy w kolumnie drugiej większy fragment tekstu, zaobserwujemy zachowanie się zawartości strony przy zmianie szerokości okna przeglądarki.

Mało praktyczne jest projektowanie strony, która będzie się rozciągała do granic możliwości monitora. Lepiej zaprojektować stronę tak, aby powyżej pewnej szerokości ekranu jej szerokość już się nie zwiększała, a poniżej pewnej szerokości okna przeglądarki – nie zmniejszała. Aby uzyskać ten efekt, należy zdefiniować minimalną i maksymalną szerokość bloku zewnętrznego (w powyższym przykładzie identyfikator ramka).

```
</section>

<footer id="stopka">Pracę wykonał: Jan Kowalski</footer>

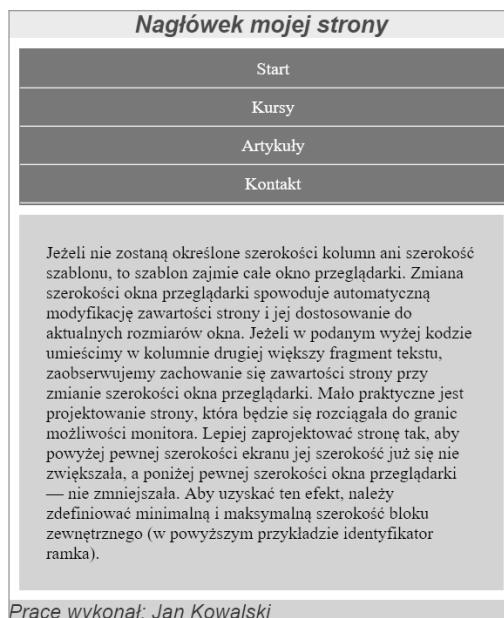
</div>

</body>

</html>
```

W podanym przykładzie, jeżeli szerokość okna, w którym jest wyświetlana zawartość strony, będzie mniejsza od 620px, nastąpi zmiana w sposobie wyświetlania zawartości strony (rysunek 3.69). Menu, które było wyświetlane z lewej, zostanie wyświetcone nad panelem głównym strony. Wynika to z zastosowania w kodzie strony reguły:

```
@media only screen and (max-width:620px) {
    #kol1 {width:100%;}
}
```



**Rysunek 3.69.** Układ strony po zmianie szerokości wyświetlonej strony

Liczba reguł @media zależy od rodzaju strony internetowej. Przy bardzo rozbudowanych stronach zdefiniowanie 3–5 reguł nie jest rzadkością i mogą one znacząco zmieniać wygląd strony nawet przy niewielkich zmianach rozdzielczości.

Jedną z zasad podczas projektowania responsywnej strony internetowej jest ukrywanie wybranych elementów przy mniejszych rozdzielczościach ekranu. Do ukrywania określonych elementów strony używamy atrybutu `display: none`.

### Przykład 3.133

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Strona responsywna</title>
    <style>
        * {
            margin: 0px;
            padding: 0px;
        }
        body {
            font: 16px Verdana;
            line-height: 1.6;
            background: #e0dfef;
        }
        #blok {
            width: 90%;
            overflow: hidden;
            margin: 0 auto;
        }
        a {
            color: #354458;
        }
        #header {
            height: 100px;
            background: #33415d;
            margin: 0 0 10px 0;
        }
        #header h1 {
```

```
padding: 20px;  
color: #e9e0d6;  
}  
  
#kon {  
width: 75%;  
background: #c0c0c0;  
float: left;  
padding: 1%;  
overflow: hidden;  
}  
  
#menu {  
background: #bbc0cb;  
width: 20%;  
float: right;  
padding: 1%;  
overflow: hidden;  
}  
  
#menu h4 {  
padding: 3px 3px 15px 3px;  
}  
  
#menu ul li {  
list-style-type: none;  
padding: 0 0 0 10px;  
}  
  
li a {  
text-decoration: none;  
}  
  
@media only screen and (max-width: 680px) {  
#menu {  
width: 88%;  
position: absolute;  
height: 30px;  
}  
  
#menu h4 {  
display: none;  
}  
  
#menu ul li {
```

```
display: inline;  
}  
  
#kon {  
width: 98%;  
position: relative;  
top: 30px;  
margin-top: 3%;  
}  
}  
</style>  
</head>  
<body>  
<header id="header">  
    <h1>Nagłówek mojej strony</h1>  
</header>  
<div id="blok">  
    <section id="kon">  
        <h3>Opis</h3>  
        <p>  
            Jeżeli nie zostaną określone szerokości kolumn ani szerokość szablonu,  
            to szablon zajmie całe okno przeglądarki. Zmiana szerokości okna przeglądarki  
            spowoduje automatyczną modyfikację zawartości strony i jej dostosowanie do aktu-  
            alnych rozmiarów okna.  
            Jeżeli w podanym wyżej kodzie umieścimy w kolumnie drugiej większy frag-  
            ment tekstu, zaobserwujemy zachowanie się zawartości strony przy zmianie szero-  
            kości okna przeglądarki.  
            Mało praktyczne jest projektowanie strony, która będzie się rozciągała  
            do granic możliwości monitora. Lepiej zaprojektować stronę tak, aby powyżej  
            pewnej szerokości ekranu jej szerokość już się nie zwiększała, a poniżej pewnej  
            szerokości okna przeglądarki – nie zmniejszała. Aby uzyskać ten efekt, należy  
            zdefiniować minimalną i maksymalną szerokość bloku zewnętrznego (w powyższym  
            przykładzie identyfikator ramka).  
        </p>  
    </section>  
    <section id="menu">  
        <h4>Menu</h4>  
        <ul>  
            <li><a href="start.html">Start</a></li>
```

```

<li><a href="nowe.html">Kursy</a></li>
<li><a href="serwis.html">Artykuły</a></li>
<li><a href="uslugi.html">Kontakt</a></li>
</ul>
</section>
</div>
</body>
</html>

```

W podanym przykładzie strona składa się z dwóch kolumn. Prawa kolumna zawiera menu (rysunek 3.70). Gdy szerokość okna zmniejszy się do 680px, menu przesunie się w górę i zmieni w menu poziome (rysunek 3.71).

```

@media only screen and (max-width: 680px) {
    #menu {width:90%; position: absolute; height: 30px;}
    #menu h4 {display: none;}
    #menu ul li {display: inline;}
    #kon {width: 98%; position: relative; top: 30px; margin-top: 3%;}
}

```

Aby menu zostało przeniesione do góry, nadano mu atrybut `position: absolute`. Z kolei aby menu zmieniło się na poziome, ustwiono atrybut `width: 90%`. Zwiększono również szerokość panelu głównego (`width: 98%`). Żeby kolejne elementy menu układaly się poziomo, zmieniono atrybut `display: inline`. Dodatkowo przy zmianie menu na poziome ukryty został jego tytuł (tekst `Menu`) — atrybut `display: none`.

### Rysunek 3.70.

Wygląd strony z dwiema kolumnami

**Nagłówek mojej strony**

Jeżeli nie zostaną określone szerokości kolumn ani szerokość szablonu, to szablon zajmie całe okno przeglądarki. Zmiana szerokości okna przeglądarki spowoduje automatyczną modyfikację zawartości strony i jej dostosowanie do aktualnych rozmiarów okna. Jeżeli w podanym wyżej kodzie umieścimy w kolumnie drugiej większy fragment tekstu, zaobserwujemy zachowanie się zawartości strony przy zmianie szerokości okna przeglądarki. Mało praktyczne jest projektowanie strony, która będzie się rozciągała do granic możliwości monitora. Lepiej zaprojektować stronę tak, aby powyżej pewnej szerokości ekranu jej szerokość już się nie zwiększała, a poniżej pewnej szerokości okna przeglądarki – nie zmniejszała. Aby uzyskać ten efekt, należy zdefiniować minimalną i maksymalną szerokość bloku zewnętrznego (w powyższym przykładzie identyfikator ramka).

**Menu**

- Start
- Kursy
- Artykuły
- Kontakt

**Rysunek 3.71.**

Nowy układ menu po zmianie szerokości wyświetlanej strony

## Nagłówek mojej strony

- [Start](#)
- [Kursy](#)
- [Artykuły](#)
- [Kontakt](#)

Jeżeli nie zostaną określone szerokości kolumn ani szerokość szablonu, to szablon zajmie całe okno przeglądarki. Zmiana szerokości okna przeglądarki spowoduje automatyczną modyfikację zawartości strony i jej dostosowanie do aktualnych rozmiarów okna. Jeżeli w podanym wyżej kodzie umieścimy w kolumnie drugiej większy fragment tekstu, zaobserwujemy zachowanie się zawartości strony przy zmianie szerokości okna przeglądarki. Mało praktyczne jest projektowanie strony, która będzie się rozciągała do granic możliwości monitora. Lepiej zaprojektować stronę tak, aby powyżej pewnej szerokości ekranu jej szerokość już się nie zwiększała, a poniżej pewnej szerokości okna przeglądarki — nie zmniejszała. Aby uzyskać ten efekt, należy zdefiniować minimalną i maksymalną szerokość bloku

**Obrazki**

Jeżeli na responsywnej stronie internetowej występują obrazki (rysunek 3.72), powinny być automatycznie skalowane wraz z szerokością strony. W tym celu można do arkusza stylów dodać atrybuty, które umożliwią ich skalowanie (rysunek 3.73).

**Rysunek 3.72.**

Strona responsywna z umieszczonym obrazkiem

## Bruksela

Bruksela jest siedzibą króla i parlamentu belgijskiego, a ponadto jest siedzibą Instytucji Unii Europejskiej, NATO i Euroatomu. Ważny ośrodek handlowy, bankowo-finansowy i kulturalno-naukowy kraju. Siedziba dwóch królewskich akademii umiejętności, Królewskiej Akademii Sztuki, Biblioteki Królewskiej; dwa uniwersytety, konserwatorium i instytuty naukowo-badawcze, teatry, 30 muzeów.



- [Start](#)
- [Państwo](#)
- [Stolica](#)
- [Władze](#)

## Przykład 3.134

```
img {
    max-width: 100%;
    height: auto;
    width: auto;
}
```

### Rysunek 3.73.

Przeskalowanie obrazka na stronie responsywnej



Zawsze, gdy tworzona jest responsywna strona internetowa, szerokość obrazków i innych grafik powinna być podawana w wartościach procentowych. Gdy nie jest to możliwe, należy taki element umieścić w kontenerze i jemu nadać wartości procentowe.

W przypadku gdy na stronie pojawiają się złożone elementy graficzne, które w wyniku skalowania stają się niewyraźne, jedynym rozwiązaniem jest ich ukrywanie i zastępowanie mniejszymi odpowiednikami.

### Element <picture>

Zastosowanie znacznika `<picture>` do wyświetlania obrazków pozwala na większą elastyczność i wybór z dostępnych obrazków tego, który najlepiej zaprezentuje się na stronie. Zamiast jednego obrazka, który jest skalowany w zależności od sposobu jego wyświetlania na stronie, można użyć wielu obrazków. Ten, który najlepiej odpowiada parametrom ekranu, zostanie wyświetlony.

W elemencie `<picture>` występują dwa rodzaje znaczników: `<source>` i `<img>`.

Znacznik `<source>` posiada następujące atrybuty:

- `srcset` — określa URL obrazka do wyświetlania (wymagany);
- `media` — definiuje zapytanie o media, które są definiowane w regule CSS `@media`;
- `sizes` — określa szerokość (`width`), występuje w zapytaniu o media i określa ograniczenia narzucone na szerokość ekranu;
- `type` — określa typ danych.

### Przykład 3.135

```
<picture>
    <source srcset="obrazek1.jpg" media="(max-width: 600px)">
    <source srcset="obrazek2.jpg">
    
</picture>
```

Przeglądarka internetowa wybierze pierwszy obrazek z listy elementów (`<source>`), który pasuje do zdefiniowanego zapytania o media, i go wyświetli.

Znacznik `<img>` jest wymagany jako ostatni element w bloku deklaracji `<picture>`. Służy również do osiągnięcia kompatybilności wstępnej i pozwala na wyświetlenie obrazka w przeglądarkach, które nie obsługują elementu `<picture>`.

### Przykład 3.136

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Obrazek responsywny</title>
    <style>
        * {
            margin: 0px;
            padding: 0px;
        }
        body {
            font: 16px Verdana;
            line-height: 1.6;
            background: #e0dfel;
```

```
}

#header {
    height: 100px;
    background: #33415d;
    margin: 0 0 10px 0;
}

#header h1 {
    padding: 20px;
    color: #e9e0d6;
}

#kon {
    width: 75%;
    background: #c0c0c0;
    padding: 1%;
}

img {
    max-width: 100%;
    height: auto;
}

</style>

</head>

<body>

<header id="header">
    <h1>Bruksela</h1>
</header>

<section id="kon">
    <h4>Najważniejsze informacje</h4>
    <p>
        Bruksela jest siedzibą króla i parlamentu belgijskiego, a ponadto jest siedzibą instytucji Unii Europejskiej, NATO i Euroatomu. Ważny ośrodek handlowy, bankowo-finansowy i kulturalno-naukowy kraju. Siedziba dwóch królewskich akademii umiejętności, Królewskiej Akademii Sztuki, Biblioteki Królewskiej; dwa uniwersytety, konserwatorium i instytuty naukowo-badawcze, teatry, 30 muzeów.
    </p>
    <picture>
```

```

<source media="(min-width: 800px)" srcset="pom1.jpg">
<source media="(min-width: 600px)" srcset="pom2.jpg">

</picture>
</section>
</body>
</html>

```

Wynik działania kodu został przedstawiony na rysunkach. Jeżeli szerokość ekranu jest większa niż 800px, zostanie wyświetlony pierwszy obrazek (rysunek 3.74). Kolejny obrazek zawiera wykadrowaną część pierwszego zdjęcia i zostanie wyświetlony, gdy ekran będzie miał szerokość mniejszą od 800px. Gdy ekran będzie miał szerokość mniejszą od 600px, zostanie wyświetlony jeszcze mniejszy wykadrowany fragment zdjęcia (rysunek 3.75).



**Rysunek 3.74.**

Rysunek dodany do strony internetowej



**Rysunek 3.75.**

Zmiana rysunku przy zmienionej szerokości ekranu

## Tabele

Dla tabel wyświetlnych na responsywnej stronie internetowej można zdefiniować pasek przewijania, który zostanie wyświetlony, gdy szerokość ekranu będzie zbyt mała, aby wyświetlić całą zawartość tabeli. Żeby uzyskać podany efekt, tabelę należy umieścić w kontenerze (np. blok `<div>`) i nadać mu atrybut `overflow-x: auto`.

### Przykład 3.137

```
<!DOCTYPE html>

<html lang="pl-PL">
<head>
<meta charset="UTF-8">
<title>Tabela responsywna</title>
<style>
table {
    width: 100%;
    border-collapse: collapse;
}
th, td {
    text-align: left;
    padding: 8px;
}
div {
    overflow-x: auto;
}
tr:nth-child(even) {
    background-color: #f2f2f2
}
</style>
</head>
<body>
<div>
<table>
<tr>
<th>Europa</th>
<th>Azja</th>
<th>Afryka</th>
```

```

<th>Ameryka Północna</th>
<th>Ameryka Południowa</th>
</tr>
<tr>
<td>Francja</td>
<td>Indie</td>
<td>Egipt</td>
<td>Meksyk</td>
<td>Peru</td>
</tr>
<tr>
<td>Hiszpania</td>
<td>Chiny</td>
<td>Nigeria</td>
<td>Kanada</td>
<td>Brazylia</td>
</tr>
<tr>
<td>Włochy</td>
<td>Japonia</td>
<td>Kongo</td>
<td>USA</td>
<td>Argentyna</td>
</tr>
</table>
</div>

```

</body>

</html>

Wynik wykonania kodu został pokazany na rysunku 3.76

Europa	Azja	Afryka	Ameryka Północna	Ameryka Południowa
Francja	Indie	Egipt	Meksyk	Peru
Hiszpania	Chiny	Nigeria	Kanada	Brazylia
Włochy	Japonia	Kongo	USA	Argentyna

**Rysunek 3.76.**  
Tabela responsywna

Po zmianie rozmiaru okna na dole bloku pojawił się pasek przewijania. Należy zwrócić uwagę na to, że również teksty umieszczone w nagłówkach kolumn zostały „złamane”, a sposób ich wyświetlania został dostosowany do szerokości okna.

W podanym przykładzie zastosowano selektor :nth-child w postaci:

```
tr:nth-child(even) {background-color: #f2f2f2;}
```

Posłużył on do pokolorowania naprzemiennie parzystych wierszy tabeli. Atrybut even określa wiersze parzyste. Dla wierszy nieparzystych powinien zostać zastosowany atrybut odd.

```
tr:nth-child(odd) {background-color: #c0c0c0;}
```



## 3.10. Pytania i zadania

### 3.10.1. Pytania

- 1.** Jaką rolę w tworzeniu strony internetowej odgrywają arkusze stylów CSS?
- 2.** W jaki sposób i gdzie definiowany jest styl lokalny?
- 3.** Do czego służy znacznik <span>?
- 4.** Podaj różnice w stosowaniu zewnętrznego i wewnętrznego arkusza stylów CSS.
- 5.** Podaj ogólną postać selektora klasy.
- 6.** Kiedy stosujemy selektor identyfikatora?
- 7.** Co oznacza zapis a:link {color: green; background: yellow;} występujący w arkuszu stylów?
- 8.** Jaką rolę w modelu blokowym CSS odgrywają Border i Padding?
- 9.** Na czym polega pozycjonowanie elementów na stronie internetowej?
- 10.** Jaką rolę podczas definiowania stylów odgrywa selektor uniwersalny \*?

### Pytania egzaminacyjne

#### Zadanie 1.

W języku CSS wcięcie pierwszej linii akapitu na 30 pikseli uzyska się za pomocą zapisu:

- A.** p {text-indent: 30px; }.
- B.** p {text-spacing: 30px; }.
- C.** p {line-height: 30px; }.
- D.** p {line-indent: 30px; }.

#### Zadanie 2.

W języku CSS zdefiniowano następujące formatowanie:

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

Kolorem czerwonym zostanie zapisany:

- A.** tylko tekst pochylony nagłówka pierwszego stopnia.
- B.** tylko tekst pochylony we wszystkich poziomach nagłówków.
- C.** cały tekst nagłówka pierwszego stopnia oraz pochylony tekst akapitu.
- D.** cały tekst nagłówka pierwszego stopnia oraz cały tekst pochylony, niezależnie od tego, w którym miejscu strony się znajduje.

### Zadanie 3.

W języku CSS, aby sformatować dowolny element języka HTML w ten sposób, że po na-jechaniu na niego kursem zmienia on kolor czcionki, należy zastosować pseudoklasę:

- A.** :active.
- B.** :hover.
- C.** :visited.
- D.** :coursor.

### Zadanie 4.

Który zapis stylu CSS ustawi tło bloku na kolor niebieski?

- A.** div {background-color: blue; }.
- B.** div {border-color: blue; }.
- C.** div {shadow: bIue; }.
- D.** div {color: bIue; }.

### Zadanie 5.

W zamieszczonym przykładzie pseudoklasa `hover` sprawi, że styl pogrubiony będzie przypisany:

```
a:hover {font-weight: bold; }
```

- A.** wszystkim odnośnikom odwiedzonym.
- B.** wszystkim odnośnikom nieodwiedzonym.
- C.** każdemu odnośnikowi niezależnie od aktualnego stanu.
- D.** odnośnikowi, w momencie kiedy najechał na niego kurSOR myszy.

### Zadanie 6.

W celu określenia wysokości obrazka wyświetlanego na stronie WWW należy wyko-rystać właściwość CSS o nazwie:

- A.** padding.
- B.** margin.
- C.** height.
- D.** width.

**Zadanie 7.**

W podanej regule CSS `h1 {color: blue;}` oznacza:

- A.** klasę.
- B.** wartość.
- C.** selektor.
- D.** deklarację.

**Zadanie 8.**

W języku CSS zapis w postaci:

```
p {background-image: url("rysunek.png");}
```

sprawi, że *rysunek.png* będzie:

- A.** tłem całej strony.
- B.** tłem każdego paragrafu.
- C.** wyświetlony obok każdego paragrafu.
- D.** wyświetlony, jeżeli w kodzie zostanie zastosowany znacznik `img`.

**Zadanie 9.**

W języku CSS zdefiniowano formatowanie:

```
p > i {color: blue;}
```

Oznacza to, że kolorem niebieskim zostanie zapisany:

- A.** cały tekst paragrafu niezależnie od jego formatowania.
- B.** pochylony tekst paragrafu.
- C.** cały tekst nagłówków niezależnie od ich formatowania.
- D.** pogrubiony tekst paragrafu.

**Zadanie 10.**

Chcąc zdefiniować formatowanie tabeli w języku CSS w taki sposób, aby wiersz, który jest aktualnie wskazywany kursorem myszy, został wyróżniony np. innym kolorem, należy zastosować:

- A.** pseudoklasę `:visited`.
- B.** pseudoklasę `:hover`.
- C.** pseudoelement `:first-line`.
- D.** nowy selektor klasy dla wierszy tabeli.

## 3.10.2. Zadania

### Zadanie 3.1

Dla witryny biura podróży GLOB zdefiniuj style wyświetlanych w niej informacji. Witryna powinna być wyświetlana zgodnie ze schematem podanym na rysunku 2.44. Podczas tworzenia witryny i definiowania dla niej arkuszy stylów uwzględnij to, że może być ona wyświetlana na różnych urządzeniach (komputerze, tablecie, telefonie komórkowym).

### Zadanie 3.2

Dla utworzonej witryny internetowej na temat konkursów przedmiotowych odbywających się w szkole należy ustalić styl wyświetlanych na stronach informacji. Uczestnicy projektu powinni określić wygląd tytułów stron, tekstów zawierających informacje, różnego rodzaju list, zachowanie się odsyłaczy do innych stron internetowych oraz sposób rozmieszczenia zdjęć na stronie. Po tych ustalenach należy utworzyć zewnętrzny arkusz stylów zawierający definicję stylów dla wszystkich elementów witryny internetowej. Przygotowany arkusz stylów powinien zostać dołączony do wszystkich dokumentów HTML utworzonych przez uczestników projektu.

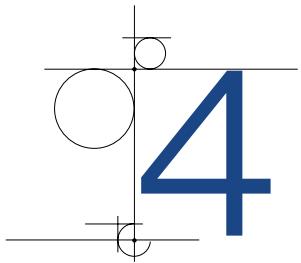
### Zadanie egzaminacyjne — cz. II

#### *Styl CSS witryny internetowej*

Styl elementów witryny zdefiniuj przy pomocy języka CSS, w osobnym pliku o nazwie [\*styl.css\*](#), plik ten zapisz w podfolderze WWW oraz prawidłowo dołącz do pliku z kodem strony.

Wymagania odnośnie do stylu CSS:

- kolor czcionki odnośników: #fffd87e,
- kolor tła banera, stopki oraz panelu lewego: #af8c4b,
- kolor tła panelu prawego: #fffd87e,
- wyrównanie tekstu banera i stopki: do środka,
- wyrównanie tekstu panelu lewego: do prawej,
- krój czcionki dla całej strony: Tahoma,
- szerokość panelu lewego: 30%,
- szerokość panelu prawego: 70%,
- wysokość paneli lewego i prawego: 500 px,
- wysokość banera: 60 px,
- wysokość stopki: 25 px,
- kolor poziomej linii w lewym panelu: #fffd87e,
- punktor list w panelu lewym: okrąg,
- wszystkie komórki tabeli obramowane ramką czarną kropkowaną o szerokości 1 px,
- włączone paski przewijania dla panelu prawego.



# Grafika na stronie internetowej

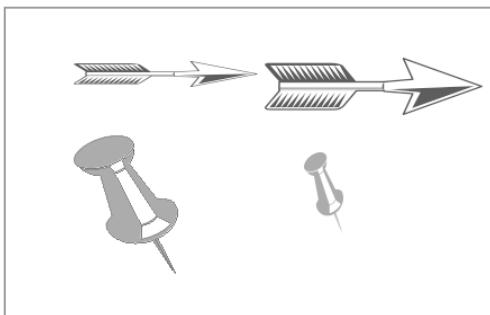
## 4.1. Rodzaje grafiki

Obrazy dostępne na komputerze mogą być tworzone przy użyciu jednej z dwóch podstawowych technik. W zależności od zastosowanej techniki wyróżniamy grafikę rastrową oraz grafikę wektorową. Grafiki te różnią się sposobem tworzenia i zapisu obrazu oraz sposobem jego edytowania.

## 4.2. Grafika wektorowa

**Grafika wektorowa** (inaczej: obiektowa) to grafika, w której obraz jest tworzony z obiektów podstawowych (prymitywnych), najczęściej figur geometrycznych (rysunek 4.1). Obrazy tej grafiki są całkowicie skalowalne, tzn. mogą być dowolnie powiększane lub zmniejszane bez straty jakości. Wadą jest to, że grafikę wektorową trzeba tworzyć od podstaw, składając ją z wielu obiektów podstawowych. Formaty plików to np. SVG i CDR. Popularne programy do tworzenia grafiki wektorowej to **CorelDRAW**, **Adobe Illustrator**, **AutoCAD**.

W grafice wektorowej każdy z obiektów jest opisywany za pomocą parametrów, np. dla odcinka podawane są współrzędne jego końców, w przypadku okręgu podaje się współrzędne środka i długość promienia. Tworzone obiekty mają atrybuty decydujące o ich wyglądzie, np. o grubości i kolorze linii, kolorze wypełnienia, stopniu przezroczystości. Atrybuty zależą głównie od stosowanego standardu opisu grafiki wektorowej.



Rysunek 4.1. Grafika wektorowa

Obrazy grafiki wektorowej mogą być łatwo konwertowane na ich odpowiedniki bitmapowe, należy jedynie podać rozdzielcość tworzonego obrazu wynikowego.

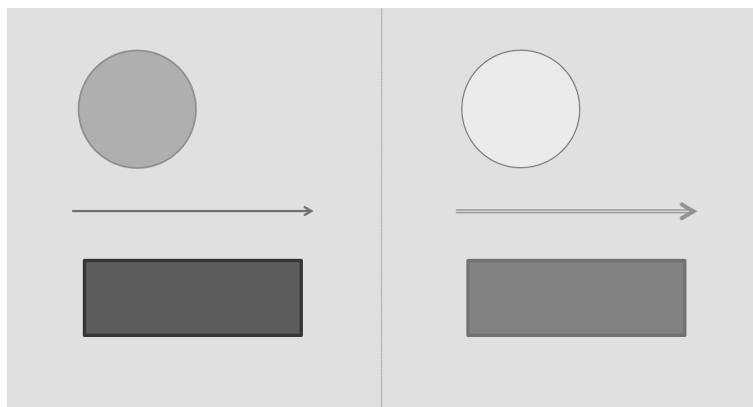
Konwersja obrazu grafiki bitmapowej na obraz grafiki wektorowej jest trudna i często nie daje oczekiwanych rezultatów.

### 4.2.1. Sposoby opisu obrazu

W grafice wektorowej sposób opisu obrazu został oparty na formułach matematycznych, dlatego obraz może być wyświetlony w maksymalnie dostępnej dla ekranu rozdzielcości. Jakość takiego obrazu jest uzależniona wyłącznie od dokładności opisu obrazu przez jego elementy (obiekty prymitywne). Obiektami prymitywnymi są najczęściej odcinki, krzywe, okręgi i wielokąty.

Jak wspomniano, zaletą tak tworzonych obrazów jest ich skalowalność (obrazy można dowolnie powiększać lub zmniejszać bez straty jakości), a także prostota opisu oraz możliwość modyfikacji poprzez zmianę parametrów obrazu. Wadą jest konieczność tworzenia obrazu od podstaw poprzez składanie go z obiektów podstawowych.

W grafice wektorowej obiekty, z których tworzony jest obraz, są opisywane za pomocą atrybutów decydujących o ich wyglądzie, np. o kolorze i grubości linii, kolorze wypełnienia. Jednak zmiana atrybutów nie wpływa na tworzony obiekt (rysunek 4.2). Można zmieniać dowolne atrybuty obiektu, nie niszcząc samego obiektu.



Rysunek 4.2. Zmiana atrybutów tworzonego rysunku

### 4.2.2. Zastosowanie grafiki wektorowej

Grafika wektorowa ma zastosowanie przy tworzeniu rysunków technicznych, schematów blokowych, diagramów i wykresów (rysunek 4.3). W technice komputerowej jest wykorzystywana do opisu czcionek, opisów grafiki trójwymiarowej, do tworzenia animacji komputerowej, do wizualizacji dużej ilości danych liczbowych (wykresy).



**Rysunek 4.3.** Zastosowanie grafiki wektorowej

### 4.2.3. Formaty plików

W grafice wektorowej wiele programów ma swoje wewnętrzne formaty zapisu, brak jednak formatów powszechnie stosowanych.

#### Format SVG

SVG (ang. *Scalable Vector Graphics*) jest uniwersalnym formatem dwuwymiarowej grafiki wektorowej. Powstał z myślą o zastosowaniu na stronach WWW.

W formacie tym można zapisywać standardowe obiekty graficzne, np.: prostokąty, elipsy, krzywe. Można również zapisywać efekty specjalne, np.: filtry, maski przezroczystości, wypełnienia gradientowe. SVG bywa też stosowany do zapisywania i odtwarzania prostych animacji.

#### Format CDR

CDR — format programu CorelDRAW.

#### Format SWF

SWF — format programu Adobe Flash. Może zawierać animacje lub apłyty o różnym stopniu interaktywności i funkcjonalności. W przeszłości technologia ta była używana do tworzenia animowanej grafiki i menu na stronach internetowych. Najpopularniejsze przeglądarki wycofują obecnie wsparcie dla Adobe Flash i nie należy go już używać. Zalecaną alternatywą są rozwiązania wprowadzone w języku HTML5.

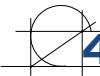
#### Format DXF

DXF — format, w którym można zapisywać zarówno rysunki 2D, jak i rysunki zawierające elementy 3D. Specyfikacja tego formatu została opracowana przez firmę Autodesk i służyła początkowo głównie do wymiany danych pomiędzy programami AutoCAD i 3D Studio.

## Programy

Istnieje wiele programów do tworzenia grafiki wektorowej. Najpopularniejsze to:

- **Adobe Illustrator** — aplikacja przeznaczona głównie do projektowania i tworzenia ilustracji.
  - **CorelDRAW** — program do edycji grafiki wektorowej. Umożliwia przygotowanie projektu do wydruku na drukarce lub w zakładzie poligraficznym.
  - **AutoCAD** — używany do komputerowego wspomagania projektowania. Umożliwia wykonanie rysunków technicznych, przygotowanie dokumentacji technicznych, pozwala na dołączenie bibliotek z szablonami i obiektami charakterystycznymi.
  - **Adobe Flash** — używany do tworzenia animacji z wykorzystaniem grafiki wektorowej na zasadzie klatek kluczowych. Wyposażony został w język skryptowy ActionScript. Umożliwia tworzenie interaktywnych animacji, reklam i prezentacji. Powstałe pliki są zapisywane w formacie SWF.
  - **Inkscape** — bezpłatny program do tworzenia grafiki wektorowej, oparty na formacie SVG opisującym statyczną i animowaną grafikę dwuwymiarową. Wykorzystywany jest do tworzenia symboli, ikon, znaków towarowych i logotypów — odpowiednik płatnego programu CorelDRAW.



## 4.3. Grafika rastrowa

**Grafika rastrowa** (inaczej: bitmapowa) to grafika, w której obraz jest tworzony z leżących obok siebie punktów (*pikseli*). To, jak gęsto na danej powierzchni są ułożone piksele, mówi o rozdzielczości obrazu. Ten rodzaj grafiki najczęściej jest stosowany do cyfrowego zapisu zdjęć lub obrazów ze skanera, gdyż wiernie oddaje wszystkie szczegóły obrazu (rysunek 4.4). Wadą grafiki rastrowej jest to, że przy dużym powiększeniu obrazu punkty (*piksele*), z których się on składa, zaczynają być widoczne (rysunek 4.5). Najczęściej spotykane formaty plików rastrowych to: BMP, JPG, PNG, GIF, TIFF. Popularne programy do tworzenia grafiki rastrowej to: Paint, GIMP, Adobe Photoshop.



**Rysunek 4.4.** Grafika rastrowa



### Rysunek 4.5.

Piksele widoczne  
przy dużym  
powiększeniu

Piksel jest najmniejszym elementem obrazu wyświetlany na ekranie monitora; to także najmniejszy element obrazu uzyskanego z aparatu cyfrowego lub ze skanera.

### 4.3.1. Sposoby zapisu obrazu

Obraz zapisany w postaci cyfrowej składa się z leżących obok siebie punktów (pikseli). Piksel jest najmniejszym elementem zapisanego obrazu. Ma postać kwadratu lub prostokąta i jest wypełniony jednym kolorem. Kolor jest definiowany dla każdego piksela osobno. W zależności od użytej palety barw punkt może przyjmować nawet do 65 tys. kolorów, a dla palety 24-bitowej nawet do 16,7 mln kolorów. Podczas zapisywania obrazu zapamiętywane są położenie punktu i jego kolor. Piksele tworzące obraz są ułożone w kolumnach i wierszach; powstaje w ten sposób siatka zwana mapą bitową. Ze względu na to, że obraz składa się z oddzielnych pikseli, przy jego powiększaniu uzyskujemy efekt schodkowania.

Przy zapisie obrazu cyfrowego są stosowane różne metody jego kompresji. Kompresja może być bezstratna (zmniejszenie objętości pliku nie powoduje pogorszenia jakości obrazu) lub stratna (po skompresowaniu następuje pogorszenie jakości obrazu).

### 4.3.2. Jakość obrazu

Na jakość obrazu wpływają takie parametry jak: rozdzielcość, rozmiar obrazu i głębia koloru.

#### Rozdzielcość

Rozdzielcość obrazu można określić na dwa sposoby.

- **Rozdzielcość geometryczna** określa liczbę pikseli przypadających na jednostkę długości obrazu. Miarą rozdzielcości jest liczba pikseli przypadająca na cal — ppi (ang. *pixels per inch*). Można stosować inną miarę rozdzielcości, np. piksel na milimetr. Wybór rozdzielcości powinien zależeć od przeznaczenia obrazu. Im większa rozdzielcość, tym lepsza jakość obrazu, ale również większy rozmiar (większa objętość) pliku. Zbyt niska rozdzielcość to pogorszenie jakości obrazu.
- **Rozdzielcość radiometryczna** określa jasność piksela. Do zapisu jasności piksela najczęściej wykorzystuje się 8 bitów i wtedy można uzyskać 256 stopni jasności. Czasami stosuje się zapis 4-bitowy, co daje 16 stopni jasności. W profesjonalnych zastosowaniach do zapisu jasności wykorzystuje się 10, 12 i więcej bitów.

#### Rozmiar obrazu

Rozmiar obrazu określa fizyczną wielkość obrazu zdefiniowaną za pomocą dwóch parametrów: szerokości (ang. *width*) oraz wysokości (ang. *height*). Można go podawać w pikselach, calach lub milimetrach.

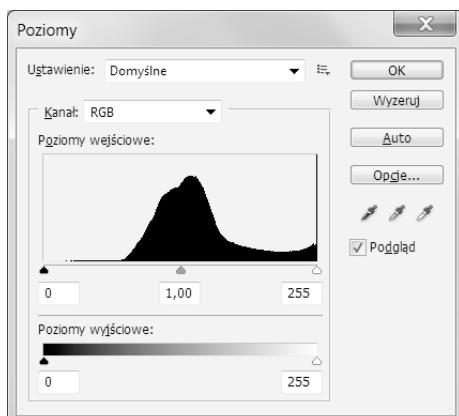
## Głębia koloru

Głębia koloru określa liczbę barw możliwych do uzyskania w obrazie. Skoro kolor każdego piksela jest zapisywany przez określoną liczbę bitów, głębia koloru jest informacją o tym, ilu bitów potrzeba, aby opisać kolor każdego piksela. Im większa liczba bitów przeznaczona na zapisanie piksela, tym większa paleta kolorów i lepsza jakość obrazu. Aby można było odwzorować kolory naturalne, liczba bitów opisująca kolor nie powinna być mniejsza niż 24. Miarą głębi koloru jest bpp (ang. *bits per pixels*).

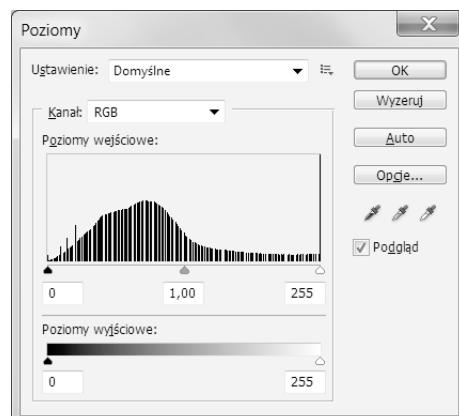
## Histogram obrazu cyfrowego

Histogram obrazu to wykres przedstawiający liczbę pikseli o określonej jasności (intensywności) występujących w obrazie. Na podstawie histogramu można określić cechy obrazu. Obraz o dobrej jakości ma histogram w miarę wyrównany, a jasności pikseli przyjmują wartości od zera do wartości maksymalnej.

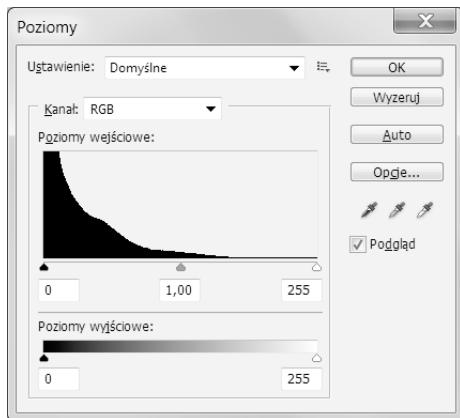
W celu poprawienia jakości obrazu można „rozciągnąć” histogram (np. z rysunków 4.6 i 4.8). Dzięki temu poprawiona zostanie rozróżnialność elementów obrazu. Niestety nie doprowadzi to do pojawienia się w obrazie informacji, których tam nie było (rysunki 4.7 i 4.9). Pozioma oś na wykresie przedstawia różne natężenie jasności — od całkowitej czerni do całkowitej bieli — pionowa zaś przedstawia liczbę pikseli należących do danej grupy jasności.



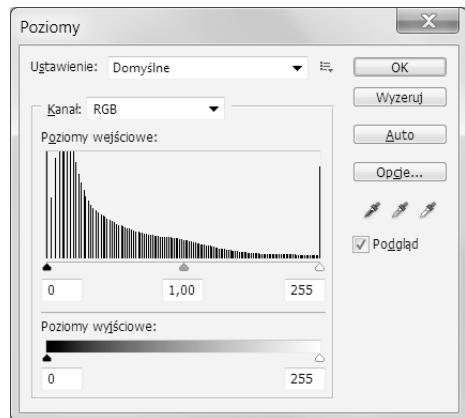
**Rysunek 4.6.**  
Histogram obrazu jasnego



**Rysunek 4.7.**  
Histogram rozciagnięty

**Rysunek 4.8.**

Histogram obrazu ciemnego

**Rysunek 4.9.**

Histogram rozciągnięty

W celu uwypuklenia szczegółów obrazu, które są mało widoczne ze względu na niewielki kontrast, można zastosować wyrównanie histogramu.

Dla zdjęcia niedoświetlonego dane na histogramie skupią się z lewej strony osi poziomej, dla prześwietłonego — z prawej strony osi poziomej. W prawidłowo naświetlonym zdjęciu wykres jest równomierny i z obu stron kończy się wraz z bokami ramki (rysunek 4.10).

**Rysunek 4.10.** Histogram dla obrazu prawidłowo naświetlonego

### 4.3.3. Metody pozyskiwania obrazów

#### Zdjęcia fotograficzne

Zdjęcia wykonujemy, korzystając najczęściej z cyfrowych aparatów fotograficznych wyposażonych w pamięci (flash). Dodatkowo aparaty cyfrowe często mają wyświetlacz LCD, który udostępnia podgląd obrazu w czasie robienia zdjęcia. Aparat cyfrowy wykorzystuje zestaw soczewek (układ optyczny), które rzutują obraz na przetwornik optoelektroniczny (matryca CMOS — ang. *Complementary Metal Oxide Semiconductor*). Za pomocą przetwornika analogowo-cyfrowego obraz jest zamieniany z postaci analogowej na binarną. Postać binarna obrazu zapisywana jest w karcie pamięci w jednym z formatów przeznaczonych dla grafiki rastrowej. Format zapisu często można wybrać w oprogramowaniu aparatu. Najpopularniejszym formatem używanym do zapisu fotografii cyfrowej jest JPG. Coraz większą popularność zdobywa format RAW.

Gotowe zdjęcia można pozyskiwać ze stron internetowych oferujących nieodpłatnie dużą liczbę zdjęć. Często są to zdjęcia bardzo wysokiej jakości, a jedynym warunkiem ich wykorzystania jest podanie pod zdjęciem nazwiska autora. Jedną z największych i najlepszych stron z darmowymi zdjęciami jest <https://pl.freeimages.com/>. Inne interesujące strony to: <http://www.flickr.comcreativecommons/>, <http://www.stockvault.net/> oraz wyszukiwarka zdjęć <http://www.everystockphoto.com/>. Ciekawą alternatywą są również internetowe bazy zdjęć dostępnych odpłatnie, np. <https://www.istockphoto.com/pl> lub <https://www.shutterstock.com/pl>.

Im większa rozdzielcość zisanego obrazu, tym lepsza jakość i tym większy format zdjęcia w pikselach. W specyfikacji aparatu fotograficznego powinna być podana maksymalna rozdzielcość matrycy (określająca maksymalną rozdzielcość zdjęć w megapikselach — Mpx) oraz maksymalna rozdzielcość obrazu (określająca maksymalny rozmiar obrazu w pikselach).

#### Skaner

Skaner to urządzenie służące do odczytania obrazu, np. z kartki papieru, i przetworzenia go na postać binarną w celu dalszej obróbki za pomocą komputera.

Podczas skanowania obraz jest oświetlany, a odbite od niego światło jest kierowane do czujników, które przetwarzają natężenie światła każdej z barw na wartości elektryczne. W przetworniku analogowo-cyfrowym wartości elektryczne są zamieniane na postać cyfrową. Postać binarna obrazu jest zapisywana na komputerze w jednym z formatów dostępnych dla grafiki rastrowej. Ustawienie formatu zapisu obrazu odbywa się w programie obsługującym skaner.

Podstawowym parametrem skanowania jest rozdzielcość. Jej wielkość zależy od układu optycznego zastosowanego w skanerze. Skanując zdjęcia i rysunki, najlepiej wybrać rozdzielcość 300 dpi i 24-bitową paletę kolorów. Osiąga się wtedy optymalny stosunek jakości obrazu do objętości pliku. Aby uzyskać obraz bardzo dobrej jakości, należy zwiększyć rozdzielcość, co jednak powoduje zwiększenie wielkości pliku wynikowego.

## 4.3.4. Formaty plików graficznych

Format pliku określa sposób jego zapisu. W grafice komputerowej o formacie pliku decyduje rodzaj tworzonej grafiki oraz przeznaczenie pliku. Najważniejszymi cechami pliku graficznego są jego rozmiar oraz jakość zapisanego obrazu. Znając przeznaczenie pliku, można jego format dobrać tak, aby pogodzić te dwie wykluczające się cechy.

### Format BMP

BMP to podstawowy format grafiki rastrowej. Każdy piksel opisany jest przez kolor w formacie RGB. W przypadku obrazów o dużej rozdzielczości (np. zdjęć z aparatu fotograficznego) rozmiary pliku mogą dochodzić do kilkudziesięciu MB. Format BMP umożliwia kompresję danych, ale i tak nie zmienia to znacznie rozmiarów pliku. Duże rozmiary plików są podstawową wadą tego formatu. Zaletą natomiast jest bezstratny zapis oraz możliwość otwarcia pliku w większości programów.

### Format JPEG

JPEG to jeden z najpopularniejszych formatów plików graficznych. Został opracowany na potrzeby internetu. Charakteryzuje się bardzo dużą kompresją. Ze względu na to, że jest oparty na kompresji stratnej, należy pamiętać o tym, że im większy jej stopień, tym gorsza jakość ostatecznego obrazu. Widoczne jest to szczególnie w grafikach zawierających wyraźne linie i krawędzie, dlatego JPEG jest nieczęściej stosowany przez grafików do zapisu efektów ich pracy. Natomiast w fotografiach, w których nie ma ostrych krawędzi i zmian barwy, błędy kompresji są widoczne dopiero przy wysokim jej współczynniku. Z tego powodu format JPEG jest często stosowany do zapisu zdjęć w aparatach cyfrowych. Dzięki niewielkim rozmiarom plików jest najczęściej używanym formatem zdjęć na stronach internetowych.

### Format GIF

GIF to format plików graficznych opracowany w celu umożliwienia zapisu prostych obrazów o niewielkiej liczbie kolorów. Jest wykorzystywany przy projektowaniu nagłówków, przycisków, ikon i prostej grafiki. Pozwala na zapisywanie i odtwarzanie prostych animacji. Umożliwia tworzenie grafik transparentnych (przezroczystych). Jedną z zalet tego formatu jest niewielki rozmiar plików. Z tych powodów GIF jest często stosowany przy projektowaniu stron internetowych.

### Format PNG

Format PNG służy do zapisu plików graficznych i stanowi połączenie zalet formatów GIF oraz JPG. Został opracowany z myślą o projektowaniu stron internetowych i w związku z wprowadzeniem opłat za wykorzystywanie w programach graficznych formatu GIF. Zaletami tego formatu są bezstratna kompresja i tworzenie plików o niewielkich rozmiarach oraz obsługiwanie kanału alfa.

## Format PSD

PSD jest formatem programu Adobe Photoshop. Zapisuje wszystkie elementy formowania tego programu (warstwy, maski, efekty, przezroczystość). Pliki zapisane w tym formacie mogą być odczytywane przez inne programy graficzne, m.in. Paint Shop Pro czy GIMP, ale przed taką operacją dokument trzeba spłaszczyć (jeżeli zawiera kilka warstw, to muszą one zostać połączone w jedną). Jest to format służący do edytowania wielu warstw, do ich tworzenia i modyfikowania. Nie powoduje utraty jakości obrazu, ale zazwyczaj tworzy pliki o dużych rozmiarach. Po opracowaniu gotowego obrazu najlepiej ostateczny efekt zapisać w innym formacie.

## Format TIFF

TIFF (ang. *Tagged Image File Format*) jest formatem plików graficznych, który służy do zapisywania grafiki rastrowej. Format ten pozwala na zapisywanie obrazów stworzonych w skali szarości oraz w wielu trybach koloru. Przechowuje ścieżki i kanały alfa, profile koloru i komentarze tekstowe. TIFF umożliwia stosowanie kompresji bezstratnej. Pliki zapisane w tym formacie mają rozszerzenie *.tif* lub *.tiff*.

## Format RAW

Format RAW (ang. *raw*) jest zapisem danych z matrycy aparatu, które nie zostały podane żadnej obróbce. Charakteryzuje się dużym zakresem tonalnym oraz kompresją bezstratną. Przez konwersję tak zapisanych danych tworzony jest plik JPG. Przekształcając format RAW w JPG, mamy nad tym procesem pełną kontrolę i możemy odzyskać pełną informację zarejestrowaną przez matrycę. Otwierając plik w formacie RAW na komputerze, odzyskujemy dane, które w przypadku JPG byłyby utracone.

Jedynym ustandaryzowanym formatem RAW jest format DNG (ang. *Digital Negative*, cyfrowy negatyw) stworzony przez firmę Adobe. Jest to standard otwarty.

Każdy producent aparatów ma własny standard RAW, np.:

- Canon — to pliki z rozszerzeniem CRW,
- Nikon — NEF,
- Minolta — MRW,
- Pentax — PEF,
- Sony — ARW.5.4.

## 4.4. Modele przestrzeni barw

Do opisu kolorów używanych w grafice komputerowej stosuje się modele przestrzeni barw. Każdy model przestrzeni barw ma ściśle określoną paletę kolorów. Najpopularniejsze modele przestrzeni barw to: **RGB, CMYK, HSV (HSB)**.

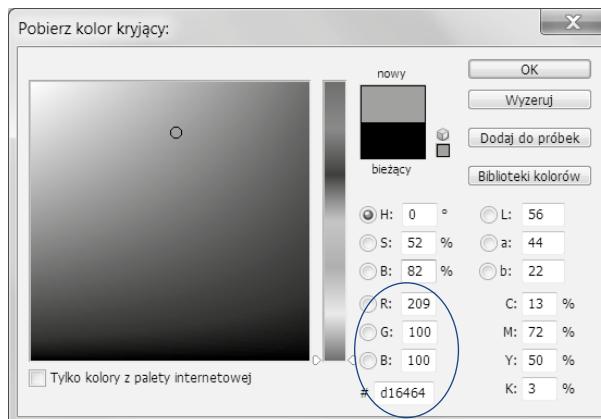
## 4.4.1. Model RGB

Podstawowym modelem barw stosowanym w grafice komputerowej jest model **RGB**. W tym modelu barw dowolny kolor można uzyskać, składając go z trzech podstawowych kolorów: czerwonego (ang. *Red*), zielonego (ang. *Green*) i niebieskiego (ang. *Blue*). Wynika to z właściwości oka ludzkiego, które reaguje na te trzy barwy. Kolory, mieszając się, dodają swoje wartości, co w zależności od natężenia każdej z barw tworzy w efekcie określony kolor. Jest to tzw. **model addytywny**. Ten model barw najczęściej jest stosowany w urządzeniach wyświetlających obraz, np.: monitorach komputerów, telewizorach, aparatach fotograficznych i skanerach.

Każda z barw składowych koloru jest zapisywana za pomocą 8 bitów (1 bajt) i przyjmuje wartości z zakresu **0–255**. Jeżeli dla wszystkich składowych barw zostanie ustawiona wartość **0**, to otrzymamy kolor czarny. Natomiast gdy dla wszystkich składowych barw zostanie ustawiona wartość **255**, to otrzymamy kolor biały. Pozostałe kolory wynikają z odpowiednich ustawień wartości kolorów składowych.

Skoro wszystkich kombinacji wartości składowych jest ponad 16 mln, to tyle kolorów można uzyskać, mieszając poszczególne barwy.

W modelu RGB kolor każdego piksela jest opisywany za pomocą wartości kolorów składowych. W grafice komputerowej czy przy tworzeniu stron internetowych często definiuje się kolory wprost, podając ich wartości w kodzie heksadecymalnym (np. `background-color: #d16464` — rysunek 4.11).



**Rysunek 4.11.** Próbnik kolorów. Możliwość wybrania składowych koloru poprzez podanie wartości R, G, B lub kodu heksadecymalnego

Należy zwrócić uwagę na to, że na różnych urządzeniach wyświetlających uzyskiwany kolor może wyglądać zupełnie inaczej. Dzieje się tak dlatego, że każde z takich urządzeń może mieć własny zakres barw możliwych do uzyskania.

## Kanał alfa

Kanał alfa definiuje przezroczyste obszary grafiki. Jest zapisywany wewnątrz obrazu razem z trzema wartościami modelu barw RGB. Wartość zerowa tego parametru oznacza pełną przezroczystość, wartość maksymalna — pełne zabarwienie.

Przy zapisie 32-bitowym kanał alfa jest zapisany na 8 bitach, a trzy pozostałe kanały 8-bitowe przechowują informacje o poziomie nasycenia kolorów RGB. Powstały w ten sposób format RGBA (RGB + Alpha) pozwala określić stopień przezroczystości oraz wartość poszczególnych kolorów modelu RGB. Nie wszystkie programy do obróbki grafiki obsługują kanał alfa. Te, które go obsługują, definiują go w różny sposób. Także nie wszystkie formaty plików graficznych po zapisaniu przechowują informacje o kanale alfa.

### 4.4.2. Model CMYK

Kolejnym modelem barw bardzo często stosowanym w grafice jest model CMY lub rozszerzony model CMYK. Jest to model barw używany przy różnego rodzaju wydrukach. Model barw RGB znakomicie nadaje się do wyświetlania grafiki na ekranie. Określone kolory uzyskujemy poprzez składanie kolorów podstawowych. Ta metoda nie sprawdza się natomiast przy mieszaniu kolorów farb w trakcie drukowania grafiki. Podstawowymi kolorami nadającymi się do uzyskania określonego koloru w procesie wydruku są kolory będące dopełnieniami barw w modelu RGB. Dopełnieniem koloru R (*Red*) jest kolor G+B (*Green + Blue*). Jest to kolor *Cyan* (turkusowy). Dopełnieniem koloru G (*Green*) jest kolor R+B (*Red + Blue*), czyli *Magenta* (purpurowy), a dopełnieniem koloru B (*Blue*) jest kolor R+G (*Red + Green*), czyli *Yellow* (żółty). Są to barwy podstawowe dla modelu CMY i na ich bazie tworzone są pozostałe kolory.

Modele RGB i CMY są modelami dopełniającymi. Oznacza to, że barwy definiowane w modelu CMY są dopełnieniem barw z RGB i w związku z tym możemy przeliczać wartości koloru z jednego modelu na wartości z drugiego. Większość programów do tworzenia grafiki robi to automatycznie, ale proces ten można przeprowadzić samodzielnie. Odejmując od liczby 255 wartość barwy np. w modelu RGB, otrzymamy wartość jej dopełnienia w modelu CMY.

W praktyce częściej od modelu CMY jest spotykany model CMYK. Oprócz barw modelu CMY zawiera on barwę *Black* (czarny). W tym modelu jest ona oznaczona literą K w celu odróżnienia jej od barwy niebieskiej (*Blue*) używanej w modelu RGB. Stosowanie modelu CMYK wynika z tego, że mimo iż kolor czarny można otrzymać z kolorów CMY, w praktyce jest to trudne (najczęściej uzyskuje się kolor ciemnobrunatny). Dodanie koloru czarnego poprawia jakość drukowanego obrazu. Innym powodem są względy finansowe — do drukowania czarnym atramentem nie trzeba zużywać kolorowych atramentów modelu CMY.

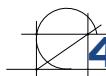
Model CMYK jest wykorzystywany w urządzeniach drukujących za pomocą farb, tonerów, atramentów — takich jak drukarki, kserokopiarki czy maszyny drukujące

stosowane w przemyśle poligraficznym. Podczas mieszania odejmowane są składowe koloru w celu uzyskania czerni. Jest to tzw. **model subtraktywny**.

### 4.4.3. Model HSV

Model barw HSV (ang. *Hue Saturation Value*) powstał na bazie sposobu, w jaki barwy są postrzegane przez człowieka. Otóż człowiek najczęściej korzysta ze standardowej palety kolorów. Dobierając barwę, wybiera z palety odpowiedni kolor farby, a dodając kolor biały lub czarny, przyjemnia lub rozmywa kolor podstawowy.

W modelu tym parametr H (*hue* — barwa) określa kolor (np. czerwony, zielony), parametr S (*saturation* — nasycenie) określa nasycenie koloru (np. czy kolor jest rozmyty, tzn. czy zawiera dużo szarości), a parametr V (*value* — wartość) — jasność koloru. Czasami w celu określenia jasności koloru używa się liter B, L lub I.



## 4.5. Ogólne zasady tworzenia grafiki na potrzeby strony internetowej

Jednym z istotnych elementów projektowanej strony internetowej jest grafika. Jeżeli strona ma być lekka, użyteczna, dobrze zoptymalizowana i prosta w obsłudze, to grafika, którą będziemy tworzyli, musi współpracować z innymi elementami witryny. Narzędzia graficzne można używać do tworzenia przycisków nawigacyjnych, ikon, tła strony, logo, tapety oraz do modyfikowania fotografii.

### 4.5.1. Zasady tworzenia grafiki

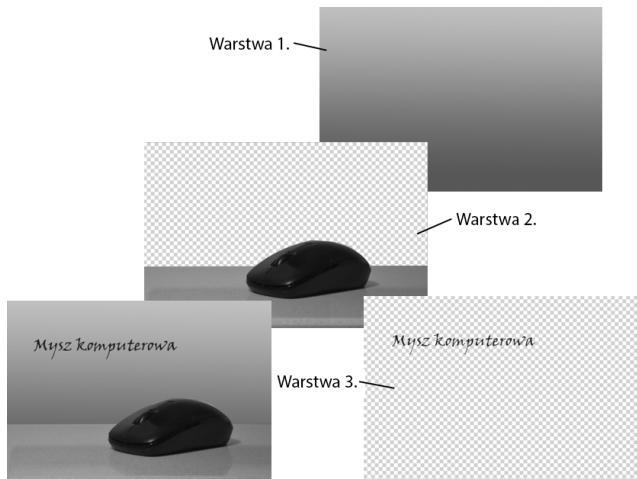
- Grafikę tworzymy w skali 1:1. Staramy się unikać skalowania obrazów, może to bowiem spowodować pogorszenie ich jakości.
- Obrazy tworzymy w tzw. rozdzielczości ekranowej. Wynosi ona 72 ppi. Taka rozdzielcość pozwoli nam dopasować grafikę do tego, co będzie widoczne na monitorze.
- Wybierając paletę kolorów, unikamy stosowania modelu CMYK. Powinien on być wykorzystywany tylko do druku. W internecie korzystamy z modelu RGB.
- Stosujemy formaty plików zgodnie z zasadą, że zdjęcia zapisujemy w formacie JPG, a grafiki z małą liczbą kolorów — w formacie GIF. Jeżeli zależy nam na jakości lub chcemy zastosować przezroczystość, zapisujemy grafikę w formacie PNG.

Tworząc grafikę na potrzeby strony internetowej, można korzystać z różnych technik. Najpopularniejsza i powszechnie stosowana w internecie jest grafika bitmapowa. Nadaje się ona do przedstawiania i modyfikowania fotografii oraz do tworzenia skomplikowanych obrazów.

## Zaawansowana grafika — warstwy

W tworzeniu zaawansowanej grafiki niezbędne są narzędzia, które pozwolą na niezależne edytowanie różnych jej elementów. Takim narzędziem dostępnym w zaawansowanych programach graficznych są warstwy.

Warstwy to odpowiedniki przezroczystych folii nakładanych na siebie. Na każdej z nich jest umieszczany fragment obrazu. Po złożeniu utworzą one całość (rysunek 4.12).



**Rysunek 4.12.** Tworzenie obrazu z warstw

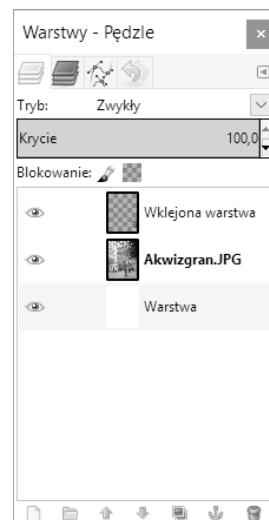
Przy tworzeniu zaawansowanej grafiki każdy fragment obrazu powinien być tworzony na osobnej warstwie. Pozwoli to na edytowanie wybranego elementu obrazu, podczas gdy pozostałe elementy pozostaną niezmienione.

Praca na warstwach umożliwia pełną kontrolę nad dworznym obrazem. Wybraną warstwę można ukryć, aby nie przesyłała innych warstw. Można zmieniać kolejność wyświetlania warstw lub usuwać te niepotrzebne.

Na rysunku 4.12 widać trzy warstwy. **Warstwa 1.** zawiera tło grafiki, **Warstwa 2.** to mysz komputerowa, **Warstwa 3.** zawiera napis „Mysz komputerowa”. Po złożeniu wszystkich warstw otrzymamy kompletny obraz.

W programie GIMP do zarządzania warstwami służy panel **Warstwy — Pędzle**, a w nim — zakładka **Warstwy** (rysunek 4.13).

W programie Photoshop do zarządzania warstwami służy panel **Warstwy**. Znajduje się on w prawej dolnej części okna programu (rysunek 4.14).



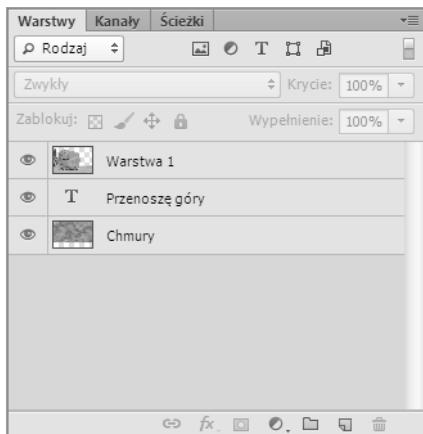
**Rysunek 4.13.**  
Panel Warstwy  
w programie GIMP

Warstwy w panelu są wyświetlane w postaci stosu. Warstwa, która znajduje się na górze, przykrywa warstwy leżące niżej. Na samym spodzie znajduje się warstwa **Tło**. Klikając warstwę, wybieramy ją do edytowania. Warstwy do edytowania można wybierać również bezpośrednio na obrazie, klikając wybrany fragment obrazu z wciskniętym klawiszem **Ctrl**. Warstwy na stosie można przemieszczać, zmieniając kolejność ich ułożenia.

## Zaawansowana grafika — maski

W zaawansowanych programach do tworzenia grafiki maski są obok warstw najistotniejszym narzędziem do tworzenia grafiki.

Maski działają podobnie jak przezroczystości warstwy. Ustawiając maskę dla wybranej warstwy, zobaczymy zamglony obraz warstw położonych niżej. Maskę możemy zastosować dla całej warstwy lub dla jej części. Jeżeli zastosujemy ją dla części warstwy, to uzyskamy efekt odkrycia warstwy położonej poniżej, tylko w tym miejscu, w którym tego potrzebujemy. Ważne jest to, że efekt maski jest odwracalny. Wymazując określony element warstwy, możemy go w dowolnym momencie przywrócić.



**Rysunek 4.14.**

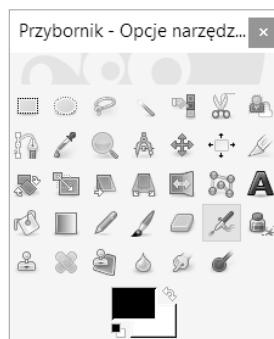
Panel Warstwy w programie Photoshop

# 4.6. Program GIMP

GIMP to bezpłatny program do edycji grafiki rastrowej. Za jego pomocą można wykonywać następujące operacje: retusz, skalowanie, kadrowanie, rysowanie, dodawanie tekstu. Do bardziej zaawansowanych narzędzi należy możliwość pracy na warstwach i kanałach, tworzenia ścieżek i wykorzystania kanału alfa. Program ten pozwala tworzyć elementy graficzne strony internetowej, takie jak: przyciski, logo, motywów czy tło strony. GIMP obsługuje podstawowe formaty graficzne: JPG, BMP, PNG, TIFF, GIF i inne.

## 4.6.1. Wprowadzenie

Po uruchomieniu programu GIMP zostaną otwarte trzy okna. Pierwsze z nich to okno **Edytora obrazów**, w którym odbywa się edycja obrazu. Drugie to **Przybornik** z narzędziami do pracy z grafiką (rysunek 4.15) i możliwością ustawiania atrybutów wybranego narzędzia.



**Rysunek 4.15.**

Okno Przybornika w programie GIMP

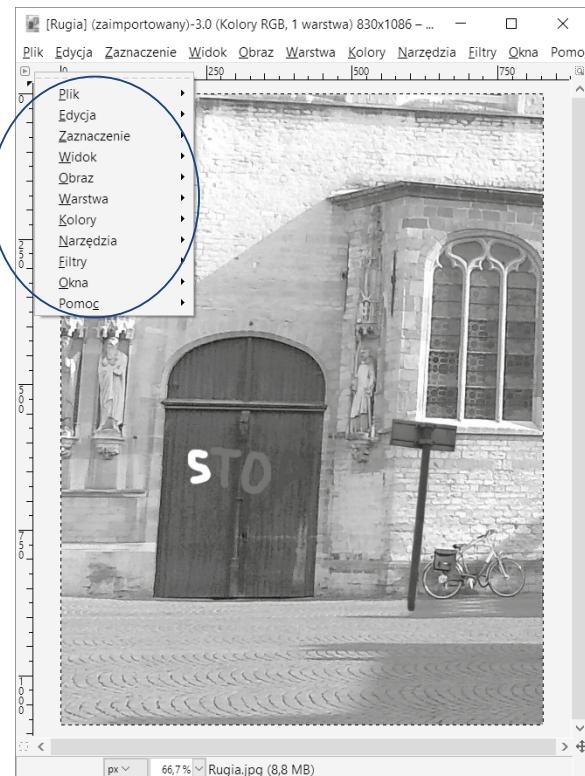
Trzecie okno zawiera kilka dodatkowych narzędzi dostępnych przez zaznaczenie zakładki obsługującej dane narzędzie (rysunek 4.16). Zakładka *Warstwy* zarządza warstwami, zakładka *Kanały* — kanałami kolorów i kanałami masek, zakładka *Ścieżki* — ścieżkami (krzywe Béziera).

W górnej części okna *Edytora obrazu* po wybraniu z menu *Plik/Otwórz* można otworzyć obraz do edycji. Dostęp do menu wyświetlnego obrazu jest możliwy po kliknięciu strzałki umieszczonej w lewym górnym rogu obrazu lub po kliknięciu prawym przyciskiem myszy w obraz (rysunek 4.17).



**Rysunek 4.16.**

Okno Warstwy w programie GIMP



**Rysunek 4.17.**

Wybór menu wyświetlnego obrazu

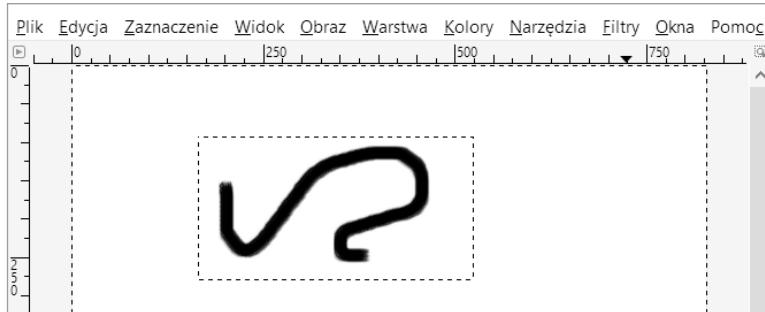
## 4.6.2. Zaznaczanie

W programie GIMP można modyfikować fragmenty obrazu tylko w obszarze zaznaczonym.

Pierwszym narzędziem dostępnym w *Przyborniku* jest *Zaznaczenie prostokątne*. Narzędzie to służy do zaznaczania regularnych kształtów. Jeżeli chcemy przesunąć zaznaczenie, należy wcisnąć klawisz *Alt* i przesunąć zaznaczony obszar za pomocą myszy.

## Ćwiczenie 4.1

Aby sprawdzić działanie narzędzia *Zaznaczenie prostokątne*, wybierz pierwszą ikonę w lewym górnym rogu *Przybornika* i w pustym dokumencie narysuj prostokąt. Następnie wybierz narzędzie *Pędzel* i maluj dowolne kształty w zaznaczonym obszarze i poza nim (rysunek 4.18). Tylko w zaznaczonym obszarze możliwe jest edytowanie obrazu.



**Rysunek 4.18.** Obszar edytowania ograniczony do zaznaczenia

## Odejmowanie/dodawanie obszarów zaznaczenia

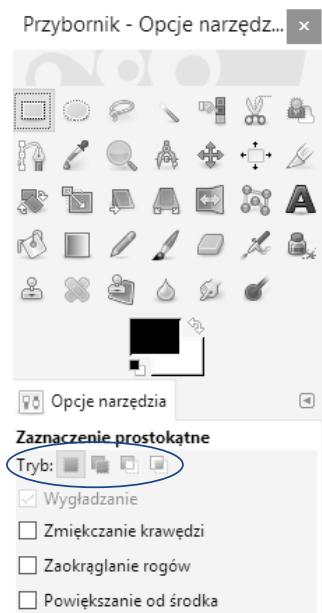
Zaznaczenie prostokątne można modyfikować przed dodanie do zaznaczonego obszaru lub odjęcie od niego określonych obszarów.

## Ćwiczenie 4.2

W *Przyborniku* wybierz *Zaznaczenie prostokątne* i narysuj prostokąt.

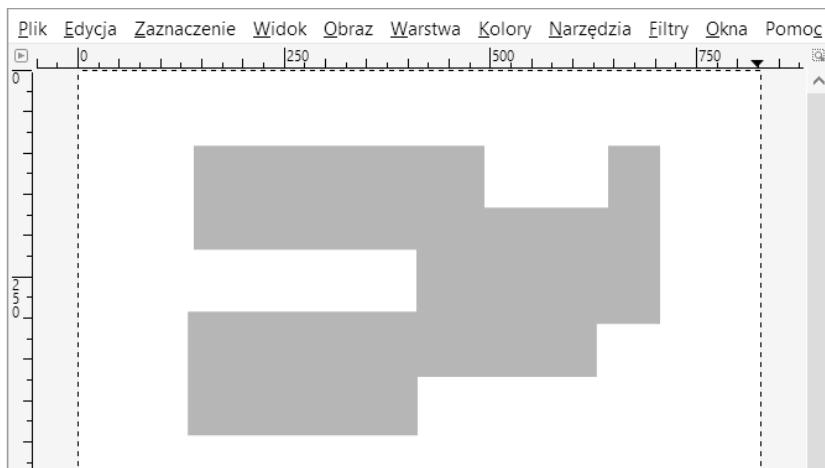
W opcjach wybranego narzędzia można określić tryb zaznaczenia (rysunek 4.19). Dostępne są: *Zastępuje bieżące zaznaczenie* (domyślny), *Dodaje do bieżącego zaznaczenia*, *Odejmuje od bieżącego zaznaczenia* i *Część wspólna z bieżącym zaznaczeniem*.

Przełączamy się do trybu *Odejmuje od bieżącego zaznaczenia* i rysujemy w zaznaczonym obszarze prostokąty. Następnie przełączamy się do trybu *Dodaje do bieżącego zaznaczenia* i rysujemy poza zaznaczonym obszarem kolejne prostokąty. Efekt tego działania został pokazany na rysunku 4.20. Zaznaczony obszar można pokolorować, wybierając narzędzie *Wypełnienie kubelkiem*.



**Rysunek 4.19.**

Wybór trybu zaznaczenia



**Rysunek 4.20.** Dodawanie i odejmowanie obszarów zaznaczenia

Kolor wypełnienia można ustawić, wybierając ikonę *Kolor pierwszoplanowy oraz tła*. Zostanie otwarte okno z paletą kolorów, z której można wybrać odpowiedni kolor.

Linia zaznaczenia zostanie usunięta po wciśnięciu klawiszy *Ctrl+Shift+A*.

Inne rodzaje zaznaczenia dostępne w *Przyborniku* to:

- *Selekcja eliptyczna* — wyodrębnia obszary o kształcie elipsy;
- *Lasso* — wyodrębnia obszary o nieregularnym kształcie;
- *Różdżka* — wyodrębnia obszary na zasadzie podobieństwa pikseli (kolor, jasność).

## Działanie różdżki

*Różdżka* to narzędzie do zaznaczania obszarów o podobnym kolorze.

### Ćwiczenie 4.3

Narzędzie *Różdżka* wykorzystaj do zaznaczenia obszarów o podobnym kolorze na wcześniej przygotowanym zdjęciu (rysunek 4.21).

Wybieramy w *Przyborniku* narzędzie *Różdżka* i w oknie z otwartym zdjęciem klikamy obszar o wybranym kolorze. Aby do zaznaczenia dodać kolejne obszary, klikamy je myszą z wciśniętym klawiszem *Shift*. Zaznaczony obszar kolorujemy za pomocą narzędzia *Wypełnienie*



**Rysunek 4.21.**

Obszary zaznaczone za pomocą narzędzia Różdżka

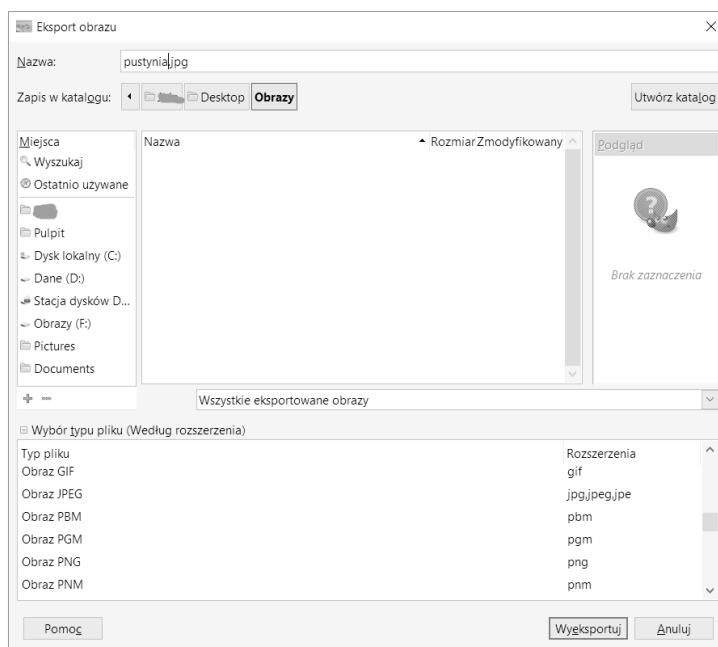
*kubikiem* kolorem ustawionym jako kolor pierwszoplanowy (rysunek 4.22).

### 4.6.3. Modyfikowanie obrazu

Zmodyfikowany obraz można zapisać, wybierając z menu *Plik/Wyeksportuj*. W otwartym oknie podajemy nazwę pliku, w dolnej jego części rozwijamy opcję *Wybór typu pliku (Według rozszerzenia)* i wybieramy format zapisu obrazu (rysunek 4.23).

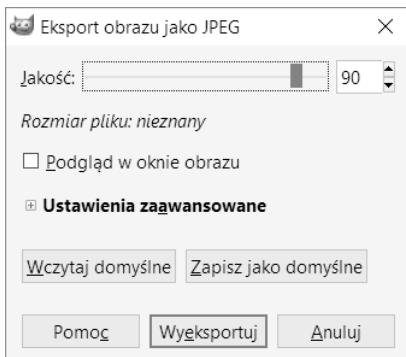


**Rysunek 4.22.**  
Wypełnienie kolorem zaznaczonego obszaru

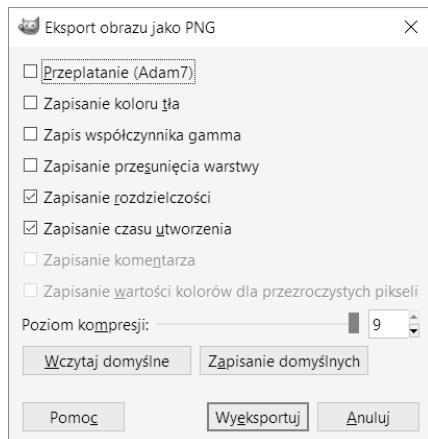


**Rysunek 4.23.** Okno eksportu obrazu do wybranego formatu

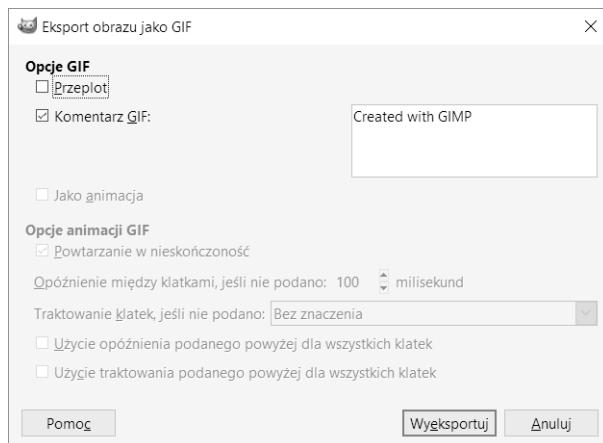
W zależności od tego, do jakiego formatu zostanie wyeksportowany plik, zostanie otwarte okno, w którym możliwe będzie ustalenie atrybutów zapisu pliku (rysunki 4.24, 4.25, 4.26). Istotnym elementem jest suwak *Jakość*, który odpowiada za ustalenie stopnia kompresji. Zasada jego działania jest prosta: im większa wartość, tym lepsza jakość pliku. Liczba 100 oznacza brak kompresji. Dobór odpowiedniej wartości w dużym

**Rysunek 4.24.**

Eksport obrazu do formatu JPEG

**Rysunek 4.25.**

Eksport obrazu do formatu PNG

**Rysunek 4.26.** Eksport obrazu do formatu GIF

stopniu zależy od konkretnego obrazka. Dlatego warto poeksperymentować w celu uzyskania najlepszego kompromisu pomiędzy jakością a rozmiarem pliku wynikowego.

## Lupa

Narzędzie *Lupa* służy do powiększania i zmniejszania podglądu obrazu. Po wybraniu *Lupy* kliknięcie w obraz spowoduje jego powiększenie. Natomiast kliknięcie w obraz z wciskniętym klawiszem *Ctrl* pomniejszy go. Ten sam efekt uzyskamy, naciskając klawisz + (powiększenie podglądu) lub klawisz – (pomniejszenie podglądu).

## Właściwości obrazu

Informacje o obrazie otwartym w programie GIMP można odczytać, wybierając z menu *Obraz/Właściwości obrazu*. Zostanie otwarte okno pokazane na rysunku 4.27.



**Rysunek 4.27.** Właściwości obrazu otwartego w programie GIMP

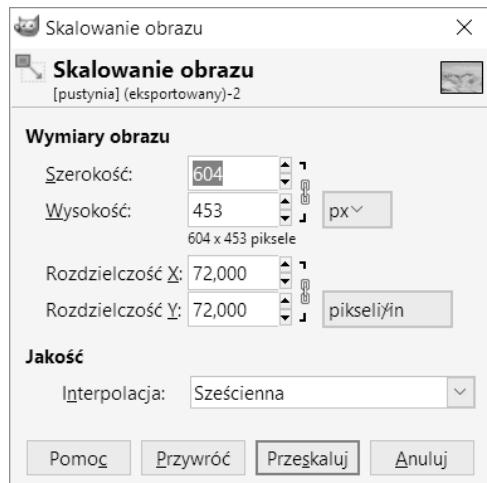
## Skalowanie obrazu

Zmiany rozmiaru obrazu można dokonać po wybraniu z menu *Obraz/Skaluj obraz*. Zostanie otwarte okno pokazane na rysunku 4.28. Poza możliwością zmiany rozmiaru obrazka ważnym elementem jest opcja *Interpolacja*, która pozwala zmienić sposób, w jaki wykonywane jest skalowanie. Każda z metod działa inaczej i wpływa na jakość pliku wynikowego.

### Ćwiczenie 4.4

Przygotuj kilka zdjęć do modyfikacji w kolejnych ćwiczeniach.

Otwieramy w programie GIMP przygotowany plik. Następnie wybieramy z menu *Obraz/Właściwości obrazu*. W otwartym



**Rysunek 4.28.**

Skalowanie obrazu

oknie odczytujemy parametry obrazu i zmieniamy jego rozmiar: szerokość **800** pikseli, wysokość **600** pikseli.

Zapisujemy obraz w formatach JPG, GIF i PNG. Używamy do tego polecenia **Plik/Wyeksportuj**. W otwartym oknie wybieramy format JPG i zaznaczamy pole **Podgląd** w oknie obrazu, a następnie ustawiamy jakość na **60** i zapisujemy obraz. Powtarzamy tę czynność dla wszystkich formatów. Na koniec porównujemy jakość oraz wielkość każdego z otrzymanych obrazów.

## Korekta jasności i kontrastu

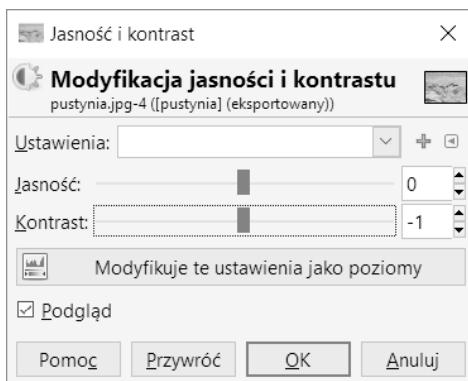
### Ćwiczenie 4.5

Otwórz w programie GIMP wybrany plik. Kliknij prawym przyciskiem myszy obraz i wybierz z menu **Kolory/Jasność i kontrast**. W otwartym oknie (rysunek 4.29), przesuwając suwaki **Jasność** i **Kontrast**, modyfikuj obraz aż do uzyskania oczekiwanej wyniku.

Korektę jasności i kontrastu można również wykonać za pomocą krzywych koloru. Klikamy prawym przyciskiem myszy obraz i wybieramy z menu **Kolory/Krzywe....** W otwartym oknie (rysunek 4.30) łapiemy myszą za krzywą i odpowiednio ją kształtuując, staramy się uzyskać oczekiwany wynik.

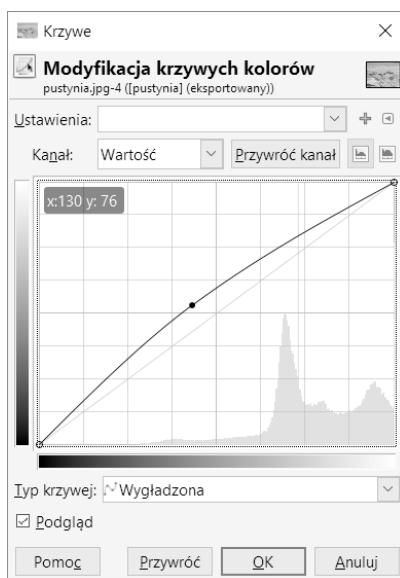
**Rysunek 4.30.**

Modyfikacja kolorów obrazu za pomocą krzywych kolorów



**Rysunek 4.29.**

Okno ustawiania jasności i kontrastu obrazu



## Ćwiczenie 4.6

Z przygotowanego zdjęcia usuń z użyciem zaznaczenia i warstw niepotrzebne elementy.

Ze zdjęcia pokazanego na rysunku 4.31 zostanie usunięte graffiti.

W oknie *Przybornik* wybieramy narzędzie *Zaznaczenie prostokątne*. W opcjach tego narzędzia ustawiamy zmiękczanie krawędzi i zaokrąglanie rogów: *Zmiękczanie krawędzi: Promień — 30, Zaokrąglanie rogów: Promień — 5*. W obszarze muru zaznaczamy obszar, który będzie można skopiować i wkleić w miejsce graffiti. Klikamy prawym przyciskiem myszy i wybieramy z menu *Edycja/Skopij*. Następnie wybieramy z menu *Edycja/Wklej jako/Nowa warstwa*. Skopiowany fragment powinien się pojawić w lewym górnym rogu zdjęcia.

W oknie *Przybornik* wybieramy narzędzie *Przesunięcie* i przesuwamy skopiowany fragment w odpowiednie miejsce. Jeżeli nie został usunięty cały napis, skopiowany element wykorzystujemy ponownie, kopując go jako nową warstwę (należy wybrać *Edycja/Wklej jako/Nowa warstwa* i przesuwać skopiowany fragment w odpowiednie miejsce). Zauważmy, że podczas kopiowania w oknie *Warstwy — Pędzle* pojawiły się nowe warstwy.

## Ćwiczenie 4.7

Otwórz w programie GIMP plik ze zdjęciem, które wymaga korekty. Do jej wykonania zastosuj poznane narzędzia. Podczas pracy dobierz odpowiednie parametry zaznaczania. Do zaznaczania nieregularnych fragmentów użyj narzędzia *Odręczne zaznaczanie obszarów*. Zapisz wynik pracy w pliku w formacie JPG.

### 4.6.4. Zaznaczanie zaawansowane

Do zaznaczania obszarów, które mają mocno skontrastowane krawędzie, można użyć narzędzia *Inteligentne nożyce*. Rozpoznawanie krawędzi i zaznaczanie obszarów jest realizowane na podstawie wartości pikseli (koloru, jasności). Narzędzie automatycznie dopasowuje brzegi zaznaczenia do krawędzi zaznaczanego obiektu.

#### Gradient

Narzędzie *Gradient* dostępne w *Przyborniku* służy do nadawania wybranym obszarom (np. selekcjom, warstwom) wypełnienia tonalnego (jeden kolor przechodzi płynnie w drugi). Kolory przejścia są ustawiane w oknie *Przybornika* jako *Kolor pierwszoplanowy oraz tła*.



**Rysunek 4.31.**

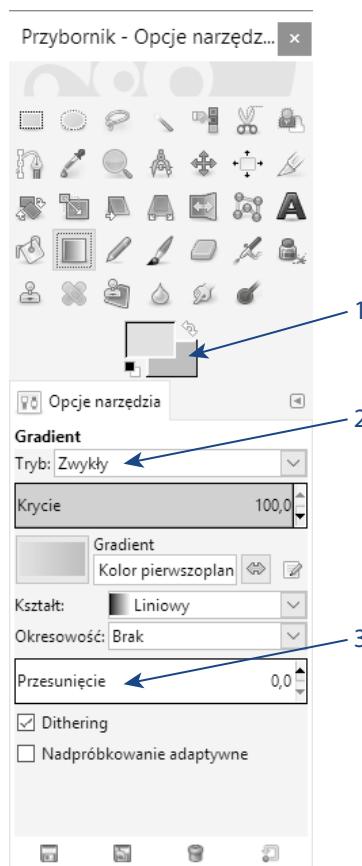
Korekta zdjęcia — usuwanie graffiti

Tworzenie gradientu składa się z kilku etapów (rysunek 4.32):

- 1.** Wyboru kolorów.
- 2.** Wyboru trybu mieszania kolorów.
- 3.** Ustawienia przesunięcia (przesunięcia koloru pierwszoplanowego).
- 4.** Przeciągnięcia myszą po obrazie w wybranym kierunku.

**Rysunek 4.32.**

Ustawienie opcji gradientu



### Ćwiczenie 4.8

Przygotuj zdjęcie z pomnikiem znanej osoby. Ze zdjęcia (rysunek 4.33) wyodrębnij figurę znajdująca się na pomniku. Dla wyodrębnionej figury zmień tło z wykorzystaniem gradientu.

W oknie *Przybornik* należy wybrać narzędzie *Inteligentne nożyce*. W opcjach nożyczek wybieramy *Zmiękczaj krawędzie* i ustawiamy promień *2 px*. Obrysowujemy figurę, klikając po jej krawędzi (nie musi to być zbyt dokładne zaznaczenie, ale jego ostatni punkt musi zostać połączony z pierwszym). Klikamy raz wewnętrz figurę, aby ją zaznaczyć, i kopiuujemy zaznaczoną figurę (*Edycja/Skopij*), a następnie ją wklejamy

(*Edycja/Wklej*). Zauważmy, że w oknie z warstwami (*Warstwy — Pędzle*) pojawiła się warstwa *Oderwane zaznaczenie* (nie jest to zwykła warstwa). W oknie *Warstwy* należy kliknąć ikonę *Nowa warstwa*. Powinna się pojawić miniaturka wklejonej warstwy — figura, a dookoła niej szachownica w dwóch odcieniach szarości (szachownica oznacza obszar przezroczysty). Aby dodać nowe tło dla figury, ponownie klikamy ikonę *Nowa warstwa* i w otwartym oknie wybieramy *Typ wypełnienia warstwy/Białe*. W oknie *Warstwy* nowo powstałą warstwę przesuwamy tak, aby znalazła się między pozostałymi warstwami, i ją zaznaczamy. Wybieramy narzędzie *Wypełnienie kubkiem*, ustawiamy dowolny kolor wypełnienia i klikamy obrazek (rysunek 4.34).

Aby przygotować tło z wykorzystaniem gradientu, utworzymy kolejną warstwę. Wybieramy narzędzie *Gradient* i w ustawieniach *Kolor pierwszoplanowy* oraz *tła* ustawiamy dowolne kolory. Należy teraz przeciągnąć gradient po obrazku (dla zaznaczonej nowo utworzonej warstwy). Po tych działańach w oknie z warstwami powinny się znajdować cztery warstwy. Zmieniając ich położenie, zmieniamy wygląd obrazka (rysunek 4.35).



**Rysunek 4.33.**

Wyodrębnienie ze zdjęcia figury



**Rysunek 4.34.**

Użycie narzędzia Inteligentne nożyce do wyodrębnienia figury



**Rysunek 4.35.**

Gradient jako tło figury

Aby uzyskać efekt przenikania warstw, zmieniamy widoczność warstwy, przesuwając suwakiem w opcji *Krycie* (*Przybornik/Opcje narzędzia*). Zapiszmy wynik pracy w nowym pliku.

## Ćwiczenie 4.9

Przygotuj zdjęcie fragmentu miasta z widoczną przestrzenią umożliwiającą wstawienie pomnika. Wyodrębnioną figurę z pomnika umieść w przestrzeni innego miasta.

Należy rozpocząć od zaznaczenia warstwy z wyodrębnioną w poprzednim ćwiczeniu figurą. Kopiujemy ją (*Edycja/Skopij*), po czym otwieramy plik z fragmentem miasta i wklejamy figurę (*Edycja/Wklej*). W oknie *Warstwy* wklejona figura zostanie opisana jako *Oderwane zaznaczenie*. Wybieramy ikonę *Nowa warstwa*, aby przekształcić ją w warstwę. Wybieramy w *Przyborniku* narzędzie *Przesunięcie*, aby wklejoną figurę umieścić w odpowiednim miejscu (rysunek 4.36).



**Rysunek 4.36.**

Połączenie obrazu z wyodrębnioną figurą

## Szybka maska

Inną metodą zaznaczania jest wykorzystanie narzędzia *Szybka maska*. Umożliwia ono utworzenie zaznaczenia przy użyciu gumki lub ołówka. Może również służyć do poprawiania zaznaczenia wykonanego inną metodą.

Wybór narzędzia *Szybka maska* nastąpi po wybraniu kwadratu zaznaczonego linią przerywaną w lewym dolnym rogu zdjęcia (rysunek 4.37).

**Rysunek 4.37.**

Miejsce przełączania się do Szybkiej maski



Po wybraniu *Szybkiej maski* cały obraz zostanie „przykryty” maską. Fragment obrazu, który powinien zostać zaznaczony, należy „wygumkować”; w tym celu należy wybrać w *Przyborniku* narzędzie *Gumka* i „gumkując”, usunąć maskę z obszaru, który ma być widoczny (rysunek 4.38). Można zmienić ustawienia tego narzędzia, wybierając rodzaj i rozmiar pędzla oraz poziom krycia.

#### 4.6.5. Retusz zdjęcia

W programie GIMP dostępne są w *Przyborniku* narzędzia *Klonowanie (Stempel)* i *Łatka (Plaster)*, za pomocą których można wykonać retusz zdjęć (rysunek 4.39). W ten sposób można poprawiać wygląd starych zdjęć, robić korektę wyglądu twarzy czy umieszczać dodatkowe efekty na obrazie.

Narzędzia *Klonowanie* i *Łatka* pozwalają w łatwy sposób usunąć niepotrzebne elementy ze zdjęć i z obrazów o różnorodnym deseniu. Ich działanie polega na umieszczeniu w miejscu usuwanego detalu skopiowanych z innego miejsca pikseli, które zamaskują element. Narzędzie *Klonowanie* dodatkowo pozwala miejsce defektu zamalować wybranym deseniem. Piksele umieszczone w miejscu defektu przy użyciu narzędzia *Łatka* dopasowują się do najbliższego obszaru — retusz jest mniej widoczny.

Przed przystąpieniem do korekty należy ustawić właściwości wybranego narzędzia (np. kształt pędzla, wielkość, krycie).

#### Klonowanie (Stempel)

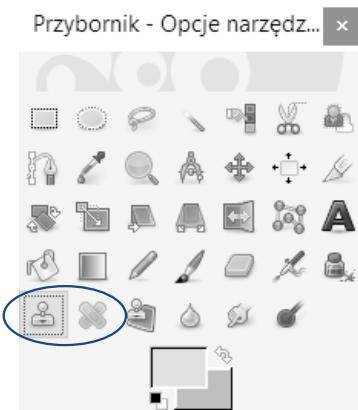
Po wybraniu narzędzia *Klonowanie* w celu pobrania próbki do klonowania należy ustawić pędzel nad obszarem o prawidłowym wyglądzie, wcisnąć na klawiaturze *Ctrl* i kliknąć wybrane miejsce lewym przyciskiem myszy. Po pobraniu próbki należy ustawić kurSOR w miejscu korekty i kliknąć lewym przyciskiem myszy. Czynność tę należy powtarzać aż do usunięcia błędów w obrazie.

W panelu *Opcje narzędzia* można dobrać końcówkę, rozmiar, krycie, tryb mieszania pikseli i inne parametry pędzla (rysunek 4.40).



Rysunek 4.38.

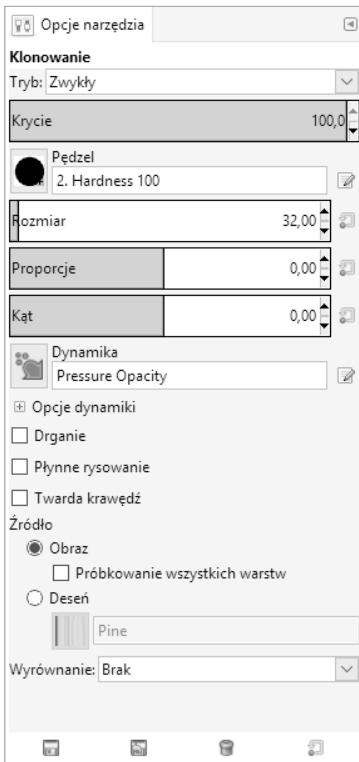
„Wygumkowanie” fragmentu maski natóżonej na obraz



Rysunek 4.39. Narzędzia Klonowanie i Łatka

**Rysunek 4.40.**

Ustawienie opcji  
narzędzia Klonowanie



W opcji *Źródło* w przypadku usuwania defektów najlepiej zaznaczyć *Obraz*. Jeżeli zostanie zaznaczona opcja *Próbkowanie wszystkich warstw*, narzędzie będzie pobierało piksele nie tylko z bieżącej warstwy, ale również z pozostałych. Jeżeli w opcji *Wyrównanie* zostanie wybrana wartość *Brak*, źródło pikseli przy każdym kliknięciu będzie stałe (tam, gdzie kliknęliśmy na początku), lecz gdy zaczniemy malować pędzlem, źródło pikseli zacznie się przemieszczać równolegle z pędzlem. Gdy zostanie wybrana wartość *Wyrównany*, źródło będzie się stale przemieszczać, a gdy wartość *Staty*, źródło pozostanie w tym samym miejscu niezależnie od ruchów pędzla.

**Ćwiczenie 4.10 (retusz zdjęcia)**

Przygotuj stare lub uszkodzone zdjęcie, które wymaga poprawy.

Wybierzmy narzędzie *Klonowanie*. Ustawmy kształt, wielkość i krycie pędzla. Następnie umieścmy pędzel w obszarze, który będzie klonowany. Wciśnijmy na klawiaturze *Ctrl* i kliknijmy lewym przyciskiem myszy miejsce, z którego zostanie pobrana próbka. Po pobraniu próbki należy ustawić kurSOR w miejscu korekty i kliknąć lewym przyciskiem myszy. Powtarzajmy tę czynność aż do skorygowania wyglądu obrazu. Mały krzyżyk określa, skąd jest pobierany fragment obrazka, a czarna strzałka i symbol stempla pokazują miejsce nakładania pobranego fragmentu obrazka. Do korygowania poszczególnych fragmentów obrazka dobierzmy odpowiednią wielkość końcówki pędzla oraz intensywność krycia. Mniejszymi końcówkami pędzla dopracujmy szczegóły.

## Łatka

*Łatka* usuwa niewielkie nieregularności z obrazu. Usuwanie niedoskonałości odbywa się przez przeniesienie obszaru wokół punktu docelowego w miejsce klonowania. Aby zastosować to narzędzie, należy wybrać odpowiedni kształt pędzla, a następnie kliknąć obszar do skopiowania, jednocześnie przytrzymując klawisz *Ctrl*.

Kolejnym krokiem jest puszczenie klawisza *Ctrl*, przeciągnięcie próbki na korygowane miejsce i ponowne kliknięcie. Jeśli defekt jest niewielki, tylko w nieznacznym stopniu odróżniający się od otoczenia, zostanie szybko naprawiony. Aby usunąć większy defekt, należy powtórzyć kliknięcia kilkakrotnie. Niestety istnieje wtedy niebezpieczeństwo rozmazania naprawianego fragmentu.

### Ćwiczenie 4.11

Przygotuj zdjęcie podobne do pokazanego na rysunku 4.41. Korzystając z narzędzia *Łatka*, usuń ze zdjęcia zbędne elementy (z rysunku 4.41 zostaną usunięte ptaki na niebie).



**Rysunek 4.41.** Zastosowanie narzędzia Łatka

W *Przyborniku* wybieramy narzędzie *Łatka* i w *Opcjach narzędzia* wybieramy kształt pędzla — koło rozmyte na krawędziach — oraz ustawiamy rozmiar pędzla na wartość **51,00**. Następnie wciskamy klawisz *Ctrl* i przytrzymując go, klikamy obszar do skopiowania znajdujący się w pobliżu ptaka. Zwalniamy klawisz *Ctrl*, przeciągamy próbkę w miejsce, gdzie znajduje się ptak, i klikamy. Jeżeli ptak został usunięty tylko częściowo, ponownie go klikamy. Postępujemy tak aż do momentu całkowitego jego usunięcia. Podobnie robimy z pozostałymi ptakami.

## 4.6.6. Wstawianie tekstu

Z pomocą dostępnego w *Przyborniku* narzędzia *Tekst* można do obrazu dodawać różnego rodzaju napisy. Po wybraniu ikony *Tekst* zostaną uaktywnione *Opcje narzędzia*,

w których można wybrać rodzaj czcionki, jej rozmiar, kolor i inne atrybuty. Po wpisaniu tekstu utworzona zostanie nowa warstwa, którą można modyfikować.

### Ćwiczenie 4.12

Przygotuj prosty druk reklamowy dla biura podróży GLOB.

Utwórzmy w programie GIMP nowy obraz, o rozmiarach 800×600 pikseli. Jako tło obrazu utworzymy dowolny gradient lub wstawimy gotowe zdjęcie. Z *Przybornika* wybieramy narzędzie *Tekst*. Ustawiamy czcionkę, rozmiar oraz kolor tekstu. Klikamy na środku obrazka i wpisujemy tekst: *Zapraszamy w podróż dookoła świata*. Zostanie utworzona nowa warstwa z tekstem, którą można modyfikować. Tekst możemy przemieszczać po obrazku tak samo jak inne warstwy — po wybraniu narzędzia *Przesunięcie*.

Wstawiamy kolejny tekst: *Podróże kształca* oraz tekst: *Biuro podróży GLOB*. Dla wstawianych tekstów wybieramy rozmiar, kolor i krój czcionki.

Dla pierwszego tekstu ustawiamy cień. Z górnego menu wybieramy *Filtr/Światło i cień/Rzucanie cienia....* W otwartym oknie opcję *Promień rozmycia* ustawiamy na 5 i zatwierdzamy to ustalenie. Cień powstanie jako nowa warstwa, którą można modyfikować.

Drugi tekst będzie tekstem wypukłym umieszczonym pionowo. Aby tekst był wypukły, wybieramy z menu *Filtr/Dekoracja/Dodaj frazę....* Z kolei aby tekst został umieszczony pionowo, wybieramy z menu *Warstwa/Przekształcenie/Obróć o 90° w prawo*. Duplikujemy utworzony tekst i przesuwamy go do prawej krawędzi. W tym celu wybieramy z menu *Warstwa/Powiel warstwę*. Powstałą warstwę obracamy o 180°.

Utworzony obraz zapiszmy jako plik w formacie JPG.

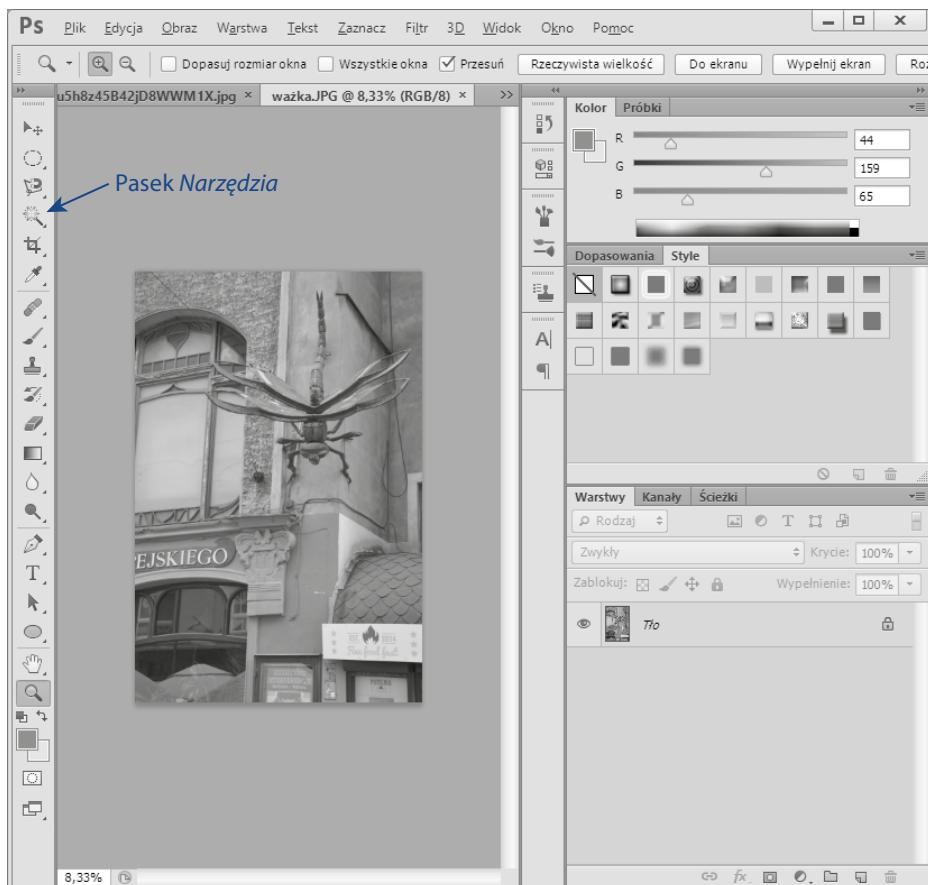
## 4.7. Program Adobe Photoshop

Narzędziem, które spełnia wymogi projektowania zaawansowanej grafiki bitmapowej, jest program Photoshop firmy Adobe.

Podstawowe okno programu Adobe Photoshop zawiera obszar roboczy, w którym wykonujemy obróbkę obrazu. U góry znajduje się pasek *Menu*, a poniżej umieszczony jest pasek *Właściwości*, którego zawartość zmienia się w zależności od wybranego narzędzia w pasku narzędzi. Pasek *Narzędzia* standardowo znajduje się z lewej strony głównego okna. Z prawej strony zaś widnieje zestaw paneli służących do edytowania grafiki (rysunek 4.42).

### 4.7.1. Nawigacja w obszarze roboczym

Aby usprawnić pracę z aplikacją, można wykorzystać narzędzia przeznaczone do nawigowania w obszarze roboczym. Narzędzia te są dostępne na pasku *Narzędzia*. Dodatkowo do większości z nich zostały zdefiniowane skróty klawiszowe. Pod ikoną większości narzędzi kryją się inne narzędzia należące do tej samej grupy.



**Rysunek 4.42.** Okno programu Photoshop

## Lupa

*Lupa* to narzędzie służące do powiększania obrazu, wybierane po kliknięciu na pasku *Narzędzia* ikony lupy. Za pomocą tego narzędzia można dowolnie zmniejszać lub powiększać obszar roboczy projektu. Można to także zrobić, wykorzystując poniższe skróty klawiszowe:

- *Ctrl+-*, który pomniejsza obraz;
- *Ctrl++*, który powiększa obraz;
- *Ctrl+0*, który dostosowuje wielkość obrazu do ekranu.

## Rączka

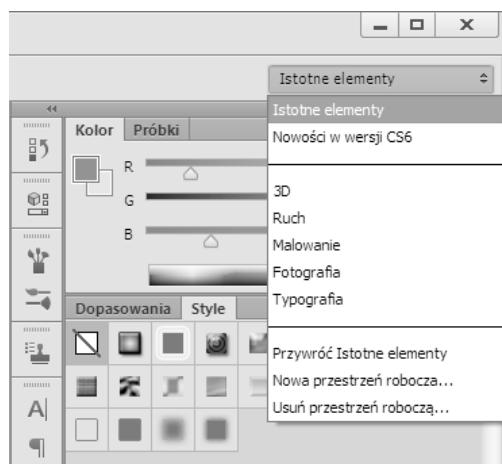
Narzędzie *Rączka* służy do przesuwania obrazu po ekranie. Dostępne jest na pasku narzędziowym po kliknięciu ikony z rączką. Można je również wywoływać po naciśnięciu i przytrzymaniu klawisza spacji na klawiaturze.

## Cofanie

Użycie skrótu [\*\*\*Ctrl+Z\*\*\*](#) cofa nasze działania o jeden krok, a jego ponowne użycie przywraca cofniętą operację. Do cofania o kilka kroków służy skrót [\*\*\*Ctrl+Alt+Z\*\*\*](#).

## Restart widoku

Jeżeli podczas edytowania zmienimy układ okien w programie, możemy zrestartować widok. W prawym górnym rogu okna programu znajduje się przycisk [\*\*\*Istotne elementy\*\*\*. Kliknięcie tego przycisku spowoduje rozwinięcie okna, w którym należy wybrać opcję \[\\*\\*\\*Przywrócić istotne elementy\\*\\*\\* \\(rysunek 4.43\\). Po tej operacji wszystkie standardowo dostępne okna powinny wrócić na swoje miejsca.\]\(#\)](#)



Rysunek 4.43. Restart widoku programu

Możemy również zresetować wszystkie ustawienia, które zdefiniowaliśmy podczas pracy z programem. W tym celu trzeba go zamknąć i uruchomić ponownie, trzymając wcisnięte klawisze [\*\*\*Ctrl+Alt+Shift\*\*\*.](#)

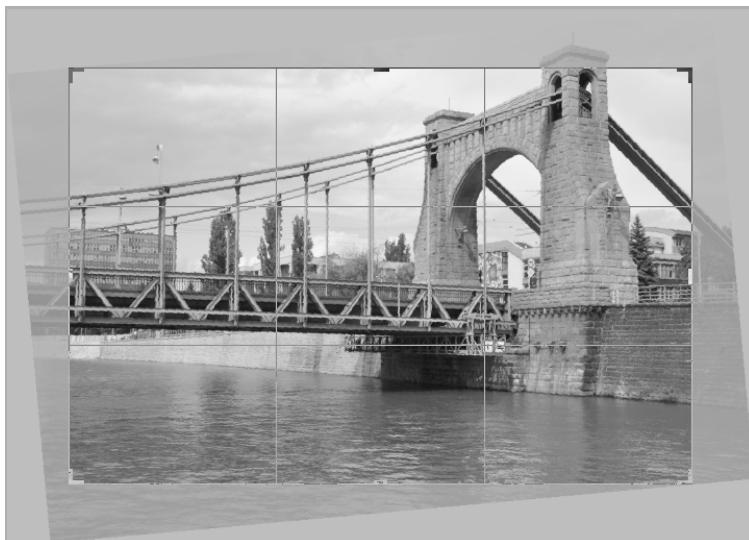
## 4.7.2. Modyfikowanie zdjęć

W programie Photoshop znajduje się wiele narzędzi, które mogą zostać wykorzystane do poprawienia wyglądu zdjęcia zapisanego w pliku. Modyfikowanie wyglądu zdjęcia może polegać na ustaleniu odpowiedniej głębi jego kolorów, wykadrowaniu lub skorygowaniu jego zawartości.

### Kadrowanie i korekcja zdjęć

Kadrowanie jest jedną z technik najczęściej stosowanych do poprawienia jakości zdjęć. Pozwala na usunięcie ze zdjęcia niepotrzebnych fragmentów oraz na jego „wyprostowanie”.

Po wybraniu narzędzia do kadrowania (ikona *Kadrowanie* na pasku *Narzędzia*) należy zaznaczyć obszar zdjęcia, którego będzie dotyczyło kadrowanie. Pojawi się siatka ułatwiająca kadrowanie (rysunek 4.44). Przy jej użyciu można dowolnie przycinać obraz, ale można również, chwytając za punkt zaczepienia znajdujący się powyżej każdego rogu obrazu, obrócić go, tak aby linie poziome były idealnie poziome, a linie pionowe były równo ustawione w pionie. Dotyczy to szczególnie obrazów, w których te linie są wyraźnie widoczne (np. schodów, kolumn).



**Rysunek 4.44.** Kadrowanie zdjęcia

## Usuwanie zbędnych elementów zdjęcia

Innym sposobem modyfikowania zdjęcia jest usunięcie zbędnych obiektów, postaci lub napisów. Mechanizm ich usuwania polega na zastąpieniu niepożądanego elementu pikselami zdjęcia pobranymi z okolic usuwanego elementu. Można do tego wykorzystać narzędzie *Stempel*, które służy do klonowania wybranego obszaru obrazu w inne miejsce. Narzędzie to wykorzystuje końcówkę pędzla, co oznacza, że przed jego użyciem musimy zdefiniować grubość oraz kształt pędzla.

### Przykład 4.1

Aby użyć narzędzia *Stempel*, należy zdefiniować obszar do klonowania. Trzeba w tym celu, przytrzymując wciśnięty klawisz *Alt*, w odpowiednim miejscu kliknąć lewym przyciskiem myszy. Następuje wówczas zdefiniowanie obszaru do klonowania. Gdy przeniesiemy kurSOR w miejsce wstawienia sklonowanego obszaru i klikniemy lewym przyciskiem myszy, zostanie zdefiniowana odległość między miejscem pobrania i miejscem wklejenia. Następnie, malując po wybranym obszarze, spowodujemy klonowanie wybranego fragmentu obrazu. Efekt użycia narzędzia *Stempel* został pokazany na rysunkach 4.45 i 4.46.



**Rysunek 4.45.** Zdjęcie przed zastosowaniem narzędzia Stempel



**Rysunek 4.46.** Efekt zastosowania narzędzia Stempel

W pokazanym przykładzie po zastosowaniu narzędzia *Stempel* została usunięta ze zdjęcia jedna z kaczek.

Podobny efekt usunięcia zbędnych elementów ze zdjęcia uzyskamy, stosując narzędzie *Pędzel korygujący*. Również to narzędzie przenosi zapamiętany fragment w inne miejsce. Różnica polega na tym, że wklejane piksele są dopasowywane pod względem jasności i koloru do otoczenia, w którym się znajdą. Wadą tego narzędzia jest to, że nie dopasowuje ono prawidłowo wyglądu pikseli na krawędziach mocno skontrastowanych.

## Czerwone oczy

Jedną z częściej wykonywanych czynności podczas korygowania błędów na zdjęciu jest usuwanie efektu czerwonych oczu. Aby wyeliminować tę wadę, należy wybrać narzędzie *Czerwone oczy*. Wskaźnik myszy przybierze kształt oka. Wystarczy kliknąć tym wskaźnikiem w miejscu wystąpienia efektu czerwonych oczu, aby go usunąć.

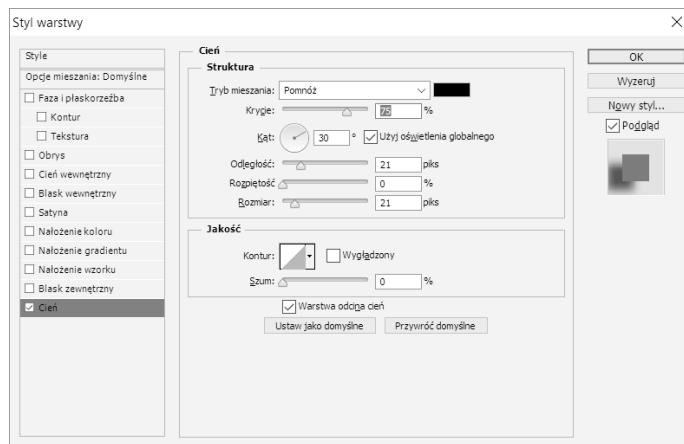
## Ćwiczenie 4.13

Przygotuj zdjęcia, które zamierzasz wykorzystać w swoim projekcie. Popraw ich wygląd, wykadruj wybrane fragmenty i usuń zbędne elementy. Zastosuj narzędzia *Kadrowanie*, *Stempel*, *Pędzel korygujący* i *Punktowy pędzel korygujący*. Przeanalizuj, w jakich sytuacjach należy stosować każde z wymienionych narzędzi.

## 4.7.3. Warstwy

W tworzeniu zaawansowanej grafiki na potrzeby strony internetowej niezbędne są narzędzia, które pozwolą na niezależne edytowanie różnych jej elementów. Takim narzędziem dostępnym w zaawansowanych programach graficznych są warstwy.

W programie Photoshop dla każdej warstwy można zdefiniować styl, który w istotny sposób wpływa na wygląd obrazu. W dolnej części panelu *Warstwy* po wybraniu ikony *fx (dodaj styl warstwy)* pojawi się lista dostępnych opcji, a po wybraniu jednej z nich otwarte zostanie okienko, w którym można definiować styl warstwy (rysunek 4.47).

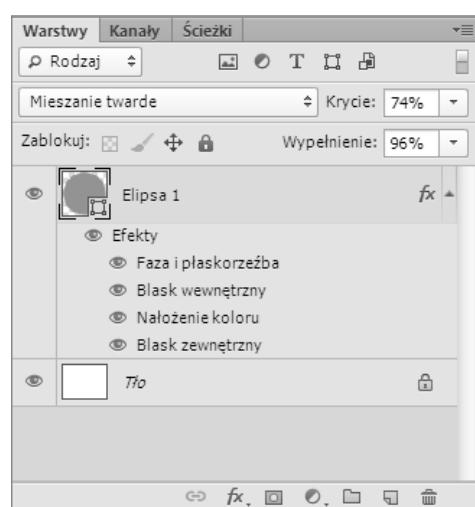


**Rysunek 4.47.** Okno Styl warstwy

Nawet prosty obraz po zdefiniowaniu dostępnych w tym oknie parametrów warstwy może się znacznie zmienić. Informacja o efektach dodanych do warstwy jest widoczna w panelu *Warstwy* (rysunek 4.48).

**Rysunek 4.48.**

Efekty ustawione dla warstwy



Podczas pracy na wielu warstwach z wykorzystaniem różnych narzędzi przydatne są skróty klawiszowe:

- ***Ctrl+kliknięcie w obszarze obiektu (po wybraniu narzędzia *Przesunięcie*)*** — wybranie warstwy, na której znajduje się obiekt;
- ***Ctrl+T*** — zaznaczenie elementu warstwy w celu jego przekształcenia.

## Zmiana tła zdjęcia

Aby utworzyć baner reklamowy, ikonę lub tło, czasami wykorzystujemy tylko niektóre obiekty znajdujące się na zdjęciu. Jeżeli tło zdjęcia jest jednolite, a element, który należy wyodrębnić, ma wyraźnie zaznaczone krawędzie, można do zaznaczania zastosować narzędzie **Różdżka**. Służy ono do zaznaczania obszarów o podobnym kolorze. Gdy po wybraniu na pasku **Narzędzia** ikony **Różdżka** klikniemy w obszarze o podobnym kolorze, spowodujemy jego zaznaczenie. Dodanie kolejnych obszarów o zbliżonej barwie uzyskamy, zaznaczając je przy wciśniętym klawiszem **Shift**.

Jednym ze sposobów projektowania grafiki jest tworzenie kompozycji z gotowych elementów graficznych. Efektem może być graficzny element strony, baner reklamowy lub logotyp. Połączenie i dobór zdjęć oraz innych elementów graficznych powinny polegać na takim ich dopasowaniu, aby powstała grafika stanowiła spójną całość.

### Ćwiczenie 4.14

Z przygotowanego wcześniej zdjęcia usuń niebo za pomocą narzędzia **Różdżka** i w jego miejsce wstaw inne niebo, pobrane z pliku ze zdjęciem (rysunek 4.49).



**Rysunek 4.49.** Zaznaczenie obszaru w kolorze nieba

Wybieramy narzędzie **Różdżka** i z wciśniętym klawiszem **Shift** zaznaczamy kolejne obszary do usunięcia. W wyniku użycia narzędzia **Różdżka** zaznaczony został obszar nieba. Do dalszej obróbki zdjęcia potrzebny jest nam obszar odwrotny do zaznaczonego, więc wybieramy z menu opcję **Zaznacz/Odwrotność** i kopiuujemy zaznaczony obszar do

nowej warstwy (*Warstwa/Nowa/Warstwa przez kopiowanie*). Powstałą warstwę nazwujemy np. *Góra*. W panelu z warstwami wyłączamy widoczność warstwy tła. Otwieramy w programie Photoshop zdjęcie, które będzie nowym tłem dla tworzonego obrazu. Wystarczy je przeciągnąć w obszar tworzonego obrazu, aby zostało skopiowane w kolejnej warstwie. Nazwiemy ją *Niebo*. Rozmiar i położenie zdjęcia znajdującego się na warstwie *Niebo* możemy dostosować do parametrów obrazu z warstwy *Góra*. Wybieramy do edycji warstwę *Niebo* i wciskamy klawisze *Ctrl+T*. Łapiemy myszą za zaznaczony obszar i przesuwamy w odpowiednie miejsce. Podobnie łapiemy myszą za krawędzie obszaru i zmieniamy jego rozmiar. Zatwierdzamy zmiany przyciskiem *Enter* lub klikając na pasku właściwości ikonę *Zatwierdź przekształcenie* (ostatnia ikona z prawej strony). W panelu z warstwami przesuwamy warstwę *Niebo* za warstwę *Góra* (rysunek 4.50).



**Rysunek 4.50.** Obraz zmodyfikowany

Inną metodą zmiany tła może być utworzenie własnego tła za pomocą narzędzi dostępnych w programie.

### Ćwiczenie 4.15

Z przygotowanego wcześniej zdjęcia wyodrębnij jego tło.

Wykorzystując narzędzie *Różdżka*, zaznaczamy elementy tła, które należy usunąć. Wybieramy z menu *Zaznacz/Odwrotność*, aby zaznaczone zostały elementy potrzebne w dalszej obróbce grafiki. Następnie wybieramy z menu *Warstwa/Nowa/Warstwa przez kopiowanie*. Utworzona zostanie warstwa z zaznaczonych elementów. Nazwijmy ją *Grafika*.

Teraz do otrzymanego obrazka dodamy własne tło, np. błękitne niebo z chmurami.

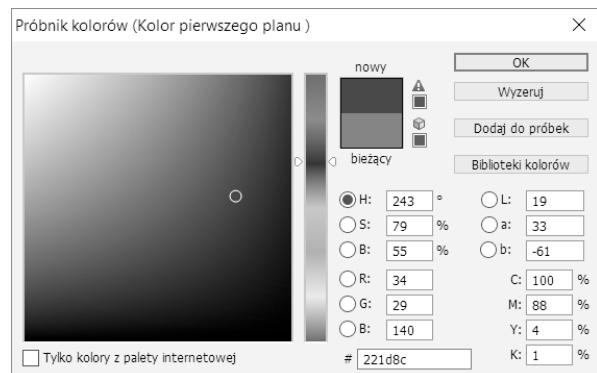
Tworzymy nową warstwę (w menu *Warstwa/Nowa/Warstwa*), o nazwie *Chmury*. Na niej spróbujemy wygenerować tło. Ustawiamy kolor pierwszoplanowy na niebieski i kolor tła również na niebieski, ale dużo jaśniejszy. Ustawienie tych parametrów jest realizowane na pasku *Narzędzia*, w obszarze *Ustaw kolor pierwszego planu* (rysunek 4.51).

Po kliknięciu ikony z górnym kwadratem otwarte zostanie okno z paletą kolorów (rysunek 4.52), w której trzeba wybrać odpowiedni kolor, następnie należy kliknąć dolny kwadrat i ponownie w otwartym oknie wybrać kolor tła.



**Rysunek 4.51.**

Ustawianie koloru pierwszoplanowego i koloru tła



**Rysunek 4.52.**

Okno palety kolorów

Z menu wybieramy *Filtr/Renderowanie/Chmury*. Utworzona warstwa zostanie wypełniona „sztucznymi” chmurami.

Kolejnym działaniem będzie uporządkowanie utworzonych warstw. Warstwę o nazwie *Tło* można usunąć (chwytamy za warstwę i przeciągamy ją do kosza — ikona w prawym dolnym rogu okna *Warstwy*). Pozostają dwie utworzone przez nas warstwy. Warstwę *Chmury* przesuwamy na dół, pod warstwę *Grafika*. Można jeszcze poprawić wygląd nieba. W tym celu wybieramy warstwę *Chmury* i zaznaczamy ją, stosując kombinację klawiszy *Ctrl+T*. Trzymając na klawiaturze wcisknięty klawisz *Ctrl*, łapiemy za górne rogi i rozciągamy niebo, aby powstał trapez (rysunek 4.53). Z perspektywy obserwatora chmury powinny się łączyć z horyzontem. Można zmieniać kształt trapezu, aby uzyskać jak najlepszy rezultat. Ustawienia zatwierdzamy, naciskając *Enter*.

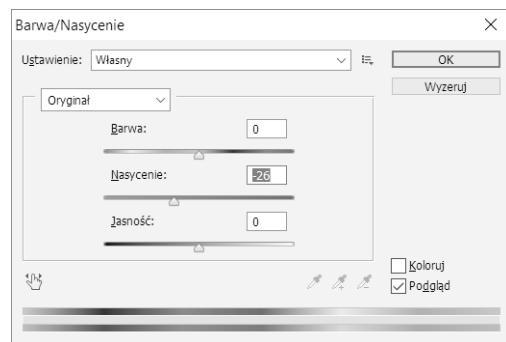
**Rysunek 4.53.**

Manipulowanie wyglądem tła



Można jeszcze bardziej poprawić wygląd chmur — przez zmianę nasycenia i jasności. Z menu wybieramy *Obraz/Dopasowania/Poziomy* lub naciskamy na klawiaturze *Ctrl+L*. W otwartym oknie w celu rozjaśnienia obrazu przesuwamy punkt szarości (szary trójkąt na środku wykresu) w lewo. Można jeszcze zmniejszyć nasycenie barwy, wybierając z menu *Obraz/Dopasowania/Barwa/Nasycenie* (rysunek 4.54).

Stosując podobne metody, można tworzyć tło dla dowolnej kompozycji, niezależnie od tego, czy to będzie tło dla zdjęcia (jak w podanym przykładzie), baner reklamowy, tło tworzonej przez nas grafiki statycznej, czy też tło strony internetowej.



**Rysunek 4.54.**

Zmiana nasycenia barwy

#### 4.7.4. Umieszczanie tekstu

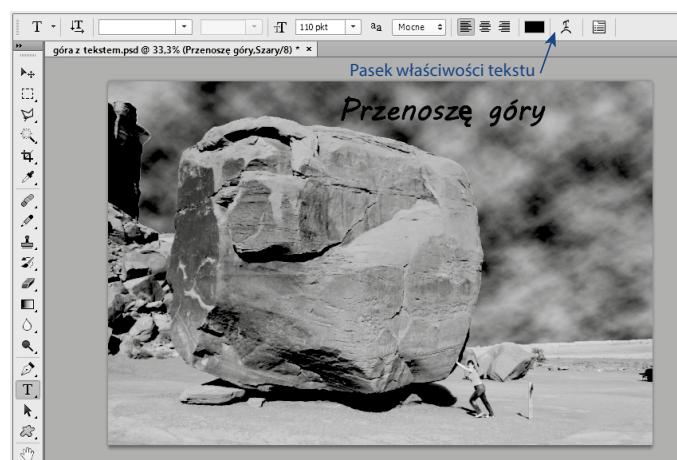
Do tworzonej grafiki można na każdym etapie pracy dodać dowolny tekst.

Aby do zaprojektowanej tapety dodać napis, należy wybrać narzędzie *Tekst*. W pasku *Właściwości* w górnej części okna programu zostaną wyświetcone właściwości tworzonego napisu (rysunek 4.55).

Przed utworzeniem napisu można ustalić: rodzinę czcionek, styl czcionki, rozmiar, metodę wygładzania, wyrównanie oraz kolor tekstu. Po kliknięciu obszaru tworzenia grafiki automatycznie powstanie nowa warstwa, na której będzie umieszczony wprowadzany tekst. Wpisany tekst należy zatwierdzić, klikając na pasku *Właściwości* ikonę z jego prawej strony. Zatwierdzmy wszystkie dotychczasowe zmiany. Tekst zapisany w oddzielnej warstwie może być modyfikowany tak jak każdy element graficzny umieszczony na warstwie.

**Rysunek 4.55.**

Właściwości tekstu



## 4.7.5. Cięcie i wstawianie tła na stronę internetową

Ze względu na czas ładowania strony projektanci stron internetowych, wykorzystując pliki graficzne, zwracają uwagę na to, aby były one jak najmniejsze. Jedną z metod zmniejszenia wielkości pliku jest wybieranie plików małych, specjalnie utworzonych na potrzeby tła. Pliki takie zawierają grafikę o niewielkich rozmiarach, najczęściej o szerokości 20 px i wysokości zdefiniowanej według potrzeb, np. 600 px. Jeżeli taki plik zostanie powtórzony, utworzy ładnie wyglądającą grafikę tła.

Do powtarzania stosowane są formuły `background-repeat: repeat-x` albo `background-repeat: repeat-y`. Wtedy plik wypełniający tło będzie powtarzał się w pionie albo w poziomie (zależnie od wybranego rozwiązania):

```
body {
    background-image: url("tlo.jpg");
    background-repeat: repeat-x;
}
```

W takich plikach grafika najczęściej opiera się na tworzeniu gradientów. Powstają obrazy, w których zmiany następują w obrębie jednego koloru i dotyczą tylko jego odcieni; czasami zmiany dotyczą przejść między kolorami.

Gdy projektujemy tło na gradientie, należy utworzyć grafikę o odpowiedniej długości (zależnej od wysokości strony), a następnie przyciąć ją do szerokości 20 px, aby tło ładowało się jak najkrócej. Kolejnym krokiem jest umieszczenie tła na stronie z użyciem polecenia `repeat`. Jeżeli ponadto określmy kolor strony (`background-color`), poza obszarem grafiki dana barwa wypełni resztę strony, tworząc przejście między zaprojektowanym tłem a pozostałą częścią strony. W takim przypadku często jako `background-color` wybiera się odcień gradientu, który będzie siedział nad resztą strony.

### Ćwiczenie 4.16

W programie Photoshop utwórz nowy dokument, o rozmiarach 960×620 px. Wykonaj tło strony internetowej.

Wybieramy narzędzie **Gradient** i w pasku właściwości klikamy opcję **Edytor gradientów** (rysunek 4.56).

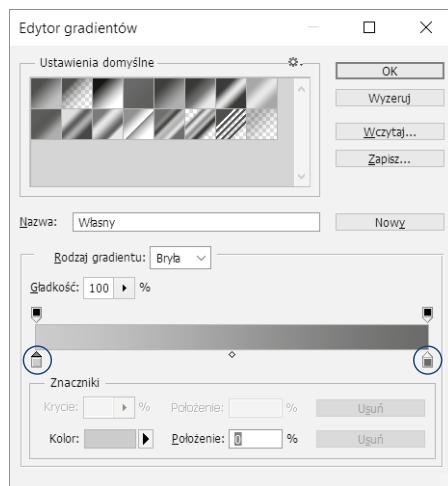


**Rysunek 4.56.** Właściwości gradientu

Zostanie otwarte okno edytowania gradientu, w którym można ustawić opcje koloru początkowego i koloru końcowego (rysunek 4.57).

**Rysunek 4.57.**

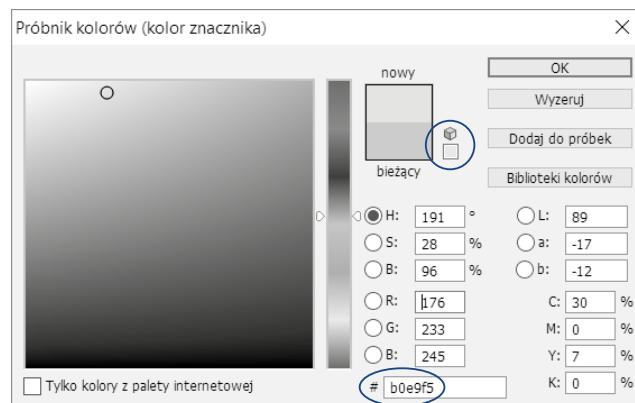
Edytowanie gradientu



W otwartym oknie klikamy dwukrotnie lewy znacznik koloru. W kolejnym oknie (rysunek 4.58) wybieramy odpowiedni kolor z palety. Możemy również wybrać kolor, podając jego wartość *hex* (np. #a5c3ca) lub wartość *RGB*. Zatwierdzamy wybrany kolor i klikamy prawy znacznik koloru. Ustawiamy drugi kolor gradientu (np. #384b66). Zapamiętujemy ustalone kolory.

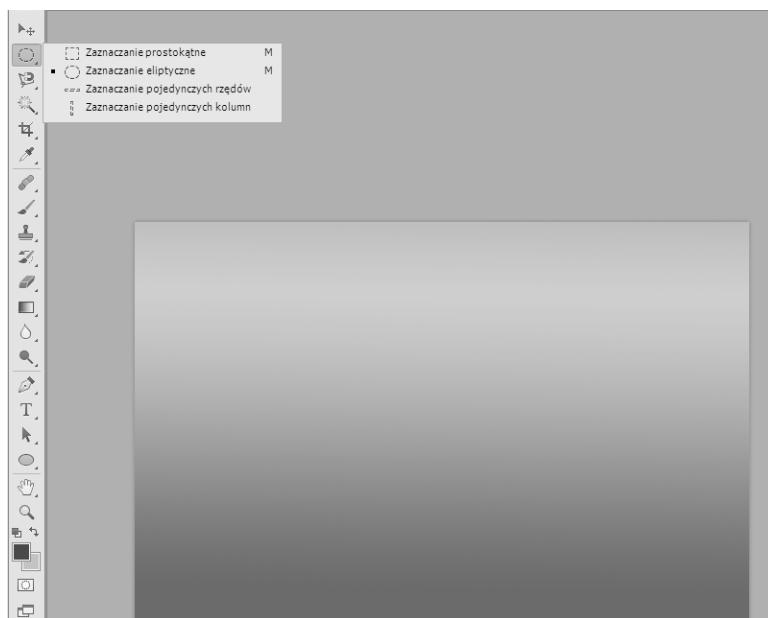
**Rysunek 4.58.**

Wybór koloru



Może się okazać, że wybrany kolor nie jest kolorem palety internetowej (w oknie koloru znacznika pojawi się ostrzeżenie). Skorygujemy to, klikając kwadrat koloru przy przycisku *Dodaj do próbek*. Zostanie automatycznie wybrany bezpieczny kolor palety internetowej. Po zatwierdzeniu kolorów przeciągamy myszką od górnej do dolnej krawędzi tworzonego obrazu. Powstanie obraz oparty na gradiencie.

Na pasku *Narzędzia* wybieramy narzędzie *Zaznaczanie pojedynczych kolumn* (rysunek 4.59), po czym zaznaczamy kolumnę i ją wycinamy. Następnie wybieramy z menu *Obraz/Kadruj*. Zapisujemy plik w formacie PNG.



**Rysunek 4.59.** Narzędzie do wycinania pojedynczych kolumn

Tak utworzony element graficzny może zostać wykorzystany do tworzenia tła strony internetowej.

### Przykład 4.2

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
body {
    background-image: url("tło.png");
    background-repeat: repeat-x;
}
</style>
</head>
<body>
</body>
</html>
```

Aby poza obszarem grafiki barwa jej dolnej części wypełniła resztę strony, należy w poleceniu `background-color` zdefiniować ten odcień gradientu. W tworzonej grafice był to kolor #384b66. Uzupełniony skrypt wygląda teraz tak:

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
body {
    background-image: url("tłol1.png");
    background-repeat: repeat-x;
    background-color: #384b66
}
</style>
</head>
<body>
</body>
</html>
```

### Zadanie 4.1

Zaprojektuj grafikę, która będzie tłem strony internetowej. Wytnij z niej pasek o niewielkiej szerokości i dowolnej wysokości. Napisz kod HTML wykorzystujący powstałą grafikę jako tło strony internetowej. Sprawdź jego działanie.

### Zadanie 4.2

Zaprojektuj kolejną grafikę, która będzie tłem strony internetowej. Wytnij z niej pasek o niewielkiej wysokości i dowolnej szerokości. Napisz skrypt wykorzystujący powstałą grafikę jako tło strony internetowej. Sprawdź jego działanie.

## 4.7.6. Tworzenie ikon

Kolejnym elementem graficznym, który może pojawić się na stronie internetowej, jest ikona. W internecie jest wiele stron, które oferują za darmo lub za niewielką opłatą zbiory ikon przeznaczonych do różnych celów. Jeżeli planujemy sami wykonać tego typu grafikę, możemy wykorzystać narzędzia do rysowania oferowane przez program Photoshop.

## Ćwiczenie 4.17

Utwórz ikony pokazane na rysunku 4.60.



**Rysunek 4.60.** Ikony na stronę internetową

Otwieramy nowy dokument. Wybieramy narzędzie *Elipsa* i trzymając wciśnięty klawisz *Shift*, rysujemy koło. Ustawiamy kolor koła — dwukrotnie klikamy w panelu *Warstwa* w powstałą warstwę i w otwartym oknie wybieramy opcję *Nałożenie koloru*, a następnie ustalamy kolor, np. szary.

Aby wyciąć w kole strzałkę, wykorzystamy narzędzie *Kształt własny*, dostępne na pasku *Narzędzia*. Po jego wybraniu na pasku *Właściwości* wyświetlona zostanie opcja *Kształt* z listą gotowych kształtów (rysunek 4.61).

**Rysunek 4.61.**

Gotowe kształty dostępne na pasku Właściwości i opcje łączenia kształtów



Kontynuując pracę, wybieramy narzędzie *Kształt własny*; rozwijamy listę kształtów i klikając ikonę strzałki, dodajemy grupę *Strzałki* i rysujemy odpowiedni kształt.

Podczas rysowania strzałki w razie potrzeby można przytrzymać klawisz spacji, aby przesunąć rysowany kształt w odpowiednie miejsce.

### Zadanie 4.3

Narysuj samodzielnie drugą ikonę z rysunku 4.60, stosując poznane metody pracy.

### Zadanie 4.4

Wykonaj ikony, które wykorzystasz na stronie internetowej. Zastanów się, jakie ikony mogłyby się pojawić na Twojej stronie, i utwórz je.

## 4.7.7. Tworzenie przycisków nawigacyjnych

Nawigacja między stronami witryny internetowej jest jedną z podstawowych funkcjonalności, jakie powinny się znaleźć na stronie.

## Ćwiczenie 4.18

Utwórz prostokątny przycisk przeznaczony na stronę internetową.

Tworzymy nowy dokument. Wybieramy narzędzie *Zaokrąglony prostokąt* i na pasku *Właściwości* ustawiamy zaokrąglenie rogów (*Promień*) oraz kolor wypełnienia (*Wypełnienie*). Po narysowaniu prostokąta powstanie nowa warstwa. Można do niej dodać zestaw efektów (ikona *fx* w panelu *Warstwy*), które zmienią kształt przycisku.

Inną metodą określania wyglądu przycisku jest zastosowanie gotowego stylu.

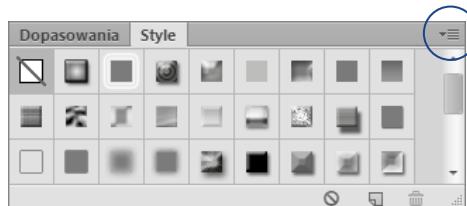
## Ćwiczenie 4.19

Po narysowaniu przycisku nadaj mu wygląd, korzystając z gotowego stylu.

W tym celu w panelu *Style* (rysunek 4.62) klikamy strzałkę w prawym górnym rogu.

**Rysunek 4.62.**

Panel Style



Z listy wybieramy opcję *Przyciski* i klikamy *Dodaj*. Wybieramy jeden z dostępnych stylów przycisków (rysunek 4.63). Dodajemy napis — wybieramy narzędzie *Tekst* i ustawiamy dla niego na pasku *Właściwości* rodzaj, wielkość i kolor czcionki.

**Rysunek 4.63.**

Projektowanie  
przycisku



Samodzielne tworzenie przycisków menu może dać lepsze efekty. Należy tylko poeksperymentować ze stylami warstwy.

## Ćwiczenie 4.20

Utwórz przycisk widoczny na rysunku 4.64.

Wykorzystamy do tego narzędzie *Zaokrąglony prostokąt*. Wygląd przycisku poprawimy, ustawiając odpowiednio parametry w opcji *Styl warstwy*.

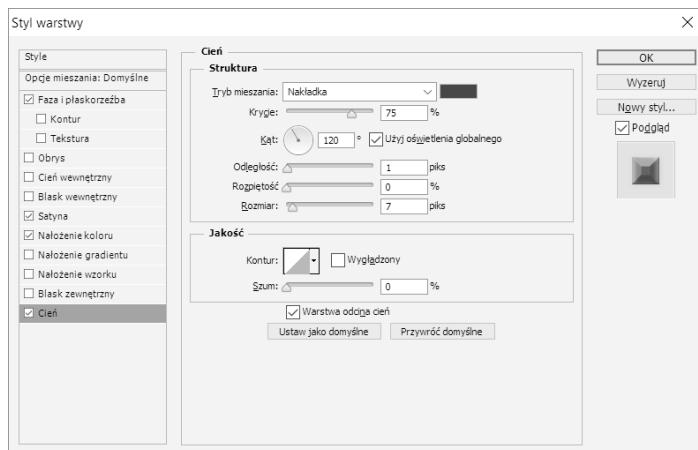
**Rysunek 4.64.**

Style warstwy wpływają  
na wygląd przycisku



Tworzymy nowy dokument, o wymiarach 200×120 px. Wypełniamy obszar tła kolorem. W tym celu z warstwy tła usuwamy kłódkę. Wybieramy *Styl warstwy* (ikona *fx* w panelu *Warstwy*) i na liście — opcję *Nałożenie koloru*. Następnie klikamy ikonę z kolorem

i ustawiamy odpowiedni kolor (np. #252a30). Wybieramy narzędzie **Zaokrąglony prostokąt**, w pasku **Właściwości** ustawiamy **Promień zaokrąglenia** (np. 20 px) i rysujemy prostokąt o odpowiednich rozmiarach. Teraz musimy nadać narysowanej figurze styl. Klikamy dwukrotnie warstwę z prostokątem i w otwartym oknie **Styl warstwy** ustawiamy wybrane parametry (rysunek 4.65).



**Rysunek 4.65.** Definiowanie stylu przycisku

**Cień** nada przyciskowi głębię. Ustawiamy jego parametry, np. tak jak na rysunku 4.65.

**Satyna** podkreśli kształty, np.: **Tryb mieszania** zwykły, **Krycie** 100%, **Kąt** 9°, **Odległość** 170 px, **Rozmiar** 60 px.

W przypadku opcji **Nałożenie koloru** można przetestować różne kolory. Wystarczy dodać napis i zapisać w pliku utworzoną grafikę.

### Ćwiczenie 4.21

Wykorzystując poznane narzędzia programu Photoshop, zaprojektuj i utwórz menu podobne do pokazanego na rysunku 4.66.

**Rysunek 4.66.**

Menu



# 4.8. Program Inkscape

Inkscape jest bezpłatnym programem do tworzenia grafiki wektorowej. Został oparty na formacie SVG, opisującym statyczną i animowaną grafikę dwuwymiarową. Wykorzystywany jest do tworzenia symboli, ikon, znaków towarowych i logotypów. Nie obsługuje modelu barw CMYK. Za jego pomocą można tworzyć różnego rodzaju kształty, linie i strzałki. Pozwala na modelowanie obiektów za pomocą krzywych Béziera.

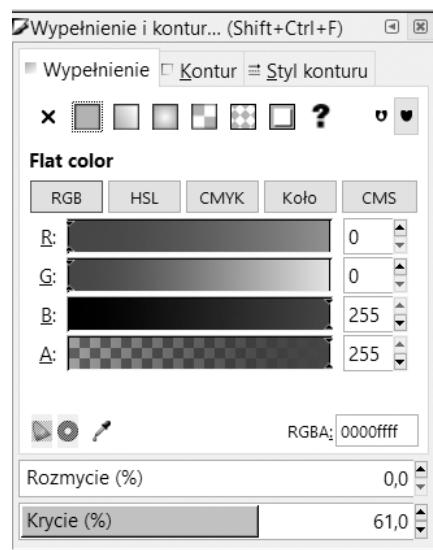
## 4.8.1. Wprowadzenie

Po uruchomieniu programu zostanie otwarte okno (rysunek 4.67) zawierające z lewej strony *Przybornik* z narzędziami do rysowania i edycji. W górnej części okna znajduje się pasek menu, a poniżej pasek poleceń zawierający przyciski będące odnośnikami do najczęściej wykonywanych w programie poleceń. Jeszcze niżej znajduje się kolejny pasek — jest to pasek właściwości, zawierający właściwości obecnie wybranego narzędzia. W dolnej części okna znajduje się pasek stanu, na którym wyświetlane są informacje związane z obecnie wykonywanymi działaniami. Powyżej niego znajduje się pasek kolorów, zawierający paletę kolorów, z której można wybierać kolory i przeciągać na obiekty, aby szybko je pokolorować. Przeciagnięcie koloru z wcisniętym przyciskiem *Shift* zmieni kolor obrysu. Wybranie ikony, która znajduje się na pozycji pierwszej od lewej, spowoduje usunięcie koloru z obiektu lub obrysu (z klawiszem *Shift*).

Wybrany kolor wypełnienia i obrysu można obserwować z lewej strony na pasku stanu. Kliknięcie tej pozycji spowoduje wyświetlenie panelu *Wypełnienie i kontur*, w którym za pomocą różnych modeli barw można wybierać kolor wypełnienia i obrysu tworzonej figury (rysunek 4.67). Oprócz kolorów można tu ustawić również kanał alfa. Zakładki *Wypełnienie* i *Kontur* służą do określenia koloru wypełnienia i konturu, w zakładce *Styl konturu* można wybrać styl i grubość linii, którą zostanie obrysowana figura.

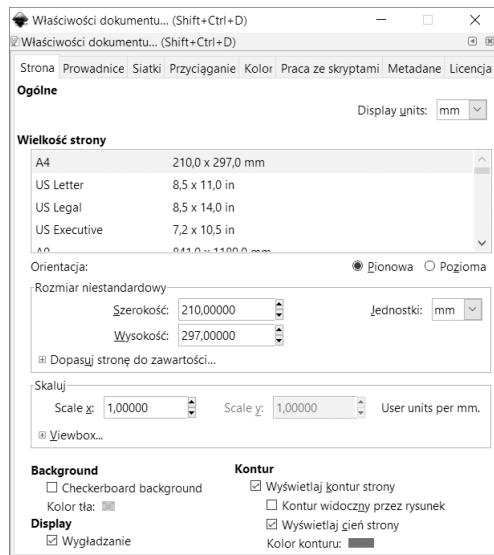
Kolejny bardzo potrzebny panel to *Wyrównaj i rozmieść...*, otwierany z paska poleceń (trzecia ikona z prawej strony). Korzystając z narzędzi dostępnych w tym panelu, można wyrównać utworzone obiekty względem siebie lub względem strony, można je też rozmieścić względem siebie.

Przydatny może się okazać panel *Właściwości dokumentu*, który można otworzyć, wybierając w menu *Plik/Właściwości dokumentu...* (rysunek 4.68). Jedną z ważniejszych opcji w tym panelu jest *Kolor tła*. Domyślnie jest on przezroczysty.



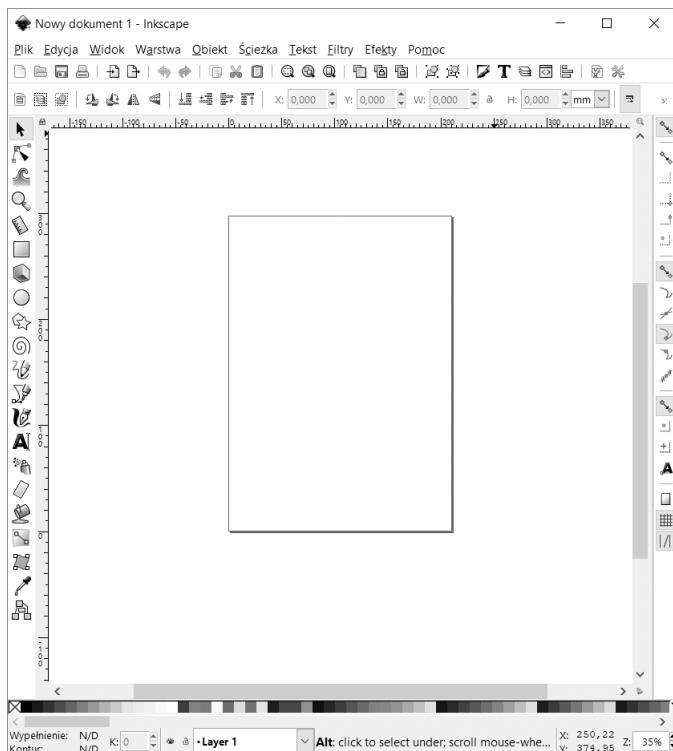
Rysunek 4.67.

Okno Wypełnienie i kontur



**Rysunek 4.68.** Panel Właściwości dokumentu...

W środkowej części okna znajduje się obszar roboczy dokumentu (rysunek 4.69).



**Rysunek 4.69.** Okno programu Inkscape

Przewijanie obszaru roboczego można wykonywać za pomocą pasków przewijania. Można też korzystać z klawiszy skrótów, używając klawisza ***Ctrl*** i klawiszy strzałek. Przybliżenie tworzonego obiektu uzyskamy za pomocą klawisza **+** (plus), oddalenie za pomocą klawisza **-** (minus). Podobny efekt można uzyskać za pomocą myszy: przybliżenie to wciśnięcie klawisza ***Ctrl*** i kliknięcie prawym przyciskiem myszy, a oddalenie — wciśnięcie klawisza ***Shift*** i kliknięcie prawym przyciskiem myszy. Można również wykorzystać znajdujące się w **Przyborniku** narzędzi **Przybliżanie/oddalenie rysunku**.

Aby utworzyć nowy dokument, należy wybrać w menu **Plik/Nowy** lub nacisnąć klawisze ***Ctrl+N***. Program otwiera kolejny dokument w oddzielnym oknie. Do przełączania się między oknami służy skrót klawiszowy ***Alt+Tab*** lub ***Ctrl+Tab***.

## 4.8.2. Tworzenie kształtów

Przybornik zawiera wiele narzędzi służących do tworzenia kształtów. Są to m.in.: **Prostokąt**, **Okrąg**, **Gwiazda**, **Spirala**.

Pierwsze narzędzie **Przybornika** to **Wskaźnik**, za pomocą którego można wybierać obiekt w obszarze roboczym tworzonego dokumentu. Po kliknięciu wybranej figury dookoła niej pojawią się uchwyty w kształcie strzałek. Za ich pomocą można przesuwać figurę, skalować ją, obracać lub pochylać. Klikając zaznaczony obiekt jeden raz, możnałączać się między trybem skalowania a trybem obracania.

### Ćwiczenie 4.22

Wybierz po kolei narzędzia kształtu i narysuj różne figury. Po narysowaniu figury wokół niej zostaną wyświetlane uchwyty w postaci rombów. Przeciągnij wybrane uchwyty i zmień kształt figury. Zmień kolor wypełnienia figury oraz kolor obrysu.

Kształt figury można również zmienić, ustawiając odpowiednie wielkości parametrów wyświetlanych na pasku właściwości. Ustawione tu wartości będą stosowane do nowo powstałych figur.

### Ćwiczenie 4.23

Za pomocą narzędzia **Prostokąt** narysuj kwadrat. Za pomocą narzędzia **Okrąg** narysuj koło, którego środek znajdzie się w górnym prawym rogu narysowanego kwadratu.

Wybierz narzędzie **Prostokąt** i narysuj figurę z wciśniętym klawiszem ***Ctrl***. Klawisz ***Ctrl*** spowoduje, że tworzony obiekt zachowa proporcje. Wybierz narzędzie **Okrąg**, ustaw kursor myszy nad prawym górnym rogiem kwadratu, wciśnij klawisz ***Ctrl*** i klawisz ***Shift***. Narysuj koło. Klawisz ***Shift*** spowoduje, że środek rysowanej figury znajdzie się w miejscu ustawienia kurSORA myszy.

### Ćwiczenie 4.24

Narysuj prostokąt, który będzie miał zaokrąglone rogi. Następnie narysuj kształt, który będzie wycinkiem koła (rysunek 4.70).



**Rysunek 4.70.** Kształty uzyskane za pomocą narzędzi Prostokąt i Okrąg

Wybierz narzędzie *Prostokąt* i narysuj figurę. Po narysowaniu prostokąta na jego rogach pojawią się małe kółka. Złap za jedno z nich i przeciągnij je wzdłuż jednego z brzegów. Spowoduje to zaokrąglenie rogów prostokąta.

Wybierz narzędzie *Okrąg* i narysuj figurę. Złap za kółeczko i przeciągnij je wzdłuż krańcza okręgu. Powstanie wycinek koła.

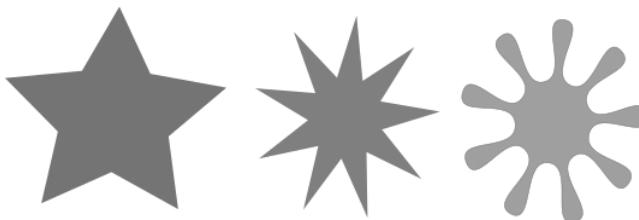
Wybierz narzędzie *Okrąg* i narysuj figurę. Złap za kółeczko i przeciągnij je wewnątrz okręgu. Jaką figurę otrzymałeś?

Kształt tworzonych wycinków można również ustawać na pasku właściwości.

### Ćwiczenie 4.25

Utwórz gwiazdę dziewięciornamienną.

Wybierz narzędzie *Gwiazda* i narysuj figurę. Powstanie gwiazda pięciornamienna. Liczbę ramion można zmienić na pasku właściwości. We właściwości *Narożniki* ustaw wartość **9**. Aby gwiazda miała węższe ramiona, można zwiększyć wartość właściwości *Proporcje ramion*. Do zaokrąglania ramion służy właściwość *Zaokrąglenia* (rysunek 4.71).

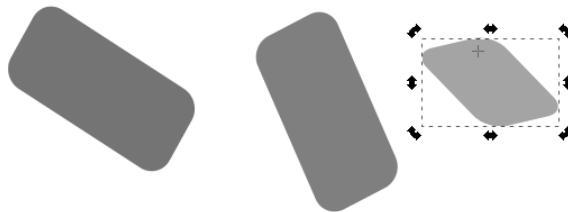


**Rysunek 4.71.** Kształty uzyskane za pomocą narzędzia Gwiazda

Za pomocą narzędzia *Gwiazda* można również tworzyć wielokąty. W tym celu po wybraniu narzędzia *Gwiazda* na pasku właściwości należy wybrać ikonę *Zamień w wielokąt foremny*. Teraz rysowana figura będzie wielokątem. Liczbę boków wielokąta można ustawić we właściwości *Narożniki*.

Każdy obiekt tworzony za pomocą programu Inkscape posiada tzw. punkt przekształcenia. Na przykład po utworzeniu obiektu możemy go zaznaczyć i przejść w tryb obracania obiektu. Na środku obiektu pojawi się punkt przekształcenia. Łapiąc za jedną ze strzałek, możemy obracać obiekt wokół punktu przekształcenia. Punkt ten możemy przesunąć w dowolne miejsce obszaru roboczego. Obiekt będzie nadal obracany względem punktu

przekształcenia (rysunek 4.72). Jeżeli przejdziemy w tryb skalowania, to skalowanie również będzie się odbywało względem ustawionego punktu przekształcenia.



**Rysunek 4.72.** Manipulowanie punktem przekształcenia figury

### 4.8.3. Tekst

Tekst jest tworzony za pomocą znajdującego się w *Przyborniku* narzędzia *Tekst*. Po wybraniu tego narzędzia należy kliknąć w miejscu wstawienia tekstu i wprowadzić napis. Aby złamać wprowadzany tekst, należy użyć klawisza *Enter*.

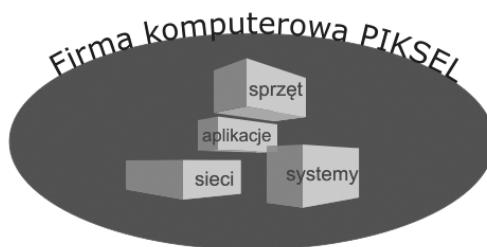
Innym sposobem wprowadzania tekstu jest zaznaczenie obszaru, w którym zostanie on wpisany. Po zaznaczeniu tekst zostanie umieszczony w utworzonym obszarze i będzie łamany automatycznie.

Ustawienie funkcji formatowania wprowadzonego tekstu jest realizowane na pasku właściwości. Można wybrać m.in. rodzaj czcionki, jej styl, wielkość, odstęp w linii oraz wyrównanie.

Wprowadzony tekst może zostać wyrównany wzdłuż wcześniej utworzonego kształtu.

#### Ćwiczenie 4.26

Wykonaj logo firmy komputerowej zgodnie z rysunkiem 4.73.



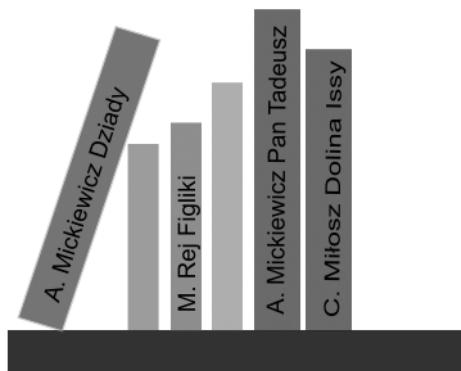
**Rysunek 4.73.** Logo firmy komputerowej

Wybierz narzędzie *Okrąg* i narysuj elipsę. Za pomocą paska kolorów wybierz kolor wypełnienia elipsy. Wybierz narzędzie *Tekst* i wprowadź tekst, który zostanie umieszczony nad elipsą. Wybierz narzędzie *Wskaźnik*, zaznacz elipsę, a następnie, trzymając wcisknięty klawisz *Shift*, zaznacz wprowadzony tekst. Przejdź do menu i wybierz *Tekst/*

**Wstaw na ścieżkę.** Tekst zostanie wyrównany wzdłuż krawędzi elipsy. Teraz skorzystaj ze znajdujących się na pasku właściwości ikon *Odbija zaznaczone obiekty poziomo* i *Odbija zaznaczone obiekty pionowo* w celu odpowiedniego ustawienia tekstu względem elipsy. Na początku tekstu wstaw kilka spacji, tak aby znalazł się on na środku górnej krawędzi elipsy. Za pomocą narzędzia *Obiekt 3D* dodaj kilka sześcianów i umieść na nich pokazane na rysunku teksty.

### Ćwiczenie 4.27

Wykonaj zgodnie z rysunkiem 4.74 grafikę przedstawiającą półkę z książkami.



**Rysunek 4.74.** Półka z książkami

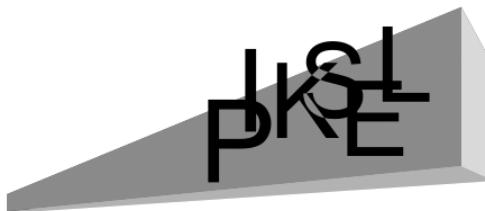
Grafika zostanie utworzona z wykorzystaniem narzędzi: *Prostokąt*, *Tekst*, *Ołówek* i *Wskaźnik*.

Wybierz narzędzie *Prostokąt* i narysuj figurę, która będzie półką. Wybierz kolor wypełnienia półki. Narysuj prostokąt, który będzie książką umieszczoną na półce. Wybierz narzędzie *Tekst* i wpisz tekst, który będzie tytułem książki. Ustaw czcionkę *Arial* i wysokość *20 pt*. Za pomocą narzędzia *Ołówek* narysuj linię prostą wzdłuż prawej krawędzi książki. Aby rysowana linia była prosta, wciśnij klawisz *Shift* i kliknij w dolnej części krawędzi książki, a następnie w jej górnej części. Wybierz narzędzie *Wskaźnik*, zaznacz utworzoną linię i trzymając wciśnięty klawisz *Shift*, zaznacz wprowadzony tekst. W menu *Tekst* wybierz opcję *Wstaw na ścieżkę*. Tekst zostanie wyrównany wzdłuż utworzonej linii. Linia pomocnicza nie musi być widoczna, więc zaznacz ją i z wciśniętym klawiszem *Shift* kliknij znajdującą się na pasku kolorów pierwszą ikonę z lewej strony. W ten sposób zostanie usunięty kolor konturu. Tak samo postępuj z pozostałymi książkami.

Aby uzyskać efekt pochylenia pierwszej książki, po utworzeniu prostokąta dla książki zaznacz go i kliknij raz, aby przełączyć się w tryb obracania uchwytów. Łapiąc za strzałkę w rogu prostokąta, obróć go.

### Ćwiczenie 4.28

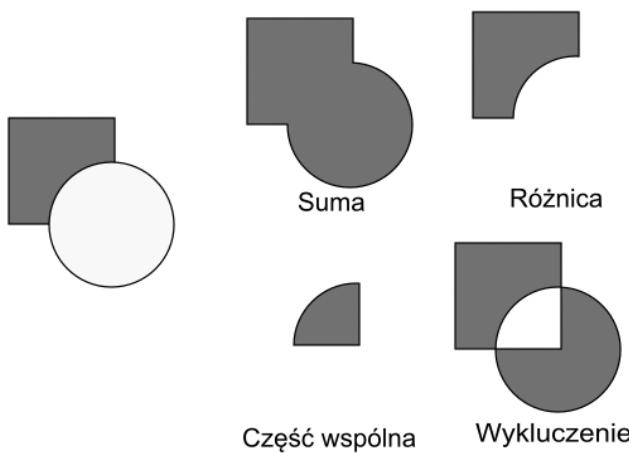
Wykonaj logotyp z nazwą firmy komputerowej zgodnie z rysunkiem 4.75.



**Rysunek 4.75.** Logotyp

Wybierz narzędzie *Obiekt 3D* i narysuj figurę, która będzie tłem napisu. Wybierz narzędzie *Tekst* i wpisz każdą literę tekstu w oddzielnym polu. Ustaw czcionkę *Arial* i wysokość *110 pt*. Wybierz narzędzie *Zaznaczanie* i przenieś po kolejne wszystkie litery na tło. Wzajemne położenie liter zostało pokazane na rysunku 4.75. Dla liter *K* i *S* zastosuj funkcję wykluczającą ich część wspólną. W tym celu zaznacz obydwie litery i wybierz w menu opcję *Ścieżka/Wykluczenie*.

W podobny sposób na obiektach zachodzących na siebie można wykonać inne operacje logiczne (rysunek 4.76).



**Rysunek 4.76.** Operacje logiczne na połączonych obiektach

Obiekty powstałe w wyniku wykonania podanych operacji logicznych zawsze mają kolor figury, która jest pod spodem.



## 4.9. Zapisywanie obrazów na potrzeby witryny internetowej

### 4.9.1. Zasady zapisywania grafiki

Najczęściej stosowane formaty plików graficznych wykorzystywane na stronach internetowych to: JPG, GIF oraz PNG.

#### Wybór formatu pliku

Obrazy proste, które mają niewielką liczbę kolorów, nie zawierają cieniowania i przejść tonalnych, zapisujemy w formacie GIF. Przy zapisie do tego formatu wybieramy jak najmniejszą liczbę kolorów, aby nie pogorszyć jakości obrazu. W tym formacie może być zapisane np. logo, które z reguły zawiera 2, 3 kolory. Aby obraz nie stracił na jakości, przy zapisie wybieramy 8–16 kolorów.

Do zapisu zdjęć, grafik zawierających przejścia tonalne itp. należy używać formatu JPG z odpowiednim stopniem kompresji. Optymalizacja w granicach 60–70% zwykle jest poziomem kompresji pozwalającym zachować dobrą jakość obrazu.

W formacie PNG zapisujemy grafiki zawierające efekt przezroczystości lub grafiki o wysokiej jakości. Mimo że format GIF również obsługuje funkcję przezroczystości, w formacie PNG obraz wygląda lepiej, a jego krawędzie są gładsze.

#### Optymalizacja plików graficznych

Optymalizacja plików graficznych to działania prowadzące do otrzymania pliku o możliwie małym rozmiarze przy zachowaniu możliwie najlepszej jakości (najlepszego wyglądu). Zaletą optymalizacji jest oszczędność miejsca na serwerze oraz szybsze ładowanie się stron.

Pliki zapisywane w formacie GIF optymalizujemy przez minimalizowanie liczby kolorów, a pliki JPG — przez procentowe określenie poziomu kompresji.

Większość programów graficznych przeznaczonych do tworzenia grafiki na potrzeby stron internetowych ma funkcje zapisu grafiki z możliwością optymalizacji pliku.

Istnieją również wyspecjalizowane programy do optymalizowania plików graficznych, np. Radical Image Optimization Tool.

### 4.9.2. Zapisywanie obrazów w programie Photoshop

Aby zapisać obraz utworzony na potrzeby internetu, należy wybrać *Plik/Zapisz dla Internetu....* Pozwoli to na kontrolowanie rozmiaru pliku i jakości obrazu.

Zostanie otwarte okno, w którym można ustawić opcje optymalizacji zapisywania pliku oraz opcje przeglądania zoptymalizowanych obrazów (rysunek 4.77).



**Rysunek 4.77.** Okno Zapisz dla Internetu...

W górnej części okna na zakładkach ustalamy sposób wyświetlania obrazu. Na przykład wybranie zakładki *2 na ekranie* spowoduje wyświetlenie oryginalnej pracy oraz obrazu zoptymalizowanego. Z lewej strony u góry została umieszczony przybornik. Po prawej stronie znajduje się zestaw narzędzi, za pomocą których określamy liczbę kolorów (format GIF) lub stopień kompresji (format JPG) oraz format zapisywanyego pliku.

W dolnej części okna można określić wielkość grafiki, która będzie wykorzystywana na stronie. Grafika powinna być dopasowana do układu strony, dlatego w opcji *Rozmiar obrazu* można podać jej wymagany rozmiar. Rozdzielcość wyświetlania zostanie automatycznie ustaliona na 72 ppi. Klikając przycisk *Zapisz*, zapisujemy zoptymalizowany plik w wybranym miejscu na dysku.

## 4.10. Pytania i zadania

### 4.10.1. Pytania

1. Podaj charakterystyczne cechy grafiki wektorowej.
2. Podaj charakterystyczne cechy grafiki rastrowej.
3. Jakie parametry wpływają na jakość obrazu w grafice rastrowej?
4. Czym charakteryzuje się model barw RGB?
5. Gdzie najczęściej jest stosowany model barw CMYK?

6. Wymień zalety stosowania do zapisu obrazu formatu PNG.
7. Jakich narzędzi dostępnych w programie graficznym Adobe Photoshop można użyć do usunięcia ze zdjęcia zbędnych elementów?
8. W jakim celu w programach graficznych stosowane są warstwy?
9. Wymień kilka metod tworzenia tła strony internetowej.
10. Na czym polega optymalizacja plików graficznych przy ich zapisywaniu na potrzeby strony internetowej?

## Pytania egzaminacyjne

### Zadanie 1.

Model opisu przestrzeni barw o parametrach *odcień, nasycenie i jasność* to:

- A. HSV.
- B. RGB.
- C. CMY.
- D. CMYK.

### Zadanie 2.

Wskaż model barw, który stosuje się do wyświetlania kolorów na ekranie monitora komputerowego:

- A. HLS.
- B. RGB.
- C. CMY.
- D. CMYK.

### Zadanie 3.

Który parametr obiektu graficznego ulegnie zmianie po modyfikacji wartości kanału alfa?

- A. Nasycenie barw.
- B. Przezroczystość.
- C. Ostrość krawędzi.
- D. Kolejność wyświetlania pikseli.

### Zadanie 4.

Który z formatów graficznych pozwala na zapis przejrzystego tła?

- A. GIF.
- B. RAW.
- C. BMP.
- D. JPEG

**Zadanie 5.**

Kolor 255 12 12 w modelu RGB na stronie WWW powinien być zapisany w postaci:

- A.** #2551212.
- B.** #EE0C0C.
- C.** #AB1A1D.
- D.** #FF0C0C.

**Zadanie 6.**

CMYK to zestaw czterech podstawowych kolorów farb drukarskich:

- A.** turkusowego, błękitnego, białego, różowego.
- B.** turkusowego, purpurowego, białego, czarnego.
- C.** czerwonego, purpurowego, żółtego, szarego.
- D.** turkusowego, purpurowego, żółtego, czarnego.

**Zadanie 7.**

Cechą formatu PNG jest:

- A.** obsługa animacji.
- B.** bezstratna kompresja.
- C.** brak obsługi kanału alfa.
- D.** reprezentacja grafiki wektorowej.

**Zadanie 8.**

Które ze zdań jest prawdziwe w stosunku do grafiki rastrowej?

- A.** Podczas przekształcania polegającego na skalowaniu skalowany obraz nie zmienia jakości.
- B.** Zapisany obraz jest opisywany za pośrednictwem figur geometrycznych umieszczonych w układzie współrzędnych.
- C.** Grafika rastrowa nie jest zapisywana w formacie WMF (ang. *Windows Metafile Format* — format metaplików Windowsa).
- D.** Jest to prezentacja obrazu za pomocą pionowo-poziomej siatki odpowiednio kolorowanych pikseli na monitorze komputera, drukarce lub innym urządzeniu wyjściowym.

**Zadanie 9.**

Aby zbadać rozkład ilościowy poszczególnych kolorów zdjęcia, należy użyć:

- A.** desaturacji.
- B.** histogramu.
- C.** balansu kolorów.
- D.** rozmycia Gaussa.

**Zadanie 10.**

Wymiary:	JPG
Format:	4272x2848px
Rozdzielcość:	72dpi

W ramce przedstawiono właściwości pliku graficznego. W celu optymalizacji czasu ładowania rysunku na stronę WWW należy:

- A.** zmniejszyć wymiary rysunku.
- B.** zwiększyć rozdzielcość.
- C.** zmienić format grafiki na CDR.
- D.** zmienić proporcje szerokości do wysokości.

## 4.10.2. Zadania

**Zadanie 1.**

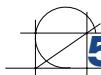
Utwórz dokument reklamujący biuro podróży GULIWER (rysunek 4.78). Przygotuj pliki ze zdjęciami. Umieść je w dowolnym programie do tworzenia grafiki na warstwach. Obrazy przeskaluj i obróć tak, aby zmieściły się obok siebie. Ustaw dla każdego obrazka dodatkowe efekty (np. cień). Utwórz dla nich tło. Do utworzonego obrazu dodaj teksty reklamujące biuro podróży. Teksty na obrazie porozmieszczaj w różny sposób. Ustaw rozmiar, kolor i rodzaj czcionki. Dołącz do tekstów efekty specjalne.



**Rysunek 4.78.** Grafika ze zdjęciami i z tekstem

# 5

# Graficzny projekt strony internetowej



## 5.1. Opracowanie projektu

Przystępując do tworzenia projektu graficznego strony internetowej, powinniśmy dążyć do uzyskania przejrzystej i łatwej w obsłudze struktury. Użytkownik, który znajdzie się na stronie, nie powinien mieć problemu z okrešleniem, czego dotyczy witryna, jaka jest jej nazwa, jakie kategorie podstron są dostępne, jakie są opcje wyboru oraz w jaki sposób można czegoś poszukać.

Dobrym rozwiązaniem jest zastosowanie się do ogólnie przyjętych zasad. Ułatwi to użytkownikom poruszanie się po witrynie i znalezienie odpowiednich informacji.

Standardem są:

- umiejscowienie logo w lewym górnym rogu;
- łatwy do znalezienia panel nawigacji, najlepiej znajdujący się u góry strony;
- wyróżnienie elementu, który jest linkiem;
- zamieszczanie zdjęć bardzo dobrej jakości;
- umieszczenie w stopce danych kontaktowych i informacji o prawach autorskich.

### 5.1.1. Użyteczność strony internetowej

Użyteczność strony sprowadza się do:

- intuicyjnej nawigacji,
- łatwości wyszukiwania informacji,
- komunikacji zrozumiałej dla użytkownika.

## 5.1.2. Układ strony

Projektowana strona internetowa powinna być dostosowana do wymagań różnych urządzeń, takich jak komputery, notebooki, tablety i smartfony. Jedną z metod rozwiązania problemu może być utworzenie kolejnej wersji strony internetowej, która byłaby wyświetlana na urządzeniach o niewielkich rozmiarach ekranów. Jednak tworzenie kilku wersji tej samej strony internetowej może się okazać i kłopotliwe, i kosztowne. Innym rozwiązaniem może być zaprojektowanie strony responsywnej (ang. *Responsive Web Design* — RWD). W tym rozwiązaniu zakłada się, że przygotowana strona internetowa będzie się automatycznie dopasowywała do urządzenia, na którym będzie wyświetlana. Podczas projektowania strony internetowej należy dodatkowo określić tzw. punkty zmian (ang. *breakpoints*) oraz liczbę elementów, które będą podlegały zmianom. Strona, w zależności od rozdzielczości ekranu urządzenia, na którym będzie przeglądana, będzie zmieniała swój wygląd; zmiany będą dotyczyć położenia poszczególnych elementów, wyświetlanych treści oraz sposobu wyświetlania nawigacji.

Punktów zmian w projektowanej stronie może być wiele, ale najczęściej przygotowywane są wersje na tablet, smartfon i komputer PC.

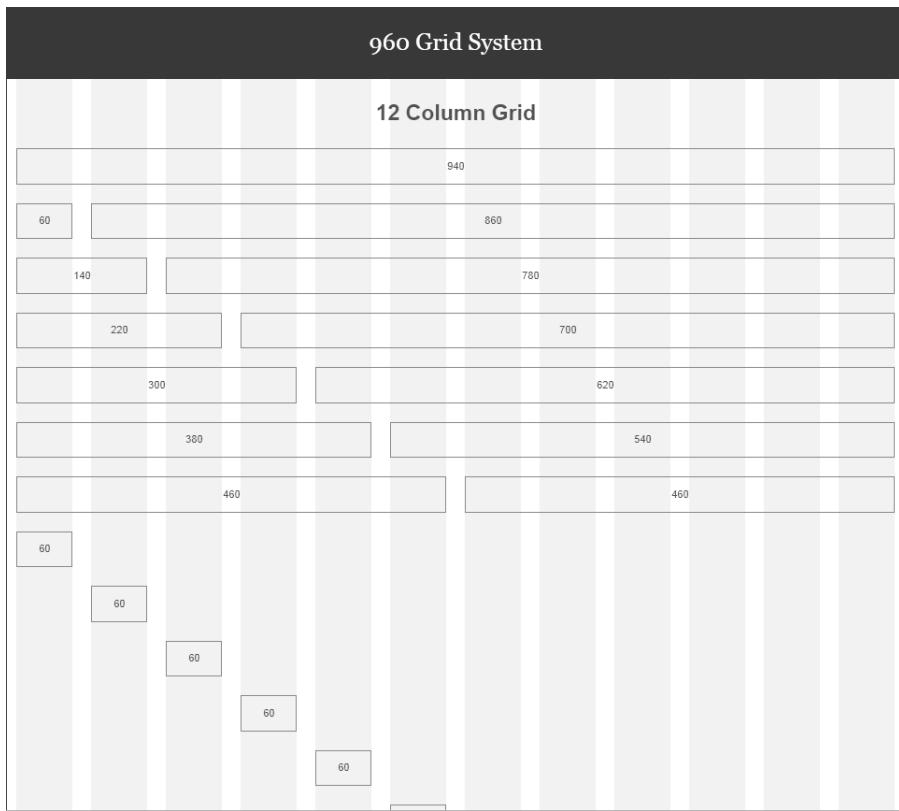
Standard stosowany na stronach internetowych to, jak wcześniej wspomniano, logo jako główny element strony, umieszczony w jej lewym górnym rogu. Menu jest umieszczane z prawej strony logo lub pod nim. Kolejny element spotykany na większości stron to tzw. slider. Jest to kilka dobrej jakości i właściwie wyeksponowanych zdjęć, które zmieniają się automatycznie. Te kilka zdjęć ma ogromny wpływ na pozytywny odbiór strony. Slider to ważne narzędzie, które umożliwia komunikację z użytkownikiem strony. Tu umieszcza się tekst i zdjęcia, które są najważniejsze w odbiorze witryny. Jeżeli slider został dobrze zaprojektowany, to bez konieczności zaglądania na kolejne podstrony użytkownikowie się, czego dotyczy witryna. Poniżej tego elementu powinna się znaleźć treść strony, najczęściej w postaci kolumn z tekstem. Układ kolumn przyczynia się do tego, że informacje są uporządkowane i łatwiej przyswajane przez odbiorcę.

Ostatnim elementem strony jest stopka. Czasami jest rozbudowana i zawiera wiele różnych informacji, np.: menu (aby nie trzeba było wracać na górę strony), ikony (np. z łączami do akcji społecznościowych), opcje logowania, czyli wszystko to, co jest potrzebne na stronie, ale nie jest tak istotne, żeby zamieszczać to w głównej jej części.

Projektowanie wyglądu strony ułatwia tzw. *960 GRID SYSTEM* (dostępny na stronie <http://960.gs>). Zawiera on szablony pomagające w projektowaniu strony (rysunek 5.1).

Grid to system poziomych i pionowych linii pomocniczych dzielących przestrzeń projektu graficznego na bloki. Pozwala w logiczny sposób uporządkować elementy strony internetowej, takie jak nagłówki, teksty, przyciski, pola tekstowe, zdjęcia i ikony.

Szablon GRID SYSTEM daje możliwość projektowania stron z podziałem na wiele kolumn. Standardowo GRID proponuje podział na 12 lub 16 kolumn. Opierając się na tym szablonie i stosując się do zaproponowanych podziałów, w prosty sposób utworzymy uporządkowaną stronę.



**Rysunek 5.1.** Szablon 960 GRID SYSTEM

Z podanej strony można pobrać zestaw szablonów, klikając przycisk *Big ol' DOWNLOAD button* — w folderze *templates* znajdzie się zestaw szablonów przygotowanych dla różnych aplikacji. W celu wykorzystania szablonu w programie Photoshop wybieramy folder *photoshop*, a w nim szablon dla 16 kolumn. Zostanie otwarty szablon pomocny w tworzeniu graficznego projektu strony internetowej. Zawiera on linie pomocnicze, przy użyciu których łatwo można budować wybrane segmenty strony.

W podobny sposób, wybierając folder *gimp*, można wykorzystać szablon w programie GIMP.



## 5.2. Projektowanie strony

Mając przygotowany na kartce szkic strony, przechodzimy do jej projektowania w programie graficznym.

### Przykład 5.1

Nasz projekt graficzny zostanie utworzony w programie Photoshop, w którym został już otwarty szablon GRID. Aby wygodniej pracować, powiększamy obszar

roboczy — wybieramy z menu *Obraz/Rozmiar* rozmiar obszaru roboczego i w otwartym oknie ustawiamy szerokość na 1280 px, a wysokość np. na 2000 px, tak aby nie przejmować się długością projektowanej strony.

Przyciemnione kolumny widoczne w szablonie są niepotrzebne, można je więc usunąć po odblokowaniu warstwy *16col Grid*. Następnym krokiem powinno być utworzenie nowej warstwy i usunięcie warstwy *Background*. Tak przygotowany szablon jest gotowy do dalszej obróbki.

Rozpoczynamy tworzenie projektu graficznego.

Wybieramy narzędzie *Prostokąt* i rysujemy prostokąt w górnej części szablonu. Po zaznaczeniu narysowanego obiektu i kliknięciu *Ctrl+T* w pasku *Właściwości* możemy dokładnie określić w pikselach wielkość prostokąta (np. *640,00 piks*; rysunek 5.2). Po ustawieniu odpowiednich wielkości zatwierdzamy je klawiszem *Enter*.



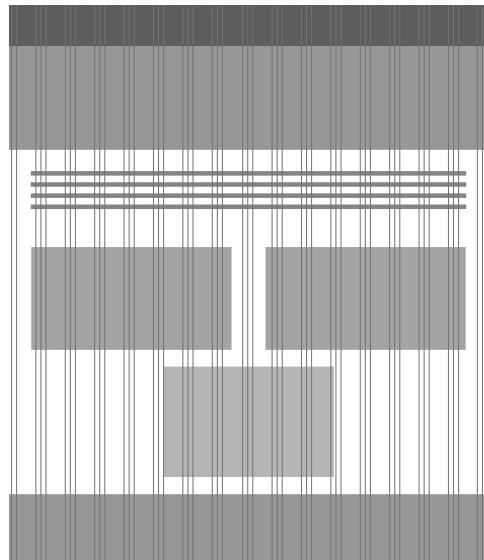
**Rysunek 5.2.** Ustawienie rozmiaru elementu

Można powięlić utworzony prostokąt i dostosować odpowiednio jego wysokość. Załączamy prostokąt i trzymając wcisnięty na klawiaturze klawisz *Alt*, przeciągamy figurę w dół. Po skopiowaniu prostokąta zostanie utworzona kolejna warstwa. Na pasku *Właściwości* ustalamy wysokość nowego elementu (np. 100 px). Zatwierdzamy ustawienia. Nowy element przenosimy w góre, dosuwając go do pierwszego elementu. Zmieniamy kolorystykę, aby nowy element odróżniał się od poprzedniego. To będzie obszar logo.

Podobnie tworzymy kolejny element, o wysokości np. 240 px. Będzie to slider. Zmieniamy kolor prostokąta i odsuwamy go od poprzedniego. W podobny sposób budujemy następne bloki projektu.

Tworzymy jeszcze warstwę tła, przesuwamy ją na dół i wypełniamy kolorem białym, wykorzystując kombinację klawiszy *Shift+F5* i w otwartym oknie wybierając *Kolor pierwszego planu*. Prycinamy wielkość projektu do planowanej wielkości strony — wybieramy narzędzie *Kadrowanie* i zaznaczamy obszar strony. Zatwierdzamy ustawienia klawiszem *Enter* (rysunek 5.3).

Jest to pierwszy etap pracy nad projektem graficznym strony. Przenieśliśmy układ zaprojektowany na kartce papieru do programu graficznego.



**Rysunek 5.3.**

Projekt strony w Photoshopie

Teraz należy wypełnić projekt treścią. Zaczniemy od wpisania tekstów.

Wpisujemy teksty na pierwszej warstwie (rysunek 5.4). Jeżeli linie pomocnicze przeszczadzają nam w pracy, można je wyłączyć, wybierając z menu *Widok/Usuń linie pomocnicze*.



**Rysunek 5.4.** Górnny fragment projektowanej strony

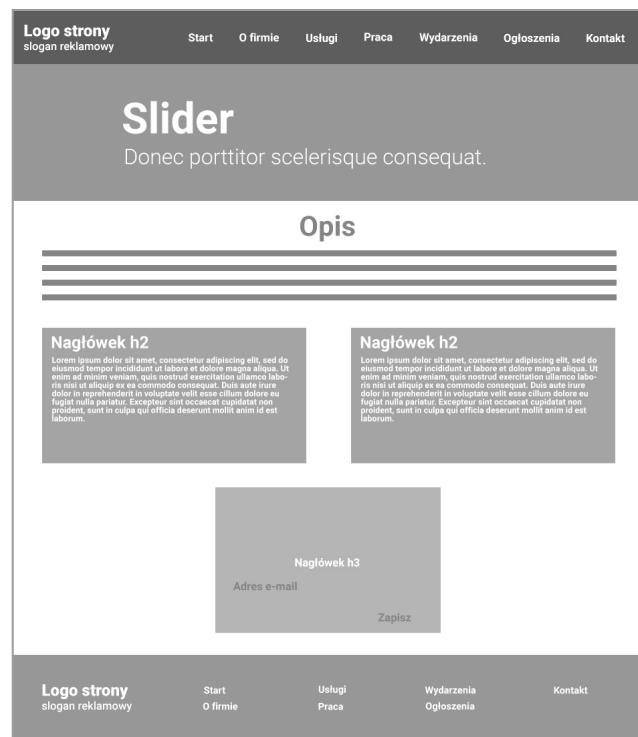
Efekt tworzenia typografii strony może wyglądać tak jak na rysunku 5.5. Warstwy zawierające podobne elementy powinny zostać pogrupowane.

Tak przygotowany projekt strony należy uzupełnić elementami graficznymi. Na stronę najczęściej wstawia się ikony, zdjęcia i przyciski.

Ikony wzbogacają wizualnie stronę i ułatwiają nawigację. Ikona powinna być streszczeniem każdej sekcji i wyrażać to, co chcemy przekazać tekstem.

Zdjęcia są bardzo ważnym elementem w sliderze. Zdjęcie użyte w tym bloku powinno być bardzo dobrej jakości. Jeżeli nie mamy odpowiednich zdjęć, możemy skorzystać z dostępnych stocków graficznych (banków zdjęć). Tam znajdują się zdjęcia naprawdę wysokiej jakości.

Wracamy do projektu strony przygotowanego w programie Photoshop. W pierwszej kolejności do tekstów dodamy ikony. Otwieramy folder z wcześniej przygotowanymi



**Rysunek 5.5.** Przykład typografii strony

ikonami i po kolej przeciagamy je do bloków z tekstem. Należy dobrać wielkość ikon (np. 48×48 px) i umieścić je w odpowiednim miejscu (rysunek 5.6). Można wstawić linię pomocniczą, aby wyrównać ikony, które powinny się znaleźć na tej samej wysokości.

Kolejny rodzaj elementów to przyciski. Tworzymy je, wykorzystując narzędzie *Prostokąt*. Trzeba pamiętać o umieszczaniu takich elementów pod warstwą tekstu.

Ostatni etap pracy to dodanie zdjęć, pokolorowanie bloków i dodanie efektów.

Wybieramy wcześniej przygotowane zdjęcie i przeciagamy je nad warstwę slidera. Dopasowujemy rozmiar zdjęcia do warstwy.

Zmieniamy kolorystykę. Dopasowanie kolorystyki strony wymaga przetestowania. Przykładowa strona może wyglądać tak jak na rysunku 5.7.

## Ćwiczenie 5.1

Przygotuj szkic witryny internetowej, która będzie zawierała informacje na temat Twojej szkoły, i utwórz na tej podstawie projekt graficzny strony WWW.

**Rysunek 5.6.**

Dodanie ikon i przycisków do projektu strony

**Rysunek 5.7.**

Projekt przykładowej strony

## 5.3. Struktura strony internetowej

Tworzona strona internetowa powinna posiadać odpowiednią strukturę, odpowiedni wygląd (kolorystykę, elementy dekoracyjne, krój czcionki) i odpowiednie rozmieszczenie treści.

Podstawowe pytanie, na które musi odpowiedzieć projektant strony internetowej, dotyczy układu strony.

Najpopularniejsze rodzaje struktury strony internetowej to:

- Układ onepage — często spotykany na stronach firmowych, blogach i stronach związanych ze sklepami internetowymi. Jest to układ składający się z jednej strony. Najczęściej w takim układzie pozycje menu nie odsyłają do kolejnych podstron, ale do poszczególnych sekcji tej samej strony. Użytkownik przewija stronę, od początku do końca odczytując jej zawartość.
- Układ dwóch kolumn — najpopularniejszy układ stron, który składa się z dwóch kolumn. Jedna z kolumn zazwyczaj jest wąska (tzw. sidebar) i zawiera menu, reklamy, dodatkowe widżety. Druga zwykle obejmuje 70 – 80% szerokości strony i jest podstawową kolumną, na której publikowana jest treść główna. Najczęściej kolumna sidebar umieszczana jest po lewej stronie witryny.
- Układ wielu kolumn — podobny do układu dwóch kolumn, ale strona dzielona jest na większą liczbę bloków. Często są to trzy kolumny, a czasami jest ich cztery lub pięć. Często w takim układzie nie ma wyeksponowanej kolumny głównej. W takim przypadku szerokość kolumn najczęściej jest równa.
- Układ slider + kolumny — składa się z kolumn, do których dodany został slider. Slider zamieszczany jest w górnej sekcji strony i odpowiada za jej wygląd. Poniżej umieszczane są kolumny z treścią. Czasami zamiast slidera w górnej części strony umieszczany jest baner (obrazek lub grafika).

Slider może zostać wykorzystany również w układzie onepage i w układzie z dwoma kolumnami.

## 5.4. Dobór palety barw

Istotnym elementem projektowania witryny internetowej jest optymalny dobór kolorów.

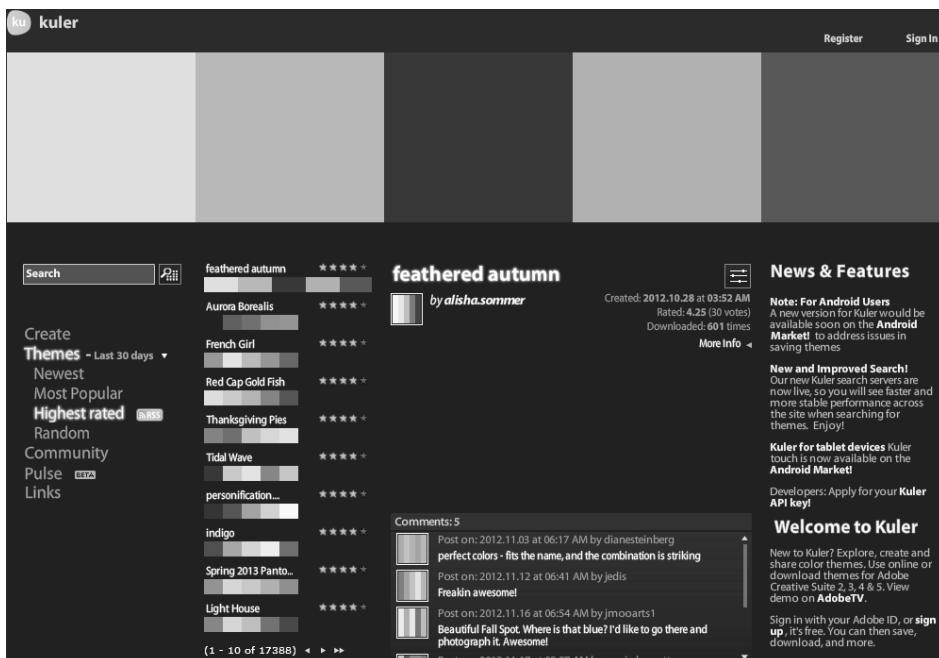
Zasadą stosowaną przy projektowaniu stron WWW jest wybieranie jak najmniejszej liczby kolorów. Powinny to być trzy lub cztery kolory, czasami pięć. Zastosowanie większej liczby barw, szczególnie gdy są źle dobrane, stwarza wrażenie niespójności.

Istnieje wiele narzędzi dostępnych w internecie, które pomagają w wyborze właściwej palety barw.

## 5.4.1. Kuler

Kuler to opracowany przez firmę Adobe program, który można wykorzystać do szybkiego tworzenia palet kolorystycznych używanych na stronach internetowych. Ale przede wszystkim jest to baza wielu tysięcy zestawów kolorów. Można też zaprojektować własne palety kolorystyczne.

Z programu można korzystać bezpośrednio na stronie internetowej, można także pobrać go na komputer i używać jako samodzielnej aplikacji. Dostępny jest w witrynie <https://kuler.adobe.com/>, gdzie mamy do dyspozycji zestawy kolorów widocznych w menu z lewej strony. Są to zestawy: najnowsze, najpopularniejsze, najwyższej oceniane i wyświetlane losowo (rysunek 5.8).

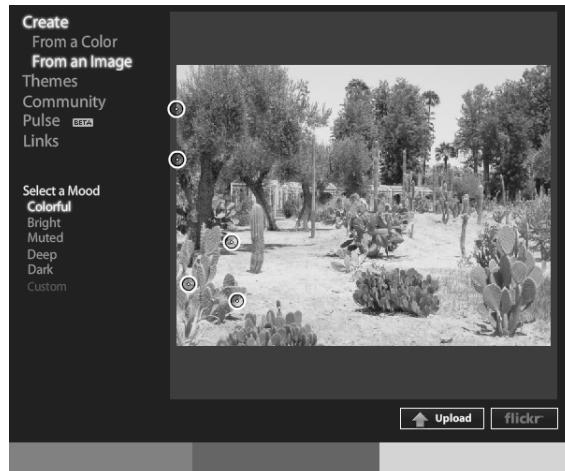


**Rysunek 5.8.** Okno programu Kuler

Aby stworzyć własną paletę kolorów, należy wybrać opcję *Create*. Paletę można wygenerować na podstawie kolorów (*From a Color*) lub z obrazu (*From an Image*). Po kliknięciu opcji *From an Image* można do dalszej pracy wybrać własne zdjęcie lub fotografię z serwisu Flickr.

### Przykład 5.2

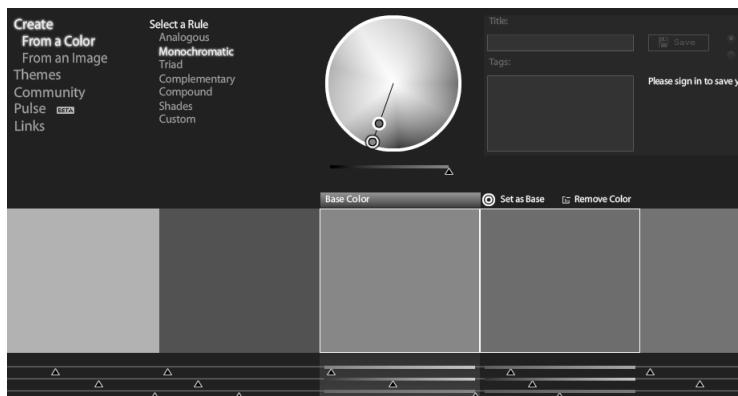
Uruchom narzędzie Kuler. Utwórz paletę z własnego zdjęcia — kliknij przycisk *Upload*, aby wczytać obraz. Na zdjęciu zostaną zaznaczone miejsca próbkowania (rysunek 5.9).



**Rysunek 5.9.** Kuler — miejsca próbkowania zdjęcia

Miejsca próbkowania można dowolnie przemieszczać na zdjęciu, wskazując kolory, które powinny się znaleźć w paletie. W dolnej części okna wyświetli się paleta wybranych kolorów.

Gdy klikniemy opcję *From a Color*, w oknie pojawi się koło podstawowych kolorów; dostępna będzie opcja *Select a Rule* z rodzajami palet oferowanymi przez program: *Analogous*, *Monochromatic*, *Triad*, *Complementary*, *Compound*, *Shades* i *Custom*. Decydujemy się np. na paletę monochromatyczną. Przesuwając wskaźnikiem po kole, możemy wybrać interesującą nas paletę kolorów (rysunek 5.10).



**Rysunek 5.10.** Paleta Monochromatic

Przesuwając wskaźnik w stronę środka, uzyskujemy łagodniejsze kolory; bliżej obwodu kolory są bardziej wyraziste.

Wybranie palety analogowej spowoduje pojawienie się na kole większej liczby wskaźników. Manipulując nimi, możemy ustalić własną paletę kolorów (rysunek 5.11).



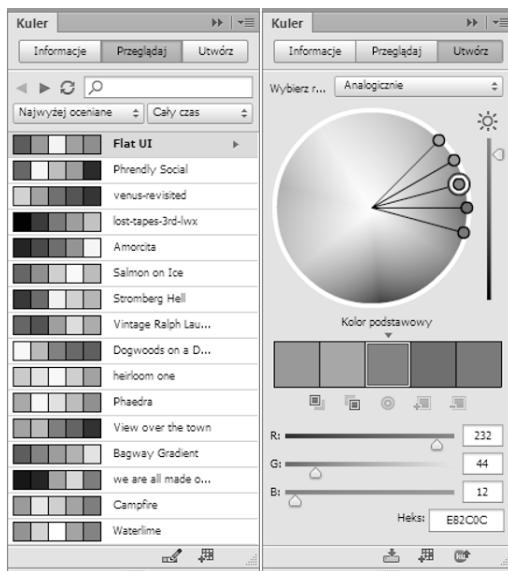
**Rysunek 5.11.** Paleta Analogous

Kuler jest wbudowany w program Photoshop — od wersji CS4. Aby uruchomić to narzędzie w Photoshopie, należy z menu wybrać *Okno/Rozszerzenia/Kuler* (rysunek 5.12).

Zakładka *Utwórz* pozwala na tworzenie własnej palety kolorów.

W polu *Wybierz* można określić rodzaj palet oferowanych przez program.

Dobór palety barw zaczynamy od ustalenia jej podstawowego koloru. Zwykle jest to kolor ze zdjęcia będącego dominującym elementem strony, np. tego, które wybraliśmy do warstwy slidera. Kolor można określić, podając jego wartość liczbową lub klikając pipetą w danym miejscu zdjęcia.



**Rysunek 5.12.**  
Kuler w programie Photoshop

## 5.4.2. Accessibility Color Wheel

Przydatne w doborze kolorów tła oraz tekstu może być narzędzie Accessibility Color Wheel, dostępne na stronie <http://gmazzocato.altervista.org/colorwheel/wheel.php>. Po otwarciu strony można określić kolor tła i kolor tekstu, aby się przekonać, czy tekst będzie dobrze widoczny na danym tle. Dodatkowo program poinformuje nas odpowiednim komunikatem, czy prawidłowo dobraliśmy kolory (rysunek 5.13).

Use: Choose a foreground color by pointing the mouse over the wheel or the vertical grey gradation strip and click or, if you have a touch screen, just touch them. Then click the "Background" button and choose a background color the same way. If a checkmark becomes visible the color pair is good for accessibility. Otherwise change one color or both by selecting foreground or background with the buttons.

**Normal**

1 \* #ffff93    2 \* #1c6700    Contrast ratio [6.7:1 ok]    Update

\* You can enter an hex value (3 or 6 digits) in this box. Then just click Update

+ Share |

**Values and examples**

Deutanope	Protanope	Tritanope
1 #ffff93    2 #5b3900 Contrast ratio 9.2:1 ok	1 #ffff92    2 #7e4e00 Contrast ratio 6.5:1 ok	1 #ffecf0    2 #28455b Contrast ratio 8.8:1 ok
Deutanopia is insensitivity to green. This box simulates the vision of deutanope (partially color blind) people. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque pede felis, consequat sit amet, congue in, 1234 ultrices id, orci. Phasellus quam lacus, mollis nec, interdum et, malesuada nec, mauris. Nulla facilisi. Ut pharetra dignissim risus. Etiam at sapien et leo porta accumsan. Praesent lacus lectus, elementum quis, lobortis vitae, egestas non, dul.	Protanopia is insensitivity to red. This box simulates the vision of protanope (partially color blind) people. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque pede felis, consequat sit amet, congue in, 1234 ultrices id, orci. Phasellus quam lacus, mollis nec, interdum et, malesuada nec, mauris. Nulla facilisi. Ut pharetra dignissim risus. Etiam at sapien et leo porta accumsan. Praesent lacus lectus, elementum quis, lobortis vitae, egestas non, dul.	Tritanopia is very rare and is insensitivity to blue. This box simulates the vision of tritanope (partially color blind) people. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque pede felis, consequat sit amet, congue in, 1234 ultrices id, orci. Phasellus quam lacus, mollis nec, interdum et, malesuada nec, mauris. Nulla facilisi. Ut pharetra dignissim risus. Etiam at sapien et leo porta accumsan. Praesent lacus lectus, elementum quis, lobortis vitae, egestas non, dul.

**Rysunek 5.13.** Testowanie koloru tekstu i koloru tła

Gdy wybierzemy kolor pierwszoplanowy i kolor tła, w prawej części okna wyświetli się podgląd zaproponowanej kolorystyki, a niżej pojawi się informacja — w postaci heksadecymalnej — o wybranych kolorach oraz o ich kontraste.

## 5.5. Dobór czcionki

Obecnie strony internetowe zawierają wiele różnych elementów, takich jak grafika, zdjęcia, animacje. Mimo to tekst jest najczęściej głównym elementem strony internetowej. Strona internetowa powinna być czytelna nie tylko na komputerze, ale też na każdym urządzeniu mobilnym. Dlatego dobór odpowiedniego koloru, kroju i stylu czcionki jest tak istotny.

### 5.5.1. Rozmiar czcionki

Informacje o rozmiarze czcionki są umieszczane w plikach ze stylami CSS. Jeżeli rozmiar nie zostanie określony przez projektanta strony, to standardowo ustawiana jest domyślna wartość dla wybranej przeglądarki. Przeważnie jest to **16px**.

### Przykład 5.3

```
p {  
    font-size: 1.2em;  
}
```

Użyta jednostka *em* pozwala na skalowanie rozmiaru czcionki w zależności od tego, jak zmienia się rozmiar dokumentu, czyli na jakim urządzeniu wyświetlana jest zawartość strony.

Jeśli wielkość czcionki wynosi *16px*, to *1em* odpowiada 100%, czyli też wynosi *16px*. Podobnie przy zapisie `font-size: 1.2em;` wielkość czcionki wynosi około *20px*.

Wybierając wielkość czcionki, powinniśmy:

- określić podstawowy rozmiar czcionki dla dokumentu;
- używać skalowalnych rozmiarów czcionek w odniesieniu do określonego rozmiaru podstawowego;
- określić interlinię pomiędzy kolejnymi wierszami tekstu — zaleca się, aby wynosiła ona *1.2em* (`line-height`);
- używać niewielu rodzajów i wielkości czcionek na stronie WWW.

### Podstawowy rozmiar czcionki

Podstawowy rozmiar czcionki dla strony internetowej lub aplikacji powinien wynosić *16px*.

### Przykład 5.4

```
body {  
    font-size: 16px;  
}  
  
lub  
  
html {  
    font-size: 16px;  
}
```

### Relatywny rozmiar czcionki

Do definiowania wielkości czcionki dla nagłówków typu H1, H2 lub H3, akapitów i innych bloków tekstu powinny być używane jednostki em lub procenty. Gwarantuje to zawsze prawidłową wielkość tekstu, bez względu na to, gdzie strona będzie wyświetlana, i zachowanie zależności między różnymi blokami tekstu.

## Odległość między kolejnymi wierszami tekstu

Podczas odczytywania zawartości strony na niewielkich ekranach niewielka odległość między liniami tekstu powoduje, że jest on mało czytelny. Z tego powodu zaleca się stosowanie między wierszami odległości minimum 1,2 razy większej niż wysokość tekstu. Jeżeli podstawowa czcionka ma wysokość *16px*, to odległość między wierszami powinna wynosić więcej niż *19px*.

### Przykład 5.5

```
p {
    font-size: 16px;
    line-height: 1.2em;
}
```

## Ograniczona liczba i wielkość czcionek

Podczas tworzenia strony powinna zostać określona minimalna liczba potrzebnych rodzajów czcionek i powinny zostać ustalone wielkości tych rodzajów czcionek.

### Przykład 5.6

```
body {
    font-size: 16px;
}

p {
    font-size: 1.1em;
}

h1 {
    font-size: 2em;
}

h2 {
    font-size: 1.5em;
}
```

## 5.5.2. Krój czcionki

Krój czcionki wybrany na stronę internetową powinien pasować do stosowanego na stronie stylu lub szablonu, ale również do treści tworzonej strony.

Powinien także zawierać wszystkie znaki, które będą używane na stronie. Jeżeli strona jest tworzona w języku polskim, należy zwrócić szczególną uwagę na to, czy w zestawie występują polskie znaki diakrytyczne. Poza tym należy sprawdzić, czy występują i jak wyglądają inne specyficzne znaki, np. polskie cudzysłowy, znak zapytania, twarda spacja.

Czcionki możemy podzielić na kilka grup:

### **Czcionki szeryfowe**

Takie czcionki zawierają niewielkie poprzeczne lub ukośne linie, które zwiększą czytelność tekstu. Nadają się do publikacji papierowych. Najpopularniejsze z nich to Times New Roman i Georgia.

Na stronach internetowych sprawdzają się jako elementy logotypu, nagłówków, podtytułów, cytatów pisanych kursywą. Nadają stronie elegancki wygląd.

### **Czcionki bezszeryfowe**

To czcionki pozbawione ozdobników. Końcówki znaków są proste. Najpopularniejsze czcionki bezszeryfowe to Arial i Verdana.

W publikacjach papierowych nadają się do tworzenia tytułów i nagłówków. Często używane są na stronach internetowych, ponieważ nie męczą wzroku i są lepiej przy stosowane do czytania na ekranie komputera.

### **Czcionki typu slab**

Są to czcionki belkowe z szeryfami w postaci jednolitych prostokątów. Nadają się do różnego rodzaju logotypów. Raczej nie są stosowane do głównych tekstów na stronie.

### **Czcionki scriptowe**

To czcionki ozdobne, cechujące się dopracowanym i starannym wykończeniem. Odwzorowują charakter pisma odręcznego. Stosowane są w nagłówkach, tytułach blogów.

### **Zestawy czcionek**

Projektując stronę internetową, należy wybierać takie czcionki, które będą dobrze do siebie pasować. Najlepiej zacząć od wyboru czcionki do logotypu i do niej dopasować pozostałe czcionki.

Ze względu na wygląd i funkcjonalność popularne są czcionki Google Fonts. Łatwo się z nich korzysta. Google Fonts oferuje ponad 600 różnych czcionek, których można użyć na tworzonej stronie internetowej.



## **5.6. Ułatwienia dostępu**

Dostępność serwisów internetowych jest ważna dla wszystkich bez wyjątku. Ale ma szczególne znaczenie dla osób narażonych na wykluczenie cyfrowe, np. seniorów czy osób z niepełnosprawnością.

WCAG 2.0 to zbiór dokumentów opracowany przez organizację World Wide Web Consortium (W3C). Jest on uznawany za najważniejszy dokument określający zasady dostępności serwisów internetowych.

Zasady te mówią, że każdy serwis internetowy powinien być dostępny dla każdego, bez względu na:

- poziom jego sprawności,
- wiek,
- sprzęt czy oprogramowanie, którego używa.

Oto najistotniejsze zasady dostępności wynikające z WCAG 2.0:

- Wszystkie elementy graficzne powinny mieć zwięzły tekst alternatywny (alt), który opisuje, co znajduje się na grafice lub — jeśli grafika jest odnośnikiem — dokąd prowadzi ten odnośnik. Jeśli grafiki są czysto dekoracyjne, powinny mieć „pusty atrybut alt”.
- Należy unikać animowanych elementów i poruszających się tekstów, ponieważ rozpraszały one wszystkich użytkowników, nie tylko niepełnosprawnych. Niektóre szczególnie agresywnie i szybko animowane grafiki mogą stanowić zagrożenie dla osób cierpiących na padaczkę fotogenną.
- Wszystkie pliki dźwiękowe (np. audycje, wywiady, wykłady) powinny być uzupełnione o transkrypcję tekstową. Odtwarzacze tych plików zamieszczone na stronie powinny dać się obsłużyć za pomocą klawiatury i być dostępne dla osób niewidomych.
- Wszystkie pliki wideo powinny być uzupełnione o napisy dla osób niesłyszących. Odtwarzacze powinny być dostępne dla osób niewidomych i osób korzystających wyłącznie z klawiatury.
- Wszelkie pliki multimedialne i flash powinny być dostępne lub udostępnione w postaci alternatywnej.
- Pliki PDF, DOC i inne popularne pliki do ściągnięcia powinny być przygotowane jako dostępne. Na przykład pliki PDF powinny mieć strukturę, która pomaga osobom niewidomym przeglądać takie dokumenty.
- Teksty zamieszczone w serwisie powinny być napisane w miarę możliwości w jak najprostszy sposób, aby dostęp do nich miały także osoby mniej wykształcone oraz osoby z upośledzeniem intelektualnym.
- Teksty powinny być opublikowane w czytelny sposób — podzielone na paragrafy, listy i inne sekcje, nieustawiane do prawej strony. Skróty literowe powinny być rozwinięte w pierwszym wystąpieniu na każdej stronie. Tekst powinien być uzupełniony o nagłówki (H1 – H6), aby osoby niewidome mogły sprawnie przejść do interesującej je sekcji.
- Nawigacja (menu) powinna być spójna, logiczna i niezmienna w obrębie serwisu. Nawigacja w obrębie całego serwisu powinna być dostępna z poziomu klawiatury.
- 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**). Zaleca się wzmacnianie domyślnego fokusa tak, aby był dobrze widoczny także dla osób niedowidzących.

- Wszystkie odnośniki powinny być unikalne i zrozumiałe, także poza kontekstem. Nie należy używać linków w postaci: „>” czy „więcej” albo „kliknij tutaj”. Odnośniki nie mogą otwierać się w nowym oknie lub zakładce przeglądarki bez ostrzeżenia.
- Zaleca się zastosowanie usprawnienia w postaci „skip links”, czyli możliwości przejścia bezpośrednio do treści pojedynczej strony. Jest to szczególnie ważne w serwisach, które mają kilkadziesiąt linków w nawigacji czy menu głównym.
- Kontrast kolorystyczny wszystkich elementów przekazujących treść (tekstów, linków, banerów) lub elementów funkcjonalnych musi mieć stosunek jasności tekstu do tła co najmniej 4,5 do 1, a najlepiej, jeśli nie jest on mniejszy niż 7 do 1.
- Stronę powinno dać się znacząco (o co najmniej 200%) powiększyć narzędziami przeglądarki. Najlepiej, jeśli wówczas strona cały czas mieści się poziomo w oknie przeglądarki i nie pokazuje się poziomy pasek przewijania ekranu. Powiększona strona nie może „gubić” treści.
- Wszystkie tytuły (*title*) stron muszą być unikalne i informować o treści podstrony, na jakiej znajduje się użytkownik. Układ treści w tytule powinien być zbudowany według schematu: [Tytuł podstrony] — [Nazwa instytucji].
- Wszystkie podstrony powinny być oparte na nagłówkach. Nagłówki (H1 – H6) są podstawowym sposobem porządkowania treści na stronie. Nagłówek H1 powinien być tytułem tekstu głównego na stronie.
- Do porządkowania treści w tekstach i elementów nawigacji należy wykorzystywać listy nieuporządkowane i uporządkowane.
- Język strony oraz język fragmentów obcojęzycznych powinien być określony atrybutem lang.
- Cytaty powinny być odpowiednio wyróżnione — co najmniej cudzysłowami.
- Kod serwisu powinien być zgodny ze standardami i nie powinien korzystać z tabel jako elementu konstrukcyjnego strony.
- Tabele służące do przekazywania danych powinny być zbudowane w możliwie najprostszy sposób i zawierać nagłówki.
- Wszystkie ramki powinny być odpowiednio zatytułowane.
- Wszystkie skrypty i apety powinny być dostępne dla osób niewidomych i osób korzystających wyłącznie z klawiatury.
- Formularze, w tym formularz wyszukiwarki, powinny być zbudowane zgodnie ze standardami. Wszystkie pola formularzy i przyciski powinny być właściwie opisane.
- Serwis powinien być dostępny w przeglądarkach i urządzeniach z wyłączoną obsługą CSS.

Wersja standardu WCAG 2.0 została wprowadzona w roku 2008. Od tego czasu zmienił się sposób projektowania stron internetowych, w związku z czym niektóre proponowane w standardzie rozwiązania są przestarzałe.

W roku 2018 została wprowadzona aktualizacja standardu do wersji WCAG 2.1. Wersja ta reguluje m.in. dostępność stron na urządzeniach mobilnych. Inne wprowadzone

zasady określają odstępy w treści, możliwość włączenia lub wyłączenia skrótów klawiszowych, informowanie o czasie i występowaniu sesji.

Podane zasady określają cztery podstawowe kryteria, zgodnie z którymi osoby o różnych rodzajach niepełnosprawności będą mogły dotrzeć do prezentowanych treści. Są to:

**Funkcjonalność** — wymagane jest, aby prezentowana informacja była w pełni dostępna za pomocą klasycznych metod, ale również odczytywana w całości za pomocą możliwie największej liczby narzędzi wspomagających.

**Kompatybilność** — sposób prezentowania informacji powinien być dobrany tak, by umożliwiał współpracę z różnymi urządzeniami za pomocą rozmaitych narzędzi wspierających.

**Postrzegalność** — informacje powinny być prezentowane w sposób umożliwiający ich postrzeganie za pomocą różnych zmysłów.

**Zrozumiałość** — należy zapewnić, aby wszelkie prezentowane informacje, a także sposób dotarcia do nich, były zrozumiałe dla każdego użytkownika.

## 5.7. Pytania i zadania

### 5.7.1. Pytania

1. Jakich zasad należy przestrzegać podczas tworzenia projektu graficznego strony internetowej?
2. Jakiego rodzaju elementy graficzne powinny się znaleźć na projektowanej stronie?
3. Do czego podczas tworzenia projektu graficznego strony internetowej można wykorzystać program Kuler?

### Pytania egzaminacyjne

#### Zadanie 1.

Projektowanie logicznego układu witryny polega na:

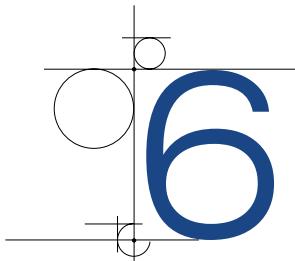
- A. zdefiniowaniu treści witryny.
- B. opracowaniu zestawu grafik dla witryny.
- C. ustaleniu adresów URL dla podstron witryny.
- D. rozmieszczeniu elementów w konkretnych miejscach witryny.

### 5.7.2. Zadania

#### Zadanie 1.

Członkowie grupy tworzącej projekt opracują projekt graficzny strony głównej dla witryny zawierającej informacje na temat szkoły oraz projekt graficzny, który będzie

obowiązywał na pozostałych stronach witryny. Zostanie zaprojektowana typografia strony. Zostanie ona uzupełniona elementami graficznymi typu ikony, zdjęcia i przyciski. Pracujący nad projektem, korzystając z poznanych narzędzi, dopasują kolorystykę strony do umieszczonych na niej elementów graficznych.



# Animacja na stronie internetowej

Animacja komputerowa wiąże się z tworzeniem ruchomych obrazów przy wykorzystaniu komputera. Jest to połączenie grafiki i zaprogramowanego ruchu.

Przy użyciu narzędzi, które są dostępne na komputerze, tworzone są obiekty nieznacznie różniące się od siebie. Jeżeli zostaną one wyświetcone jeden po drugim dostatecznie szybko, to będziemy mieć wrażenie, że obraz jest ruchomy. Aby animacja była płynna, obrazy powinny się zmieniać z odpowiednią częstotliwością, liczoną w klatkach na sekundę — **fps** (ang. *frames per second*). Liczba klatek dająca wrażenie płynnego ruchu to 18–24 na sekundę. W filmach standardem jest 24 fps. W animacjach komputerowych przyjmuje się 25–35 klatek na sekundę.

## 6.1. Metody tworzenia animacji

Animacja obiektu może dotyczyć jego położenia, kształtu i rozmiaru oraz barwy i tekstury. W tworzeniu animacji komputerowych stosowane są różne metody. Są to:

- **Animacja poklatkowa** — kolejne klatki animacji tworzone są po kolei. Każda klatka zawiera pełny obraz. Jest to metoda czasochłonna i mało efektywna.
- **Animacja z zastosowaniem klatek kluczowych** — tworzone są wybrane klatki, pozostałe są interpolowane przez komputer na podstawie ramek kluczowych.
- **Skrypty** — opisują dynamiczną zmianę właściwości obiektów, np. położenia czy wyglądu, w wyniku zaistniałego zdarzenia.

### 6.1.1. Sposoby tworzenia animacji

#### ActionScript

ActionScript to obiektowy język programowania wykorzystywany do sterowania animacjami Flash. Kod jest tworzony w trakcie projektowania animacji i zapisywany

w pliku razem z animacją. Zwykle wykorzystywany jest przy zdarzeniach związanych z myszą (takich jak kliknięcie, najechanie kursorem, usunięcie kurSORA). ActionScript nadaje animacjom interaktywność.

## Animowany GIF

Animowany GIF to seria połączonych obrazów zapisana w pliku GIF. Po uruchomieniu takiego pliku obrazy są wyświetlane kolejno jeden po drugim, co tworzy animację. Do tworzenia animowanych GIF-ów wykorzystywana jest metoda animacji poklatkowej.

## Animacje HTML5 i CSS3

Coraz częściej elementy animowane na stronie internetowej są tworzone za pomocą znaczników języka HTML5 oraz właściwości CSS3. Dzięki nim można tworzyć animowane, interaktywne treści, które działają na różnych urządzeniach.

## 6.1.2. Narzędzia do tworzenia animacji

Istnieje wiele aplikacji, które mogą zostać wykorzystane do tworzenia animacji. Są to m.in.: Adobe Animate, Tumult Hype, Google Web Designer.

### Adobe Animate

**Adobe Animate** to narzędzie do tworzenia rozbudowanych animacji zarówno na potrzeby stron internetowych, jak i funkcjonujących samodzielnie. Pozwala na importowanie dokumentów HTML i grafik w formatach: SVG, PNG, JPG i GIF. Za pomocą stylów CSS3 pozwala na zmianę ich wyglądu oraz animowanie tworzonych obiektów. Utworzone animacje działają w większości popularnych przeglądarek i na urządzeniach mobilnych.

### Tumult Hype

**Tumult Hype** to narzędzie, za pomocą którego można tworzyć animacje i interaktywną zawartość. Działa na komputerach stacjonarnych, smartfonach i iPadach.

### Google Web Designer

**Google Web Designer** to narzędzie do projektowania i tworzenia animacji na strony internetowe. Z jego pomocą możliwe jest tworzenie głównie reklam i banerów umieszczanych na stronach internetowych.

Program posiada cały pakiet narzędzi do tworzenia animowanych i interaktywnych elementów na strony internetowe. Są to m.in. funkcje do zarządzania kolorami, manipulowania właściwościami poszczególnych elementów, jak również dostępnymi komponentami, zdarzeniami i arkuszami stylów CSS.

Graficzny edytor z osią czasu pozwala na tworzenie rozbudowanych animacji. Za pomocą edytora tekstuowego można dopracowywać tworzony projekt. Bazuje na mechanizmach stworzonych w technologiach: HTML, JS, CSS oraz XML.

## 6.1.3. Formaty plików animowanych

### Format GIF

Format **GIF** (ang. *Graphics Interchange Format*) to format umożliwiający zapis grafiki z kompresją bezstratną pozwalającą zminimalizować rozmiar pliku. Format ten zachowuje przezroczystość obrazów oraz pozwala na utworzenie animacji poklatkowej. Jest stosowany na stronach internetowych do zapisywania obrazów, takich jak ikony, przyciski, logo, ale pozwala również na tworzenie prostych animacji.

### Format SVG

Format **SVG** (ang. *Scalable Vector Graphics*) to otwarty standard W3C przeznaczony do zapisu statycznej i animowanej dwuwymiarowej grafiki wektorowej. Powstał z myślą o wykorzystaniu na stronach WWW oraz jako niezależny format grafiki wektorowej. Jest stosowany do tworzenia grafiki wektorowej, animacji oraz prezentacji.

## 6.1.4. Podstawowe formy animacji

Strona internetowa powinna przyciągać uwagę użytkownika, ale nadmierna liczba efektów specjalnych czy animacji często przeszkadza w szukaniu konkretnych informacji. Animacje „ożywiają” strony internetowe, ale należy je tworzyć tak, aby były integralną częścią witryny.

### Baner reklamowy

Animowany baner reklamowy jest bardzo dobrą formą reklamy w internecie. Zaprojektowanie przyciągającego uwagę banera nie jest łatwe, ponieważ taki element musi spełniać wiele wymagań. Powinien być niewielki, przyciągać wzrok i budzić zaciekawienie. Metod na wzbudzenie zainteresowania jest wiele — np. dobranie odpowiednich kolorów i ciekawych tekstów, efektowna animowana grafika. Ważną cechą takiej animacji jest jej dynamika. Animacje i banery tworzone za pomocą HTML5 i CSS3 to nowy standard przygotowywania materiałów reklamowych na stronę internetową.

### Top na stronę

Animowany top tworzony na stronę internetową spełnia głównie funkcje reprezentacyjną i ozdabiającą. Tego typu animacja powinna być dostosowana do strony i tworzyć z nią graficzną całość.

### Animowane przyciski i menu

Animowane przyciski to jedna z form tworzenia przycisków związanych z interakcją. W momencie gdy umieścimy kurSOR myszy na takim przycisku, przycisk zmienia swój wygląd, a jego kliknięcie wywołuje jakieś działanie. Na stronie mogą się pojawić pojedyncze przyciski lub przyciski będące elementami menu nawigacyjnego.

## Animowane mapy

Przykładem takiej mapy jest animowana mapka Polski. Najechanie kursorem na obszar województwa powoduje zmianę koloru, a kliknięcie takiego obszaru — powiększenie województwa. Dla każdego województwa można dołączyć otwieranie wybranej strony. To animacja prosta graficznie i często używana na różnych stronach.



## 6.2. Tworzenie animacji GIF

Animowane GIF-y to najczęściej banery reklamowe, które pojawiają się na stronach internetowych. Jednym z programów, który umożliwia ich wykonywanie, jest GIMP. Tworzenie animacji polega na budowaniu kolejnych warstw. Każdy element animacji powinien się znaleźć na oddzielnej warstwie.

### 6.2.1. Tworzenie animacji w programie GIMP

Program GIMP służy głównie do tworzenia grafiki, ale może zostać użyty do tworzenia prostej animacji metodą animacji poklatkowej.

W programie tym utworzony zostanie baner reklamowy zawierający animowany napis. Będzie to napis *Czwarta rewolucja przemysłowa 4.0*. Słowa występujące w napisie będą się kolejno pojawiały podczas jego wyświetlania.

#### Ćwiczenie 6.1

Tworzymy nowy obrazek o wymiarach planowanego banera, np. 700×80 pikseli (rysunek 6.1).



**Rysunek 6.1.** Tworzenie nowego dokumentu w programie GIMP

Kolejne warstwy obrazka będą kolejnymi klatkami animacji. Warstwa *Tło* będzie pierwszą warstwą, na której zostaną umieszczone następne. Każda z warstw będzie kolejną klatką tworzonego banera. Animacja będzie się składała z czterech klatek.

Ustawiamy kolor, rozmiar i styl czcionki. Wstawiamy napis *Czwarta* (rysunek 6.2). Zostanie utworzona nowa warstwa ze wstawionym tekstem.



**Rysunek 6.2.** Wstawianie napisu do animacji

Kolejne warstwy najlepiej tworzyć przez duplikowanie poprzedniej, aby zachować wszystkie warstwy tekstu na tej samej wysokości.

W panelu *Warstwa* po zaznaczeniu warstwy z tekstem wybieramy ikonę *Tworzy kopię warstwy i dodaje ją do obrazu*. Zaznaczamy utworzoną warstwę i za pomocą narzędzia *Przesunięcie* przesuwamy ją w prawo, używając klawisza strzałki. Następnie wybieramy narzędzie *Tekst*, zaznaczamy napis i zmieniamy treść oraz kolor tekstu. Tworzymy kolejne dwie warstwy i postępujemy podobnie, zmieniając tekst oraz jego parametry (rysunek 6.3). Możemy podejrzeć utworzoną animację poprzez wybranie *Filtры/Animacja/Odtwarzanie*.

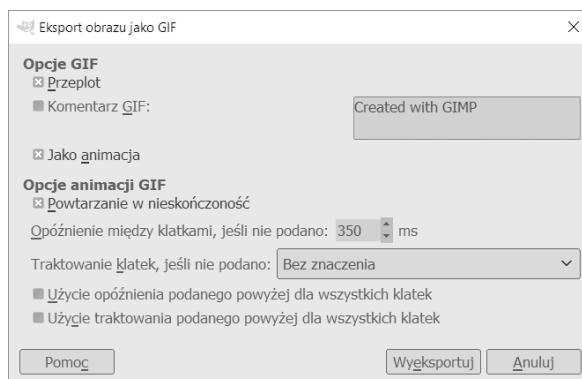


**Rysunek 6.3.** Napis umieszczony na kolejnych warstwach

Jeżeli animacja została poprawnie przygotowana, można ją zapisać, wybierając *Plik/Wyeksportuj....* W otwartym oknie należy wybrać jako typ pliku *Obraz GIF* i kliknąć przycisk *Wyeksportuj*. Zostanie otwarte okno (rysunek 6.4), w którym możemy ustawić m.in. *Przeplot*, sposób zapętlenia oraz opóźnienie między kolejnymi klatkami.

**Rysunek 6.4.**

Opcje zapisu pliku z animacją



Należy pamiętać, że utworzony plik musi mieć rozszerzenie *.gif*.

Po utworzeniu wszystkich potrzebnych klatek można określić czas ich wyświetlania oraz sposób ich pojawiania się (czy zawartość klatki ma być dodawana do poprzedniej, czy ma ją zastępować).

### Ustawianie parametrów animacji

Czas animacji podawany jest w milisekundach (*ms*). 100 ms to 0,1 sekundy, a 1000 ms to 1 sekunda.

Sposób pojawiania się klatek określa wpis:

- *replace* — zastępowanie poprzedniej klatki bieżącą,
- *combine* — dołączanie bieżącej klatki do poprzedniej.

Podane parametry animacji należy wpisać w nawiasach za nazwą każdej warstwy.

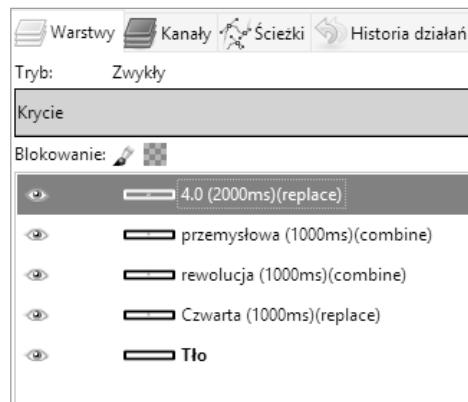
W tworzonym przykładzie parametry klatek określamy w następujący sposób:

- cztery pierwsze klatki będą wyświetlane przez 1 sekundę i będą do siebie dodawane;
- ostatnia klatka będzie wyświetdana przez 2 sekundy i będzie zastępować wszystkie wcześniejsze klatki.

Pierwsza klatka, nawet gdy tego nie zdefiniujemy, zawsze zastępuje ostatnią klatkę.

Kod, który należy umieścić obok każdej warstwy, ma następującą postać (rysunek 6.5):

```
Klatka1(1000ms) (replace),
Klatka2(1000ms) (combine),
Klatka3(1000ms) (combine),
Klatka4(1000ms) (combine),
Klatka5(2000ms) (replace).
```



**Rysunek 6.5.**

Warstwy ze zdefiniowanymi parametrami animacji

### Ćwiczenie 6.2

Kolejny animowany tekst będzie banerem reklamowym, w którym litery będą się pojawiały po kolej, sprawiając wrażenie automatycznie tworzonego napisu. Tekstem, na którym przecwickujemy wykonanie banera, będzie napis *Informatyka*.

W programie GIMP tworzymy nowy obraz (*Plik/Nowy*). Ustalamy jego szerokość na 600 pikseli, a wysokość na 120 pikseli. Następnie tworzymy tło i określamy jego kolor.

Przechodzimy do tworzenia animacji. Jeżeli chcemy stworzyć animowany napis w danej postaci, to każda litera tekstu musi się znaleźć na oddzielnej warstwie. Chcąc animować napis *Informatyka*, musimy utworzyć 11 warstw.

Wybieramy narzędzie **Tekst** i we właściwościach ustawiamy np. czcionkę **Magneto Bold Semi-Expanded** o rozmiarze 75 pikseli, koloru np. czarnego.

Dla każdej litery tworzymy oddzielną warstwę. Dla ułatwienia zadania kolejne warstwy z literą można tworzyć przez duplikowanie warstwy poprzedniej (**Warstwa/Powiel warstwę**) i przesunięcie jej w prawo za pomocą klawisza strzałki.

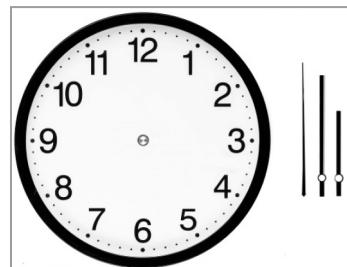
Po takim przygotowaniu tekstu można podejrzeć animację, wybierając **Filtры/Animacja/Odtwarzanie**.

Jeżeli utworzona animacja nam odpowiada, można ją zapisać. Wybieramy **Plik/Wyeksportuj** i podajemy nazwę pliku z rozszerzeniem **.gif**, np. **animacja.gif**. Klikamy przycisk **Wyeksportuj** i w otwartym oknie zaznaczamy opcję **Jako animacja**. Domyślnie powinna być włączona opcja **Powtarzanie w nieskończoność** (inaczej animacja zostanie wykonana tylko raz). Opcja **Opóźnienie między klatkami** oznacza czas przejścia z jednej klatki do drugiej, mierzony w milisekundach. W tworzonej animacji ustawimy ten czas na 300 ms. Po ustawieniu opcji wciskamy przycisk **Wyeksportuj**.

Plik z animowanym GIF-em można uruchomić w dowolnej przeglądarce internetowej, klikając go prawym przyciskiem myszy i wybierając opcję **Otwórz za pomocą**.

### Ćwiczenie 6.3

Wykonaj animację grafiki pokazanej na rysunku 6.6. Przygotuj grafikę przedstawiającą zegar bez wskazówek oraz grafiki ze wskazówkami zegara. Utwórz animację, która pokaże ruch wskazówek na zegarze.



**Rysunek 6.6.**

Zegar jako tło animowanych wskazówek

## 6.3. Tworzenie animacji za pomocą CSS i HTML

Elementy animowane na stronie internetowej coraz częściej są tworzone za pomocą znaczników języka HTML5 oraz właściwości CSS3. Dzięki zastosowaniu tych narzędzi tworzone animowane i interaktywne treści mogą funkcjonować na różnych urządzeniach.

Animacje tworzone za pomocą języka CSS można podzielić — w zależności od tego, w jaki sposób powstają — na:

- animacje powstałe przez płynne zmiany właściwości elementów tworzone za pomocą właściwości `transition`;
- animacje poklatkowe tworzone za pomocą właściwości `animation`.

### 6.3.1. Transformacje CSS

Wraz z rozwojem języka CSS pojawiły się właściwości, które wymagają podania nazwy metody oraz jej parametrów. Właściwość zdefiniowana za pomocą metody i jej parametrów wpłynie na zachowanie się elementu strony internetowej. Do właściwości definiowanych przy użyciu metod należy m.in. właściwość `transform`, która może zostać wykorzystana do animowania treści strony poprzez zmianę wyglądu elementu.

Transformacje w języku CSS umożliwiają elementowi strony internetowej zmianę kształtu, wielkości i położenia w przestrzeni 2D lub 3D.

#### Transformacje 2D

Do zdefiniowania transformacji elementu strony internetowej w przestrzeni 2D używana jest właściwość `transform` zapisana w postaci:

```
transform: none | metoda transformacji | initial | inherit;
```

gdzie

- `none` — brak transformacji;
- `initial` — ustawia wartość właściwości na wartość domyślną;
- `inherit` — wartość właściwości jest dziedziczona z elementu rodzica.

#### **UWAGA**

##### **inherit**

Jeżeli umieścimy jeden element HTML w drugim, to będą one tworzyły hierarchię typu element „rodzic” i element „dziecko”. Niektóre właściwości CSS mogą być automatycznie dziedziczone od elementu „rodzica” przez kolejnych jego „potomków” (np. właściwość `color`). Dla elementów HTML istnieją takie właściwości, które nie mogą być automatycznie dziedziczone (np. właściwość `border`). Dzięki właściwości `inherit` możemy „wymusić” dziedziczenie tych właściwości. W takim przypadku dziedziczenie dotyczy tylko „dzieci” danego elementu, a nie wszystkich jego „potomków”.

Na przykład:

```
#dziecko {border:inherit; margin-bottom:2px; padding:1px;
background-color:#c0c0c0}
```

Do niedawna żadna z przeglądarek internetowych nie obsługiwała właściwości `transform` w postaci źródłowej. Każda z nich wymagała użycia innego prefiksu dla podanej właściwości:

- *Chrome, Safari* — prefiksu `-webkit-`,
- *Firefox* — prefiksu `-moz-`,
- *Opera* — prefiksu `-o-`,
- *IE* — prefiksu `-ms-`.

Dla właściwości `transform` można używać następujących metod:

### **rotate()**

Metoda `rotate()` służy do obrócenia elementu strony. Parametrem tej metody jest wartość określająca, o ile stopni należy dany element obrócić. Stosowaną jednostką jest `deg`.

#### **Przykład 6.1**

```
-ms-transform: rotate(30deg); /*IE*/  
-webkit-transform: rotate(30deg); /*Chrome*/  
transform: rotate(30deg);
```

### **skew()**

Metoda `skew()` służy do pochylenia elementu o określzoną liczbę stopni. W metodzie tej występują dwa parametry, określające odchylenie w stosunku do osi `x` (pierwszy parametr) oraz odchylenie w stosunku do osi `y` (drugi parametr). Metoda może wystąpić w postaci `skewX()` i wtedy pochyla element tylko wzdłuż osi `x` oraz w postaci `skewY()` i wtedy pochyla element tylko wzdłuż osi `y`.

#### **Przykład 6.2**

```
-ms-transform:skew(15deg,20deg);  
-webkit-transform:skew(15deg,20deg);  
transform:skew(15deg,20deg);
```

### **translate()**

Metoda `translate()` przesuwa element z bieżącej pozycji według podanych parametrów dla osi `x` i osi `y`.

#### **Przykład 6.3**

```
-ms-transform:translate(50px, 100px);  
-webkit-transform:translate(50px, 100px);  
transform:translate(50px, 100px);
```

### **scale()**

Metoda `scale()` zmniejsza lub zwiększa rozmiar elementu w zależności od podanych parametrów.

#### **Przykład 6.4**

```
-ms-transform:scale(3, 2);  
-webkit-transform:scale(3, 2);  
transform:scale(3, 2);
```

## matrix()

Metoda `matrix()` łączy wszystkie metody transformacji 2D. Oparta została na macierzy sześcioelementowej zawierającej parametry obracania, skalowania, przenoszenia i pochylenia elementu. Są to następujące parametry:

```
matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY())
```

## Ćwiczenie 6.4

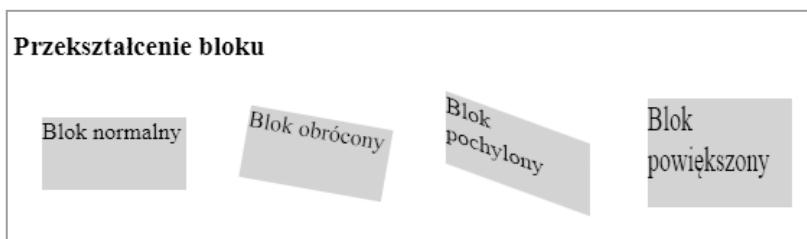
Poznane metody właściwości `transform` zostaną zastosowane do przekształcenia zdefiniowanego w języku HTML znacznika `<div>`.

```
<!DOCTYPE html>

<html lang="pl-PL">
<head>
<title>Przekształcanie bloku</title>
<meta charset="utf-8">
<style>
div.blok_n {
    width: 100px;
    height: 50px;
    background-color: #c0c0c0;
    margin: 20px;
    float:left
}
div.blok1 {
    width: 100px;
    height: 50px;
    background-color: #c0c0c0;
    -ms-transform: rotate(10deg);
    -webkit-transform: rotate(10deg);
    transform: rotate(10deg);
    margin: 20px;
    float:left
}
div.blok2 {
    width: 100px;
    height: 50px;
```

```
background-color: #c0c0c0;  
-ms-transform: skewY(20deg);  
-webkit-transform: skewY(20deg);  
transform: skewY(20deg);  
margin: 20px;  
float:left  
}  
  
div.blok3 {  
width: 100px;  
height: 50px;  
background-color: #c0c0c0;  
-ms-transform: scaleY(1.5);  
-webkit-transform: scaleY(1.5);  
transform: scaleY(1.5);  
margin: 20px;  
float:left  
}  
  
</style>  
</head>  
<body>  
<h3>Przekształcenie bloku</h3>  
<div class="blok_n">Blok normalny</div>  
<div class="blok1">Blok obrócony</div>  
<div class="blok2">Blok pochylony</div>  
<div class="blok3">Blok powiększony</div>  
</body>  
</html>
```

Wynik działania kodu został pokazany na rysunku 6.7.



**Rysunek 6.7.** Przekształcenia elementu blokowego strony internetowej

## 6.3.2. Animacje transition

Kolejną właściwością dodaną do języka CSS jest `transition`, która pozwala na utworzenie na stronie efektu animacji.

Właściwość ta umożliwia zmianę wartości niektórych właściwości CSS dla danego elementu HTML. W efekcie następuje przejście jednych wartości właściwości stylów CSS w inne (np. białe tło elementu zmieni się na niebieskie). Z tego powodu właściwość `transition` została nazwana efektem przejścia. Tak zdefiniowana animacja jest wykonywana tylko jeden raz.

Aby uzyskać efekt przejścia, należy dla zdefiniowanego elementu HTML utworzyć dodatkowy wygląd oraz określić sposób aktywowania tego wyglądu. Najczęściej są do tego stosowane selektory `:hover` i `:focus`.

Podana właściwość `transition` jest skróconą wersją czterech właściwości:

- `transition-property` — określa nazwę właściwości, dla której definiowane jest przejście;
- `transition-duration` — określa, ile sekund lub milisekund powinien trwać efekt przejścia;
- `transition-timing-function` — określa prędkość efektu przejścia;
- `transition-delay` — określa, kiedy rozpocznie się efekt przejścia.

Właściwość `transition-duration` powinna zawsze być określona. Jej brak oznacza czas trwania przejścia równy 0 i wówczas przejście nie będzie miało żadnego efektu.

### Przykład 6.5

```
transition-property: width;  
transition-duration: 2s;  
transition-timing-function: linear;  
transition-delay: 1s;
```

W podanym przykładzie ustawienie `transition-property: width;` określa właściwość, której zmiana ma zostać wykonana. Jeżeli występuje więcej właściwości, oddziela my je przecinkiem lub używamy określenia `all`. Ustawienie `transition-duration: 2s;` określa, jak długo powinna trwać zmiana. Ustawienie `transition-timing-function: linear;` określa sposób animacji. W tym przypadku jest to animacja liniowa. Można jeszcze wybrać `ease`, `ease-in`, `ease-out`, `ease-in-out`, `cubic-bezier`. Ustawienie `transition-delay: 1s;` określa opóźnienie efektu przejścia.

Ten sam rezultat uzyskamy, stosując właściwość `transition` w skróconej wersji:

### Przykład 6.6

```
transition: width 2s linear 1s;
```

Utworzmy znacznik <div> oraz zdefiniujemy dla niego za pomocą stylów CSS wygląd podstawowy i wygląd dodatkowy. Wygląd dodatkowy zostanie aktywowany, gdy wystąpi określone zdarzenie. W naszym przykładzie zdarzeniem będzie ustawienie kurSORA myszy nad elementem (selektor :hover).

Styl podstawowy dla znacznika <div>:

### Przykład 6.7

```
div {  
    width: 35%;  
    height: 60px;  
    background-color: white;  
    padding: 10px;  
    font-size: 1.2em;  
    border: 3px solid #000;  
}
```

Styl dla znacznika <div> po ustawieniu nad nim kurSora myszy:

### Przykład 6.8

```
div:hover {  
    width: 80%;  
    background-color: blue;  
}
```

Dla tak zdefiniowanych stylów można określić sposób przejścia z jednego wyglądu w drugi.

### Ćwiczenie 6.5

Zastosuj zdefiniowane w przykładzie style do pokazania ich działania dla utworzonego za pomocą języka HTML znacznika <div>.

### Rozwiązanie

```
<!DOCTYPE html>  
<html lang="pl-PL">  
<head>  
    <title>Zmiana wyglądu bloku</title>  
    <meta charset="utf-8">  
    <style>  
        div {  
            width: 100px;  
            height: 60px;
```

```
background-color: #c0c0c0;
transition-duration: 2s;
}

div:hover {
width: 300px;
background-color: #f0c0c0;
}

</style>

</head>

<body>
<h3>Zmiana wygladu bloku</h3>
<p>Ustaw kursor myszy nad elementem</p>
<div></div>
</body>
</html>
```

W wyniku wykonania kodu zostanie wyświetlony kwadrat w kolorze szarym. Ustawienie kurSORA myszy nad tym elementem spowoduje zwiększenie jego rozmiaru (width: 300px) oraz zmianę koloru (background-color: #f0c0c0).

Innym przykładem zastosowania właściwości transition jest zmiana wyglądu przycisku poleceń po ustawieniu nad nim kurSORA myszy.

## Ćwiczenie 6.6

W języku HTML utworzony zostanie przycisk poleceń. Poprzez ustawienie nad nim kurSORA myszy spowodujemy zmianę jego wyglądu (selektor :hover). W wyniku odsunięcia kurSORA myszy nastąpi powrót do poprzedniego stanu, a kliknięcie przycisku (selektor :focus) spowoduje zachowanie zmienionej postaci przycisku aż do wybrania innego elementu strony.

## Rozwiązanie

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
<title>Animowany przycisk</title>
<meta charset="utf-8">
<style>
#blok {
display: inline-block;
```

```
padding: 15px 20px;  
border: 0;  
border-radius: 5px;  
background: #338171;  
color: #ffffff;  
font-family: sans-serif;  
font-size: 16px;  
box-shadow: 0 0 0 0 #03211b;  
transition: 0.5s background, 1s box-shadow;  
}  
  
#blok:hover, #blok:focus {  
background: #af1bd;  
box-shadow: 0 10px 0 0 #0aa687;  
}  
  
</style>  
</head>  
<body>  
  <button type="button" id="blok">  
    Ustaw kurSOR myszy lub kliknij  
  </button>  
</body>  
</html>
```

W podanym rozwiązaniu za pomocą poleceń języka HTML utworzony został przycisk (`button type="button" id="blok"`). Przy użyciu stylów CSS został określony jego wygląd. Zastosowana właściwość `border-radius` nadaje obramowaniu przycisku zaokrąglone rogi. Właściwość `box-shadow` definiuje dla elementu cień. Początkowo cień jest niewidoczny (`box-shadow: 0 0 0 0 #03211b;`). Po ustawnieniu kurSora myszy lub kliknięciu elementu cień się pojawi (`box-shadow: 0 10px 0 0 #0aa687;`), a po usunięciu myszy lub odznaczeniu elementu cień znika.

Kolejnym przykładem animacji z zastosowaniem właściwości `transition` jest użycie właściwości `transform` do zdefiniowania zmiany wyglądu elementu strony internetowej.

## Ćwiczenie 6.7

W języku HTML utworzone zostaną dwa bloki `<div>`. Po ustawnieniu nad pierwszym z nich kurSora myszy jego rozmiar ulegnie zmianie. Z chwilą odsunięcia kurSora myszy nastąpi powrót do poprzedniego rozmiaru. Po ustawnieniu kurSora myszy nad drugim

blokiem ulegnie on przekształceniu przez pochylenie. W wyniku odsunięcia kursora myszy nastąpi powrót do poprzedniego wyglądu. Zmiany w wyglądzie bloków zostaną zrealizowane za pomocą właściwości transform.

## Rozwiążanie

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
    <title>Przekształcanie bloków</title>
    <meta charset="utf-8">
    <style>
        #b1 {
            width: 100px;
            height: 60px;
            background: #c0c0c0;
            -webkit-transition: 1s ease-in-out;
            -moz-transition: 1s ease-in-out;
            -o-transition: 1s ease-in-out;
            transition: 1s ease-in-out;
        }
        #b1:hover {
            -ms-transform: scale(3, 1)
            -webkit-transform: scale(3, 1)
            transform: scale(3, 1)
        }
        #b2 {
            width: 120px;
            height: 80px;
            background: #f0c0c0;
            -webkit-transition: 1s ease-in-out;
            -moz-transition: 1s ease-in-out;
            -o-transition: 1s ease-in-out;
            transition: 1s ease-in-out;
        }
        #b2:hover {
            -ms-transform: skewY(20deg)
            -webkit-transform: skewY(20deg)
        }
    </style>
</head>
<body>
    <div id="b1"></div>
    <div id="b2"></div>
</body>
</html>
```

```

        transform: skewY(20deg)
    }
</style>
</head>
<body>
    <h3>Zmiana rozmiaru bloku</h3>
    <p>Ustaw kursor myszy nad elementem</p>
    <div id="b1"></div>
    <h3>Pochylenie bloku</h3>
    <p>Ustaw kursor myszy nad elementem</p>
    <div id="b2"></div>
</body>
</html>

```

W podanym rozwiązaniu za pomocą polecień języka HTML utworzone zostały dwa bloki: b1 i b2. Przy użyciu stylów CSS zdefiniowany został wygląd tych bloków (`#b1{...}` i `#b2{...}`) oraz sposób ich animowania (`transition:`). Dodatkowo zdefiniowany został ich wygląd (`transform: scale(3, 1)` i `transform: skewY(20deg)`) po zaistnieniu zdarzeń (`#b1:hover` i `#b2:hover`).

### 6.3.3. Animacje animation

Tworząc animacje za pomocą właściwości `animation`, mamy dużo większe możliwości, ponieważ definiowanie zmian wartości właściwości możemy w dowolnym momencie animacji rozpocząć i w dowolnym jej momencie zakończyć.

Tworząc animację za pomocą właściwości `animation`, musimy zdefiniować parametry animacji oraz określić przy użyciu reguły `@keyframes` selektory poszczególnych klatek animacji elementu oraz przypisane tym klatkom style.

#### Właściwość `animation`

Właściwość `animation`, podobnie jak `transition`, jest skróconą wersją innych właściwości.

##### **animation-duration**

Właściwość `animation-duration` określa czas trwania animacji:

`animation-duration: czas| initial| inherit;`

gdzie

- `initial` — ustawia wartość właściwości na wartość domyślną;
- `inherit` — wartość właściwości jest dziedziczona z elementu rodzica.

## animation-timing-function

Właściwość `animation-timing-function` określa tempo animacji. Wartościami tej właściwości mogą być:

- `linear` — animacja ma taką samą prędkość od początku do końca;
- `ease` — animacja zaczyna się powoli, potem przyspiesza, a na końcu zwalnia (wartość domyślna);
- `ease-in` — animacja zaczyna się powoli;
- `ease-out` — animacja kończy się powoli;
- `ease-in-out` — animacja zaczyna się powoli i kończy się powoli;
- `cubic-bezier(x, x, x, x)` — własna krzywa prędkości animacji z krzywej Béziera.

## animation-delay

Właściwość `animation-delay` określa opóźnienie rozpoczęcia animacji:

```
animation-delay: czas | initial | inherit;
```

## animation-iteration-count

Właściwość `animation-iteration-count` określa liczbę powtórzeń animacji:

```
animation-iteration-count: liczba | infinite | initial | inherit;
```

gdzie

- `liczba` — określa liczbę powtórzeń animacji;
- `infinite` — oznacza odtwarzanie animacji bez końca.

## animation-direction

Właściwość `animation-direction` określa sposób powtarzania się oraz rozpoczęcia animacji:

```
animation-direction: sposób powtarzania animacji | initial | inherit;
```

Wartościami określającymi sposób odtwarzania animacji w kolejnych cyklach mogą być:

- `normal` — animacja jest odtwarzana normalnie (wartość domyślna);
- `reverse` — animacja jest odtwarzana odwrotnie;
- `alternate` — animacja jest odtwarzana na przemian, normalnie w cyklach nieparzystych i odwrotnie w cyklach parzystych;
- `alternate-reverse` — animacja jest odtwarzana na przemian, odwrotnie w cyklach nieparzystych i normalnie w cyklach parzystych.

## animation-fill-mode

Właściwość `animation-fill-mode` określa dodatkowy wygląd elementu HTML przed wykonaniem animacji lub po jej wykonaniu:

---

```
animation-fill-mode: none| forwards| backwards| both| initial| inherit;
```

Wartościami `animation-fill-mode` są:

- `forwards` — po zakończeniu odtwarzania animacji dla elementu zostaną ustawione wartości właściwości, które obowiązywały przy zakończeniu animacji;
- `backwards` — przed początkiem odtwarzania animacji dla elementu zostaną ustawione wartości, które zostały ustawione dla pierwszej klatki kluczowej animacji;
- `both` — przed początkiem odtwarzania animacji dla elementu zostaną ustawione wartości, które zostały ustawione dla pierwszej klatki kluczowej animacji, a po zakończeniu odtwarzania animacji zostaną ustawione wartości właściwości, które obowiązywały przy zakończeniu animacji.

### **animation-play-state**

Właściwość `animation-play-state` określa zatrzymanie lub wznowienie animacji:

```
animation-play-state: paused| running| initial| inherit;
```

Wartości to:

- `paused` — animacja jest wstrzymana;
- `running` — animacja jest uruchomiona (wartość domyślna).

### **animation-name**

Właściwość `animation-name` określa nazwę animacji dla klatek kluczowych:

```
animation-name: nazwa| none| initial| inherit;
```

Wszystkie podane właściwości animacji można określić za pomocą zbiorczej deklaracji właściwości `animation`:

```
animation: -name -duration -timing-function -delay -interation-count -direction  
-fill-mode -play-state | initial | inherit
```

### **reguła @keyframes**

Aby animacja zadziałała, należy najpierw zdefiniować obiekt i określić dla niego sposób animowania, a następnie określić klatki kluczowe animacji i przypisać tym klatkom odpowiednie style. Reguła `@keyframes` definiuje selektory kolejnych klatek kluczowych animacji oraz przypisuje tym klatkom style.

Przy tworzeniu reguły po słowie kluczowym `@keyframes` należy umieścić nazwę animacji, a następnie w nawiasach klamrowych zapisać reguły animacji za pomocą odpowiednich właściwości stylów CSS. Przed każdą regułą należy umieścić selektor czasu animacji. Jego wartość jest określana w procentach (np. `0%`, `25%`, `50%`, `100%`) lub za pomocą słów kluczowych `from` i `to`, które odpowiadają wartościom procentowym `0%` i `100%`. Selektory czasu animacji należy umieszczać w kolejności rosnącej (od najmniejszej wartości `0%` do największej wartości `100%`). Podczas działania animacji zdefiniowane właściwości elementu są stopniowo zmieniane z bieżącego na nowy styl.

Ogólna postać reguły @keyframes jest następująca:

```
@keyframes nazwa_animacji {keyframes-selector {css-styles;}}
```

gdzie

- **keyframes-selector** — określenie klatek kluczowych animacji w postaci **%** lub słów **from** i **to**; w jednej animacji może wystąpić wiele selektorów klatek kluczowych;
- **css-styles** — jedna lub wiele deklaracji stylów.

### Przykład 6.9

```
@keyframes anim {  
    from {top: 0px;}  
    to {top: 200px;}  
}
```

### Przykład 6.10

```
@keyframes anim {  
    0% {top: 0px;}  
    25% {top: 200px;}  
    50% {top: 100px;}  
    100% {top: 200px;}  
}
```

## 6.3.4. Zastosowania animacji na stronie internetowej

### Animacja tekstu

#### Ćwiczenie 6.8

W języku HTML utworzony zostanie blok **<div>** z tekstem *Animacja komputerowa*. Za pomocą klatek kluczowych dla utworzonego bloku zdefiniujemy animowanie umieszczonego w nim tekstu.

```
<!DOCTYPE html>  
<html lang="pl-PL">  
<head>  
    <title>Animacja komputerowa</title>  
    <meta charset="utf-8">
```

```
<style>
div#box {
    color: #338171;
    font-family: sans-serif;
    font-size: 24px;
    position: relative;
    top: 20px;
    left: 150px;
    width: 300px;
    height: 40px;
    animation: box 10s ease 2s infinite alternate;
    -webkit-animation: box 10s ease 2s infinite alternate;
}
@keyframes box {
    0% {top: 20px; left: 150px;}
    20% {top: 20px; left: 150px;}
    55% {top: 5px; left: 350px;}
    100% {top: 80px; left: 0px;}
}
@-webkit-keyframes box {
    0% {top: 20px; left: 150px;}
    20% {top: 20px; left: 150px;}
    55% {top: 5px; left: 350px;}
    100% {top: 80px; left: 0px;}
}
</style>
</head>
<body>
    <div id="box">Animacja komputerowa</div>
</body>
</html>
```

Zdefiniowana w ćwiczeniu animacja nazywa się *box*. Będzie ona trwała 10s, zacznie się powoli, potem przyspieszy, a na końcu zwolni. Przyspieszenie animacji wyniesie 2s i będzie ona odtwarzana bez końca. Animacja będzie odtwarzana na przemian, normalnie w cyklach nieparzystych i odwrotnie w cyklach parzystych (animation:

box 10s ease 2s infinite alternate). Za pomocą reguły @keyframes zostaną zdefiniowane klatki kluczowe (selektory: 0%, 20%, 55%, 100%) i zostaną określone dla nich style (położenie bloku z tekstem).

## Tworzenie animowanego menu

Aby utworzyć animowane menu, należy zrobić trzy rzeczy. Najpierw trzeba utworzyć strukturę menu w języku HTML.

### Przykład 6.11

```
<ul class="menu">
  <li><a href="#">Start</a></li>
  <li><a href="#">Kursy</a></li>
  <li><a href="#">Artykuły</a></li>
  <li><a href="#">Aktualności</a></li>
  <li><a href="#">Kontakt</a></li>
</ul>
```

Następnie należy dodać style dla znacznika `<ul>` i elementów listy.

### Przykład 6.12

```
.menu {
  padding: 0px;
  margin: 0 0 10px 0;
  position: relative;
}

.menu li {
  font-size: 50px;
  display: block;
}
```

Ostatnim działaniem powinno być zdefiniowanie animacji.

### Przykład 6.13

```
.menu li a {
  display: block;
  -webkit-transition: all 0.2s ease-in-out;
  -moz-transition: all 0.2s ease-in-out;
  -o-transition: all 0.2s ease-in-out;
  -ms-transition: all 0.2s ease-in-out;
```

```
transition: all 0.2s ease-in-out;  
}  
  
.menu li a:hover {  
color: #f0f0f0;  
padding-left: 10px;  
}
```

Po połączeniu wszystkich utworzonych kodów powstanie animowane menu.

### Ćwiczenie 6.9

Utworzmy animowane pionowe menu, które będzie się składało z pięciu pozycji. Wybrana pozycja menu będzie wyraźnie widoczna z cieniem w kolorze niebieskim, a pozostałe pozycje będą lekko rozmazane z cieniem w kolorze białym. Wybrana pozycja menu dodatkowo ulegnie przesunięciu w stosunku do pozostałych (rysunek 6.8).



**Rysunek 6.8.** Menu animowane strony

### Rozwiązanie

```
<!DOCTYPE html>  
  
<html lang="pl-PL">  
  
<head>  
  
    <title>Animowane menu</title>  
  
    <meta charset="utf-8">  
  
    <style>  
body {  
  
background-image: url("a1.jpg");
```

```
}

.menu {
    padding: 0px;
    margin: 0 0 10px 10px;
    position: relative;
}

.menu li {
    font-size: 30px;
    display: block;
}

.menu li a {
    color: #c0c0c0;
    display: block;
    text-transform: uppercase;
    text-shadow: 0px 0px 4px #fff;
    letter-spacing: 1px;
    -webkit-transition: all 0.2s ease-in-out;
    -moz-transition: all 0.2s ease-in-out;
    -o-transition: all 0.2s ease-in-out;
    -ms-transition: all 0.2s ease-in-out;
    transition: all 0.2s ease-in-out;
}

.menu:hover li a {
    text-shadow: 0px 0px 6px #fff;
}

.menu li a:hover {
    color: #f0f0f0;
    text-shadow: 0px 0px 1px #142b49;
    padding-left: 10px;
}

</style>

</head>

<body>

<ul class="menu">
    <li><a href="#">Start</a></li>
```

```

<li><a href="#">Kursy</a></li>
<li><a href="#">Artykuły</a></li>
<li><a href="#">Aktualności</a></li>
<li><a href="#">Kontakt</a></li>
</ul>
</body>
</html>

```

Do przekształcenia tekstu w duże litery użyta została właściwość transformacji tekstu `text-transform: uppercase` (wartość `uppercase` zamienia wszystkie litery na duże). W celu ustalenia dla tekstu cienia zastosowaliśmy właściwość `text-shadow: 0px 0px 4px #ffff` (pierwsze dwie wartości określają przesunięcie cienia w prawo i w dół, kolejna określa jego wielkość, a ostatnia kolor). Właściwość `transition` określa sposób animacji.

## Animacja — wyścigi samochodów

Animacja komputerowa wiąże się z tworzeniem ruchomych obrazów przy wykorzystaniu komputera. Jest to połączenie grafiki i zaprogramowanego ruchu. Za pomocą języków HTML oraz CSS potrafimy przekształcać elementy i zmieniać ich położenie. W kolejnej animacji wykorzystamy poznane narzędzia do przygotowania wyścigów samochodowych.

### Ćwiczenie 6.10

Z pomocą programu graficznego (np. Photoshopa) przygotuj proste rysunki pięciu samochodów lub pobierz odpowiednią grafikę z internetu. Zmień wielkość obrazów, dostosowując ją do rozmiarów  $100 \times 60$  pikseli. Obrazki zapisz w formacie `.png` z wykorzystaniem kanału `alfa`. Przygotowane rysunki zostaną użyte do stworzenia animowanej wersji wyścigów samochodowych (rysunek 6.9).

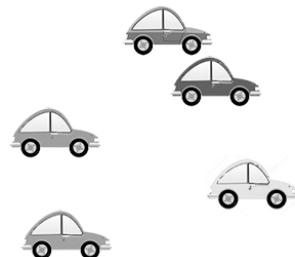
### Rozwiązanie

```

<!DOCTYPE html>
<html lang="pl-PL">
<head>
<title>Wyścigi samochodów</title>
<meta charset="utf-8">
<style>
div {
  width: auto;

```

Wyścigi samochodów



**Rysunek 6.9.**

Wyścig samochodów  
animowany za pomocą  
HTML i CSS

```
height: 50px;
position: relative;
-webkit-animation: sam 10s infinite;
animation: sam 10s infinite;
}

#div1 {-webkit-animation-timing-function: linear;}
#div2 {-webkit-animation-timing-function: ease;}
#div3 {-webkit-animation-timing-function: ease-in;}
#div4 {-webkit-animation-timing-function: ease-out;}
#div5 {-webkit-animation-timing-function: ease-in-out; }

#div1 {animation-timing-function: linear;}
#div2 {animation-timing-function: ease;}
#div3 {animation-timing-function: ease-in;}
#div4 {animation-timing-function: ease-out;}
#div5 {animation-timing-function: ease-in-out; }

@-webkit-keyframes sam {
  from {left: 0px;}
  to {left: 900px;}
}

@keyframes sam {
  from {left: 0px;}
  to {left: 900px;}
}

</style>

</head>
<body>

<h3>Wyścigi samochodów</h3>
<div id="div1"></div>
<div id="div2"></div>
<div id="div3"></div>
<div id="div4"></div>
<div id="div5"></div>

</body>
</html>
```

Przygotowane wcześniej obrazki z samochodami zostały za pomocą kodu HTML umieszczone w znacznikach `<div>` (`<div id="div1"></div>`). Za pomocą stylów CSS został określony wygląd tych znaczników oraz sposób animacji dla pozycji startowej samochodów: `animation: sam 10s infinite` (animacja o nazwie `sam`, trwająca 10 sekund i powtarzana w nieskończoność). Następnie został zdefiniowany oddzielnie ruch dla każdego z samochodów (`#div1 {animation-timing-function: linear;}`) oraz za pomocą reguły `@keyframes` zostały określone klatki kluczowe animacji (`@keyframes sam {from {left: 0px;} to {left: 900px;}}`). Wyścig samochodów będzie się rozpoczętał od pozycji startowej i po dotarciu do mety (900px) wszystkich samochodów rozpocznie się ponownie.

## 6.4. Pytania i zadania

### 6.4.1. Pytania

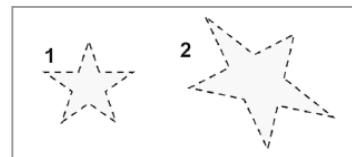
1. Wymień metody stosowane do tworzenia animacji komputerowych.
2. Jakiego rodzaju animacje mogą się znaleźć na stronie internetowej?
3. Jakie zastosowanie ma kanał alfa w grafice komputerowej i animacji?
4. Jaką metodą tworzone są animacje GIF?
5. W jaki sposób tworzone są animacje z wykorzystaniem języka HTML i stylów CSS?
6. Jak działa animacja tworzona z wykorzystaniem stylów CSS i właściwości `transition`?
7. Jak działa animacja tworzona z wykorzystaniem stylów CSS i właściwości `animation`?

### Pytania egzaminacyjne

#### Zadanie 1.

Najprostszy sposób zamiany obiektu oznaczonego cyfrą 1 na obiekt oznaczony cyfrą 2 polega na:

- A. animowaniu obiektu.
- B. zmianie warstwy obiektu.
- C. narysowaniu docelowego obiektu.
- D. geometrycznym transformowaniu obiektu.



### Zadanie 2.

Który z formatów **nie pozwala** na zapis plików animowanych?

- A.** GIF.
- B.** ACE.
- C.** SWF.
- D.** SVG.

### Zadanie 3.

Prosta animacja może być zapisana w formacie:

- A.** TIFF.
- B.** BMP.
- C.** PSD.
- D.** GIF.

### Zadanie 4.

Plik graficzny należy zapisać w formacie GIF, jeżeli:

- A.** jest to grafika wektorowa.
- B.** jest to obraz stereoskopowy.
- C.** jest potrzeba zapisu obrazu bez kompresji.
- D.** jest potrzeba zapisu obrazu lub animacji.

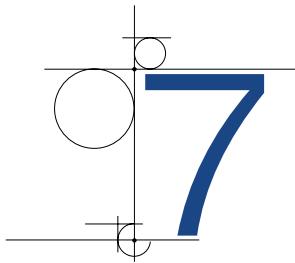
## 6.4.2. Zadania

### Zadanie 1.

Na potrzeby tworzonej witryny internetowej przygotuj baner reklamowy. W zadaniu wykorzystaj narzędzia do tworzenia animacji GIF.

### Zadanie 2.

Na potrzeby tworzonej witryny internetowej przygotuj menu poziome z animowanymi przyciskami polecień. Do opracowania menu wykorzystaj narzędzia do tworzenia animacji za pomocą języka HTML i stylów CSS.



# Dźwięk i wideo



## 7.1. Dźwięk

### 7.1.1. Formaty plików dźwiękowych

Aby można było odtworzyć dźwięk, musi on być zapisany w pliku we właściwym formacie.

Do podstawowych formatów audio należą:

- plik audio AIFF (rozszerzenie [.aiff](#)),
- plik audio AU (rozszerzenie [.au](#)),
- plik audio MP3 (rozszerzenie [.mp3](#)),
- plik **Wave Form Audio Format** (rozszerzenie [.wav](#)),
- plik **Windows Media Audio** (rozszerzenie [.wma](#)),
- plik w formacie **Vorbis** (rozszerzenie [.ogg](#)).

#### Format WAVE

WAVE jest najprostszym formatem plików dźwiękowych. W formacie tym dźwięk nie jest poddawany żadnej kompresji. Jego wadą jest duży rozmiar plików, a zaletą — brak utraty jakości dźwięku. Dodatkowym walorem jest łatwość edytowania plików tego typu. Jest to format obsługiwany przez większość programów do edytowania dźwięków, w związku z czym często bywa wykorzystywany jako standard przenoszenia danych między aplikacjami.

#### Format MP3

Format MP3 jest standardem kodowania audio. Jest to format kompresji stratnej wykorzystujący niedoskonałości ludzkiego ucha. Z pliku dźwiękowego zostają usunięte częstotliwości, których człowiek nie jest w stanie usłyszeć. Kompresja pozwala na znaczne zmniejszenie rozmiaru pliku w zależności od tego, jaki rodzaj dźwięku jest kompresowany. Na przykład kompresja tekstu mówionego zapisanego na potrzeby

internetu może być większa niż kompresja muzyki, przy której standardów jakości nie powinniśmy obniżać. Główną zaletą formatu MP3 jest niewielki rozmiar pliku przy dobrej jakości dźwięku. Jego wada to utrata pewnych częstotliwości z oryginalnego materiału, a co za tym idzie — uniemożliwienie odbiorcy usłyszenia niektórych dźwięków.

## Format WMA

Jest to format firmy Microsoft wprowadzony na rynek jako konkurencja dla formatu MP3. Zasada kompresji jest podobna w obu formatach, również jakość dźwięku jest porównywalna.

## Format MIDI

W formacie tym nie jest zapisywany sam dźwięk, ale informacje o tym, jaki dźwięk powinien zostać odtworzony, jaka ma być jego częstotliwość i jaki ma być instrument odtwarzania.

## Format AIFF

Jest to format plików dla komputerów Macintosh, ale może być również wykorzystywany na innych komputerach. Pliki nie są kompresowane. Zaletą tego formatu jest dobra jakość dźwięku oraz to, że jest obsługiwany przez wiele przeglądarek bez konieczności instalowania wtyczek. Wadą są duże rozmiary plików, co ogranicza ich wykorzystanie.

## Format RealMedia

Format ten został opracowany z myślą o zastosowaniach w internecie. Jest wykorzystywany do strumieniowego przesyłania danych, takich jak muzyka czy filmy wideo. Zaletą tego formatu jest duży współczynnik kompresji przy dobrej jakości. Do odtwarzania plików zapisanych w tym formacie służy odtwarzacz RealPlayer.

## Format Ogg Vorbis

Ogg Vorbis to format kompresji plików audio. Umożliwia zachowanie wysokiej jakości dźwięku przy niewielkiej objętości. Jest alternatywą dla popularnego formatu MP3. Skompresowane pliki mają mniejszy rozmiar niż pliki MP3 przy podobnej jakości ich odtwarzania. Jest bezpłatny, otwarty (kod udostępniany na zasadzie licencji GNU) i nieopatentowany.

## 7.1.2. Kompresja dźwięku

Pliki zawierające muzykę mają duże rozmiary. Do zapisania jednej minuty muzyki nieskompresowanej potrzeba około 10 MB, a utwory muzyczne trwają zwykle dłużej. Przesyłanie tak obszernych plików przez internet jest problemem. Kompresja, czyli pakowanie danych dźwiękowych, pozwala na zmniejszenie ich rozmiaru. Kompresję dźwięku dzielimy na stratną i bezstratną.

## Kompresja stratna

Kompresja stratna to usunięcie z zapisu dźwięków niesłyszalnych lub słabo słyszalnych w celu zmniejszenia rozmiaru pliku. Od stopnia kompresji zależy nie tylko rozmiar pliku, ale również jakość dźwięku. Im większy stopień kompresji, tym gorsza jakość dźwięku. Kompresja stratna jest procesem nieodwracalnym.

## Kompresja bezstratna

Kompresja bezstratna to metoda zmniejszenia objętości pliku z możliwością odtworzenia dźwięku w wersji identycznej z wersją pierwotną. Najczęściej polega na zmianie sposobu zapisu danych, dzięki czemu plik zajmuje mniej miejsca.

W standardowych formatach zapisywania dźwięku (np. w plikach WAV) stosuje się ten sam sposób zapisu, niezależnie od tego, czy dźwięk ma skomplikowany przebieg, czy panuje cisza. Zmieniając sposób zapisu dźwięku (np. przez specjalne oznaczenie ciszy), można zaoszczędzić sporo miejsca.

### 7.1.3. Odtwarzanie audio

Każdy system operacyjny jest wyposażony w odtwarzacz plików audio, który obsługuje najpopularniejsze formaty plików. W systemie Windows jest to Windows Media Player. W internecie można znaleźć wiele odtwarzaczy komercyjnych i bezpłatnych, np.: Winamp, QuickTime, iTunes, RealPlayer. Również programy do nagrywania zawierają odtwarzacze multimedialne.

#### Windows Media Player

**Windows Media Player** to program, który poza tym, że służy odtwarzania, ułatwia wyszukiwanie plików multimedialnych na komputerze i w sieci. Umożliwia odtwarzanie strumieni multimedialnych z internetu, zgrywanie muzyki z płyt CD-Audio oraz jej nagrywanie na płytach CD.

#### QuickTime

QuickTime jest popularnym odtwarzaczem multimedialnym opracowanym przez firmę Apple dla systemów Windows i Mac OS. Pozwala na odtwarzanie plików audio i wideo udostępnianych strumieniowo za pośrednictwem internetu. Obsługuje format H.264 i różne standardy dźwięku przestrzennego, umożliwiając odtwarzanie multimedialów w bardzo dobrej jakości. QuickTime automatycznie analizuje przepustowość łącza, co pozwala na oglądanie materiałów w najlepszej jakości.

#### iTunes

iTunes to aplikacja opracowana przez firmę Apple jako odtwarzacz multimedialny. Umożliwia odtwarzanie muzyki, a także słuchanie radia internetowego, nagrywanie płyt CD, katalogowanie nagrań i konwertowanie plików audio.

## RealPlayer

RealPlayer firmy Real Networks to aplikacja do odbioru transmisji audio i wideo przez internet w czasie rzeczywistym. Stosowane przez program formaty RealAudio i RealVideo są bardzo popularnymi standardami transmitowania audycji radiowych i telewizyjnych w internecie. Program umożliwia konwersję plików audio i wideo na wiele formatów, odtwarzanie płyt DVD oraz nagrywanie płyt CD-Audio. Ma wbudowaną przeglądarkę stron WWW.

## 7.2. Edytowanie i obróbka dźwięku

Do prostego edytowania dźwięku może zostać wykorzystany rejestrator dźwięku obecny w systemie operacyjnym. Przy jego użyciu użytkownik może nagrać np. swój głos. W internecie dostępne są programy służące do nagrywania, edytowania, mikowania i zapisywania plików dźwiękowych w wielu formatach.

### 7.2.1. Przetwarzanie dźwięku — program Audacity

Prostym edytorem plików dźwiękowych jest program **Audacity**, który umożliwia nagrywanie i edytowanie dźwięku w formatach: WAV, AIFF, MP3 oraz AU, a także ma wbudowane narzędzia do konwersji formatów, edytor obwiedni i narzędzia do nakładania dodatkowych efektów, np. dynamiki, korekcji, modulacji. Może służyć do usuwania usterek w gotowych plikach muzycznych. Współpracuje z kodekiem LAME, który umożliwia tworzenie plików MP3.

#### Instalowanie Audacity

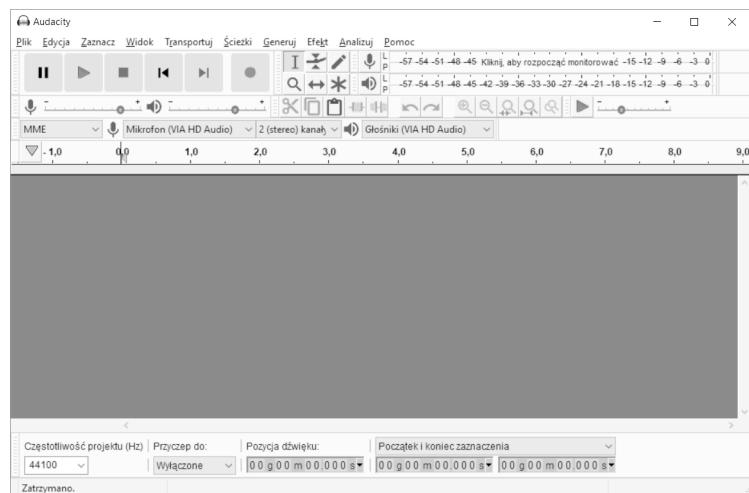
Program Audacity jest bezpłatny i można go pobrać ze strony <http://www.audacityteam.org>. Znajduje się na niej najnowsza wersja Audacity. Po pobraniu i zainstalowaniu program Audacity zostanie uruchomiony automatycznie (rysunek 7.1).

Pierwszym krokiem powinno być sprawdzenie i w razie potrzeby zmodyfikowanie ustawień dotyczących urządzeń do odtwarzania i nagrywania dźwięku. Wybieramy w menu *Edycja/Ustawienia...*, w otwartym oknie zaznaczamy opcję *Urządzenia* (rysunek 7.2) i wskazujemy urządzenie do odtwarzania dźwięku (w polu *Odtwarzanie*) oraz urządzenie do nagrywania (w polu *Nagrywanie*). W polu *Kanały* ustawiamy dwa kanały stereo.

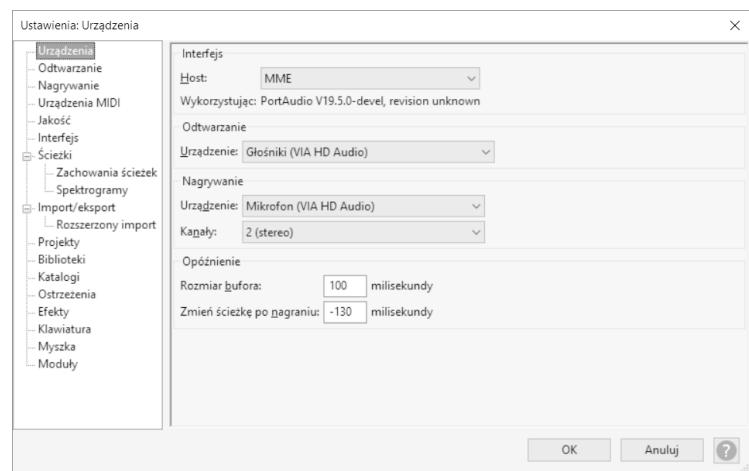
W starszych wersjach Audacity należało pobrać (z podanego powyżej adresu) dodatkowo kodęk MP3 LAME i zainstalować go. Następnie w oknie *Ustawienia* należało wybrać opcję *Biblioteki* (rysunek 7.3).

**Rysunek 7.1.**

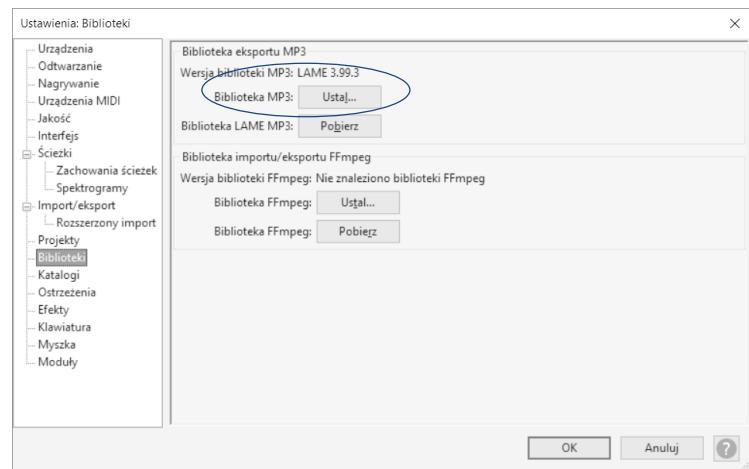
Okno programu  
Audacity

**Rysunek 7.2.**

Ustawienia  
urządzeń  
odtwarzania  
i nagrywania

**Rysunek 7.3.**

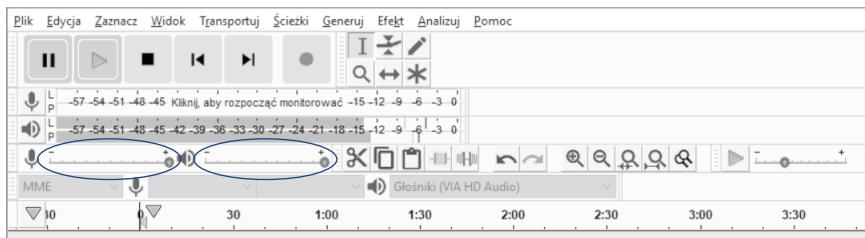
Ustawienia  
panelu Biblioteki



Biblioteki LAME zostaną automatycznie dołączone. W otwartym oknie pojawi się opis dołączonej biblioteki lub komunikat, że biblioteka eksportu MP3 nie została znaleziona. W takim przypadku należy kliknąć przycisk *Ustaw...* i znaleźć kodek LAME zainstalowany na komputerze. Jeżeli kodek nie jest zainstalowany, trzeba kliknąć przycisk *Pobierz* i zainstalować wymagane oprogramowanie. Zatwierdzamy wprowadzone ustawienia, klikając przycisk *OK*.

Jeżeli ustawienia dla bibliotek zostały poprawnie zdefiniowane, program Audacity może odczytywać i zapisywać pliki MP3.

W oknie programu można zmienić ustawienia wzmacnienia głośników lub słuchawek (*Głośność odtwarzania*) oraz ustawienia wzmacnienia mikrofonu (*Głośność nagrywania*), przesuwając odpowiednio suwaki tych opcji (rysunek 7.4).



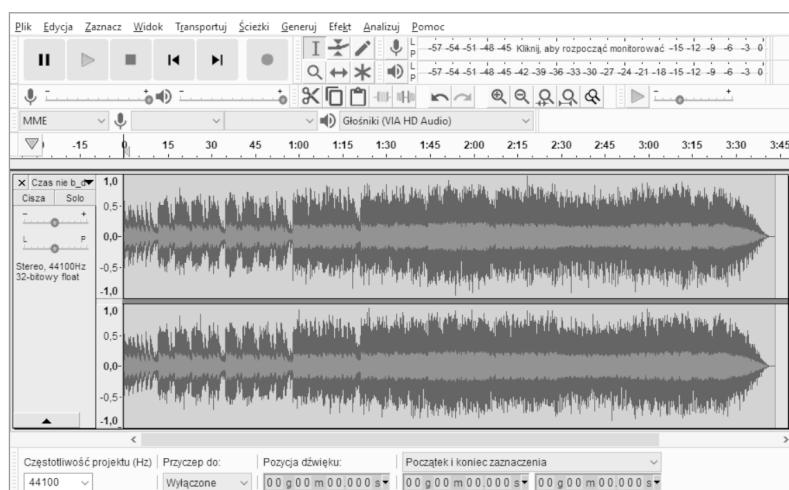
**Rysunek 7.4.** Ustawienie wzmacnienia głośników (słuchawek) i mikrofonu

## 7.2.2. Modyfikowanie dźwięku

Modyfikowanie plików zapisanych w formacie MP3 jest jedną z najczęściej wykonywanych czynności związanych z obróbką dźwięku. Zawartość wczytanego pliku (z menu *Plik/Otwórz*) zostanie wyświetlona w programie w postaci wykresu spektrograficznego (rysunek 7.5).

**Rysunek 7.5.**

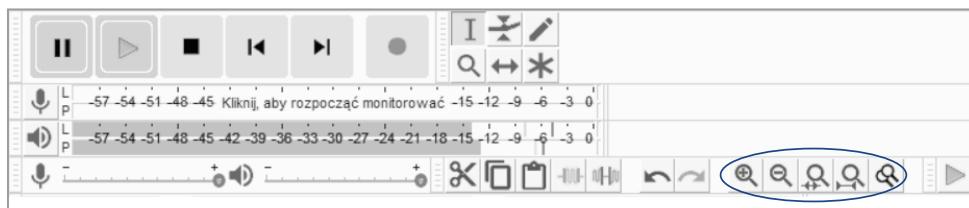
Plik MP3  
odtwarzany  
w programie  
Audacity



Ze względu na to, że utwór został nagrany w trybie stereo, wyświetlane są dwie ścieżki. Jeśli klikniemy w którymś miejscu w obrębie wykresu dźwięku, zostanie tam wstawiona pionowa linia. Od tego miejsca po kliknięciu przycisku *Odtwarzaj* rozpocznie się odtwarzanie muzyki.

## Lupa

Zawartość pliku dźwiękowego jest wyświetlana w programie w postaci wykresu. Do łatwiejszego operowania na wykresie można użyć narzędzi z paska *Edycja*. Pasek edytowania zawiera zestaw lupy, które pozwalają manipulować widocznym w oknie fragmentem utworu. Narzędzie *Dopasuj projekt do szerokości* (lupa z kreską) pozwoli na dostosowanie wykresu utworu do wielkości okna. Narzędzie *Dopasuj zaznaczenie do szerokości* (lupa ze strzałkami) dopasuje zaznaczony fragment do wielkości okna (rysunek 7.6).



**Rysunek 7.6.** Lupy na pasku edytowania

Pracę z dźwiękiem zawsze wykonujemy na zaznaczonym fragmencie. Zaznaczenie nastąpi po kliknięciu w dowolnym miejscu utworu i przeciągnięciu myszą w prawo lub w lewo po wykresie.

## Wyciszenie

Jedną z metod modyfikowania utworów muzycznych jest wyciszenie początkowego lub końcowego fragmentu utworu.

### Przykład 7.1

Wczytujemy plik muzyczny przeznaczony do modyfikacji. Chcemy wyciszyć początkowy fragment utworu. Odsłuchujemy go i określamy, jaki fragment zostanie wyciszczyony. Klikamy na wykresie w miejscu końca wyciszenia i przeciągamy myszą do samego początku utworu. Wybieramy ikonę lupy ze strzałkami i powiększamy zaznaczony fragment. Odtwarzamy ten fragment. Przed wykonaniem jakiejkolwiek operacji powinniśmy go odsłuchać, aby upewnić się, że zaznaczenie zostało wykonane prawidłowo.

Aby wyciszyć zaznaczony fragment, wybieramy ikonę *Wycisz zaznaczony dźwięk* na pasku *Edycja* (rysunek 7.7).



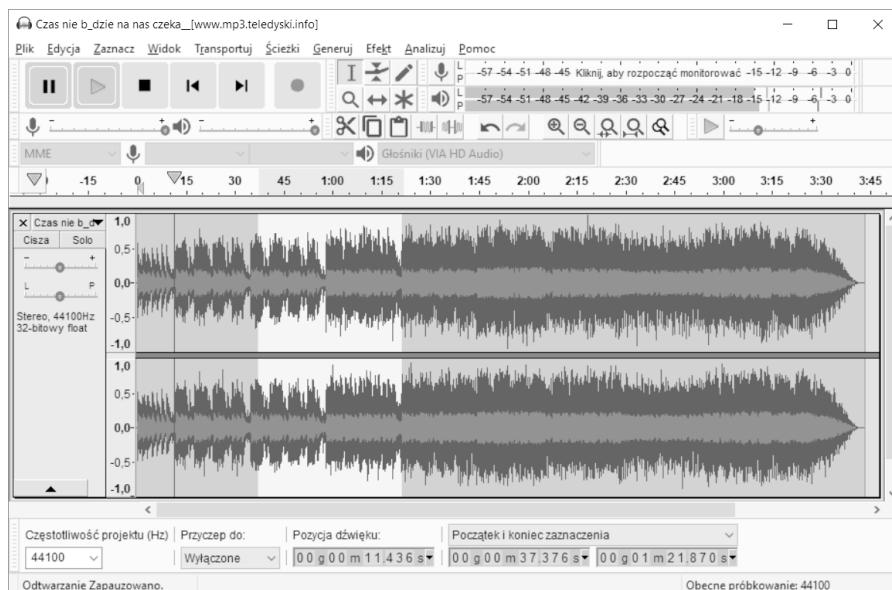
**Rysunek 7.7.** Ikona wyciszenia

## Wycinanie fragmentu

Często wycinamy część utworu w celu dalszego wykorzystania tylko tego fragmentu.

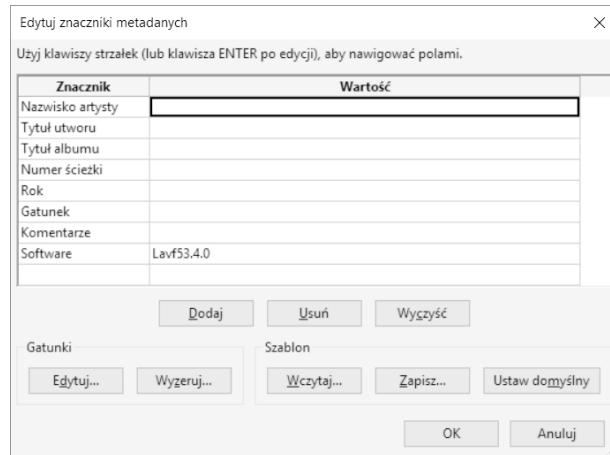
### Przykład 7.2

Otwieramy plik dźwiękowy i uruchamiamy go. Odsłuchując utwór, na skali czasu (nad wykresem) ustawiamy czas odpowiadający fragmentowi wycinanemu z całości. Następnie klikamy w obszarze wykresu, aby ustawić znacznik początku fragmentu wycięcia, i przeciągając myszą, zaznaczamy koniec fragmentu wycięcia (rysunek 7.8).



**Rysunek 7.8.** Wybrany fragment zaznaczony na wykresie

Zapisujemy zaznaczoną część utworu i poddajemy dalszej obróbce tylko ten fragment. Wybieramy *Plik/Eksportuj/Eksportuj zaznaczony dźwięk...* i w otwartym oknie wpisujemy nazwę pliku, wskazujemy format MP3 i klikamy *Zapisz*. Pojawi się okno do wprowadzenia dodatkowych informacji o zapisywany fragmentie (rysunek 7.9). Możemy wypełnić je dowolną treścią.



**Rysunek 7.9.** Dodatkowe informacje o utworze

Zapisany fragment ponownie otwieramy. Odsłuchujemy początkową i końcową część. Jeżeli utwór kończy się zbyt gwałtownie, można to skorygować. Zaznaczamy ostatnie trzy sekundy utworu (należy próbować z różnymi wartościami, aż uzyskamy oczekiwany efekt), wybieramy z menu opcję *Efekt* i z listy efekt *Ściszanie*. Przy odtwarzaniu końcowego fragmentu głośność będzie stopniowo malała. Zapisujemy utwór, wybierając z menu *Plik/Eksportuj /Eksportuj dźwięk....*

Inaczej możemy postąpić, jeśli zaznaczony fragment utworu chcemy po prostu usunąć.

### Przykład 7.3

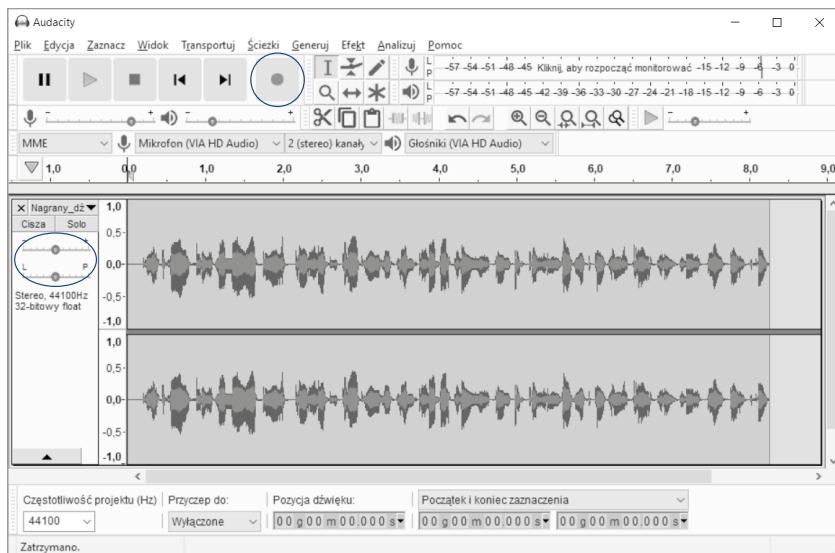
Planujemy usunięcie z wybranego utworu 30-sekundowego fragmentu początkowego. Otwieramy plik muzyczny. Zaznaczamy fragment przeznaczony do wycięcia i klikamy na pasku *Edycja* ikonę *Wytnij* (rysunek 7.10). Zaznaczony fragment zostanie usunięty. Trzeba jeszcze odsłuchać utwór i — jeżeli to konieczne — skorygować wycięcie, a następnie zapisać plik.



**Rysunek 7.10.** Ikona Wytnij

### Nagrywanie dźwięku

Korzystając z programu Audacity, użytkownik może nagrać np. swój głos i odpowiednio go zmodyfikować. Służy do tego przycisk *Nagrywaj* na pasku *Transport* (rysunek 7.11).



**Rysunek 7.11.** Narzędzia nagrywania głosu

Gdy głos jest już nagrany, za pomocą suwaków z lewej strony wykresu można podwyższyć lub obniżyć głos. Nagrany głos można zapisać w pliku MP3.

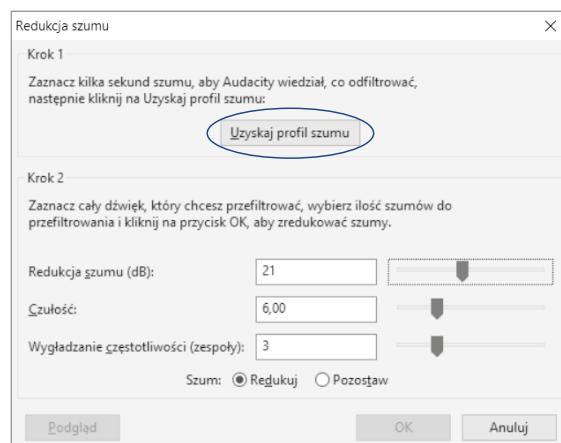
## Usuwanie szumów

### Przykład 7.4

Otwieramy plik dźwiękowy zawierający szумy, które chcemy usunąć. Zaznaczamy fragment ścieżki dźwiękowej — najlepiej, żeby był to kilkusekundowy fragment, na którym zarejestrowana jest wyłącznie cisza (czyli sam szum mikrofonu) — następnie z menu wybieramy opcję *Efekt/Redukcja szumu*. W otwartym oknie klikamy przycisk *Uzyskaj profil szumu* (rysunek 7.12) i zatwierdzamy przyciskiem *OK*. Uzyskana przez program Audacity próbka szumu zostanie wykorzystana przy jego usuwaniu.

### Rysunek 7.12.

Definiowanie profilu szumu



Naciskamy klawisze ***Ctrl+A***, żeby zaznaczyć całą ścieżkę dźwiękową. Ponownie wybieramy opcję ***Efekt/Redukcja szumu***, w otwartym oknie (rysunek 7.12) ustawiamy suwak ***Redukcja szumu (dB)*** na ***0***. Przed zapisaniem klikamy przycisk ***Podgląd***, aby odsłuchać ścieżkę po zastosowaniu ustawień. W razie potrzeby możemy manipulować pozostałymi suwakami aż do uzyskania pożądanych rezultatów.

## Łączenie plików muzycznych

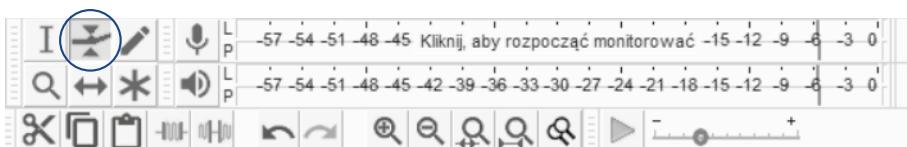
Domyślnie kolejne pliki audio są otwierane w nowych oknach programu Audacity. Jeżeli zamierzamy połączyć kilka plików dźwiękowych, należy otworzyć je w jednym oknie. Najprostszą metodą takiego otwarcia plików jest wybranie z menu opcji ***Plik/Otwórz....***. Zostanie otwarty pierwszy plik muzyczny. Kolejne pliki będą dodawane do następnych ścieżek tego samego okna po wybraniu z menu opcji ***Plik/Importuj/Dźwięk....***. Importowane pliki mogą mieć różne poziomy głośności. Do dopasowania głośności utworów może zostać wykorzystane ***Narzędzie obwiedni***.

## Obwiednia

Obwiednia to zapis cyfrowy amplitudu dźwięku — zapis ten możliwie jak najlepiej odzwierciedla oryginalne brzmienie dźwięku.

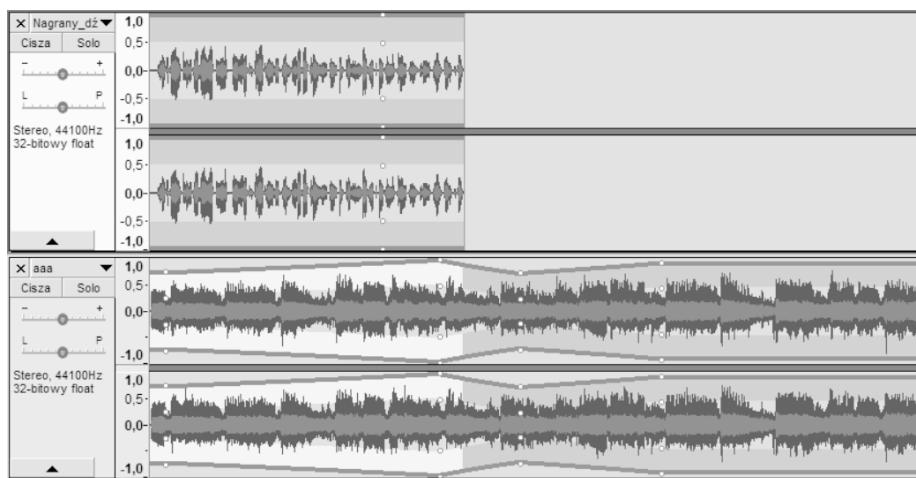
### Przykład 7.5

Importujemy do programu Audacity dwa pliki dźwiękowe. Jeden zawiera muzykę, drugi — nagrany wcześniej głos. Jeżeli muzyka jest za głośna w stosunku do nagranego głosu, to do jej przyciszenia można wykorzystać ***Narzędzie obwiedni***. Jego ikona znajduje się na pasku ***Narzędzia*** (rysunek 7.13).



**Rysunek 7.13.** Ikona narzędzia obwiedni

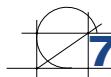
Po wybraniu tej ikony ustawiamy kurSOR myszy nad wykresem ścieżki z muzyką, chwytamy za górną krawędź obwiedni wykresu i przeciągamy myszą w dół. Powoduje to ściszenie utworu. Klikając w kolejnym miejscu obwiedni, tworzymy na niej następny punkt i podnosimy myszą obwiednię do góry, co powoduje zwiększenie głośności muzyki. W ten sposób można tworzyć kolejne punkty na obwiedni i odpowiednio modelować głośność utworu (rysunek 7.14).



**Rysunek 7.14.** Definiowanie obwiedni

### Zadanie 7.1

Przygotuj trzy pliki dźwiękowe. Umieść je na osobnych ścieżkach. Stosując poznane metody pracy z dźwiękiem, zmodyfikuj kolejne ścieżki (przytnij je i wycisz), tak by stworzyć miks muzyczny.



## 7.3. Pliki wideo

Pliki wideo służą do zapisywania filmów. Do pliku wideo mogą być dołączone napisy — np. informacje o filmie lub jego autorze — komentarze lub paski informacyjne.

### 7.3.1. Standardy

**Rozdzielcość obrazu** mówi o jego jakości. Im wyższa rozdzielcość, tym wyraźniejszy obraz. W systemie PAL (ang. *Phase Alternating Line*), wykorzystywanym w Europie, obraz jest wyświetlany w rozdzielcości  $704 \times 576$  pikseli. W systemie NTSC (ang. *National Television System Committee*), stosowanym w USA, rozdzielcość obrazu wynosi  $720 \times 480$  pikseli.

W wykorzystywanym w telewizji wysokiej rozdzielcości standardzie HDTV (ang. *High Definition TV*) obraz jest wyświetlany w większej rozdzielcości niż w standardach PAL i NTSC. Obecnie stosowane są standardy: **HDTV 720p** (rozdzielcość obrazu wynosi  $1280 \times 720$  pikseli), **FullHD 1080i/1080p** (rozdzielcość obrazu wynosi  $1920 \times 1080$  pikseli) oraz **Ultra HD** (rozdzielcość obrazu wynosi  $3840 \times 2160$  pikseli).

**Liczba klatek na sekundę** określa liczbę klatek filmu wyświetlanych w czasie jednej sekundy. Przyjmuje się, że aby obraz zmieniał się płynnie, ten parametr musi wynosić 24–30 klatek na sekundę. W systemie PAL wynosi on 25 klatek na sekundę, w systemie NTSC — 30 klatek na sekundę.

**Format obrazu** to stosunek szerokości obrazu do jego wysokości. Stosuje się dwa formaty obrazu — 4:3 i 16:9. Format 4:3 jest używany w telewizji w systemach PAL i NTSC, a format 16:9 w telewizji HDTV (wysokiej rozdzielczości) i telewizji cyfrowej.

## 7.3.2. Odtwarzanie wideo

Do odtworzenia filmu wideo potrzebne jest odpowiednie oprogramowanie. Dostępne odtwarzacze mają zwykle pakiet kodeków do odtwarzania różnych formatów plików wideo. Jeżeli one nie wystarczają, można doinstalować dodatkowe kodeki. Niektóre aplikacje odtwarzają tylko wybrany format plików (RealPlayer, QuickTime), inne pozwalają na odtwarzanie wielu typowych formatów plików (Windows Media Player).

## 7.3.3. Formaty plików wideo

### Format AVI

Format AVI został opracowany przez firmę Microsoft. Jest stosowany do zapisu dźwięku i sekwencji wideo. Dane można zapisywać w postaci skompresowanej lub nieskompresowanej. Aby można było odtwarzać pliki zapisane w tym formacie, należy zainstalować właściwy kodек. Pliki mają rozszerzenie [.avi](#).

### Format MPEG

Jest to format stratnej kompresji plików audio i wideo. Jego zaletą jest duża kompresja bez straty jakości obrazu. Kolejne wersje tego standardu to: **MPEG-1**, **MPEG-2** i **MPEG-4**. Format **MPEG-4** został przystosowany do kompresji danych strumieniowych, format **MPEG-7** to standard opisu danych multimedialnych, a format **MPEG-21** jest standardem, który ma być wprowadzony w przyszłości; przyczyni się on do dalszej standaryzacji treści multimedialnych. Pliki mają rozszerzenie [.mpg](#) lub [.mpeg](#).

### Format Windows Media Video

**Windows Media Video** to format opracowany przez firmę Microsoft dla plików audio i wideo. Charakteryzuje się dużym stopniem kompresji. Pliki wideo mają rozszerzenie [.wmv](#).

### Format DivX

Jest to format stratnej kompresji plików audio i wideo. Obecne wersje zgodne z **MPEG-4** pozwalają zapisać na płycie kompaktowej filmy o długości do 90 minut i jakości nie-wiele ustępującej DVD-Video. Format ten jest wykorzystywany do przesyłania filmów przez internet.

### Format QuickTime

Format **QuickTime** został opracowany przez firmę Apple dla plików multimedialnych. Pliki w tym formacie mogą zawierać nagrania audio, wideo, animacje, obiekty

3D. Jest to format często stosowany do publikowania filmów w internecie. Pliki mają rozszerzenie [.qt](#) lub [.mov](#).

## 7.3.4. Programy do edytowania wideo

### Adobe Premiere

**Adobe Premiere** to bardzo ceniona aplikacja do nieliniowego edytowania wideo. Jest wyposażona w zaawansowane narzędzia i funkcje do łączenia obrazu i dźwięku w jedną całość. Praca odbywa się w czasie rzeczywistym. Narzędzia do edytowania plików audio i wideo umożliwiają precyzyjną kontrolę nad każdym etapem tworzenia projektu. Z łatwością można manipulować obrazem, dźwiękiem, animacjami i grafiką, aby tworzyć filmy wysokiej jakości.

### Avid Media Composer

**Avid Media Composer** to jedna z najpopularniejszych aplikacji przeznaczonych do montażu i edytowania wideo w czasie rzeczywistym. Program oferuje: zestaw narzędzi do profesjonalnej korekcji barw, narzędzie do usuwania greenscreena (SpectraMatte), natywne wsparcie dla formatów: P2, XDCAM, RED i QuickTime. Ponadto edytor umożliwia wysyłanie plików wideo do popularnych serwisów społecznościowych, ma funkcję stabilizowania obrazu oraz pozwala na obsługę wielu pluginów i filtrów.

### Avid Studio

**Avid Studio** to aplikacja przeznaczona do nieliniowego edytowania wideo. Dostępna jest w polskiej wersji językowej. Główne jej zalety to prostota obsługi oraz setki wbudowanych efektów (w tym specjalnych) i przejść. Umożliwia pracę z nieograniczoną liczbą ścieżek audio i wideo, ma wbudowany stabilizator obrazu, pozwala eksportować filmy do najpopularniejszych formatów, np. wykorzystywanych w serwisie YouTube, czy do aplikacji QuickTime HD. Oferuje również profesjonalne narzędzia do edytowania kolorów.

### Sony Vegas Movie Studio

**Sony Vegas Movie Studio** to okrojona wersja aplikacji Sony Vegas przeznaczona do amatorskich i półprofesjonalnych zastosowań. Pozwala na montaż i edytowanie wideo w czasie rzeczywistym. Ponadto edytor umożliwia wysyłanie plików wideo do bieżącego w chmurze serwisu Sony oferującego udostępnianie zdjęć i filmów w internecie.

### Kdenlive

**Kdenlive** to jeden z najpopularniejszych edytorów wideo na Linuksa. Umożliwia nielinowy montaż wideo. Obsługuje formaty: QuickTime, AVI, WMV, MPEG.

## 7.4. Planowanie filmu

Przystępując do tworzenia filmu, należy starannie zaplanować etapy pracy.

- **Pierwszy etap** — określenie, co powinien zawierać film.
- **Drugi etap** — opracowanie scenariusza filmu (opis scen i wykorzystywanych materiałów).
- **Trzeci etap** — gromadzenie materiałów (przygotowywanie plików filmowych, plików muzycznych, zdjęć, tekstów).
- **Czwarty etap** — montaż filmu.

Montaż filmu to nie tylko połączenie zebranych materiałów w całość, ale również wykrywanie niepotrzebnych scen z plików wideo, dodawanie efektów, dodawanie napisów, synchronizacja ścieżki dźwiękowej.

### Zadanie 7.2

Przygotuj materiały do filmu pod tytułem *Spacer po moim mieście*. Wykonaj zdjęcia miejsc, które chciałbyś pokazać w filmie. Nakrć klipy filmowe. Do ich utworzenia możesz wykorzystać kamerę, aparat cyfrowy lub telefon komórkowy z funkcją rejestracji filmów. Przygotuj pliki muzyczne. Opracuj scenariusz filmu.

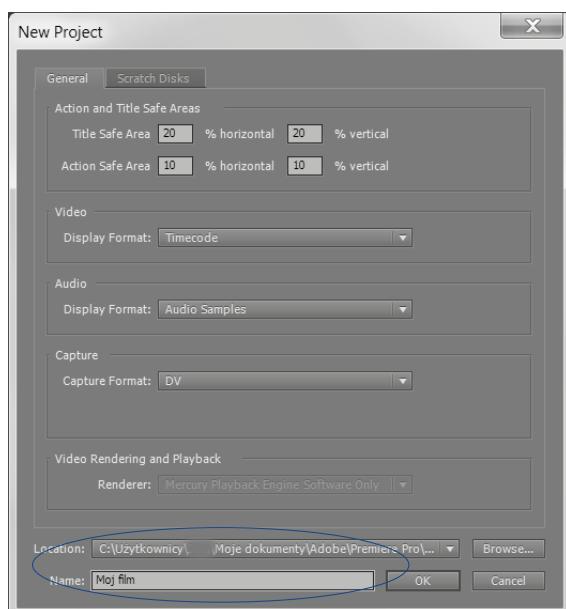
## 7.5. Tworzenie filmu w programie Adobe Premiere

Adobe Premiere to narzędzie dla profesjonalistów tworzących filmy wysokiej jakości, ale również mniej doświadczony użytkownik tego programu doceni jego możliwości i prostotę obsługi.

### 7.5.1.

#### Środowisko pracy

Po uruchomieniu programu i wybraniu opcji *New Project* zostanie otwarte okno (rysunek 7.15); należy w nim podać nazwę tworzonego projektu i wskazać folder, w którym zostanie zapisany.

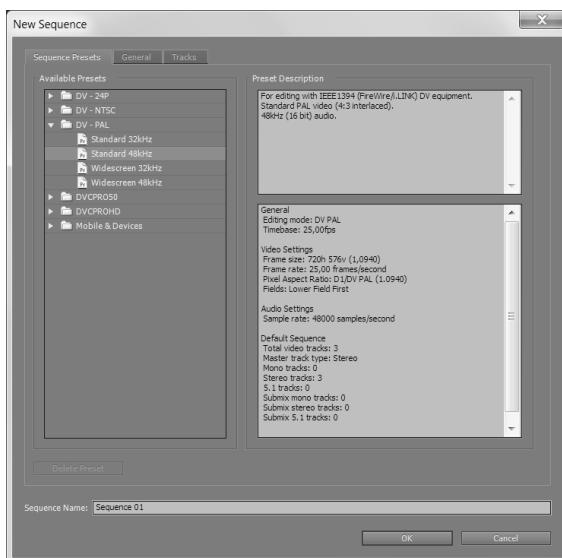


Rysunek 7.15.

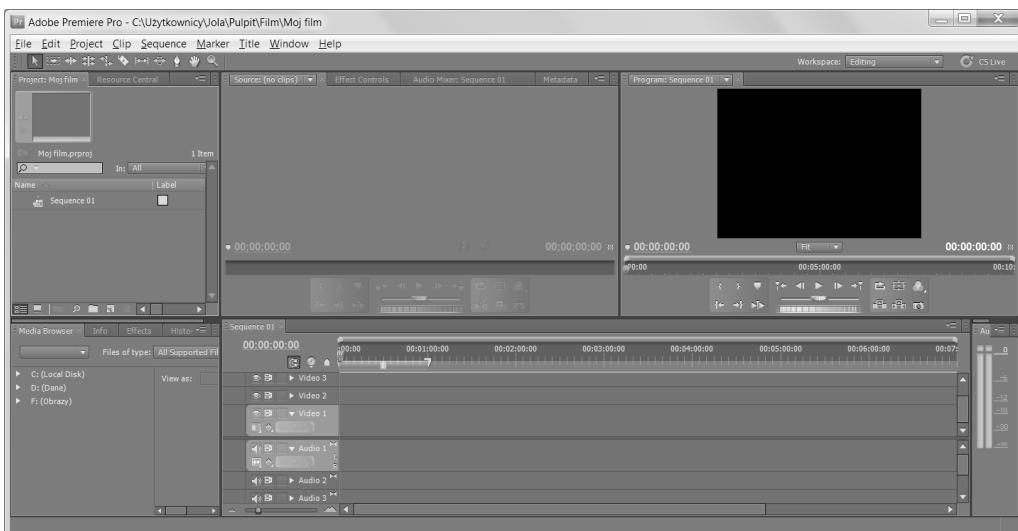
Okno New Project

Gdy klikniemy przycisk **OK**, pojawi się kolejne okno, w którym wyświetla się profile dostępne w programie (rysunek 7.16).

Wybieramy profil **DV-PAL, Standard 48kHz** i podajemy dowolną nazwę sekwencji. Zostanie otwarte okno pracy w programie Adobe Premiere (rysunek 7.17).



**Rysunek 7.16.**  
Definiowanie sekwencji



**Rysunek 7.17.** Okno pracy w programie Adobe Premiere

Okno pracy składa się z kilku paneli. Najważniejszy jest panel **Timeline** (linia czasu), znajdujący się z prawej strony, w dolnej części okna. W nim będą umieszczone odpowiednio zmodyfikowane sekwencje wideo i audio. Tu będziemy przycinać materiał, nakładać efekty i rozmieszczać materiał na ścieżkach wideo i audio. Powyżej tego panelu znajduje się z prawej strony panel **Program** (okno programu), gdzie będzie można oglądać rezultat

wykonanego montażu oraz na bieżąco śledzić działanie wprowadzanych modyfikacji. W lewym górnym rogu widoczny jest panel *Project* (okno projektu), w którym zostaną umieszczone materiały potrzebne do tworzenia projektu. Kolejny panel to *Media Browser*, znajdujący się poniżej panelu *Project*. W tym panelu wyświetlana jest struktura katalogów, w której możemy wyszukiwać potrzebne do pracy pliki. Pod zakładką *Effects* znajduje się panel z listą efektów podzielonych zgodnie z ich przeznaczeniem; wykorzystuje się je do modyfikowania audio i wideo lub do definiowania przejść. Powyżej panelu *Project* widnieje panel *Tools*, zawierający narzędzia do przycinania, modyfikowania i powiększania sekwencji wideo lub audio oraz do wykonywania innych działań. W górnej części, pomiędzy panelami *Project* i *Program*, znajduje się panel zawierający m.in. okno *Effect Controls*. W panelu tym będą wyświetlane parametry efektów dodawanych do sekwencji wideo lub audio.

W programie Adobe Premiere aktywny panel jest zaznaczany żółtym obramowaniem. Aby wyświetlić jego zawartość bez zmiany układu pozostałych paneli, należy po jego zaznaczeniu nacisnąć na klawiaturze klawisz ze znakiem ~ (bez klawisza *Shift*) — wybrany panel zostanie powiększony do rozmiarów okna programu Adobe Premiere. Ponowne naciśnięcie klawisza ze znakiem ~ przywraca poprzedni rozmiar panelu.

## 7.5.2. Montaż filmu

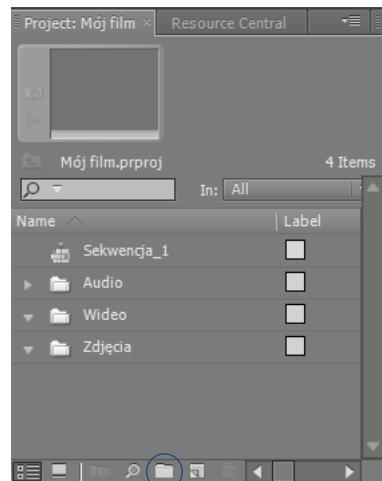
Tworzenie filmu rozpoczynamy od przygotowania materiałów. Powinniśmy dysponować własnym materiałem filmowym, własnymi zdjęciami i plikami z muzyką.

### Import plików

Pierwszym etapem pracy jest zimportowanie przygotowanego materiału do okna *Project*. W tym celu wybieramy z menu *File/Import* (skrót *Ctrl+I*) i w otwartym oknie znajdujemy potrzebne pliki. Zaznaczamy je i importujemy do panelu *Project*. Pliki można wybierać również w oknie *Media Browser*. Zimportowane elementy trzeba uporządkować. W panelu *Project* można tworzyć foldery za pomocą ikony *New Bin* (rysunek 7.18). Dobrym rozwiązaniem jest utworzenie oddzielnych folderów dla plików wideo, plików audio oraz dla zdjęć.

### Zadanie 7.3

Zimportuj do projektu w programie Adobe Premiere wykonane przez siebie zdjęcia i klipy filmowe oraz pliki audio przygotowane do filmu *Spacer po moim mieście*.



**Rysunek 7.18.**

Ikona New Bin w oknie Project

## Przycinanie

Podstawowe narzędzia do montażu to *Selection Tool* i *Razor Tool*, dostępne w panelu *Narzędzia*.

Narzędzie *Selection Tool* jest wywoływanie klawiszem **V** i służy do wyboru elementu, a narzędzie *Razor Tool* jest wywoływanie klawiszem **C** i służy do przycinania fragmentów sekwencji wideo.

Po zimportowaniu materiałów do programu można rozpocząć pracę nad nimi.

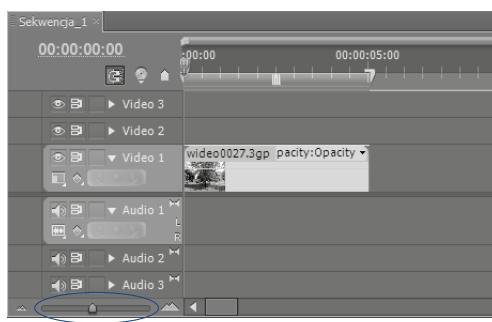
Przenosimy pierwszą sekwencję wideo do panelu linii czasu. W panelu tym domyślnie widoczne są trzy ścieżki wideo i trzy ścieżki audio. Zaznaczamy wybrany element w panelu *Project* i przeciągamy go do panelu *Timeline*, na ścieżkę *Video 1*. Jeżeli widoczna w panelu *Timeline* sekwencja wideo ma niewielkie rozmiary, można powiększyć jej widok, wybierając na klawiaturze klawisz **+** (plus) bez naciskania klawisza *Shift*. Aby zmniejszyć widok, należy wybrać klawisz **-** (minus), również bez naciskania klawisza *Shift*. Ten sam rezultat można osiągnąć, zmieniając położenie suwaka znajdującego się w dolnej części panelu *Timeline* (rysunek 7.19).

Przycinanie sekwencji wideo z przodu lub z tyłu można zrealizować z wykorzystaniem wskaźnika myszy. Jeżeli ustawiemy kurSOR myszy na początku bądź na końcu sekwencji wideo, to zmieni on swój wygląd na czerwony prostokąt. Przy przesuwaniu wskaźnika w prawo lub w lewo pojawi się informacja, o ile klatek element zostanie przycięty. Przycięty materiał można odzyskać, przesuwając wskaźnik myszy w przeciwną stronę.

Drugim narzędziem do przycinania jest *Razor Tool*. Po jego wybraniu kurSOR myszy zmieni się w symbol żyletki — można wtedy w dowolny sposób przyciąć materiał filmowy.

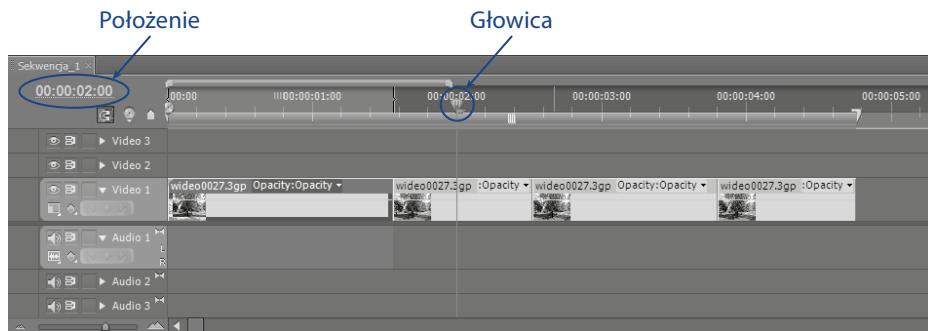
Montaż materiału nie polega na przypadkowym jego przycinaniu, ale na przycinaniu go w zaplanowanych miejscach. Do określenia miejsca przycięcia służy głowica znajdująca się nad materiałem filmowym. Gdy głowica jest przesuwana, w lewym górnym rogu panelu *Timeline* jest wyświetlane jej położenie (rysunek 7.20).

Po ustaleniu położenia głowicy wystarczy kliknąć narzędziem *Razor Tool* w wyznaczonym miejscu (bez specjalnego dopasowania), aby to miejsce zostało przycięte. W celu usunięcia niepotrzebnego fragmentu trzeba przejść w tryb wskazywania, zaznaczyć wybrany fragment filmu i usunąć go (np. klawiszem *Delete*). Usunięty materiał zawsze można odzyskać. Po usunięciu niepotrzebnych fragmentów między fragmentami klipu powstaną puste miejsca. Można je usunąć, klikając w pustym miejscu i naciskając klawisz *Delete*.



**Rysunek 7.19.**

Okno Timeline z suwakiem do powiększania sekwencji wideo



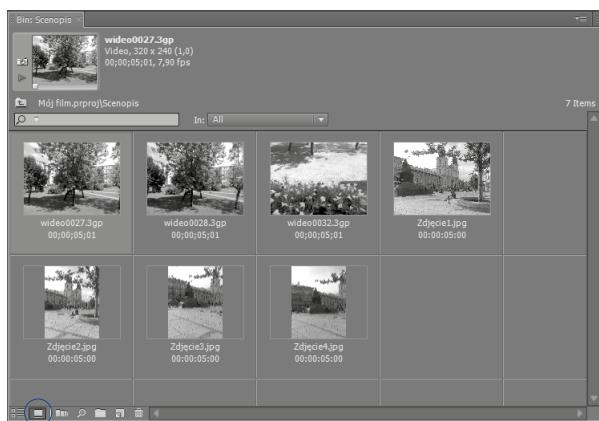
**Rysunek 7.20.** Przycinany materiał filmowy oraz położenie głowicy

## Scenopis

Jeżeli tworzony film będzie się składał z wielu elementów (kilku sekwencji filmowych, kilku zdjęć), należy je uporządkować i określić, w jakiej kolejności na linii czasu będą odtwarzane. W tym celu możemy wszystkie materiały umieścić w jednym folderze. Następnie wykorzystujemy narzędzia programu Adobe Premiere do przygotowania scenopisu.

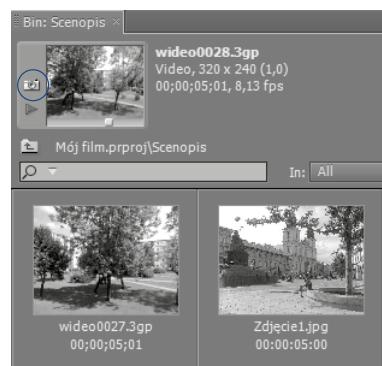
W panelu *Project* wybieramy folder ze zgromadzonymi zdjęciami i klipami filmowymi. Klikamy ten folder dwukrotnie, aby go otworzyć. W dolnej części otwartego panelu znajduje się ikona *Icon View*. Po jej kliknięciu zmieni się sposób wyświetlania zawartości folderu (rysunek 7.21). W tym widoku zawartość folderu może być modyfikowana — można zmieniać kolejność elementów lub powielać je, aż powstanie gotowy scenopis.

Zdarza się, że różne fragmenty tego samego klipu filmowego są wykorzystywane w kilku miejscach filmu. W panelu scenopisu można skopiować klip i umieścić jego kopię w wybranych miejscach. To, który fragment klipu zostanie odtworzony, ustala się w panelu podglądu (góra część otwartego okna), zatrzymując przeglądanie sekwencji wideo w odpowiednim miejscu. Po kliknięciu ikony *Poster Frame* miejsce to zostanie zapamiętane w scenopisie (ikona obok panelu podglądu — rysunek 7.22).



**Rysunek 7.21.**

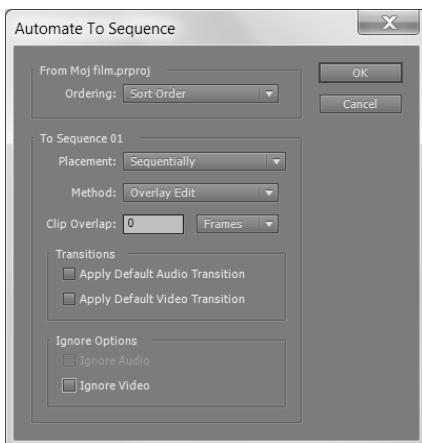
Tworzenie scenopisu — ikona *Icon View*



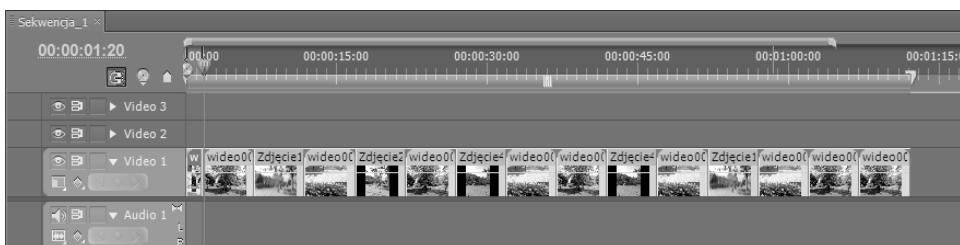
**Rysunek 7.22.**

Fragment okna scenopisu

Po ustaleniu scenariusza filmu można automatycznie cały materiał przenieść do panelu ***Timeline***. Aby to zrobić, po zaznaczeniu wszystkich elementów do przeniesienia wybieramy w dole panelu scenopisu ikonę ***Automate to Sequence***, co spowoduje otwarcie panelu do modyfikacji ustawień przenoszonego materiału (rysunek 7.23). Po ustaleniu odpowiednich parametrów klikamy ***OK*** i zamykamy panel scenopisu. Nasz materiał znalazł się w panelu ***Timeline*** (rysunek 7.24).



**Rysunek 7.23.** Ustawienia parametrów scenopisu



**Rysunek 7.24.** Elementy scenopisu w oknie

Kolejny etap pracy to przycinanie i usuwanie niepotrzebnych fragmentów z przygotowanego scenopisu (narzędzia *Selection Tool* i *Razor Tool*).

Pracę może usprawnić narzędzie *Ripple Edit Tool* (z panelu narzędzi), które nie tylko przycina film, ale również usuwa niepotrzebne fragmenty klipu.

W celu zastosowania tego narzędzia należy ustawić przy użyciu głowicy miejsce cięcia, następnie chwycić myszą za krawędź początkową lub końcową modyfikowanego fragmentu i przeciągnąć do miejsca ustawienia głowicy. W wyniku takiego działania materiał zostanie przycięty, a niepotrzebny fragment filmu — usunięty.

### Zadanie 7.4

Utwórz scenopis filmu *Spacer po moim mieście*. Wykonaj montaż filmu.

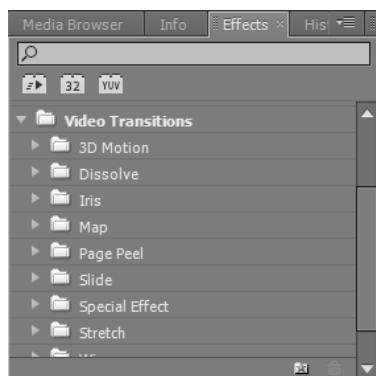
## Efekty przejść wideo

W celu uatrakcyjnienia filmu zastosujemy efekty przejść. Pozwolą one na łagodne rozpoczęcie lub zakończenie filmu oraz na uzyskanie efektownych przejść między kadrami.

Efekty przejść mieścią się z lewej strony panelu *Timeline*, w panelu *Effects*. Wybierając na liście efektów opcję *Video Transitions*, znajdziemy pogrupowane efekty dotyczące sekwencji wideo (rysunek 7.25).

Dodawanie efektu polega na znalezieniu go w panelu *Effects* i przeciagnięciu do panelu *Timeline*, w odpowiednie miejsce filmu.

Efekty mogą być umieszczane albo na lewej, albo na prawej krawędzi klipu lub na połączeniu między klipami (rysunek 7.26). Nie można ich umieszczać na klipie. Oprócz położenia efektu możemy ustalać czas jego trwania. Najprostszą metodą jest zaznaczenie efektu przejścia w panelu *Timeline* i odpowiednie jego wydłużenie. Czas trwania efektu można również ustawić w panelu *Effect Controls* (powyżej panelu *Timeline*), zmieniając opcję *Duration* (rysunek 7.27).

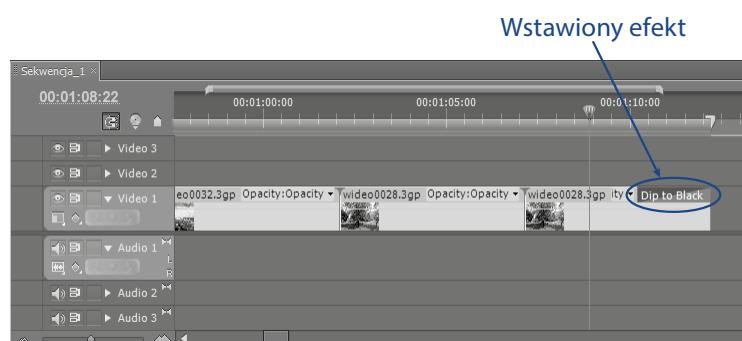


Rysunek 7.25.

Okno efektów wideo

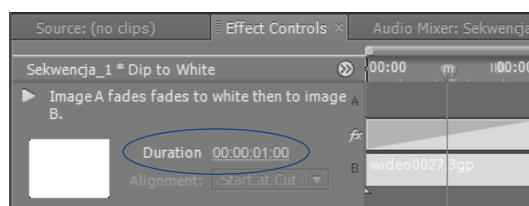
Rysunek 7.26.

Napołożenie efektu  
Dip to Black  
na końcu filmu



Rysunek 7.27.

Czas trwania efektu  
w panelu  
Effect Controls

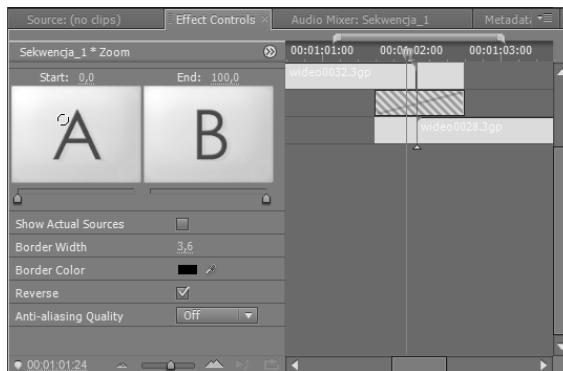


## Najczęściej stosowane efekty

- Efekt *Dip to Black* — przejście z obrazu do koloru czarnego. Efekt ten jest zwykle umieszczany na końcu filmu, ponieważ powoduje łagodne wygaszenie obrazu. Może być również umieszczany na początku filmu lub jako przejście między klipami.

- Efekt **Cross Dissolve** — łagodne przejście z jednego kadru w drugi. Efekt jest nakładany na przejście między klipami.
- Efekt **Dip to White** — przejście z obrazu do koloru białego. Jeżeli znacznie skrócimy czas trwania efektu, to uzyskamy efekt błysku między dwoma kadrami. Efekt jest nakładany na przejście między klipami.

Efektów stosowanych do wideo jest bardzo dużo. W trakcie pracy nad filmem zwykle przegląda się i testuje te, które mogłyby zostać użyte. Dostępne są też efekty z dodatkowymi parametrami, które można modyfikować w panelu *Effect Controls* (rysunek 7.28).



**Rysunek 7.28.** Okno Effect Controls

### Zadanie 7.5

Wybierz efekty przejść i dodaj je do tworzonego filmu *Spacer po moim mieście*. Przetestuj ich działanie.

## 7.6. Pytania i zadania

### 7.6.1. Pytania

1. Wymień kilka podstawowych formatów plików audio.
2. Na czym polega kompresja stratna plików audio?
3. Na czym polega kompresja bezstratna plików audio?
4. Wymień standardowe rozdzielczości obrazu w plikach wideo.
5. Wymień podstawowe formaty plików wideo.
6. Na czym polega montaż materiału wideo?
7. Wymień kilka efektów dostępnych w aplikacji Adobe Premiere, które można stosować podczas montażu materiału wideo.

## Pytania egzaminacyjne

### Zadanie 1.

Który z formatów zapewnia największą redukcję rozmiaru pliku dźwiękowego?

- A.** WAV.
- B.** PCM.
- C.** MP3.
- D.** CD-Audio.

### Zadanie 2.

Aby dopasować dźwięk do danego poziomu głośności, należy użyć efektu:

- A.** wyciszenia.
- B.** normalizacji.
- C.** podbicia basów.
- D.** usuwania szumów.

### Zadanie 3.

Formatem bezstratnej kompresji dźwięku jest:

- A.** MP3.
- B.** AAC.
- C.** WWA.
- D.** FLAC.

### Zadanie 4.

W czasie przetwarzania dźwięku, aby pozbyć się niechcianych odgłosów spowodowanych złą jakością mikrofonu, należy zastosować narzędzie:

- A.** echa.
- B.** obwiedni.
- C.** wyciszenia.
- D.** usuwania szumów.

### Zadanie 5.

Programem do edycji dźwięku jest:

- A.** Brasero.
- B.** Winamp.
- C.** Audacity.
- D.** RealPlayer.

### Zadanie 6.

Aby obraz w filmie zmieniał się płynnie, liczba klatek (nieprzenikających się wzajemnie) na sekundę musi się znajdować w zakresie:

- A.** 16–19 fps.
- B.** 20–23 fps.
- C.** 24–30 fps.
- D.** 31–36 fps.

### Zadanie 7.

W standardzie HDTV jest stosowana rozdzielcość:

- A.**  $704 \times 576$  px.
- B.**  $720 \times 480$  px.
- C.**  $1280 \times 1024$  px.
- D.**  $1920 \times 1080$  px.

## 7.6.2. Zadania

### Zadanie 1.

Na potrzeby tworzonej witryny internetowej opracuj pliki dźwiękowe, które mogą zostać umieszczone na stronie (np. krótkie rozmowy, informacje lub pliki muzyczne). Przygotuj również materiały filmowe. Mogą to być zdjęcia lub klipy filmowe miejsc, które należałyby pokazać na stronie internetowej. Pliki dźwiękowe odpowiednio zmodyfikuj. Na podstawie zgromadzonego materiału filmowego wykonaj również montaż filmu. Do filmu dodaj efekty przejść i przetestuj ich działanie.

# 8

# Systemy zarządzania treścią (CMS)

## 8.1. Wprowadzenie

Systemy zarządzania treścią (CMS, ang. *Content Management System*) to aplikacje internetowe, które pozwalają na łatwe tworzenie i modyfikowanie oraz rozbudowę strony internetowej. Systemy CMS generują przy wykorzystaniu odpowiednich szablonów gotowe strony internetowe. Tworzenie strony oparte na szablonach pozwala na łatwą i szybką zmianę jej wyglądu, aktualizowanie i późniejszą rozbudowę. Tworzenie treści i sposobu ich prezentacji odbywa się za pomocą prostego w obsłudze interfejsu użytkownika, zazwyczaj poprzez rozbudowane formularze i moduły.

Systemy zarządzania treścią budowane są zgodnie z zasadą oddzielenia treści (zawartości informacyjnej witryny) od wyglądu (sposobu jej prezentacji). Wprowadzane treści są zapisywane w bazie danych. Następnie system CMS generuje dynamicznie strony internetowe na podstawie treści pochodzących z bazy danych oraz odpowiednio opracowanych szablonów. Pozwala to na łatwe i elastyczne zarządzanie treścią.

Tworzenie stron oparte na wykorzystaniu szablonów sprawia, że zmiana wyglądu witryny internetowej polega na przygotowaniu i wymianie szablonu. Dodatkowo te same dane mogą być prezentowane jednocześnie w różny sposób.

### 8.1.1. Dostępne systemy zarządzania treścią

Do systemów zarządzania treścią należą m.in.: WordPress, Joomla! oraz Drupal.

## WordPress

WordPress jest najpopularniejszym darmowym CMS do prowadzenia blogów, tworzenia galerii zdjęć i stron internetowych. Pozwala na dowolną edycję treści, a dzięki dostępności darmowych wtyczek oraz szablonów umożliwia tworzenie stron internetowych bez znajomości języków HTML i CSS. W pierwotnej wersji został zaprojektowany głównie do obsługi blogów.

Wykorzystuje język PHP i współpracuje z bazą danych MySQL. Rozpowszechniany jest na licencji GNU i jest dostępny bezpłatnie.

## Joomla!

Joomla! (wymowa „dżumla”) to system zarządzania treścią napisany w języku PHP, wykorzystujący bazę danych MySQL, rozprowadzany na zasadach wolnego oprogramowania.

System Joomla! został wyposażony w funkcje umożliwiające tworzenie wielu kont użytkowników i zarządzanie uprawnieniami użytkowników do różnych części witryny, tworzenie banerów reklamowych i zarządzanie nimi oraz tworzenie wielopoziomowych menu.

## Drupal

Drupal to zaawansowany system zarządzania treścią i framework. Służy głównie do tworzenia blogów, portali informacyjnych oraz stron biznesowych i edukacyjnych. Wyposażony został w funkcje obsługi blogów i wspólnej pracy nad projektem. Za jego pomocą można tworzyć galerie zdjęć, bieletyny i fora dyskusyjne. Automatycznie tworzy kanały informacyjne RSS. Umożliwia też proste ustalanie aliasów, tak aby adresy stron były łatwiejsze do zapamiętania przez użytkowników. Posiada narzędzia pozwalające na łatwą zmianę wyglądu strony.

Drupal jest oprogramowaniem o otwartym kodzie źródłowym, dlatego możliwe jest rozszerzanie jego funkcjonalności za pomocą tzw. modułów. Niektóre z nich są dostępne od razu po instalacji, inne trzeba dodatkowo pobrać i zainstalować.

Drupal wykorzystuje język PHP, współpracuje z bazami danych takimi jak: MySQL, PostgreSQL i Microsoft SQL Server za pośrednictwem biblioteki PDO oraz serwerami, na których można uruchomić PHP, np. Apache, IIS.

Najnowsza wersja, Drupal 8, charakteryzuje się wysoką wydajnością, modułową architekturą, przyjaznym interfejsem użytkownika, responsywnym panelem zarządzania i prostym zarządzaniem elementami na stronie.

## 8.2. WordPress

WordPress jest bezpłatną aplikacją blogową, która często jest traktowana jak CMS i jest używana do tworzenia stron internetowych. Główną zaletą WordPressa jest jego prostota i intuicyjność. Mimo że jest bardziej blogiem niż typową platformą CMS, to

poprzez ciągłą modernizację i dostępność tysięcy wtyczek stał się narzędziem, które coraz bardziej przypomina CMS. Część wtyczek jest dostępna za darmo na stronie platformy. Na stronie tej dostępne są również szablony graficzne stron internetowych. Część z nich jest płatna, inne są bezpłatne.

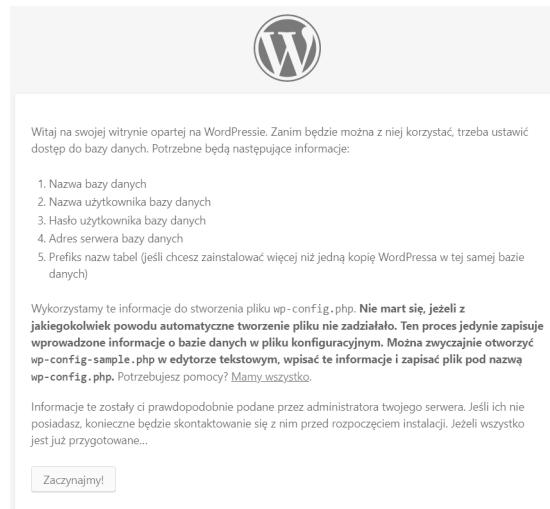
Aby WordPress był podobny pod względem funkcjonalności do innych platform CMS (Joomla!, Drupal), wymaga instalacji wielu wtyczek, co wpływa niekorzystnie na wydajność aplikacji.

## 8.2.1. Instalacja programu WordPress na serwerze

Plik instalacyjny pakietu WordPress można pobrać ze strony <https://pl.wordpress.org/>. Po pobraniu tego pliku należy go rozpakować, a następnie przesłać wszystkie pliki za pomocą klienta FTP na serwer. Pliki powinny zostać przesłane do katalogu publicznego serwera, z którego korzystamy (np. `public_html` lub `www`).

Kolejnym krokiem powinno być utworzenie bazy danych, w której aplikacja WordPress będzie przechowywała dane. Sposób tworzenia bazy danych zależy od serwera, z którego korzystamy (np. w panelu administracyjnym, na stronie głównej, w sekcji `Bazy danych MySQL` może znajdować się ikonka `Utwórz bazę`). Po utworzeniu bazy danych należy utworzyć jej użytkownika. Wybieramy `login` i `hasło` i klikamy przycisk `Utwórz użytkownika`. Pozostało jeszcze nadanie utworzonemu użytkownikowi pełnych uprawnień do bazy danych.

Następnym krokiem będzie instalacja aplikacji WordPress. Aby uruchomić instalator tego programu w przeglądarce, należy wpisać adres tworzonej strony internetowej. Po wpisaniu adresu zostanie wyświetlona strona instalatora programu WordPress (rysunek 8.1).



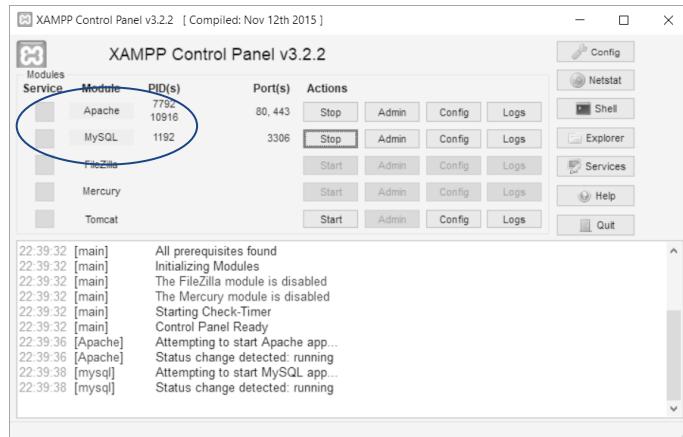
**Rysunek 8.1.** Instalator programu WordPress

W kolejnym oknie instalacji podajemy dane dotyczące utworzonej bazy i jej użytkownika. W polu *Adres serwera bazy danych* powinniśmy wpisać adres serwera bazodanowego. Najczęściej domyślna wartość *localhost* wystarczy, ale niektóre firmy hostingowe wymagają podania innego adresu. Aby uzyskać taki adres, należy zgłosić się do usługodawcy. Jeżeli wprowadzone dane są prawidłowe, będzie można przejść do kolejnego okna instalowania. W oknie tym podajemy tytuł strony (pole *Tytuł witryny*) i tworzymy konto głównego administratora strony (pole *Nazwa użytkownika* oraz pole *Hasełko*). Po wprowadzeniu tych danych i wciśnięciu przycisku *Zainstaluj WordPressa* program zostanie zainstalowany.

## 8.2.2. Instalacja programu WordPress lokalnie

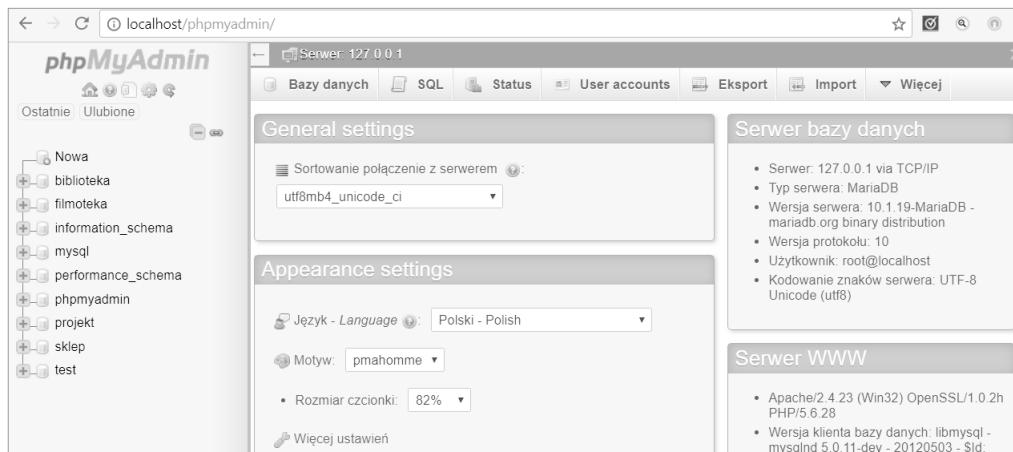
### Przygotowanie instalacji

W celu przetestowania możliwości programu WordPress lub nowych funkcjonalności tworzonej strony internetowej można zainstalować program na lokalnym komputerze. Aby to zrobić, na komputerze musi być zainstalowany serwer obsługujący PHP oraz serwer baz danych MySQL. Jednym z rozwiązań jest zainstalowanie pakietu XAMPP, który zawiera wszystkie potrzebne składniki (instalacja pakietu XAMPP została omówiona w czwartej części podręcznika). Po zainstalowaniu pakietu należy uruchomić serwer WWW Apache oraz serwer baz danych MySQL (rysunek 8.2).



**Rysunek 8.2.** Uruchomienie serwera Apache oraz serwera MySQL

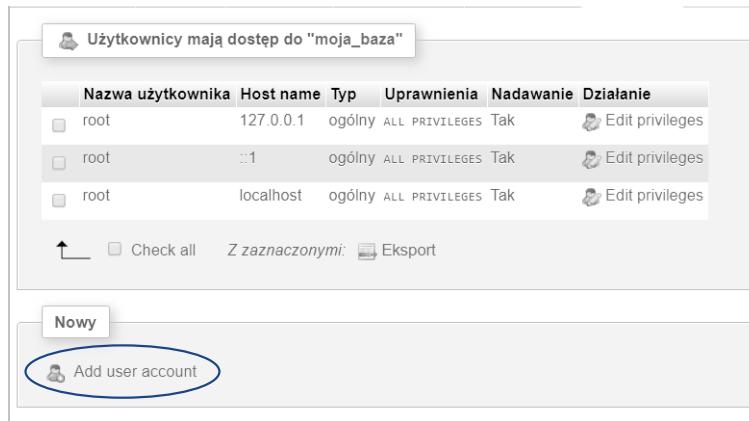
Po uruchomieniu serwerów należy utworzyć bazę danych. W pakiecie XAMPP bazę danych tworzy się i obsługuje za pomocą narzędzia **phpMyAdmin**. Należy otworzyć dowolną przeglądarkę i wpisać adres <http://localhost/phpmyadmin/>. Zostanie uruchomiony program phpMyAdmin (rysunek 8.3).



**Rysunek 8.3.** Okno programu phpMyAdmin

W prawym panelu wybieramy zakładkę **Bazy danych** i w otwartym oknie wybieramy opcję **Utwórz bazę danych**. Wprowadzamy nazwę bazy i klikamy przycisk **Utwórz**.

W lewym panelu wybieramy nowo utworzoną bazę danych i w górnym menu wybieramy zakładkę **Uprawnienia**. Następnie w otwartym oknie wybieramy opcję **Add user account**, aby dodać nowego użytkownika bazy (rysunek 8.4).



**Rysunek 8.4.** Zakładka Uprawnienia w panelu programu phpMyAdmin

Zostanie otwarte kolejne okno, w którym należy podać dane tworzonego użytkownika i nadać mu pełne prawa do utworzonej bazy danych (rysunek 8.5).

**Dane użytkownika**

Nazwa użytkownika:	<input type="text" value="oskar"/> Użyj pola tekstowego	<input type="text" value="oskar"/>
Host name:	<input type="text" value="Host lokalny"/> Host lokalny	<input type="text" value="localhost"/> localhost
Hasło:	<input type="text" value="Brak hasła"/> Brak hasła	<input type="password"/>
Powtórz:	<input type="password"/>	
Authentication Plugin	<input type="text" value="Native MySQL authentication"/> Native MySQL authentication	
Wygeneruj hasło:	<input type="button" value="Generuj"/>	<input type="password"/>

**Database for user account**

Utwórz bazę danych z taką samą nazwą i przyznaj wszystkie uprawnienia.

Przyznaj wszystkie uprawnienia do baz danych o nazwach pasujących do maski (nazwaużytkownika\_%).

Przyznanie wszystkich uprawnień do bazy danych "moja\_baza".

**Rysunek 8.5.** Tworzenie nowego użytkownika bazy danych i nadanie mu pełnych uprawnień

## Przebieg instalacji

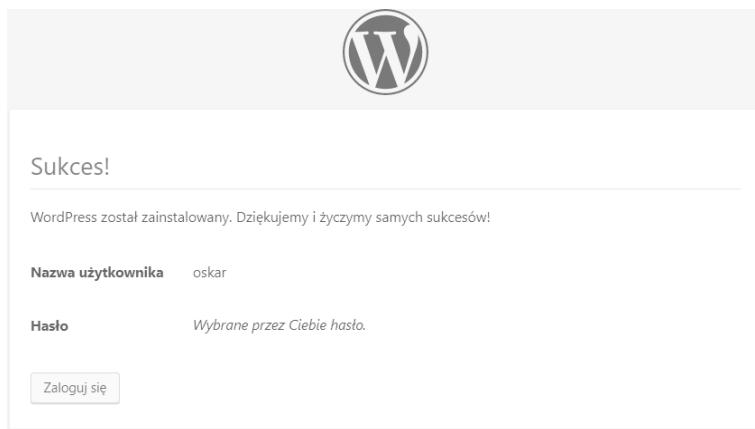
Następnie przechodzimy do katalogu `C:\xampp\htdocs`. Tworzymy w nim nowy katalog, np. `C:\xampp\htdocs\wpress`, i kopujemy bezpośrednio do niego wszystkie pliki instalacyjne WordPressa. Po przekopiowaniu plików wpisujemy w przeglądarce adres, np. <http://localhost/wpress/>.

Jeżeli wszystkie wykonywane czynności przebiegały prawidłowo, zostanie wyświetlone okno instalatora programu WordPress. W dostępnych polach należy wpisać tytuł strony internetowej oraz dane użytkownika (administratora strony). Po przejściu do kolejnego okna zostanie wyświetlony komunikat, który pozwoli rozpoczęć instalację programu (rysunek 8.6).



**Rysunek 8.6.** Okno instalatora WordPressa

Po podaniu informacji na temat użytkownika program zostanie zainstalowany i pojawi się podsumowanie instalacji (rysunek 8.7).



**Rysunek 8.7.** Podsumowanie instalacji programu WordPress

Po pomyślnym zainstalowaniu programu można przystąpić do pracy z WordPressem.

### 8.2.3. Zarządzanie panelem administratora

Aby rozpocząć pracę z programem WordPress i uzyskać dostęp do panelu administratora strony internetowej, należy się zalogować. W tym celu trzeba w przeglądarce internetowej wpisać adres swojej strony internetowej i dodać na końcu adresu tekst `wp-admin` (np. <http://domena.com/wp-admin>) lub — jeżeli pracujemy w trybie lokalnym — wpisać adres <http://localhost/katalog/wp-admin/>, gdzie katalog to miejsce, do którego zostały skopiowane pliki programu WordPress. Zostaniemy przekierowani do panelu administratora programu.

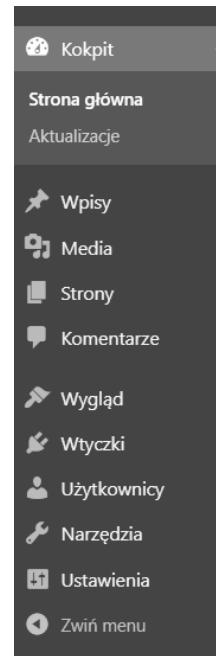
#### Panel administratora

Panel administratora to miejsce, gdzie można zorganizować i zaprojektować swoją stronę internetową. Po lewej stronie znajduje się główne menu nawigacyjne, zawierające dostępne funkcje administracyjne (rysunek 8.8).

#### Kokpit

Główna część okna panelu administratora (*Kokpit*) składa się z kilku części (*widgetów*), którymi są:

- *Panel powitalny*,
- *Aktywność*,
- *Szybki szkic*,
- *Wydarzenia i nowości*,
- *W skrócie*.



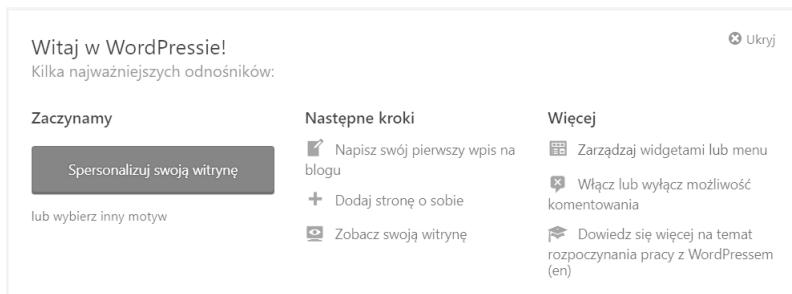
**Rysunek 8.8.**

Menu nawigacyjne panelu administratora programu



## Panel powitalny

W panelu powitalnym znajdują się linki, które prowadzą do dostępnych funkcji i ustawień programu WordPress (rysunek 8.9).



**Rysunek 8.9.** Panel powitalny programu WordPress

## Aktywność

Panel *Aktywność* pokazuje ostatnią aktywność na obsługiwanej stronie internetowej. W panelu tym można odczytać treść komentarza oraz informację o tym, przez kogo komentarz został dodany. Można również za pomocą dołączonych linków przejść do moderacji komentarzy (rysunek 8.10).

The screenshot shows the 'Activity' section of the dashboard. It displays a recent comment from a user named 'Cześć', with a link to view it in the comments section. Below this, there are links for 'All' (1), 'Pending' (0), 'Approved' (1), 'Spam' (0), and 'Trash' (0).

**Rysunek 8.10.**  
Panel Aktywność programu WordPress

## Szybki szkic

Panel *Szybki szkic* daje możliwość stworzenia szybkiego szkicu dla prostej publikacji (rysunek 8.11).

The screenshot shows the 'Quick Draft' editor. It has a title field, a text area for 'What are you thinking?', and a 'Save Draft' button at the bottom.

**Rysunek 8.11.**  
Panel Szybki szkic programu WordPress

Aby utworzyć szybki szkic, należy wypełnić pole *Tytuł*, podając tytuł tworzonej publikacji, oraz wprowadzić jej krótką treść. Po zapisaniu szkicu pojawi się pod spodem jego zawartość (rysunek 8.12). Tytuł wpisu zmieni się w link, który przekierowuje do pełnego edytora, gdzie można tworzyć nowe wpisy oraz artykuły.

Szybki szkic

O czym myślisz?

Zapisz szkic

Twoje ostatnie szkice

Przemysł 4.0 21 kwietnia 2018  
Czwarta rewolucja przemysłowa to zanikanie barier między ludźmi i maszynami...

**Rysunek 8.12.**

Zawartość utworzonego szkicu

## Wydarzenia i nowości

W panelu *Wydarzenia i nowości* można zobaczyć najnowsze wiadomości z oficjalnego bloga WordPressa. Informuje on na bieżąco m.in. o aktualizacjach do najnowszej wersji (rysunek 8.13).

Wydarzenia i nowości

Weź udział w nadchodzących wydarzeniach w pobliżu ciebie. ↗

WordCamp Poznań 6.7.2018, piątek  
Poznań, Poland

WordPress 4.9.4 – Wydanie poprawiające ważny błąd  
WordPress 4.9.4 – Wydanie poprawiające ważny błąd  
WordPress 4.9.3 – Wydanie poprawiające błędy  
WordPress 4.9.2 – Wydanie poprawiające bezpieczeństwo i błędy

Spotkania ↗ | WordCampy ↗ | Wiadomości ↗

**Rysunek 8.13.**

Panel Wydarzenia i nowości programu WordPress

## W skrócie

W panelu *W skrócie* znajduje się krótkie podsumowanie dotyczące strony tworzonej w WordPressie. Umieszczony jest tutaj licznik wpisów, stron oraz komentarzy. Na samym dole wyświetlana jest wersja programu WordPress oraz nazwa wykorzystywanego motywu (rysunek 8.14).

W skrócie

1 wpis 1 strona

1 komentarz

Używasz WordPressa 4.9.5 z motywem Twenty Seventeen.

**Rysunek 8.14.**

Panel W skrócie programu WordPress

## Opcje

W górnej części okna z prawej strony znajdują się dwie zakładki: *Opcje ekranu* oraz *Pomoc*.

## Opcje ekranu

W zakładce *Opcje ekranu* można modyfikować sposób wyświetlania paneli w kokpicie. Aby usunąć lub dodać aktywny panel, wystarczy zaznaczyć lub odhaczyć wybrany element.

## Pomoc

Zakładka *Pomoc* zawiera zbiór informacji dotyczących sposobu korzystania z programu WordPress.

## 8.2.4. Tworzenie bloga

### Ćwiczenie 8.1

Przygotuj stronę internetową, która będzie zawierała blog poświęcony rewolucji przemysłowej 4.0. Na stronie bloga znajdą się wpisy autora na ten temat. Będzie możliwe przeglądanie wpisów oraz dodawanie komentarzy.

### Pasek narzędzi

W górnej części okna panelu administratora znajduje się pasek narzędzi. Zawiera on łącza do informacji o programie, a także szybkie łącza do tworzenia nowych wpisów, stron i linków, dodawania nowych wtyczek i użytkowników, przeglądania komentarzy i alertów do dostępnych aktualizacji wtyczek i motywów na tworzonej stronie.

Po kliknięciu linku z nazwą tworzonej strony zostanie wyświetlona jej zawartość. Układ, który jest widoczny na stronie (rysunek 8.15), to jeden z dostępnych w programie gotowych motywów. U góry znajduje się nagłówek z tytułem strony, niżej — obszar zawartości, z prawej strony mamy kilka tytułów i linki, a na dole — stopkę.

**Rysunek 8.15.**

Układ strony utworzonej w programie WordPress



## Użytkownicy

Po wybraniu w menu głównym panelu administratora opcji *Użytkownicy* zostanie wyświetcone nowe okno z listą wszystkich użytkowników. W oknie tym można dodawać i zmieniać istniejące konta użytkowników i autorów.

Po kliknięciu opcji menu *Twój profil* zostanie wyświetlone okno, w którym można zmieniać informacje o autorze i administratorze strony. Po zmodyfikowaniu ustawień należy kliknąć przycisk *Aktualizuj profil*, aby zapisać wprowadzone zmiany.

## Zmiana wyglądu strony

### Motyw

*Motyw* to ogólny projekt witryny, który obejmuje kolor, grafikę i tekst.

Po wybraniu w menu głównym panelu administratora opcji *Wygląd/Motyw* można zmienić wygląd strony przy użyciu dostępnych stylów prezentacji. Aby szybko zmienić motyw, należy kliknąć przycisk *Włącz* pod wybranym motywem. Aby podejrzeć, jak wygląda strona z nowym motywem, należy kliknąć nazwę swojej strony na pasku narzędzi.

### Nagłówek

Opcja *Nagłówek* umożliwia zarządzanie obrazem wyświetlany w nagłówku wybranego motywu. Opcja ta jest dostępna tylko wtedy, gdy w tworzonej kompozycji jest możliwe konfigurowanie nagłówka.

## Zarządzanie postami

*Wpisy* (posty) dotyczą wpisów na blogach. Wpisy to pisma, kompozycje, dyskusje i dyskursy właściciela bloga i współautorów. Są one pokazywane w kolejności powstawania od najnowszego wpisu do najstarszego. Wpisy posiadają opcje społecznościowe, istnieje też możliwość ich komentowania. Każdy utworzony wpis jest również widoczny i aktualizowany na kanale RSS.

Aby dodać nowy post w panelu administratora, z menu głównego wybieramy opcję *Wpisy*. Do zarządzania wpisami służą następujące podopcje: *Wszystkie wpisy*, *Dodaj nowy*, *Kategorie*, *Tagi*.

Po wybraniu opcji *Dodaj nowy* (rysunek 8.16) zostanie otwarte okno do tworzenia nowych wpisów. Należy wprowadzić tytuł wpisu oraz jego treść. Po wprowadzeniu treści należy kliknąć przycisk *Opublikuj*, żeby wprowadzone zmiany zostały zapisane na stronie.

**Rysunek 8.16.**

Nowy wpis na stronę internetową



Po wpisaniu kilku postów można je obejrzeć na stronie, wybierając na pasku narzędzi jej nazwę (rysunek 8.17).

The screenshot shows a blog homepage with two main posts:

- Internet rzeczy** (26 maja 2018, ośkar, 0 kom.)
- Przyszłość** (21 kwietnia 2018, ośkar, 0 kom.)

The sidebar contains the following sections:

- NAJNOWSZE WPISY**: Internet rzeczy, Przyszłość, Nie tylko w przemyśle
- NAJNOWSZE KOMENTARZEE**: ośkar o Przyszłość
- ARCHIWA**: Maj 2018

**Rysunek 8.17.** Podgląd strony internetowej — dodane wpisy

Po wybraniu w panelu administratora opcji *Wszystkie posty* można wybrać post lub posty, które będą edytowane, zostaną usunięte albo wyświetlane. Różne opcje wyszukiwania i filtrowania pozwalają znaleźć posty, które mają być edytowane lub powinny zostać usunięte.

Opcja *Kategorie* umożliwia klasyfikację istniejących wpisów i tworzenie dla nich hierarchii grup i podgrup.

Opcja *Tagi* pozwala wyodrębnić słowa kluczowe występujące we wpisach.

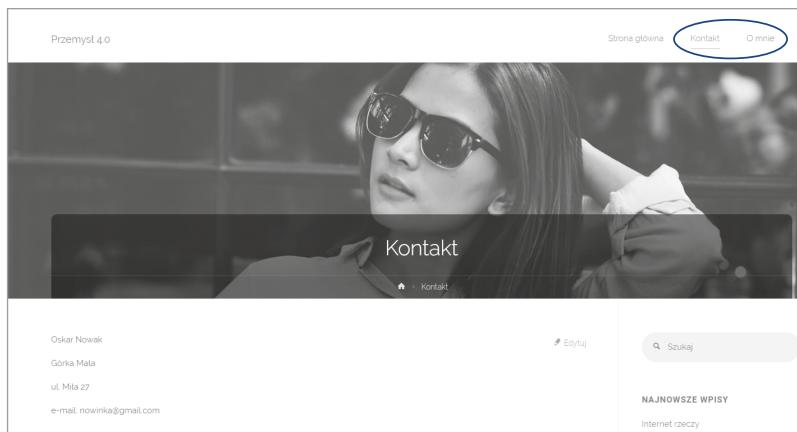
## Strony (statyczna zawartość)

Sekcja *Strony* jest kolejnym dostępnym w menu głównym narzędziem, za pomocą którego można dodawać zawartość do WordPressa. Służy do przedstawienia statycznych informacji typu kontakt lub opis. Strony funkcjonują poza normalną chronologią blogów i w związku z tym nie są wyświetlane wraz z postami, lecz oddzielnie.

Po wybraniu opcji *Dodaj nową* zostanie otwarte okno do tworzenia nowych stron. Należy wprowadzić tytuł strony oraz jej treść. Po wprowadzeniu treści należy kliknąć przycisk *Opublikuj*.

Opcja *Wszystkie strony* pozwala wybrać stronę, która będzie edytowana lub zostanie usunięta. Do usunięcia i edycji można wybrać wiele stron. Podobnie jak w przypadku postów, narzędzia do edycji pozwalają aktualizować wiele pól dla pojedynczej strony.

Utworzona nowa strona zostanie dodana do strony WordPressa ze zdefiniowaną treścią (rysunek 8.18).



**Rysunek 8.18.** Podgląd strony internetowej — dodane nowe podstrony

## Komentarze — opinie czytelników

Komentarze są tą funkcją blogów, która pozwala czytelnikom odpowidać na posty. Zazwyczaj czytelnicy przedstawiają swoje uwagi dotyczące treści wstawionego wpisu, ale użytkownicy mogą również udostępniać linki do innych zasobów lub rozpoczęć dyskusję.

Komentarze mogą być kontrolowane przez autora za pomocą filtrów dla języka i treści, mogą być także udostępniane dopiero po zatwierdzeniu przez administratora.

Po wybraniu z menu głównego opcji *Komentarze* zostanie wyświetcone okno z listą dodanych komentarzy. Komentarze można edytować, usuwać lub oznaczać jako spam (rysunek 8.19). Komentarze oczekujące na moderację można oznaczyć jako zatwierdzone. Można zaznaczyć i zatwierdzić wiele komentarzy. W górnej części okna komentarzy wyświetlana jest liczba oczekujących na moderację i liczba zatwierdzonych komentarzy.

**Rysunek 8.19.** Okno komentarzy dodanych na stronie

## Ustawienia

Za pomocą sekcji *Ustawienia* dostępnej w panelu głównym można konfigurować podstawowe ustawienia tworzonej strony. Są tu dostępne opcje: *Ogólne*, *Pisanie*, *Czytanie*, *Dyskusja*, *Media*, *Bezpośrednie odnośniki*.

### Ogólne

Wybranie opcji *Ogólne* pozwoli na skonfigurowanie podstawowych ustawień naszej strony, takich jak: tytuł strony, adres e-mail, na który będą wysyłane m.in. powiadomienia, konfiguracja sposobu wyświetlania daty i czasu na stronie.

### Pisanie

Opcja *Pisanie* zawiera ustawienia związane z tworzeniem wpisów.

### Czytanie

Opcja *Czytanie* zawiera ustawienia dotyczące sposobu wyświetlania wpisów oraz ich widoczności.

### Dyskusja

Po wybraniu opcji *Dyskusja* można zdefiniować ustawienia dotyczące komentarzy i ich moderacji. Są tu dostępne opcje, które pozwalają na włączanie lub blokowanie komentarzy do tworzonych wpisów, określenie, kto może komentować wpisy, oraz na włączenie moderacji komentarzy.

### Media

Opcja *Media* zawiera ustawienia związane z wyświetlaniem multimedialów i zarządzaniem nimi.

### Bezpośrednie odnośniki

Opcja *Bezpośrednie odnośniki* zawiera ustawienia dotyczące formatu linków i pozwala na definiowanie własnych schematów linków dostępnych na stronie.

## Wtyczki

Wtyczki umożliwiają dodawanie nowych funkcji, które standardowo nie są dostarczane z domyślną instalacją programu WordPress.

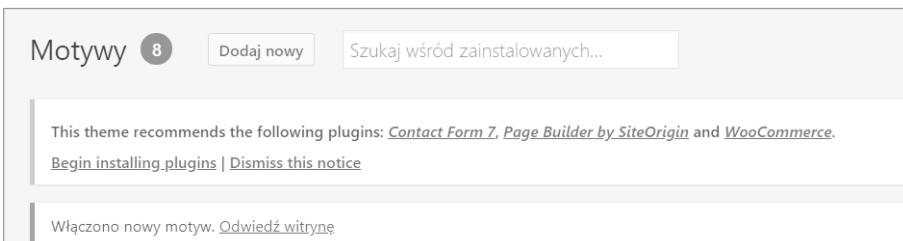
Wybierając w menu głównym opcję *Wtyczki*, można przeglądać zainstalowane wtyczki, a po wybraniu opcji *Dodaj nową* można doinstalować potrzebną wtyczkę poprzez wybranie jej z listy dostępnych wtyczek.

## 8.2.5. Tworzenie strony internetowej

### Ćwiczenie 8.2

Utwórz witrynę internetową z najświeższymi informacjami na temat nowoczesnych technologii. Jedna z zaprojektowanych stron będzie zawierała wcześniej opracowany blog dotyczący przemysłu 4.0. W skład witryny internetowej wejdą także strona startowa, strona zawierająca informacje o nowych technologiach, strona z galerią zdjęć oraz strona z kontaktem do autora witryny.

Po zalogowaniu się do programu WordPress określmy wygląd strony poprzez wybór odpowiedniego motywu. W menu głównym wybieramy *Wygląd/Motyw*. Zostanie wyświetcone okno z dostępnymi motywami. Jeżeli żaden z nich nam nie odpowiada, możemy kliknąć w górnej części okna przycisk *Dodaj nowy*, a zostanie wyświetlona kolekcja innych motywów. Można również w polu *Szukaj motywu* wpisać odpowiednią frazę, aby wyświetlić motywy z nią związane. Wprowadzamy tekst *magazine* i spośród motywów wybieramy np. *timagazine*. Przed zainstalowaniem motywu można wybrać opcję *Szczegóły i podgląd*, aby zapoznać się z opisem motywu i zobaczyć jego podgląd. W celu zainstalowania motywu klikamy przycisk *Zainstaluj*. Po zainstalowaniu motywu najczęściej pojawia się okno, które informuje, że do motywu zalecane są określone wtyczki (rysunek 8.20). Klikamy odnośnik *Begin installing plugins* i w otwartym oknie zaznaczamy wtyczki wybrane do zainstalowania.



**Rysunek 8.20.** Lista wtyczek do zainstalowania po wybraniu motywu

Aby zwiększyć możliwości programu WordPress, można również doinstalować wtyczki, wybierając z menu głównego *Wtyczki/Dodaj nową*. Jedną z polecanych jest wtyczka *Elementor*, która pomaga w tworzeniu i umieszczaniu na stronie elementów takich jak: kolumny, nagłówki, obrazy, przyciski czy ikony. Wpisujemy w polu *Szukaj wtyczek Elementor* i po wybraniu wtyczki instalujemy ją, klikając przycisk *Zainstaluj teraz*, a następnie przycisk *Włącz*. Na liście zainstalowanych wtyczek zostaną wyświetlane wtyczki ostatnio zainstalowane (rysunek 8.21).

Wtyczka	Opis
<input type="checkbox"/> Akismet antyspam Włącz   Usuń	Used by millions, Akismet is quite possibly the best way in the world to protect your blog from spam. It keeps your site protected even while you sleep. To get started: activate the Akismet plugin and then go to your Akismet Settings page to set up your API key. <a href="#">Wersja 4.0.3</a>   Autor: Automatic   Szczegóły
<input type="checkbox"/> Contact Form 7 Ustawienia   Wyłącz	Kolejna wtyczka formularzy kontaktowych. Prosta, lecz elastyczna. <a href="#">Wersja 5.0.1</a>   Autor: Takayuki Miyoshi   Szczegóły
<input type="checkbox"/> Elementor Ustawienia   Wyłącz   Poznaj PRO	Najbardziej zaawansowany edytor stron "przeciągnij i upuść" w trybie "live". Twórz z rekordową prędkością wysokiej jakości strony dopracowane co do piksela. Współpracuje z każdym motywem i każdym projektem. <a href="#">Wersja 2.0.8</a>   Autor: Elementor.com   Szczegóły   Dokumentacja i FAQ   Poradniki Wideo
<input type="checkbox"/> Hello Dolly Włącz   Usuń	To nie jest zwykła wtyczka – symbolizuje ona nadzieję i entuzjazm całego pokolenia, podsumowane w dwóch słowach piosenki, której najpopularniejszym wykonawcą jest Louis Armstrong: „Hello, Dolly”. Po włączeniu tej wtyczki, w prawym górnym rogu każdego ekranu kokpitu będzie wyświetlany losowy wers Hello, Dolly. <a href="#">Wersja 1.7</a>   Autor: Matt Mullenweg   Szczegóły
<input type="checkbox"/> Page Builder by SiteOrigin Wyłącz   Support Forum   Newsletter   Addons	A drag and drop, responsive page builder that simplifies building your website. <a href="#">Wersja 2.6.5</a>   Autor: SiteOrigin   Szczegóły
<input type="checkbox"/> WooCommerce Ustawienia   Wyłącz	Zestaw narzędzi e-commerce, który pomoże ci sprzedać wszystko. Pięknie. <a href="#">Wersja 3.3.5</a>   Autor: Automatic   Szczegóły   Dokumentacja   Dokumentacja API   Wsparcie Premium

**Rysunek 8.21.** Lista zainstalowanych wtyczek w programie WordPress

Nastepnym krokiem jest utworzenie stron, które będą wchodziły w skład witryny internetowej. Wybieramy z menu głównego *Strony/Dodaj nową*, wprowadzamy nazwę strony *Start* i klikamy przycisk *Opublikuj*. Podobnie postępujemy z kolejnymi stronami, tworząc strony: *Kontakt, Nowe, Galeria i Blog*.

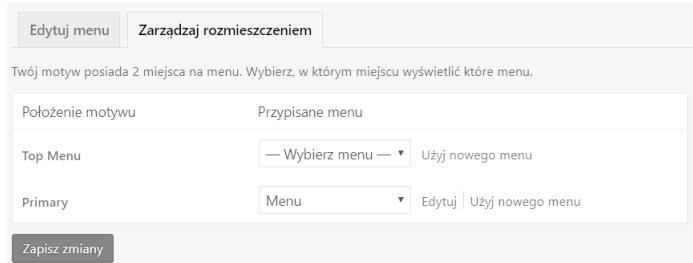
Aby przechodzić między stronami, utworzymy menu. Wybieramy w menu głównym *Wygląd/Menu*. Następnie w otwartym oknie, w polu *Nazwa menu*, wpisujemy jego nazwę, np. *Menu*, i klikamy przycisk *Zapisz*. W obszarze *Strony* wyświetlona została lista utworzonych stron naszej witryny. Zaznaczamy wszystkie pozycje listy i klikamy przycisk *Dodaj do menu* (rysunek 8.22).

Wybrane strony zostaną przeniesione do obszaru *Struktura menu*. W obszarze tym elementy można poukładać, lapiąc za wybrany element i przenosząc go na odpowiednią pozycję (rysunek 8.23). Po wprowadzeniu zmian zapisujemy je, klikając przycisk *Zapisz menu*.

**Rysunek 8.22.**  
Lista stron do menu

**Rysunek 8.23.**  
Struktura menu witryny internetowej

W kolejnym kroku należy określić, gdzie na stronie będzie wyświetlane menu. Przełączamy się na zakładkę **Zarządzaj rozmieszczeniem** i wybieramy położenie menu. Dla opcji **Primary** wybieramy z listy **Menu** (rysunek 8.24) i klikamy przycisk **Zapisz zmiany**.



**Rysunek 8.24.** Wybór położenia menu na stronie

Przechodzimy do strony i sprawdzamy, czy menu pojawiło się w jej górnej części (rysunek 8.25).

**Rysunek 8.25.**  
Wygląd tworzonej strony po dodaniu menu

Chcemy, aby po wpisaniu adresu w przeglądarce strona **Start** otwierała się jako pierwsza. Po wyświetleniu się zawartości strony internetowej wybieramy na pasku narzędzi opcję **Personalizacja**. W otwartym oknie wybieramy z menu **Ustawienia strony głównej** i w obszarze **Strona główna wyświetla** zaznaczamy opcję **statyczną stronę**. Jako stronę główną, wyświetlana po wpisaniu w przeglądarce adresu witryny, wybieramy z listy pozycję **Start**, a jako stronę z wpisami — pozycję **Blog** (rysunek 8.26) i klikamy przycisk **Opublikuj**, aby opublikować zmiany.

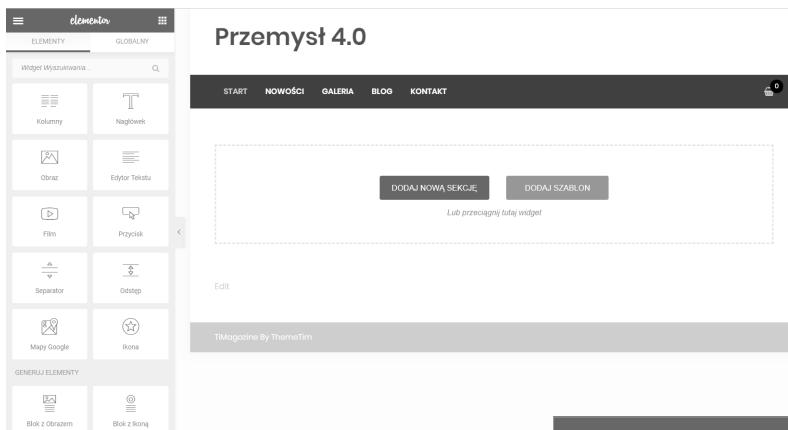
Zamykamy okno ustawień i przechodzimy do wyświetlania tworzonej strony.

**Rysunek 8.26.**

Ustawienia strony głównej

Następny etap pracy polega na zaprojektowaniu wyglądu strony. Działania te zostaną zrealizowane za pomocą zainstalowanej wtyczki *Elementor*.

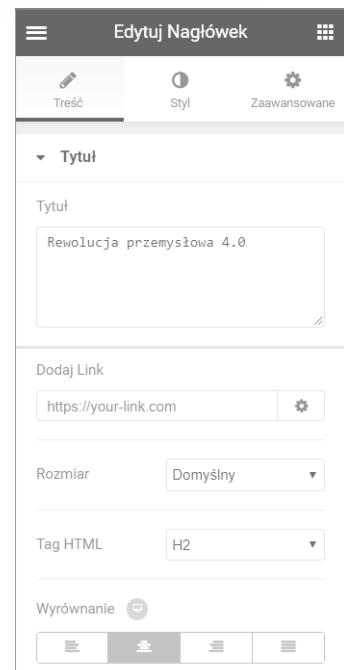
Wracamy do panelu administratora, wybieramy z menu głównego *Strony/Wszystkie strony*, zaznaczamy stronę *Start* i klikamy opcję *Edytuj*. W otwartym oknie klikamy przycisk *Edytuj w Elementorze*. Po uruchomieniu wtyczki pojawią się nowe narzędzia, które można wykorzystać przy tworzeniu strony. W otwartym oknie w panelu z lewej mamy zbiór elementów, które można umieszczać na stronie, a w centrum wyświetlana jest projektowana strona (rysunek 8.27).



**Rysunek 8.27.** Narzędzia wtyczki Elementor

Po kliknięciu przycisku *Dodaj nową sekcję* pojawi się zbiór struktur do umieszczenia na stronie. Pierwszym elementem, który chcemy umieścić, jest nagłówek z tytułem strony. Wybieramy pierwszą strukturę i aby umieścić w niej napis w bloku z elementami, wybieramy *Nagłówek* i przeciągamy go nad utworzoną strukturę. Wprowadzamy tekst nagłówka (rysunek 8.28). Można to zrobić bezpośrednio na stronie lub w bloku z lewej.

W opcji *Wyrównanie* wybieramy wyśrodkowanie tekstu i przechodzimy do zakładki *Styl*. W zakładce tej można wybrać kolor tekstu (*Kolor Tekstu*), parametry czcionki, takie jak: rodzina czcionek, rozmiar, grubość, styl i inne (*Typografia*), oraz cień (*Cień Tekstu*). Po ustaleniu parametrów zapisujemy stronę, klikając przycisk *UPDATE* w dolnej części bloku *Edytuj Nagłówek*. Tworzona strona została pokazana na rysunku 8.29.



**Rysunek 8.28.**  
Edytowanie tekstu wstawionego do nagłówka



**Rysunek 8.29.** Wygląd tworzonej strony

W celu usunięcia zbędnego napisu *Przemysł 4.0* oraz tekstu w stopce należy przejść do opcji *Personalizacja*.

Wracamy do panelu administratora i w menu głównym wybieramy *Wygląd/Personalizacja*. Tutaj również zostanie wyświetlony podgląd strony. Klikamy ikonę *Kliknij aby edytować element* przy tekście *Przemysł 4.0* (rysunek 8.30) i w panelu z lewej strony usuwamy tekst zapisany w polu *Tytuł witryny* oraz tekst zapisany w polu *Opis*. Klikamy przycisk *Opublikuj*, aby zapisać wprowadzone zmiany.



**Rysunek 8.30.** Fragment okna Personalizacja

Poprawimy jeszcze wygląd stopki. W lewym panelu wracamy do okna *Personalizujesz*, wybieramy opcję *Footer/Footer Bottom*, usuwamy tekst wpisany w polu *Footer Bottom Copyright Text* i wprowadzamy własny. W opcji *Footer Bottom BG Color* zmieniamy kolor tła. W opcji *Footer/Footer Top/Footer Top Background Color* zmieniamy kolor tła w górnej części stopki. W podobny sposób możemy postąpić z nagłówkiem, wybierając w oknie *Personalizujesz* opcję *Header*. Po wprowadzeniu zmian ponownie klikamy przycisk *Opublikuj* i przechodzimy do strony, aby sprawdzić jej wygląd.

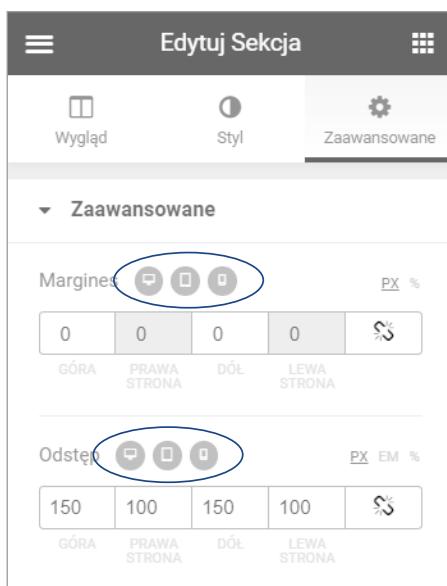
Do tworzonej strony zostanie dodane tło w postaci wcześniej przygotowanego obrazka. Ponownie wracamy do panelu administratora, wybieramy opcję *Strony/Wszystkie strony*, zaznaczamy stronę *Start* i ponieważ dostępna jest opcja *Edytuj w Elementorze*, wybieramy ją. Zaznaczamy utworzoną sekcję i wybieramy umieszoną nad nią ikonę *Edytuj Sekcję*. W panelu *Edytuj Sekcję*, w zakładce *Wygląd*, ustawiamy opcję *Rozciągnij sekcję* na *Tak*,

aby sekcja obejmowała całą stronę. Następnie przełączamy się na zakładkę *Styl* i w opcji *Rodzaj Tła* wybieramy *Klasyczny*. Po kliknięciu w obszarze *Obraz* zostanie otwarte okno *Załaduj Media*. Po wybraniu zakładki *Dodaj pliki* możemy odnaleźć plik, który zostanie umieszczony jako tło strony. Jeżeli nie posiadamy własnych grafik, możemy wykorzystać dostępne w internecie zdjęcia utworzone przez profesjonalistów. Po dodaniu zdjęcia należy ustawić parametry określające sposób jego wyświetlania na stronie. Wybieramy parametr *Rozmiar* i ustawiamy wartość *Pokryj*. Przechodzimy do zakładki *Zaawansowane*, ustawiamy *Margines PRAWA STRONA* i *LEWA STRONA* na wartość *0px*, *Odstęp GÓRA* na *150px*, *DÓŁ* na *150px*, a *PRAWA STRONA* i *LEWA STRONA* na *100px* (rysunek 8.31).

Zaznaczamy tekst umieszczony na stronie i w zakładce *Styl* zmieniamy jego rozmiar i kolor tak, aby był on widoczny na tle zdjęcia. Rezultat wprowadzanych zmian przy wyświetleniu strony na różnych urządzeniach możemy obserwować, wybierając ikony wyświetlane przy opcjach *Margines* i *Odstęp* (rysunek 8.31). Klikamy przycisk *UPDATE*, aby zapisać wprowadzone zmiany. Wracamy do panelu administratora, przechodzimy do strony i sprawdzamy jejgląd. Będzie on podobny do tego pokazanego na rysunku 8.32.

### Rysunek 8.32.

Wygląd strony projektowanej w programie WordPress

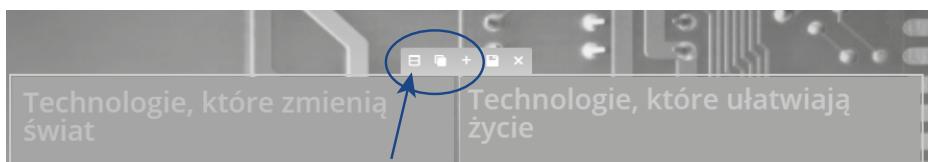


### Rysunek 8.31.

Ustawienie sposobu wyświetlania zdjęcia jako tła strony

Do utworzonej strony dodamy poniżej istniejącego nagłówka dwie kolumny z tekstem. Wracamy do edycji strony w *Elementorze*. W głównym oknie wybieramy przycisk *Dodaj nową sekcję* i wybieramy strukturę z dwiema kolumnami. Kolumny wypełnimy tekstem składającym się z nagłówka i treści. W panelu z elementami wybieramy *Nagłówek* i umieszczaemy go w pierwszej kolumnie, następnie wybieramy *Edytor tekstu* i umieszczaemy go poniżej *Nagłówka*. Dla utworzonego bloku ustawimy kolor tła i umieścimy odpowiednie teksty. Wybieramy ikonę *Edytuj Kolumna* i w panelu z edycją bloku wybieramy zakładkę *Styl*, a w niej opcję *Tło/Rodzaj Tła/Klasyczny* i ustawiamy wybrany kolor. Dodajemy do *Nagłówka* i treści przygotowane teksty. Następnie klikamy przycisk *UPDATE*, aby zapisać wprowadzone zmiany. Aby zachować ten sam styl ustawień w drugiej kolumnie, wystarczy skopiować pierwszą z nich i tylko zmienić jej treść. W panelu głównym wybieramy w lewym górnym rogu dla pierwszej kolumny ikonę *Duplikuj Kolumna*. Natomiast niepotrzebną trzecią kolumnę możemy usunąć. W drugiej kolumnie zmieniamy kolor tła i inne ustawienia treści. Zapisujemy zmiany i sprawdzamy wygląd strony.

Chcielibyśmy jeszcze zmienić położenie na stronie utworzonych bloków. Wracamy do edycji strony i zaznaczamy całą sekcję z kolumnami (rysunek 8.33).



**Rysunek 8.33.** Sposób zaznaczenia sekcji z kolumnami

W panelu *Edytuj Sekcję* wybieramy zakładkę *Zaawansowane* i w opcji *Margines* ustawiamy wartość górnego marginesu tak, aby sekcja z kolumnami nałożyła się na sekcję z nagłówkiem (rysunek 8.34).

Po ustawieniu marginesów zapisujemy zmiany.

W kolejnej sekcji umieścimy zdjęcie dotyczące tematu witryny. Tworzymy nową sekcję i wybieramy strukturę jednokolumnową. W panelu z elementami wybieramy blok *Obraz*, przeciągamy go do tworzonej sekcji i upuszczamy. W panelu *Edytuj Obraz* klikamy opcję *Wybierz Obraz* i w otwartym oknie wybieramy obraz wcześniej umieszczony w *Bibliotece mediów* lub po wybraniu zakładki *Dodaj pliki* szukamy pliku z potrzebnym obrazem.

Dodamy jeszcze do strony ikony mediów społecznościowych. Tworzymy nową sekcję i wybieramy strukturę jednokolumnową.



**Rysunek 8.34.**

Ustawienie marginesów sekcji z kolumnami

W panelu z elementami wybieramy blok *Ikonki Społecznościowe* i przeciągamy ikonki do utworzonej sekcji. Zostaną dodane ikony Facebooka, Twittera i Google Plus. Należy utworzyć do nich odnośniki. W tym celu zaznaczamy sekcję z ikonami, w panelu *Edytuj Ikonki Społecznościowe* wybieramy np. opcję *Facebook* i w polu *Dodaj Link* umieszczamy link do odpowiedniej strony (rysunek 8.35).

W podobny sposób można dodawać do projektowanej strony kolejne elementy dostępne we wtyczce *Elementor*. Ustawienia związane z wyglądem samego szablonu można modyfikować w menu głównym panelu administratora, wybierając opcję *Wygląd/Personalizacja*. Dostępne w tym miejscu opcje zależą od wybranego motywu, a jeżeli jest to motyw darmowy, część opcji może być zablokowana.

## 8.3. Joomla!

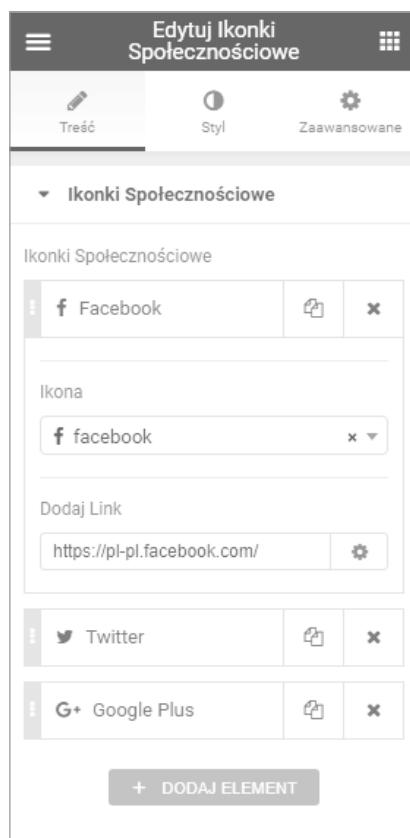
Joomla! jest kolejnym bezpłatnym programem typu CMS. Może być on stosowany w celach prywatnych lub komercyjnych bez dodatkowych opłat.

### 8.3.1. Instalacja programu Joomla!

Joomla! jest systemem instalowanym na serwerze stron internetowych. Dlatego wcześniej należy na komputerze zainstalować serwer WWW (np. Apache), bazę danych (np. MySQL) i język PHP. Może to być również platforma webowa XAMPP dla systemu Windows lub LAMP dla systemu Linux, względnie Microsoft Web Platform. Innym rozwiązaniem jest korzystanie z bezpłatnego lub komercyjnego konta założonego u jednego z dostawców usług internetowych, który oferuje dostęp do podanych wyżej zasobów.

Do tworzenia stron internetowych z wykorzystaniem systemu Joomla! nie jest wymagana znajomość języków: PHP, HTML i CSS oraz baz danych SQL, gdyż praca odbywa się w trybie wizualnym, a kody HTML i CSS są tworzone automatycznie. Dla systemu Joomla! dostępne są w internecie liczne rozszerzenia, które rozbudowują możliwości witryny.

Dodawanie nowych treści odbywa się przy pomocy przeglądarki internetowej. Wszystkie dodawane treści przechowywane są w bazie danych. Wszystkie prace związane



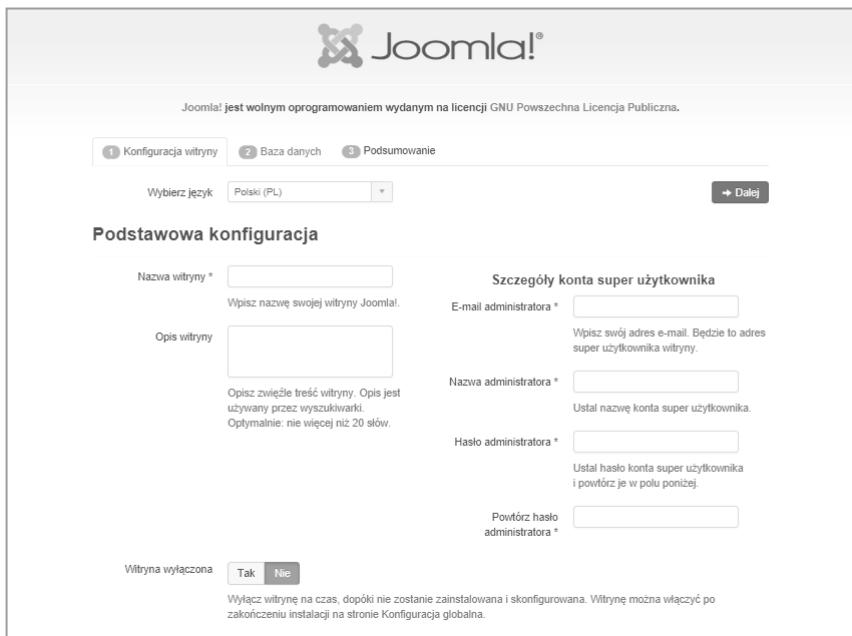
**Rysunek 8.35.**

Ustawianie linków do ikon stron społecznościowych

z umieszczaniem nowych treści i zarządzaniem serwisem odbywają się za pomocą uruchamianego w przeglądarce internetowej interfejsu. Modyfikacje zawartości są oznaczane datą i danymi zmieniającego.

Plik instalacyjny pakietu można pobrać ze strony <http://www.joomla.pl/pobierz-joomla>. Jeżeli pakiet jest instalowany na lokalnym komputerze, na którym został zainstalowany pakiet XAMPP, plik należy rozpakować w podkatalogu `htdocs` katalogu, w którym został zainstalowany XAMPP, np. `C:\xampp\htdocs\joomla`.

Instalacja pakietu jest prosta, ponieważ Joomla! posiada działający w przeglądarce instalator, który przeprowadzi nas przez cały proces instalacji. Podczas instalowania należy podać nazwę logowania i hasło do bazy danych, nazwę tworzonej witryny oraz opcjonalnie (jeżeli ta funkcjonalność będzie wykorzystywana) nazwę i hasło do konta FTP. Podczas instalowania systemu Joomla! niezbędne są uruchomiony serwer WWW oraz przeglądarka internetowa. Po wpisaniu w przeglądarce adresu, pod którym zostały umieszczone rozpakowane pliki Joomla!, np. <http://localhost/joomla>, wyświetli się okno instalatora (rysunek 8.36).



**Rysunek 8.36.** Okno instalatora Joomla!

Należy sprawdzić, czy został wybrany język polski, i wypełnić pola konfiguracji witryny. Po kliknięciu przycisku *Dalej* zostanie udostępnione okno konfigurowania dostępu do bazy danych.

Konfigurowanie dostępu do bazy danych wymaga utworzenia bazy, w której będą przechowywane m.in. treści dodawane do strony internetowej, ustawienia Joomla! i hasła użytkowników. Baza danych jest integralną częścią tworzonej strony.

W nowym oknie przeglądarki internetowej należy wpisać adres <http://localhost/php-myadmin>. Zostanie uruchomione narzędzie pakietu XAMPP, które służy do tworzenia baz danych i zarządzania nimi. Wybieramy zakładkę *Bazy danych* i w obszarze *Utwórz bazę danych* wpisujemy nazwę tworzonej bazy, np. *baza\_joomla*.

Wracamy do instalatora Joomla!. W polu *Nazwa serwera bazy danych* należy wpisać *localhost*, a w polu *Nazwa użytkownika — root*, pole *Hasło* zostaje puste, a w polu *Nazwa bazy danych* należy wpisać nazwę bazy utworzonej w phpMyAdmin.

W kolejnym kroku zostaną wyświetcone opcje konfiguracyjne oraz ustawienia systemu. Jeżeli któryś z tych ustawień jest wyróżnione kolorem czerwonym, kontynuowanie instalacji nie będzie możliwe. Trzeba sprawdzić, z jakiego powodu dalsza instalacja jest niemożliwa, i usunąć przyczynę. Nie należy zaznaczać opcji wczytywania przykładowych danych, ponieważ trzeba będzie je usunąć przed rozpoczęciem tworzenia strony internetowej (rysunek 8.37).



**Rysunek 8.37.** Okno konfiguracji Joomla!

Po wybraniu przycisku *Instaluj* pakiet zostanie zainstalowany. Jeżeli instalacja przebiegła prawidłowo, wyświetli się kolejne okno (rysunek 8.38). Aby móc zakończyć instalację, w oknie tym należy wybrać opcję usunięcia folderu instalacyjnego.

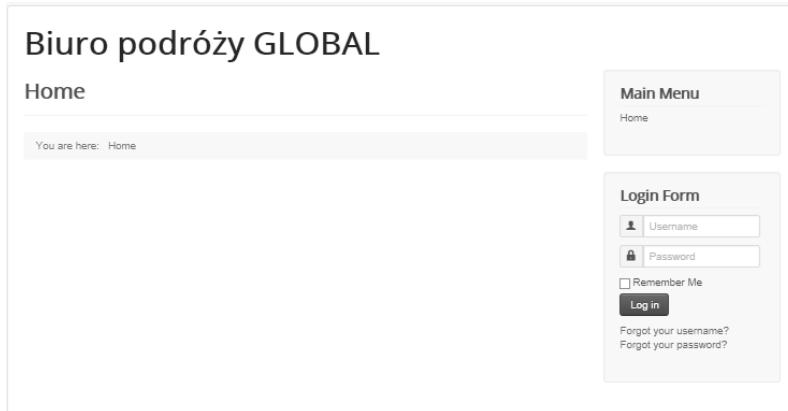


**Rysunek 8.38.** Okno podsumowujące instalację Joomla!

Po zainstalowaniu pakietu można rozpocząć pracę z witryną. Mamy dostęp do dwóch składowych systemu Joomla!. Są to: *Zaplecze* i *Witryna*. *Zaplecze* służy do zarządzania portalem. Pozwala na tworzenie struktury strony, dodawanie menu, zmianę wyglądu. *Witryna* pokazuje, jak będzie wyglądała wyświetlona w przeglądarce tworzona strona.

## Witryna

Po wpisaniu w przeglądarce adresu <http://localhost/joomla/> zostanie wyświetlona strona główna tworzonej witryny, zawierająca tytuł strony i formularz logowania (rysunek 8.39). Są to elementy domyślne tworzone przez system, które zostaną rozbudowane w trakcie tworzenia strony.



**Rysunek 8.39.** Okno tworzonej witryny

## Zaplecze (część administracyjna)

Po wpisaniu w kolejnym oknie przeglądarki adresu <http://localhost/joomla/administrator/> zostanie wyświetlony ekran logowania (rysunek 8.40) do części administracyjnej portalu.

W polu *Username* należy wpisać, zdefiniowaną w trakcie instalacji, nazwę konta administratora, a w polu *Password* — jego hasło. Po kliknięciu przycisku *Log in* zostanie wyświetlona część administracyjna witryny.

Przed rozpoczęciem pracy można zainstalować pakiet językowy zawierający spolszczoną wersję systemu Joomla!. W tym celu po zalogowaniu się do panelu administracyjnego należy wybrać z menu opcję *Extensions/Language(s)/Installed*, a następnie w otwartym oknie kliknąć przycisk *Install Languages*, odnaleźć plik zawierający spolszczenie i zainstalować go.

W oknie przeglądarki wyświetli się informacja potwierdzająca zainstalowanie pakietu języka polskiego (rysunek 8.41).



**Rysunek 8.40.** Okno logowania do panelu administracyjnego

Language	Language Tag	Version	Details URL
Irish	ga-IE	3.8.13.1	<a href="https://update.joomla.org/language/details3/ga-IE_details.xml">https://update.joomla.org/language/details3/ga-IE_details.xml</a>
Italian	it-IT	3.9.10.1	<a href="https://update.joomla.org/language/details3/it-IT_details.xml">https://update.joomla.org/language/details3/it-IT_details.xml</a>

**Rysunek 8.41.** Informacja potwierdzająca zainstalowanie pakietu języka

Kolejnym krokiem powinno być wybranie języka polskiego jako języka domyślnego dla witryny. W tym celu należy ponownie wybrać z menu *Extensions/Language(s)/Content Languages*, w otwartym oknie wybrać opcję z językiem polskim i zaznaczyć dla niego opcję *Default*. Dzięki temu językiem domyślnym dla witryny stanie się język polski (rysunek 8.42). Wprowadzone zmiany będą widoczne na stronie głównej tworzonej witryny po jej odświeżeniu.

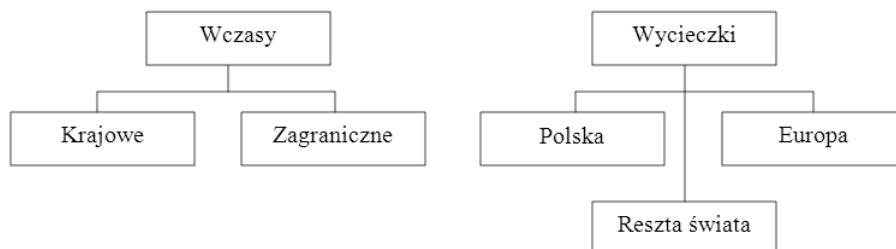
**Rysunek 8.42.** Ustawienie języka polskiego dla witryny

Aby język polski był dostępny również w panelu administracyjnym, z listy znajdującej się nad językami (rysunek 8.42) należy wybrać opcję *Administrator* i w taki sam sposób ustawić język polski jako domyślny dla części administracyjnej.

## 8.3.2. Tworzenie treści portalu

Treść jest najważniejszą częścią portalu. Powinna zostać zorganizowana tak, aby użytkownik witryny miał do niej łatwy dostęp. W Joomla! publikacje dostępne w witrynie internetowej są zorganizowane w postaci struktury złożonej z artykułów i kategorii. Artykułem jest każda publikacja, która powinna znaleźć się na stronie. Każdy artykuł musi zostać przypisany do jednej z kategorii. W kategorii można umieścić dowolną liczbę artykułów, a każda kategoria może zawierać inne kategorie. Struktura kategorii może być dowolnie rozbudowywana, w zależności od potrzeb tworzonej witryny. Po instalacji systemu Joomla! w witrynie pojawia się specjalna kategoria domyślna: *Uncategorised*. W niej będą zamieszczane artykuły, które nie pasują do żadnej innej kategorii. Mogą to być publikacje przeznaczone na stronę tytułową lub inne pojedyncze teksty.

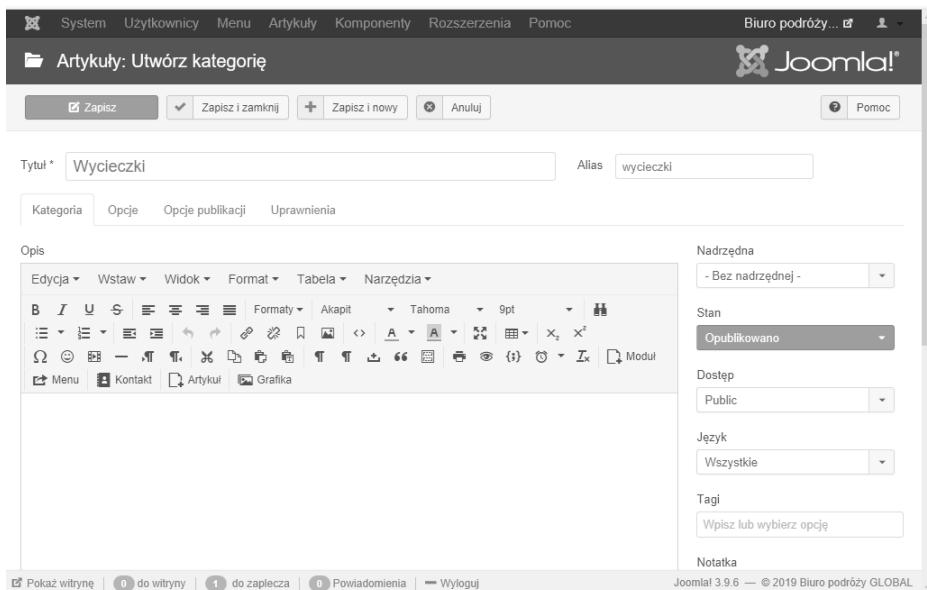
Jeżeli założymy, że tworzona witryna będzie dotyczyła biura podróży, to podział na kategorie może mieć postać pokazaną na rysunku 8.43.



**Rysunek 8.43.** Przykładowy schemat kategorii

### 8.3.3. Tworzenie kategorii

Po określeniu kategorii witryny można przystąpić do tworzenia struktury jej treści. Po zalogowaniu się do części administracyjnej należy wybrać z menu opcję *Artykuły/Kategorie/Utwórz kategorię*. W otwartym oknie zostanie wyświetlane okno *Edytora kategorii* (rysunek 8.44).



**Rysunek 8.44.** Okno edytowania kategorii

W polu *Tytuł* należy wpisać nazwę kategorii (np. *Wycieczki*). Można używać polskich liter. Nazwa, która tutaj zostanie wpisana, będzie wyświetlana w serwisie.

W polu *Alias* należy wpisać nazwę kategorii, która będzie wykorzystywana w adresach służących do nawigacji po witrynie. Nie należy używać polskich liter ani spacji. Ta nazwa będzie widoczna w polu adresu przeglądarki. Jeżeli pole pozostanie niewypełnione, system sam utworzy wewnętrzny alias.

W opcji *Nadrzędna* z prawej strony okna powinna zostać wybrana pozycja *Bez nadzędnej*. Oznacza to, że tworzona kategoria nie będzie miała kategorii nadzędnej.

Natomiast dla opcji *Stan* powinna zostać wybrana pozycja *Opublikowano*. Oznacza to, że kategoria będzie publikowana na stronie.

Po kliknięciu przycisku *Zapisz* kategoria zostanie utworzona i zapisana. Jeżeli będą tworzone następne kategorie, można kliknąć przycisk *Zapisz i nowy*. Jeżeli kończymy pracę z edytorem kategorii, można wybrać przycisk *Zapisz i zamknij*.

W podobny sposób można utworzyć kolejne kategorie. Jeżeli tworzona będzie kategoria podzonna, to po rozwinięciu listy w opcji *Nadrzędna* należy wybrać kategorię, do której

powinna zostać przypisana tworzona kategoria. Po utworzeniu wszystkich kategorii należy kliknąć przycisk **Zapisz i zamknij**, co spowoduje zapisanie kategorii, opuszczenie **Edytora kategorii** i przejście do panelu **Artykuły: Kategorie** (rysunek 8.45).

	Stan	Tytuł	Dostęp	Język	ID
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Uncategorised (Alias: uncategorised)	0 0 0 0	Public	Wszystkie 2
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Wycieczki (Alias: wycieczki)	0 0 0 0	Public	Wszystkie 8
<input type="checkbox"/>	<input checked="" type="checkbox"/>	– Polska (Alias: polska)	0 0 0 0	Public	Wszystkie 9
<input type="checkbox"/>	<input checked="" type="checkbox"/>	– Europa (Alias: europa)	2 0 0 0	Public	Wszystkie 10
<input type="checkbox"/>	<input checked="" type="checkbox"/>	– Reszta świata (Alias: reszta-swiata)	0 0 0 0	Public	Wszystkie 11
<input type="checkbox"/>	<input checked="" type="checkbox"/>	– Wczasy (Alias: wczasy)	0 0 0 0	Public	Wszystkie 12
<input type="checkbox"/>	<input checked="" type="checkbox"/>	– Krajowe (Alias: krajowe)	1 0 0 0	Public	Wszystkie 13
<input type="checkbox"/>	<input checked="" type="checkbox"/>	– Zagraniczne (Alias: zagraniczne)	0 0 0 0	Public	Wszystkie 14

**Rysunek 8.45.** Lista kategorii w Menedżerze kategorii

Jeżeli jakaś kategoria powinna zostać zmodyfikowana lub usunięta, to te czynności można wykonać w panelu **Artykuły: Kategorie**. Widoczna jest tu lista wszystkich utworzonych kategorii wraz z informacjami o nich. Widoczne u góry przyciski pozwalają na wykonanie operacji na kategoriach: dodawanie nowych, usuwanie istniejących, ustawianie parametrów. Aby całkowicie usunąć kategorię z systemu, należy najpierw przenieść ją do kosza, a następnie opróżnić kosz.

### Ćwiczenie 8.3

Utwórz strukturę witryny biura podróży zgodną z rysunkiem 8.43.

### Rozwiązanie

Przejdź do części administracyjnej tworzonej witryny. Wybierz w menu opcję **Artykuły/Kategorie/Utwórz kategorię**. W otwartym oknie w polu **Tytuł** wpisz **Wycieczki**, w polu **Alias** powtórz tytuł. W polu **Nadrzędna** z prawej strony okna wybierz **Bez nadrzędnej**. W polu **Stan** wybierz **Opublikowano**. Kliknij przycisk **Zapisz i nowy**. W kolejnym oknie w polu **Tytuł** wpisz **Polska** i w polu **Alias** powtórz tytuł. W polu **Nadrzędna** wybierz **Wycieczki**, a w polu **Stan** wybierz **Opublikowano**. Ponownie kliknij przycisk **Zapisz i nowy**. Postępując w ten sposób, utwórz całą strukturę biura podróży, a następnie kliknij przycisk **Zapisz i zamknij**. W otwartym oknie zostaną wyświetlane utworzone w ćwiczeniu kategorie.

### Zadanie 8.1

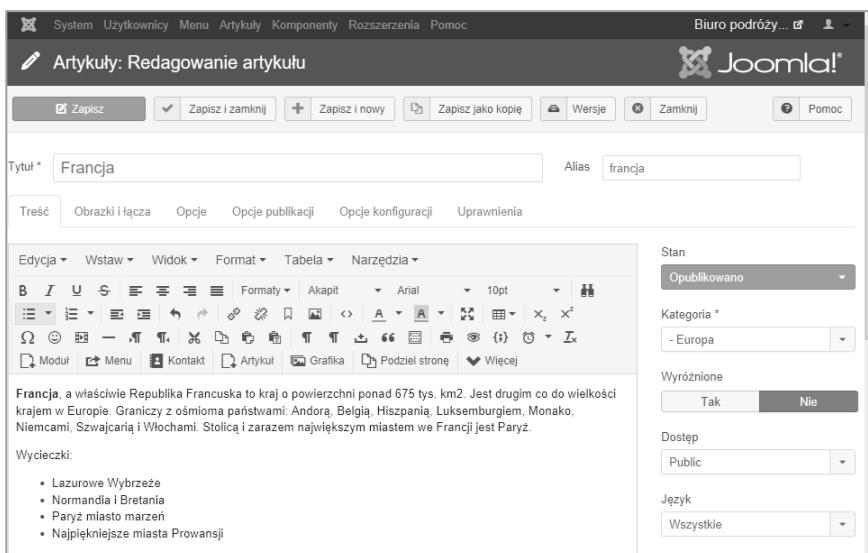
Zaprojektuj własną strukturę witryny, którą utworzysz za pomocą systemu Joomla!. Zebrane podczas projektowania witryny materiały podziel na kategorie. W Joomla! utwórz kategorie zgodne z przygotowanym projektem.

## 8.3.4. Tworzenie artykułów

Artykuły są najważniejszą częścią witryny. Jest to dowolna treść umieszczana na stronie. Może zawierać kilka zdań lub być kilkustronicowym tekstem opisującym jakieś zagadnienie. Treść artykułu może być dowolnie formatowana.

Aby utworzyć nowy artykuł, należy w części administracyjnej wybrać z menu opcję **Artykuły/Artykuły**. W otwartym oknie wyświetli się wszystkie utworzone artykuły. Początkowo lista jest pusta i aby dodać nowy artykuł, trzeba kliknąć przycisk **Utwórz**. Zostanie wyświetlony **Edytor artykułu**.

W polu **Tytuł** należy wpisać tytuł artykułu. W polu **Kategoria** trzeba wybrać z listy kategorię, do której będzie należał artykuł. Aby artykuł był widoczny dla użytkowników witryny, opcję **Stan** należy ustawić na wartość **Opublikowano**. Aby artykuł nie został wyświetlony na stronie głównej witryny, opcja **Wyróżnienie** powinna być ustawiona na wartość **Nie**. Po tych ustawieniach trzeba wpisać treść artykułu. W polu wpisywania treści dostępne są opcje formatowania tekstu, których można używać tak jak w innych edytورach tekstu (rysunek 8.46). Aby zapisać artykuł i przejść do panelu **Artykuły**, należy kliknąć przycisk **Zapisz i zamknij**. Jeżeli będą tworzone kolejne artykuły, można wybrać przycisk **Zapisz i nowy**.



**Rysunek 8.46.** Edytowanie artykułu

Domyślnie udostępniany w systemie Joomla! **Edytor artykułów** zawiera podstawowe funkcje edycji tekstu. Jeżeli potrzebne są bardziej zaawansowane funkcje formatowania, można przełączyć się do trybu zaawansowanego. W tym celu z menu należy wybrać **Rozszerzenia/Dodatki** i w panelu **Dodatki** wybrać z listy edytor **TinyMCE**. Jest to domyślny edytor systemu Joomla!. W oknie ustawień edytora po wybraniu zakładki **Opcje podstawowe** należy wybrać z listy **Tryb pracy** pozycję **Rozszerzony** i kliknąć przycisk **Zapisz i zamknij**.

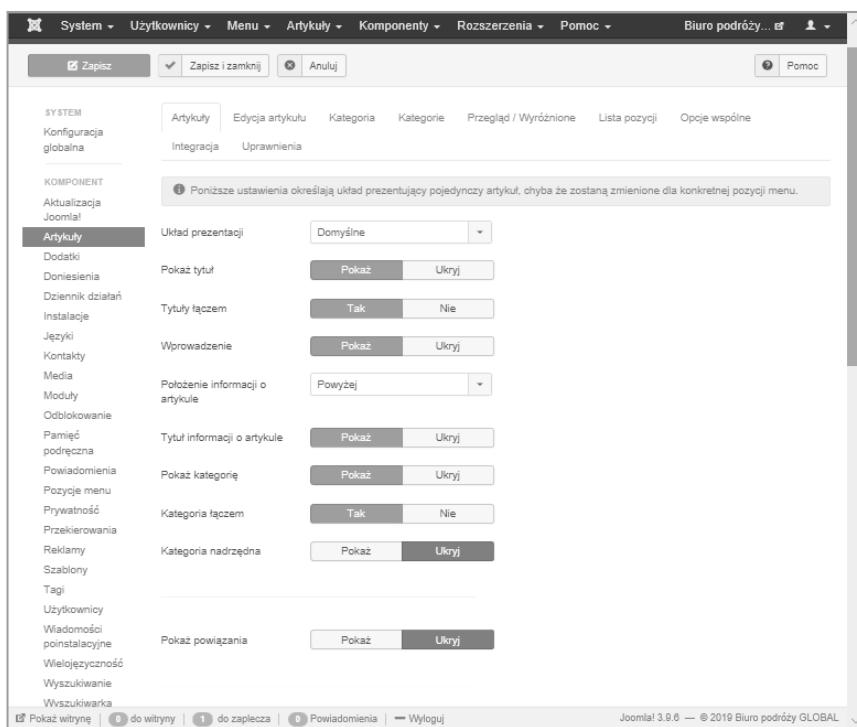
## Artykuł na stronie głównej

Jeżeli tworzony artykuł będzie wyświetlany na stronie głównej witryny, wystarczy podczas jego edycji w opcji *Wyróżnione* wybrać wartość *Tak*, a w opcji *Kategoria* — pozycję *Uncategorised*. Spowoduje to umieszczenie artykułu w grupie specjalnej, która domyślnie jest wyświetlana na stronie głównej.

## Atrybuty artykułów

Podczas tworzenia witryny internetowej można określić, jakie atrybuty artykułów będą pokazywane przy tekście, a także zmodyfikować atrybuty grupy artykułów lub pojedynczego artykułu. Konfiguracja globalna pozwala ustawić te cechy, które będą takie same dla wszystkich artykułów. Później cechy pojedynczych artykułów mogą być modyfikowane według potrzeb.

Globalna konfiguracja jest dostępna w panelu *Artykuły* po wybraniu przycisku *Opcje*. Wyświetlony ekran konfiguracji składa się z kilku części (rysunek 8.47), w których pokazywany jest aktualny stan atrybutów dla wszystkich artykułów. Modyfikacja dostępnych parametrów będzie dotyczyła wszystkich artykułów.



**Rysunek 8.47.** Globalna konfiguracja artykułów

Atrybuty pojedynczego artykułu można modyfikować po przejściu do trybu jego edycji. Wtedy w górnej części dostępna jest zakładka *Opcje publikacji*. Zawiera ona kilka

przydatnych opcji. W polu *Rozpocznij publikację* można określić, kiedy ma się rozpocząć publikacja artykułu (rysunek 8.48). Domyślnie jest to data umieszczenia artykułu w systemie. W polu *Zakończ publikację* można określić, kiedy ma się zakończyć publikacja artykułu. Domyślnie artykuł nie ma ustawionych ograniczeń na czas publikacji. *Wersja* określa liczbę modyfikacji artykułu w systemie. Pole *Zmodyfikował* pokazuje nazwę użytkownika, który zmodyfikował artykuł.

The screenshot shows the Joomla! article editing screen. At the top, there's a toolbar with buttons for saving (Zapisz, Zapisz i zamknij, Zapisać nowy, Zapisz jako kopię, Wersje, Zamknij), a help button (Pomoc), and a user icon. Below the toolbar, the title field contains "Francja". The alias field is set to "francja". There are several tabs at the top: "Treść", "Obrazki i łącza", "Opoje", "Opcje publikacji" (which is selected), "Opcje konfiguracji", and "Uprawnienia". The "Opcje publikacji" tab contains fields for "Rozpocznij publikację" (set to 15-07-2019 21:25:01), "Zakończ publikację" (empty), "Data utworzenia" (set to 15-07-2019 21:25:01), "Utworzony przez" (Super User), "Inna nazwa autora" (empty), "Data modyfikacji" (set to 15-07-2019 21:58:24), "Zmodyfikował" (Super User), "Wersja" (2), and "Odsłony" (0). To the right of these fields are corresponding input fields for "Opis-streszczenie", "Słowa kluczowe", "Klucz odnośnika", "Wyszukiwarki" (set to Globalnie), "Autor", "Prawa do treści", and "Odniesienie zewnętrzne". At the bottom of the form, there are links for "Pokaz witryny", "do witryny", "do zaplecza", "Powiadomienia", and "Wyloguj". The footer displays "Joomla! 3.9.6 — © 2019 Biuro podróży GLOBAL".

**Rysunek 8.48.** Opcje publikacji artykułu

Zakładka *Opcje publikacji* zawiera takie same opcje jak opcje konfiguracji globalnej. Można zmienić ich wartość dla wybranego artykułu.

#### Ćwiczenie 8.4

Przygotuj krótkie teksty dotyczące oferty biura podróży. Mogą to być teksty opisujące państwa, do których biuro podróży organizuje wycieczki, lub miejscowości, w których organizowane są wczasy. Utwórz po jednym artykule dla kategorii: *Wczasy→Krajowe*, *Wczasy→Zagraniczne*, *Wycieczki→Polska*, *Wycieczki→Reszta świata*, dwa artykuły dla kategorii *Wycieczki→Europa* oraz artykuł *Oferta* bez określonej kategorii.

#### Rozwiążanie

Pierwszy artykuł będzie dotyczyć wycieczki do Francji. W menu wybierz opcję *Artykuły/Artykuły/Utwórz artykuł*. W tytule wpisz *Francja*. Wpisz krótki tekst i podaj listę wycieczek. W polu *Kategoria* wybierz *Europa*, w polu *Stan* ustaw *Opublikowano*, w polu *Wyróżnienie* wybierz *Nie*. Sformatuj tekst, wybierając czcionkę Arial, 10px. Dla listy wycieczek ustaw wypunktowanie. Zmień kolor czcionki. Aby zapisać artykuł, kliknij

przycisk *Zapisz i nowy*. W podobny sposób opracuj kolejne artykuły. Zakończ, klikając przycisk *Zapisz i zamknij*. W otwartym oknie panelu *Artykuły* zostaną wyświetcone utworzone w ćwiczeniu artykuły.

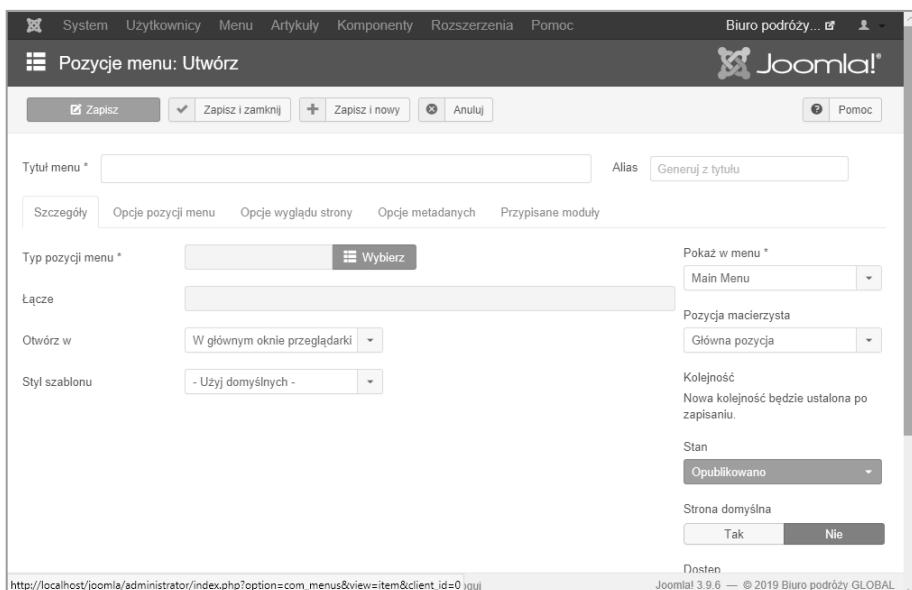
## Zadanie 8.2

Na podstawie zgromadzonych materiałów opracuj dla tworzonej przez siebie witryny internetowej artykuł i poprzedzaj je do odpowiednich kategorii.

### 8.3.5. Tworzenie menu

Menu witryny internetowej jest elementem, dzięki któremu użytkownik będzie mógł poruszać się po różnych częściach witryny i odnajdywać interesujące go treści. Po zainstalowaniu systemu dostępne jest w witrynie podstawowe menu z jedną pozycją: *Home*. Ponieważ w tworzonej witrynie występują już kategorie i artykuły, można utworzyć menu, dzięki któremu użytkownik będzie miał do nich dostęp.

Aby móc modyfikować menu, należy w części administracyjnej wybrać w menu *Menu/ Main Menu*. Zostanie otwarte okno panelu *Pozycje menu*: Początkowo będzie tu tylko wpis *Home*. Aby utworzyć nową pozycję menu, należy kliknąć przycisk *Utwórz*. Uručomi się edytor pozycji menu (rysunek 8.49).



**Rysunek 8.49.** Edytor menu witryny internetowej

W polu *Tytuł menu* należy wpisać nazwę wybranej pozycji menu (np. *Wycieczki*). Dla pola *Typ pozycji menu* należy kliknąć przycisk *Wybierz*. Wyświetli się lista pogrupowanych tematycznie typów dostępnych w systemie.

Typ pozycji określa, co się wyświetli na stronie po wybraniu określonej pozycji menu. Jeden typ może wskazywać na pojedynczy artykuł i po wyborze przez użytkownika takiej pozycji wyświetli się tekst artykułu. Inny typ może wskazywać na grupę artykułów w danej kategorii i wtedy zostanie wyświetlona lista artykułów danej kategorii. Typy menu są zdefiniowane w systemie — podczas tworzenia menu należy tylko wskazać odpowiedni typ. Po wybraniu typu pozycji menu zostanie udostępniona opcja *Wybierz nadziedną kategorię*. Z wyświetlonej listy należy wybrać kategorię, która odpowiada tworzonej pozycji menu (np. *Wycieczki*). W celu utworzenia pozycji menu trzeba kliknąć przycisk *Zapisz i zamknij*. W podobny sposób można tworzyć kolejne pozycje menu.

## Typy pozycji menu

W systemie Joomla! każda pozycja menu jest połączona ze specjalnym komponentem, który odpowiada za generowanie treści. Dostępne komponenty pozwalają tworzyć dowolnie rozbudowaną strukturę witryny (np. artykuły, kategorie artykułów, formularze kontaktowe).

Każda pozycja menu wskazuje na określony zakres treści, np. artykuł, kategorię, kilka kategorii. Wskazanie określonego zakresu treści definiuje również formę, w jakiej treść będzie prezentowana.

Wybranie w polu *Typ pozycji menu* danego typu określi zakres treści, które pojawią się po wyborze pozycji menu, jak również formę, w jakiej treść będzie prezentowana.

W polu tym mamy do dyspozycji ponad 20 typów, które pomogą w sprawnym zarządzaniu strukturą witryny. Nazwy typów odzwierciedlają zakres i formę treści. Pierwszy człon wskazuje, w jakiej postaci zostanie zaprezentowana treść, drugi określa jej rodzaj, np.: Lista – zawartość kategorii, Tabela – artykuły z kategorii, Łącze – kontakt. *Typ pozycji menu* oznacza więc sposób organizacji elementów struktury treści oraz szablon ich formatowania.

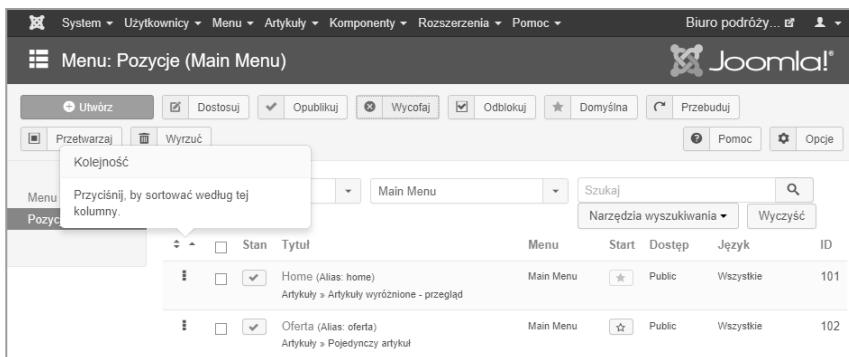
## Modyfikowanie menu

Pozycje menu mogą być dowolnie modyfikowane. Można zmieniać nazwę wybranej pozycji, miejsce jej ustawienia lub konfigurację. Aby zmienić nazwę lub konfigurację, wystarczy w panelu *Menu: Pozycje* kliknąć ikonę z nazwą wybranej pozycji. Otworzy się okno edytowania, w którym można dowolnie zmieniać wprowadzone ustawienia. Aby usunąć pozycję menu, należy w panelu *Menu: Pozycje* zaznaczyć pozycję i kliknąć ikonę *Wyrzuć*. Kolejność menu można zmienić, chwytając ikonę pozycji w kolumnie *Kolejność* i przeciagając ją do góry lub w dół (rysunek 8.50).

## Rozbudowa menu

Tworzone menu może zawierać pozycje podrzędne, które — po podziale artykułów na podkategorie — umożliwią dostęp z poziomu menu do wybranej grupy artykułów.

Aby utworzyć w menu pozycję podrzędną, należy przejść do tworzenia menu (*Menu/Main Menu/Utwórz pozycję menu*), wpisać tytuł, jako *Typ pozycji menu* wybrać z listy



**Rysunek 8.50.** Ustalanie kolejności pozycji menu

**Artykuły**, a następnie **Pojedynczy artykuł**. Później w opcji **Pozycja macierzysta** należy z listy pozycji menu wybrać pozycję, która będzie nadziedna w stosunku do tworzonej. W kolejnym kroku trzeba w opcji **Wybierz kategorię** wybrać z listy kategorię odpowiadającą tworzonej pozycji menu. Aby zapisać wprowadzone ustawienia, należy kliknąć przycisk **Zapisz i zamknij**. Na liście pozycji menu nowa pozycja powinna zostać pokazana jako podziedna w stosunku do wybranej pozycji głównej. W podobny sposób można tworzyć kolejne pozycje menu podziednego.

Po wprowadzeniu zmian można przetestować menu, przechodząc do strony głównej witryny. Ponieważ pozycje podziedne menu domyślnie są ukryte, wyświetlona witryna wygląda tak, jakby nic się nie zmieniło. Dopiero po kliknięciu pozycji nadziednej wyświetla się wszystkie pozycje podziedne. Modyfikacja pozycji menu podziednych odbywa się podobnie jak modyfikacja pozycji menu głównego.

## Ćwiczenie 8.5

Dla projektowanej witryny biura podróży:

1. Utwórz menu zawierające pozycje: *Oferta*, *Wczasy* i *Wycieczki*.
2. Ukryj dodatkowe informacje o stronie (kto utworzył artykuł, kiedy został on opublikowany oraz ile razy był otwierany) dla wszystkich artykułów.
3. Wyświetlaj tylko na stronie *Oferty* informację o tym, ile było odsłon.
4. Utwórz artykuł dla strony głównej i przypnij go w menu do pozycji *Home*.

## Rozwiążanie

### Ad 1.

Wybierz w menu *Menu/Main Menu*. W otwartym oknie kliknij przycisk *Utwórz*. Uruchomi się edytor pozycji menu. W polu *Tytuł menu* wpisz nazwę pierwszej pozycji menu *Oferta*. W polu *Typ pozycji menu* kliknij przycisk *Wybierz* i z wyświetlonej listy wybierz *Artykuły*, a następnie *Pojedynczy artykuł*. Później w polu *Wybierz artykuł* kliknij

przycisk *Wybierz* i odszukaj artykuł z ofertą. Kliknij wybrany artykuł, a potem kliknij przycisk *Zapisz i nowy*.

Utwórz następną pozycję menu. W polu *Tytuł menu* wpisz *Wczasy*. W polu *Typ pozycji menu* kliknij przycisk *Wybierz* i z wyświetlanej listy wybierz *Artykuły*, a następnie *Jedna kategoria artykułów - przegląd*. Następnie w polu *Wybierz kategorię* kliknij przycisk *Wybierz* i wybierz kategorię *Wczasy*. Kliknij przycisk *Zapisz i nowy* i w podobny sposób utwórz pozycję menu *Wycieczki*.

Po przełączeniu się do panelu witryny i jej odświeżeniu na liście *Main Menu* pojawią się utworzone pozycje. Po kliknięciu wybranej pozycji zostaną wyświetcone przygotowane artykuły.

#### Ad 2.

Oprócz zawartości artykułu na stronie została wyświetlona informacja o tym, kto utworzył artykuł, kiedy został on opublikowany oraz ile razy był wyświetlany. Te informacje należy ukryć. Zrobimy to globalnie dla wszystkich utworzonych artykułów.

Wróć do panelu administracyjnego. Wybierz z menu opcję *Artykuły/Artykuły* i kliknij znajdujący się z prawej strony okna przycisk *Opcje*. Na wyświetlanej liście odszukaj po kolejne opcje: *Pokaż autora*, *Pokaż kategorię*, *Pokaż datę opublikowania*, *Pokaż odsłony* i dla każdej z nich kliknij przycisk *Ukryj*. Można jeszcze ukryć ikony, wybierając opcje *Pokaż ikony* i *Pokaż ikonę Drukuj*, a następnie kliknąć przycisk *Ukryj*. Po ustawieniu podanych opcji kliknij przycisk *Zapisz i zamknij*.

Przejdź do panelu witryny, odśwież stronę i sprawdź, czy dodatkowe informacje zniknęły.

#### Ad 3.

Wróć do panelu administracyjnego. Wybierz z menu opcję *Menu/Main Menu* i przejdź od edycji *Oferty*, wybierając z listy pozycję *Oferta*. W otwartym oknie wybierz zakładkę *Opcje*. Odszukaj pozycję *Pokaż odsłony* i wybierz opcję *Pokaż*, a następnie kliknij przycisk *Zapisz i zamknij*.

#### Ad 4.

Wybierz w menu *Artykuły/ Artykuły*. W otwartym oknie kliknij przycisk *Utwórz*. Urichomisie edytor artykułów. W polu *Tytuł* wpisz *Główna*. Wprowadź tekst z informacją dotyczącą działalności biura podróży. Ustaw pole *Wyróżnione* na *Tak* i kliknij przycisk *Zapisz i zamknij*. Przełącz się do panelu witryny, odśwież ją i sprawdź działanie wszystkich pozycji menu.

### Parametry wyświetlania

Jeżeli wyświetlanych na stronie artykułów jest kilka, to układają się one w ten sposób, że pozycja główna jest jedna, a pozostałe artykuły są rozmieszczane w dwóch kolumnach (rysunek 8.51).

**Biuro podróży GLOBAL**

**Hiszpania**

Hiszpania, Królestwo Hiszpanii – państwo w zachodniej części Europy Południowej, położone głównie na Półwyspie Iberyjskim, obejmujące także Baleary na Morzu Śródziemnym, Wyspy Kanaryjskie na Oceanie Atlantyckim oraz tereny w północnej Afryce.

Wczasy:

- Costa Brava
- Wyspy Kanaryjskie
- Barcelona
- Ibiza

**Włochy**

Włochy – państwo położone w Europie Południowej, na Półwyspie Apenińskim, posiadające na swoim terytorium dwie enklawy: San Marino na północy i Watykan na terenie stolicy kraju, Rzymu. Włochy cechuje wyżynny i górzysty krajobraz. Praktycznie cały kraj reprezentuje klimat podzwrotnikowy i rośliność

**Francja**

Francja, właściwie Republika Francuska, to kraj o powierzchni ponad 675 tys. km<sup>2</sup>. Jest drugim co do wielkości krajem w Europie. Graniczy z ośmioma państwami: Andorą, Belgią, Hiszpanią, Luksemburgiem, Monako, Niemcami, Szwajcarią i Włochami. Stolicą i zarazem największym miastem we Francji jest Paryż.

**Main Menu**

- Home
- Oferta
- Wczasy
- Nowe wycieczki

**Login Form**

Użytkowni  
Hasło  
 Zapamiętaj  
**Zaloguj**

Nie pamiętasz nazwy?  
Nie pamiętasz hasła?

**Rysunek 8.51.** Ułożenie artykułów na stronie

Układ wyświetlanych na stronie artykułów można zmienić, wybierając z menu opcję **Artykuły/Artykuły**, a następnie klikając przycisk **Opcje**. W otwartym oknie należy wybrać zakładkę **Przegląd/Wyróżnione** i ustawić parametry określające liczbę pozycji głównych, liczbę pozostałych pozycji wyświetlanych na stronie poniżej pozycji głównych oraz liczbę kolumn, w których zostaną wyświetlane artykuły.

Dla długich artykułów można użyć przycisku **Czytaj więcej**. Należy go dodać do każdego artykułu oddzielnie. W tym celu należy wybrać z menu opcję **Artykuły/Artykuły** i otworzyć do edycji wybrany artykuł. Następnie należy ustawić kursor myszy w miejscu, od którego tekst nie będzie wyświetlany, i wybrać na pasku z ikonami do formatowania tekstu ikonę **Więcej**. W tekście pojawi się czerwona linia oznaczająca miejsce podziału artykułu. Teraz należy kliknąć przycisk **Zapisz i zamknij** i przetestować działanie ustawień witryny.

### Zadanie 8.3

Dla swojej witryny internetowej utwórz menu. Przydziel poszczególne pozycje menu do odpowiednich kategorii. Dla każdej pozycji menu określ odpowiedni typ tej pozycji.

## 8.3.6. Konfiguracja witryny

Ustawienia konfiguracyjne tworzonej witryny są realizowane w menu po wybraniu **System/Konfiguracja globalna**. Zostanie otwarte okno, w którym można ustawić tytuł wyświetlanej strony, zmieniając w obszarze **Optymalizacja dla wyszukiwarek** opcję **Nazwa witryny w tytułach stron** na wartość **Przed**. Po zapisaniu wprowadzonych ustawień na karcie strony zostanie wyświetlony jej tytuł.

Ustawienie opcji **Witryna wyłączona** na czas modyfikacji strony spowoduje zablokowanie dostępu do niej dla użytkowników. Możliwe jest wyświetlenie informacji o powodach wyłączenia witryny. W tym celu należy w polu **Komunikat o wyłączeniu** umieścić odpowiednią informację.

W obszarze **Metadane** można ustawić parametry związane z obsługą strony przez wyszukiwarki internetowe.

### 8.3.7. Grafika

Korzystając z systemu Joomla!, można do witryny dodawać dowolne grafiki, wcześniej umieszczone na serwerze. Jeżeli konto WWW jest dostępne lokalnie, wystarczy potrzebne grafiki umieścić w przeznaczonym do tego folderze.

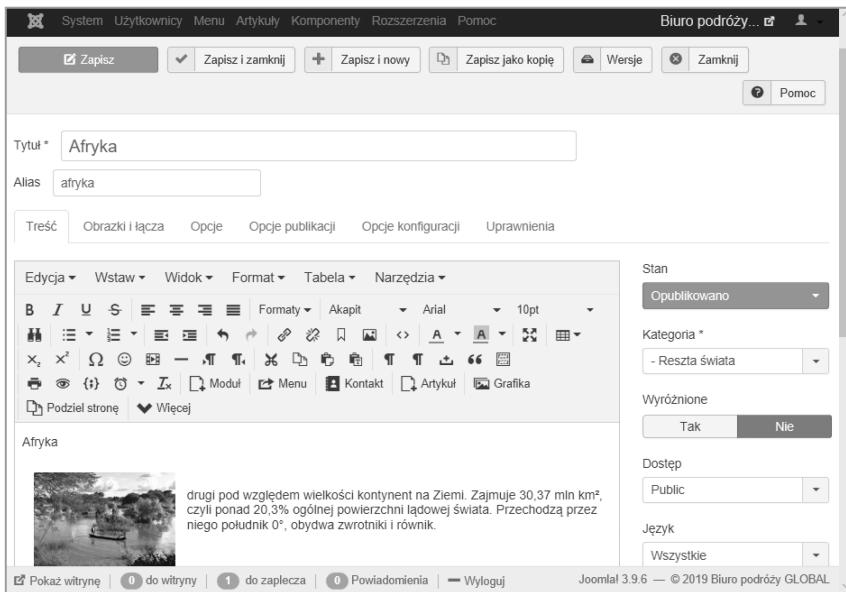
Plikami graficznymi można zarządzać za pomocą panelu **Media**. Pozwala on na tworzenie struktury katalogów, wczytywanie plików na serwer oraz wybór odpowiedniego pliku. Uruchomienie panelu **Media** nastąpi po wybraniu z menu **Artykuły/Media**. W otwartym oknie z lewej strony widoczna jest struktura katalogów z grafiką. U góry dostępne są dwa przyciski: **Miniatury** i **Szczegóły**, które pozwalają na przełączanie się między tymi widokami. Powyżej znajdują się ikony do tworzenia i usuwania katalogów. W głównej części okna widoczne są miniatury plików graficznych i katalogów. Po przełączeniu się do widoku **Szczegóły** widoczne będą szczegóły dotyczące plików. Między innymi wyświetla się informacje o rozdzielczości obrazu oraz wielkości pliku.

W systemie można tworzyć własne katalogi. Szczególnie gdy plików z grafiką jest dużo, zalecane jest tworzenie własnych katalogów i podkatalogów. Po kliknięciu przycisku **Utwórz katalog** zostanie pokazana ścieżka do tworzonego katalogu. Po wprowadzeniu nazwy należy kliknąć znajdujący się obok przycisk **Utwórz katalog**. Plik lub katalog można usunąć, klikając ikonę z krzyżykiem znajdującą się w kolumnie **Usuwanie** (widok **Szczegóły**) lub przy jego miniaturze (widok **Miniatury**). Gdy grafiki są umieszczone na serwerze lub są dostępne lokalnie, można umieścić je na wybranych stronach witryny.

#### Dodawanie grafiki do artykułu

Grafika wstawiana na stronę internetową najczęściej jest ilustracją artykułu. Przed jej wstawieniem należy utworzyć artykuł i przejść do jego edycji. Jeżeli grafika ma zostać wstawiona w określonym miejscu artykułu, trzeba ustawić w tym miejscu kurSOR myszy i kliknąć przycisk **Grafika**. W otwartym oknie należy wybrać katalog i plik do wstawienia. W dolnej części tego okna znajdują się pola, które pozwalają na edycję parametrów wstawianego obrazu. Można ustawić sposób wyrównania obrazu, wstawić jego tytuł i opis. Po kliknięciu przycisku **Wstaw** obraz znajdzie się w tekście. Wstawiony obraz można edytować. Aby zmienić jego parametry, należy go zaznaczyć i wybrać ikonę **Wstaw/edytuj obrazek**. Otworzy się okno, w którym można edytować obraz. W zakładce **Ogólne**, w polu **Wymiary** można określić rozmiar obrazka. W zakładce **Zaawansowane** można ustalić położenie obrazka. Pole **Ramka** służy do określenia pozycji obrazka względem tekstu, a pola **Odstęp pionowy** i **Odstęp poziomy** określą odstęp

między obrazem a tekstem w pionie i poziomie. Po wprowadzeniu zmian należy kliknąć przycisk *Update*. Obrazek zostanie umieszczony w artykule zgodnie z wprowadzonymi zmianami (rysunek 8.52). Po tych zmianach należy zapisać artykuł i sprawdzić, jak wygląda na stronie internetowej.



**Rysunek 8.52.** Wstawianie grafiki do artykułu

### Zadanie 8.4

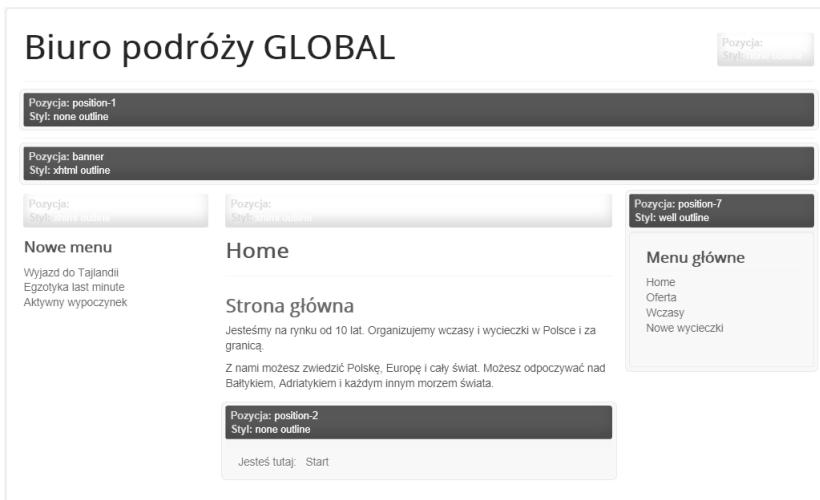
W projektowanej przez siebie witrynie przygotuj stronę zawierającą podstawowe informacje mówiące o właściwym twórcom witryny. Utwórz artykuł *O mnie*, dodaj do niego grafikę. Utwórz nową pozycję w menu i przypisz do niej utworzony artykuł.

## 8.3.8. Moduły

W tworzonej witrynie internetowej można wyodrębnić pewne części odpowiedzialne za wykonywanie określonych operacji. Każdą wyodrębnioną i niezależnie wyświetlaną część witryny można traktować jako moduł. Modułem jest menu, lista najnowszych wiadomości czy liczba osób odwiedzających stronę. Wszystko, co jest wyświetlane w witrynie, jest modułem. Tworząc witrynę internetową, należy określić, jakie moduły powinny się w niej znaleźć, np. w przypadku biura podróży mogą to być: lista atrakcyjnych wycieczek, lista nowych wycieczek, lista gości obecnych na stronie, oferta last minute. Moduł wyświetli te dane i umożliwi ich konfigurację.

Strona internetowa jest podzielona na kilka części. Tworząc nowy moduł, należy określić, w jakiej części strony powinien on zostać umieszczony. Jeżeli w jednej części zostanie umieszczonych kilka modułów, należy zdecydować o kolejności ich wyświetlania,

podając wartość porządkową modułu. Po rozpoczęciu pracy w systemie Joomla! domyślnie stosowany jest szablon *protostar*, w którym obowiązuje podział strony pokazany na rysunku 8.53. Po wybraniu w menu *Rozszerzenia/Szablony* w otwartym oknie zostanie wyświetlona lista szablonów. Jako domyślny została zaznaczony szablon *protostar*. Po wybraniu z lewej strony opcji *Szablony* można obejrzeć jego wygląd, ale podgląd jest niewidoczny. Można to zmienić, wybierając przycisk *Opcje* i włączając opcję *Podgląd pozycji modułów*. Po kliknięciu przycisku *Zapisz i zamknij* ustawienia zostaną zapisane. Teraz można wybrać podgląd domyślnego szablonu.



**Rysunek 8.53.** Szablon protostar

## Modyfikowanie modułu menu

Znajdujące się w witrynie menu jest modułem. Może ono być modyfikowane, ale można też utworzyć nowe menu. Aby można było tworzyć lub modyfikować moduł, należy przejść do trybu jego edycji. W tym celu trzeba wybrać w menu *Rozszerzenia/Moduły*. Wyświetli się panel *Moduły* z dostępnymi modułami. Początkowo są to standardowe moduły dostępne w systemie.

### Ćwiczenie 8.6

Zmień tytuł wyświetlanego na stronie modułu *Main Menu* na *Menu główne* i *Login Form* na *Panel logowania*. Panel logowania powinien być widoczny tylko na stronie głównej (*Home*) i stronie *Oferta*.

### Rozwiązywanie

Wyświetl panel *Moduły*, wybierając w menu *Rozszerzenia/Moduły*. W otwartym oknie wybierz z listy moduł *Main Menu*, który jest związany z wyświetlonym w witrynie menu. W oknie edycji tego modułu w polu *Tytuł* wprowadź nową nazwę: *Menu główne*. Pole *Pokaż tytuł* decyduje o tym, czy tytuł ten będzie wyświetlany w witrynie. Zostaw

ustawienia domyślne. Pole *Pozycja* decyduje o tym, gdzie będzie umieszczone menu. Tu również zostaw ustawienia domyślne. Po ustawieniu opcji modułu kliknij przycisk *Zapisz i nowy*. Wybierz moduł *Login Form* i w polu *Tytuł* wprowadź nazwę *Panel logowania*. Wybierz zakładkę *Przypisz do pozycji menu* i w polu *Zasada przypisania* wybierz *Tylko na zaznaczonych stronach*. Na liście pozostaw zaznaczoną tylko stronę *Home* i stronę *Oferta*. Kliknij przycisk *Zapisz i zamknij*. Przejdz na stronę główną witryny i sprawdź, jak wygląda menu po wprowadzonych zmianach.

## Dodawanie modułu menu

Przed rozpoczęciem tworzenia nowego modułu należy pamiętać, że moduł jest tylko elementem wyświetlającym dane. Dlatego przed utworzeniem modułu należy przygotować treść, która zostanie w nim pokazana.

### Ćwiczenie 8.7

Dla tworzonej witryny biura podróży utwórz nowe menu. Tytuł tego menu to *Newsy*, a pozycje to *Wyjazd do Tajlandii*, *Egzotyka last minute* i *Aktywny wypoczynek*.

### Rozwiążanie

Najpierw należy utworzyć menu, a następnie moduł, w którym będzie ono umieszczone.

Wybierz z menu *Menu/Projekt menu/Utwórz menu* i w polu *Tytuł* wpisz tytuł *Newsy*. W polu *Typ menu* wpisz dowolną wymyśloną przez siebie nazwę typu menu. Będzie ona używana tylko wewnętrznie przez system Joomla! jako identyfikator. Nie można używać spacji i polskich liter. W polu *Opis* wprowadź dowolną treść opisującą tworzone menu. Kliknij przycisk *Zapisz i zamknij*. Tytuł menu pojawi się na liście dostępnych menu. Kliknij w nowo utworzone menu, a zostanie wyświetcone znane już okno *Menu: Pozycje*. Aby utworzyć kolejne pozycje menu, kliknij przycisk *Utwórz* i w polu *Tytuł menu* wpisz pierwszą pozycję. Jako typ pozycji menu wybierz *Artykuły/Jedna kategoria artykułów - lista*. Następnie w polu *Wybierz kategorię* wybierz z listy kategorię przypisaną do pozycji menu i kliknij przycisk *Zapisz i nowy*. W ten sam sposób utwórz kolejne pozycje menu.

Ponieważ zawartość menu wyświetlana jest za pomocą modułów, utworzymy nowy moduł.

Wybierz z menu *Rozszerzenia/Moduły* i kliknij przycisk *Utwórz*. Wyświetli się okno wyboru typu modułu, w którym można wybrać typ tworzonego modułu. Ponieważ moduł będzie służył do wyświetlania menu, wybierz typ *Menu*. W kolejnym oknie wyświetcone zostały opcje konfiguracji modułu. Wpisz tytuł modułu (np. *Newsy*). W polu *Wybierz menu* wybierz menu, które będzie wyświetlane w tworzonym module (*Newsy*). W polu *Pokaż tytuł* ustaw *Pokaż*, a w polu *Pozycja* określ, w którym miejscu na stronie będzie wyświetlany moduł (np. *Lewa góra [position-7]*).

Przejdz do zakładki *Przypisz do pozycji menu* i określ, na jakich stronach menu będzie ono wyświetlane. Domyślnie wybrana jest pozycja *Na wszystkich stronach*, co oznacza, że menu będzie widoczne niezależnie od tego, która strona będzie wyświetlana. Po

wybieraniu pozycji *Tylko na zaznaczonych stronach* wyświetli się lista wszystkich pozycji menu. Zostaw zaznaczone tylko te, na których dodatkowe menu powinno zostać wyświetlone. Po ustaleniu wszystkich opcji kliknij przycisk *Zapisz i zamknij*. Aby ocenić, czy nowe menu wyświetla się prawidłowo, przejdź na stronę główną witryny i sprawdź, jak wygląda i funkcjonuje nowo utworzone menu.

## Włączanie i wyłączanie modułu

Istniejący moduł może w zależności od potrzeb zostać włączony lub wyłączony. Można w ten sposób zablokować wyświetlanie modułu na wszystkich lub na niektórych stronach. Moduł logowania standardowo jest widoczny na wszystkich stronach. Jeżeli w tworzonej witrynie nie będzie potrzebny, można go wyłączyć.

Aby wyłączyć moduł logowania, należy w menu wybrać *Rozszerzenia/Moduły* i w otwartym oknie kliknąć ikonę *Stan* z lewej strony modułu logowania. Ikona zmieni swój wygląd. Po przejściu na stronę główną zobaczymy, że formularz logowania zniknął. Aby przywrócić wyświetlanie modułu, należy ponownie wejść do panelu *Moduły*, wybrać określony moduł i kliknąć ponownie ikonę *Stan*. Po zmianie wyglądu ikony moduł będzie dostępny na stronie.

## Kolejność modułów

Kolejność wyświetlania modułów jest dowolna. W zależności od potrzeb można ją zmieniać. Po uruchomieniu panelu *Moduły* wyświetli się lista utworzonych modułów. Wystarczy wybrany moduł w kolumnie *Kolejność* chwycić myszą i przesunąć na odpowiednią pozycję.

### 8.3.9. Moduły systemu Joomla!

System Joomla! zawiera kilka standardowych modułów realizujących podstawowe funkcje związane z obsługą strony internetowej. Są to: lista najpopularniejszych artykułów, liczba osób korzystających z witryny, lista nowości, ścieżka powrotu.

#### Najczęściej czytane

Z pomocą modułu *Najczęściej czytane* można wyświetlić na stronie listę najczęściej czytanych artykułów.

#### Ćwiczenie 8.8

Dla tworzonej witryny biura podróży utwórz moduł *Najpopularniejsze artykuły* i umieść w nim listę najczęściej czytanych artykułów.

#### Rozwiążanie

Wybierz w menu *Rozszerzenia/Moduły* i kliknij przycisk *Utwórz*. Z wyświetlonej listy typów modułu wybierz pozycję *Artykuły - Popularne* i w otwartym oknie edycji modułu wpisz tytuł, pod jakim będzie wyświetlany moduł, np. *Najpopularniejsze artykuły*.

Opcję *Pokaż tytuł* zostaw ustawioną na *Pokaż*, a opcję *Stan* — na *Opublikowano*. W polu *Pozycja* wybierz pozycję modułu na stronie, np. *position-3*. W polu *Liczba artykułów* wpisz liczbę artykułów wyświetlnych w module. W polu *Kategoria* wybierz kategorie, z których będą wybierane artykuły. Przejdz do zakładki *Przypisz do pozycji menu* i na liście *Zasada przypisania* określ, kiedy moduł będzie wyświetlany, np. *Tylko na zaznaczonych stronach*. Aby zapisać wprowadzone ustawienia, kliknij przycisk *Zapisz i zamknij*. Przejdz na stronę główną witryny i sprawdź, czy nowy moduł jest wyświetlany.

## Nowości

Podczas tworzenia witryny istnieje również możliwość umieszczenia w niej informacji na temat nowo dodanych i ostatnio zmodyfikowanych artykułów.

### Ćwiczenie 8.9

Dla tworzonej witryny biura podróży utwórz moduł z listą nowych artykułów.

### Rozwiążanie

Wybierz w menu *Rozszerzenia/Moduły* i kliknij przycisk *Utwórz*. Z wyświetlonej listy typów modułu wybierz pozycję *Artykuły - Nowości*. W polu *Tytuł* wpisz tytuł, pod jakim będzie wyświetlany moduł. W polu *Pozycja* ustal pozycję modułu na stronie. W polach *Kategorie* i *Liczba artykułów* ustaw odpowiednie wartości. W polu *Kolejność* możesz ustalić, czy najpierw widoczne będą artykuły ostatnio dodane, ostatnio zmodyfikowane, czy ostatnio redagowane. Zapisz dane, klikając przycisk *Zapisz i zamknij*, i sprawdź, czy nowy moduł jest wyświetlany na stronach witryny.

### Zadanie 8.5

Stosując metodę pokazaną w poprzednich przykładach, dodaj kolejny moduł do witryny internetowej. Będzie to moduł pokazujący liczbę odwiedzających wykorzystujący typ modułu *Gościmi*.

## 8.3.10. Komponenty

Komponenty to narzędzia, które umożliwiają rozbudowę funkcjonalności witryny. Ich zadaniem jest zbieranie, przetwarzanie, przechowywanie i generowanie danych. Po odpowiednim przygotowaniu danych są one za pomocą modułu wyświetlane na stronie. W systemie Joomla! są dostępne komponenty realizujące tylko podstawowe zadania, np. zarządzanie listą kontaktów, reklam lub odnośników. Inne można pobrać z internetu jako rozszerzenia systemu. Komponenty są wyświetlane w środkowej części strony.

### Lista kontaktów

Często na stronie umieszczana jest lista kontaktów, szczególnie w witrynach różnych firm. System Joomla! pozwala nie tylko tworzyć takie listy, ale także dzielić je na kategorie i kontrolować, które kontakty powinny być wyświetlane oraz jakie dane dla każdego kontaktu powinny zostać wyświetlone.

## Ćwiczenie 8.10

Dla tworzonej witryny biura podróży utwórz komponent zawierający listę z kontaktami. Na liście powinny znaleźć się grupy: *Klienci, Pracownicy, Polecanie firmy*.

### Rozwiązanie

Wybierz z menu *Komponenty/Kontakty/Kategorie kontaktów*. Otworzy się okno zarządzania kategoriami kontaktów. Standardowo istnieje jedna kategoria: *Uncategorised*. W niej powinny być umieszczane kontakty, które nie pasują do żadnej innej kategorii. Dodaj nową kategorię, klikając przycisk *Utwórz*. W oknie, które zostanie otwarte, w polu *Tytuł* wpisz tytuł tworzonej kategorii, np. *Klienci*. W polu *Nadrzędna* na liście z prawej strony okna zostaw pozycję *Bez nadzędnej*, co oznacza, że nad tworzoną kategorią nie będzie innych. W polu *Stan* zostaw pozycję *Opublikowano*. Wprowadzone dane zapisz, klikając przycisk *Zapisz i zamknij*. Utworzona kategoria powinna się pojawić na liście kategorii. W podobny sposób postępuj, tworząc kolejne kategorie.

Następnym etapem będzie utworzenie kontaktu należącego do zdefiniowanej kategorii. W tym celu wybierz z menu *Komponenty/Kontakty/ Wizytówki*. Wyświetli się okno zarządzania kontaktami (wizytówkami). Utwórz nowy kontakt, klikając przycisk *Utwórz*. W otwartym oknie wpisz nazwę kontaktu (imię i nazwisko). W polu *Kategoria* wybierz kategorię, do której zostanie przypisany kontakt. W polu *Połącz z użytkownikiem* wybierany jest użytkownik, dla którego tworzony jest ten kontakt. Można zostawić to pole niewypełnione. W polu *Wybierz obraz* można dodać grafikę, np. zdjęcie, które zostanie przypisane do kontaktu. Przejdź do zakładki *Opcje wyglądu* i wybierając *Pokaż* lub *Ukryj*, zdecyduj, które dane zostaną wyświetcone w witrynie. Aby zapisać wprowadzone dane, kliknij przycisk *Zapisz i zamknij*. W podobny sposób dodaj kolejne kontakty.

Nastepnym krokiem będzie powiązanie utworzonego komponentu z menu. Spowoduje to po wybraniu odpowiedniej pozycji menu wyświetlenie w witrynie listy kontaktów. Przejdz do tworzenia nowej pozycji menu (wybierz *Menu/Main Menu/Utwórz pozycję menu*). W otwartym oknie w polu *Typ pozycji menu* wybierz *Kontakty/Pojedynczy kontakt* (jeżeli na liście będzie tylko jeden kontakt) lub *Jedna kategoria kontaktów - lista* (jeżeli będzie ich więcej). W tytule menu wpisz nazwę menu, np. *Klienci*. W polu *Wybierz kontakt* wybierz kontakt przypisany do pozycji menu lub kategorii i zapisz dane, klikając przycisk *Zapisz i zamknij*. Wejdź na stronę główną witryny i sprawdź, czy poprzez menu dostępne są utworzone kontakty.

### 8.3.11. Szablony

Równie istotny jak tworzenie różnych treści i rozmieszczanie ich na stronie jest sposób ich prezentacji. W systemie Joomla! treść oraz sposób jej prezentacji są od siebie oddzielone. Pozwala to na prezentowanie tej samej treści na wiele różnych sposobów. Za wygląd witryny odpowiadają szablony, co powoduje, że można modyfikować jej wygląd, zmieniając wykorzystywany szablon. W systemie Joomla! dostępnych jest kilka szablonów pozwalających zmienić wygląd witryny, ale w internecie jest bardzo dużo innych gotowych szablonów, które można zastosować. Część z nich jest płatna, a inne

są udostępniane za darmo. Ich instalacja przebiega podobnie jak instalacja pakietu językowego.

Operacje na szablonach są dostępne po wybraniu z menu *Rozszerzenia/Szablony*. W oknie *Szablony* jest wyświetlna lista dostępnych szablonów. W polu nad szablonami można wybrać szablon dla strony internetowej (opcja *Witryna*) lub szablon dla zaplecza administracyjnego (opcja *Zaplecze*). Gwiazdka umieszczona w kolumnie *Domyślny* określa, który z szablonów jest domyślny. Szablonem domyślnym jest ten, za pomocą którego wyświetlana będzie witryna internetowa. Domyślny szablon można zmienić. Wystarczy wybrać odpowiedni szablon i kliknąć gwiazdkę znajdującą się w kolumnie *Domyślny*.

Po kliknięciu wybranego szablonu otworzy się okno jego edycji. Po wybraniu zakładki *Wzbogacone* można odczytać i zmienić podstawowe parametry szablonu. W polu *Logo* można umieścić ścieżkę dostępu do pliku graficznego zawierającego logo witryny. W polu *Tytuł* należy wpisać tekst, który będzie wyświetlany, gdy zostanie wyłączone graficzne logo. W polu *Kolor wyróżnień* można ustawić kolorystykę witryny. Po wprowadzeniu zmian można zapisać szablon, klikając przycisk *Zapisz i zamknij*, lub można utworzyć kopię szablonu, klikając *Zapisz jako kopię*.

### Ćwiczenie 8.11

Dla strony internetowej biura podróży zmień szablon domyślny na Beez3.

#### Rozwiążanie

Wybierz z menu *Rozszerzenia/Szablony*. W oknie *Szablony* wybierz z listy opcję *Witryna*. Dla szablonu Beez3 zaznacz gwiazdkę w kolumnie *Domyślny*. Przejdz do strony internetowej i odśwież ją. Sprawdź, jak zmienił się jej wygląd.

### Ćwiczenie 8.12

Zmień szablon domyślny dla zaplecza administracyjnego na Hathor.

#### Rozwiążanie

Wybierz z menu *Rozszerzenia/Szablony*. W oknie *Szablony* wybierz z listy opcję *Zaplecze*. Dla szablonu Hathor zaznacz gwiazdkę w kolumnie *Domyślny*. Wygląd zaplecza administracyjnego powinien ulec zmianie.

### Szablony zewnętrzne

Jeżeli zamierzamy skorzystać z szablonu zewnętrznego, po znalezieniu w internecie odpowiedniego pliku należy pobrać go na swój komputer, a następnie zainstalować w systemie Joomla!. Instalacja przebiega tak samo jak instalacja pliku zawierającego spolszczenie.

## Zadanie 8.6

Sprawdź, jak wyglądają standardowe szablony systemu Joomla!. Poszukaj szablonów zewnętrznych i spróbuj dołączyć je do swojej witryny. Wybierz szablon, który najlepiej pasuje do treści zamieszczonych w Twojej witrynie, i zastosuj go.

# 8.4. Pytania i zadania

## 8.4.1. Pytania

1. Do czego służą systemy zarządzania treścią (CMS)?
2. Wymień podstawowe cechy systemów zarządzania treścią.
3. Jakie oprogramowanie jest niezbędne do prawidłowego działania systemu WordPress?

## 8.4.2. Zadania

### Zadanie 1.

Zaprojektuj i utwórz w programie WordPress stronę internetową, która będzie zawierała blog poświęcony przyszłości samochodów elektrycznych. Przygotuj kilka wpisów na ten temat. Wybierz motyw strony pasujący do tego zagadnienia. Ustaw moderację wstawianych komentarzy.

Podczas projektowania swojego bloga zastanów się, kto go będzie czytał. Jakie informacje chciałbyś publikować? Dla kogo tworzysz blog? Jak często powinieneś publikować nowe informacje?

### Zadanie 2.

Zaprojektuj i utwórz w programie WordPress witrynę internetową szkoły językowej prowadzącej kursy języka angielskiego. Witryna będzie się składała z czterech stron: *Główna*, *Nasza oferta*, *O nas*, *Kontakt*. Zaprojektuj wygląd poszczególnych stron oraz wybierz motyw pasujący do tematu.

### Zadanie 3.

Wykorzystując system zarządzania treścią, zaprojektuj strukturę witryny internetowej swojej szkoły. W tworzonej witrynie umieść następujące moduły: *nagłówek*, *menu*, *logowanie*, *treść*. Rozplanuj położenie modułów. Dla każdego z nich przygotuj odpowiednią treść. W module *menu* umieść trzy pozycje: *Historia szkoły*, *Patron* i *Kierunki kształcenia*. Kliknięcie każdej pozycji menu otworzy podstronę zawierającą odpowiednią treść. W module *treść* umieść podstawowe informacje o szkole. W module *nagłówek* wstaw jej zdjęcie.

# 9

## Testowanie i publikowanie strony

Mimo że strony internetowe wyglądają dobrze i działają prawidłowo, czasami zawierają różnego rodzaju błędy, dłujo się ładują lub nie są wykrywane przez wyszukiwarki internetowe. Aby ustrzec się przed takimi zdarzeniami, należy projektowaną stronę internetową wielokrotnie i w różny sposób sprawdzić i przetestować. Testowanie i sprawdzanie powinno się odbywać w następujących etapach:

- sprawdzanie poprawności strony,
- walidacja strony,
- testowanie strony w przeglądarkach,
- publikowanie strony,
- testowanie szybkości ładowania strony,
- testowanie funkcjonalności strony,
- optymalizacja strony,
- pozycjonowanie strony.

### 9.1. Sprawdzanie poprawności strony

Przed umieszczeniem strony w internecie trzeba sprawdzić jej poprawność. Nie chodzi tylko o to, czy strona prawidłowo otwiera się w kilku najpopularniejszych przeglądarkach. Należy sprawdzić, czy na stronie nie ma błędów ortograficznych, czy działają wszystkie odnośniki, czy grafika ładuje się szybko oraz czy strona nie zawiera innych błędów.

#### 9.1.1. Ortografia i interpunkcja

W pierwszej kolejności, korzystając z dostępnych programów do edytowania tekstu, należy sprawdzić, czy tekst, który umieściliśmy na stronie, nie zawiera błędów ortograficznych i stylistycznych. Niektóre programy do tworzenia stron internetowych mają

wbudowane słowniki ortograficzne, które już w trakcie tworzenia strony podkreślają błędnie wprowadzany tekst.

Nic bardziej nie zniechęca do odwiedzania strony niż źle sformatowany i zapisany z błędami tekst, niepoprawnie wstawione znaki interpunkcyjne i chaotycznie porozmieszczone spacje.

## 9.1.2. Nagłówek strony

Należy zadbać o to, aby nagłówek strony zawierał: tytuł strony, słowa kluczowe, opis strony, definicję strony kodowej oraz wersji językowej.

## 9.1.3. Poprawność hiperłączy

Należy sprawdzić poprawność wszystkich odnośników działających w obrębie strony oraz odnośników do innych stron. Poprawność odnośników trzeba sprawdzić również po umieszczeniu strony na serwerze.

## 9.1.4. Grafika

Należy pamiętać o tym, aby wszystkie ilustracje umieszczane na stronie były opatrzone zastępczym tekstem (ułatwi to m.in. nawigację osobom niepełnosprawnym, np. niewidomym korzystającym ze specjalnych czytników). W przypadku gdy grafika z jakiegoś powodu nie zostanie wyświetlona na stronie, użytkownik będzie mógł się zorientować, co ona przedstawia.

Ilustracje umieszczane na stronie powinny być zapisane w plikach o odpowiednich formatach. Proste obrazy należy zapisać w formacie GIF. Do zapisu zdjęć i grafik zawierających przejścia tonalne trzeba używać formatu JPG, wybierając odpowiedni procent kompresji. W formacie PNG należy zapisywać grafiki zawierające efekt przezroczystości.

Pliki graficzne powinny zostać zoptymalizowane, tak aby były możliwie małe przy zachowaniu możliwie najlepszej jakości (najlepszego wyglądu). Dzięki temu można zaoszczędzić miejsce na serwerze oraz przyspieszyć wczytywanie stron.

Po umieszczeniu strony na serwerze należy jeszcze raz sprawdzić, czy nie ma problemu z ładowaniem grafiki.

## 9.1.5. Przeglądarka internetowa

Nie wszyscy użytkownicy internetu korzystają z tej samej przeglądarki. Należy nie tylko sprawdzić, jak strona będzie wyświetlana w różnych przeglądarkach, ale także przetestować ją na różnym sprzęcie i w różnych systemach operacyjnych.

Również typ i rozdzielcość monitora wpływają na wygląd strony. Trzeba sprawdzić, jak strona wygląda w zależności od zastosowanej rozdzielcości ekranu oraz typu monitora.

## 9.1.6. Urządzenia mobilne

Obecnie używane urządzenia, na których wyświetlane są strony internetowe, mają bardzo różne rozmiary — od bardzo małych (telefony) do bardzo dużych (monitory komputerów stacjonarnych). Należy sprawdzić, jak dana strona będzie wyświetlana na różnych urządzeniach.

W internecie jest wiele darmowych aplikacji, które umożliwiają sprawdzenie, jak testowana strona będzie wyglądała na urządzeniach mobilnych o różnej wielkości ekranu oraz w orientacji pionowej i orientacji poziomej.

W wyniku wykonania testu na ekranie zostaną wyświetcone najczęściej monitory różnych urządzeń mobilnych z zawartością testowanej strony. Widok strony w każdym urządzeniu umożliwia przetestowanie jej zachowania. Często możliwe jest przewijanie strony oraz sprawdzenie działania linków.

## 9.2. Weryfikacja strony

Przeglądarki internetowe są wyposażone w narzędzia do „naprawy” błędów w kodzie źródłowym. Niestety nie ma gwarancji, że poprawnie zinterpretują każdy kod. Jeżeli kod zostanie zinterpretowany nieprawidłowo, strona może zostać błędnie wyświetlona lub w ogóle się nie wyświetli. Dlatego powinno się sprawdzać poprawność składni każdego dokumentu HTML i dokumentu CSS.

Do sprawdzenia, czy strona jest wolna od błędów, można użyć odpowiedniego programu lub odpowiedniej usługi dostępnej w internecie. Weryfikatory sprawdzają zgodność kodu ze standardami wyznaczonymi przez organizację W3C.

### UWAGA

Organizacja W3C (ang. [World Wide Web Consortium](#)) opracowuje i zatwierdza standardy dotyczące pisania i przesyłania stron WWW. Zrzesza ponad 400 organizacji, firm, agencji rządowych i uczelni z całego świata. Należy pamiętać, że rekomendacje publikowane przez tę organizację nie mają mocy prawnej nakazującej ich użycie.

## 9.2.1. Weryfikatory

Weryfikatory skanują dokument w poszukiwaniu niezgodności ze specyfikacją języka oraz błędów w kodzie. Wyszukują brakujące bądź nieprawidłowo użyte znaczniki lub błędnie zastosowane style.

Jak już wiemy, plik HTML powinien się zaczynać od deklaracji typu dokumentu (DTD). Specyfikuje on standard języka używanego na stronie. Na przykład:

```
<!DOCTYPE html>
```

Poza deklaracją typu dokumentu, która określa, jaka wersja HTML jest używana na stronie, należy ustalić język dokumentu:

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

Validator będzie sprawdzał zgodność kodu z zadeklarowanym standardem języka HTML używanego na stronie.

## Validatory HTML

Działanie validatora HTML polega na sprawdzeniu kodu HTML strony oraz jego zgodności ze standardami wyznaczonymi przez organizację W3C.

Są różne rodzaje validatorów. Jedne szukają tylko błędów, inne proponują sposoby ich naprawy, jeszcze inne objaśniają, które instrukcje mogą prowadzić do nieoczekiwanych błędów. Organizacja W3C opracowała własny, bezpłatny validator HTML, który udostępnia na stronie <http://validator.w3.org/>.

## Validatory CSS

Podobnie jak validator HTML działa validator kodu CSS. Sprawdza on poprawność kaskadowych arkuszy stylów pod kątem zgodności ze standardami CSS wyznaczonymi przez organizację W3C. Darmowy validator udostępniany przez organizację W3C znajduje się na stronie <http://jigsaw.w3.org/css-validator/>.

Walidacja strony gwarantuje tylko sprawdzenie kodu pod kątem poprawności składni, nie gwarantuje natomiast poprawnego wyświetlania witryny w różnych przeglądarkach. Wynika to z tego, że niektóre nowe funkcjonalności proponowane przez twórców przeglądarek nie są częścią standardu W3C. W takim przypadku validator może ostrzegać o błędach, pomimo że większość popularnych przeglądarek poprawnie wyświetli stronę.

## 9.2.2. Programy do walidacji stron

Istnieją programy do walidacji stron, z których można korzystać na komputerze użytkownika, pracując w trybie *offline*. Wtedy należy podać nazwę i lokalizację pliku, który ma zostać sprawdzony. Istnieją również programy pracujące w trybie *online*, które sprawdzają strony umieszczone na serwerze i dostępne w internecie. W takiej sytuacji wystarczy podać adres URL strony oraz opcje jej sprawdzania.

Większość validatorów, które testują stronę dostępną w internecie, oprócz sprawdzania kodu ma także możliwość sprawdzania czasu ładowania strony, poprawności odnośników oraz wyglądu strony w różnych przeglądarkach.

### Wave Accessibility Tool

Narzędzie to sprawdza kod strony pod kątem jego dostępności zgodnie z różnymi standardami. Testowaną stronę wyświetla z ikonami sygnalizującymi ewentualne problemy z dostępnością. Strona do testowania może zostać wczytana z dysku twardego lub

może być udostępniona *online*. Można również w wybranej przeglądarce zainstalować pasek narzędziowy lub zakładkę, za pomocą której aktualna strona zostanie wysłana do przetestowania.

## W3C HTML Validation Service

Jest to darmowy validator HTML udostępniany przez organizację W3C. Można wysłać bezpośrednio do validatorsa dokument zawierający stronę, która ma zostać przetestowana, lub podać adres URL strony. Jest to najbardziej wiarygodne narzędzie do sprawdzania stron pod kątem zgodności ze standardami.

## W3C CSS Validation Service

Jest to darmowy validator CSS udostępniany przez organizację W3C, przeznaczony do walidacji kaskadowych arkuszy stylów. Można go pobrać i używać na własnym komputerze. Można również wysłać do validatorsa własne arkusze stylów lub podać adres URL testowanej strony.

## W3C Link Checker

Jest to darmowy validator opracowany przez organizację W3C do testowania odnośników. Gdy podamy adres URL strony, W3C Link Checker wyszuka na niej niedziałające odnośniki.

### Zadanie 9.1

Wykonaj walidację utworzonej przez siebie strony internetowej. Przeanalizuj otrzymane wyniki. Popraw kod strony.

## Podsumowanie

Walidacja strony nie gwarantuje jej poprawnego wyświetlania. Dlatego wśród twórców stron internetowych istnieje duża grupa przeciwników walidacji. Argumentują oni, że nawet po walidacji kodu należy stronę testować w różnych przeglądarkach. Mimo że kod nie ma błędów w składni, nie zawsze działa zgodnie z naszymi oczekiwaniemi. Jeżeli natomiast strona jest poprawnie wyświetlana, to nie ma znaczenia, czy została zapisana zgodnie ze standardami. Jej użytkownik nie będzie szukał w kodzie źródłowym błędów języka HTML czy CSS. Prawdziwym testem jest to, jak strona będzie wyświetlana w przeglądarce.

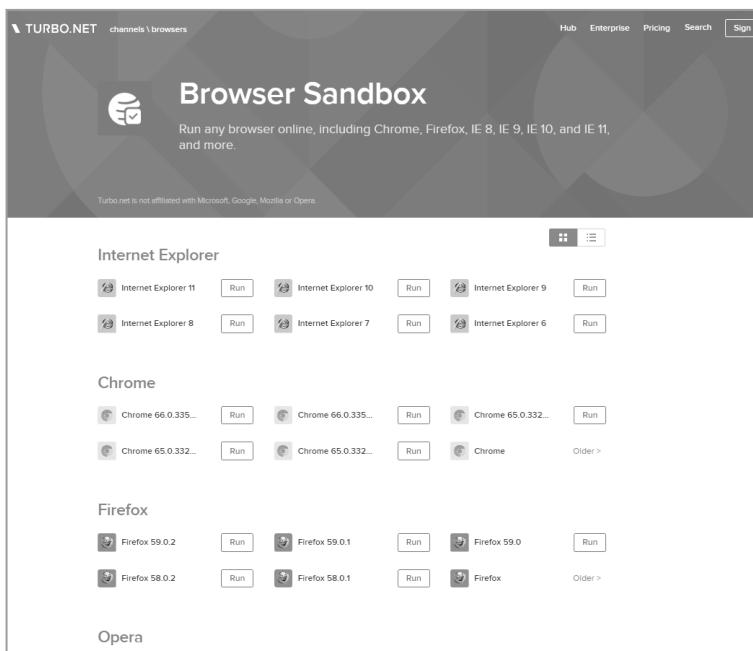
Zwolennicy walidacji twierdzą, że jeżeli strona nie zawiera błędów, to istnieje większe prawdopodobieństwo, że będzie działać dobrze niezależnie od przeglądarki, systemu operacyjnego czy sprzętu, którym dysponuje użytkownik.

Jeżeli strona zawiera błędy, to wyszukiwarki w trakcie wyszukiwania słów kluczowych w różny sposób wykonują działania związane z tymi błędami. Może to spowodować niezaindeksowanie części lub nawet całej strony.

Ostatni i najważniejszy argument to profesjonalizm strony. Błędy na stronie świadczą o niskich umiejętnościach jej twórcy.

## 9.3. Testowanie strony w przeglądarkach

Zanim witryna zostanie opublikowana, należy sprawdzić, jak będzie wyświetlana w różnych przeglądarkach. Nie zawsze istnieje możliwość zainstalowania równocześnie wielu przeglądarek na komputerze. Problemem może być również zainstalowanie wielu wersji tej samej przeglądarki. Wtedy można wykorzystać emulatory przeglądarek internetowych. Jednym z nich jest **Browser Sandbox**, dostępny w internecie pod adresem <https://turbo.net/browsers>. Na liście tego narzędzia znajdują się różne wersje najpopularniejszych przeglądarek (rysunek 9.1).



**Rysunek 9.1.** Emulatory przeglądarek

Gdy klikniemy ikonę wybranej przeglądarki, uruchomi się ona w środowisku wirtualnym. **Turbo.net** obsługuje wtyczki przeglądarek, takie jak Firebug i Internet Explorer Developer Toolbar. Ponadto Turbo.net obsługuje konsole debugowania CSS i JavaScript.

Inny emulator przeglądarek internetowych znajdziemy na stronie <http://browsershots.org/>. Przy użyciu tego narzędzia można sprawdzić wygląd strony internetowej dla różnych systemów operacyjnych i w różnych przeglądarkach. Program wykonuje zrzuty ekranu z wybranych wersji przeglądarek.

### Zadanie 9.2

Przetestuj działanie utworzonej strony internetowej w różnych przeglądarkach. Do testowania wykorzystaj dostępne w internecie emulatory przeglądarek. Porównaj efekty wyświetlania strony w tych przeglądarkach.



## 9.4. Publikowanie strony internetowej

Publikowanie strony internetowej może się odbywać na kilka sposobów:

- poprzez umieszczenie strony na płatnym serwerze,
- poprzez umieszczenie strony na darmowym serwerze,
- poprzez utworzenie własnego serwera.

### 9.4.1. Umieszczanie strony na płatnym lub darmowym serwerze

Usługodawców oferujących miejsce na serwerze jest w internecie wielu. Można wybrać jedną z następujących usług:

- hosting — wynajęcie od dostawcy internetowego miejsca na serwerze do przechowywania plików,
- serwer dedykowany — udostępnienie oddzielnego komputera jako serwera wraz z systemem operacyjnym i ze wsparciem serwisowym dla jednego usługobiorcy,
- kolokacja — wynajęcie od dostawcy internetowego miejsca na serwer będący własnością usługobiorcy.

Oferty dostępne w internecie różnią się parametrami oraz ceną. Można znaleźć również darmowe usługi hostingu. Najczęściej wiążą się one z dużymi ograniczeniami, takimi jak:

- niewystarczająca przestrzeń dyskowa,
- brak dostępu do baz danych,
- brak obsługi języków skryptowych,
- reklamy umieszczane przez usługodawcę.

Po wybraniu jednej z ofert dotyczących miejsca na serwerze należy umieścić stronę lub witrynę internetową w wyznaczonym obszarze. Do tego służy kolejna usługa internetowa — protokół FTP.

### 9.4.2. Protokoły internetowe

Do publikowania stron w internecie niezbędne jest miejsce na serwerze oraz oprogramowanie obsługujące jeden z protokołów sieciowych służących do przesyłania plików. Najpopularniejszymi protokołami sieciowymi są FTP oraz SCP.

#### Protokół FTP

FTP (ang. *File Transfer Protocol*) jest podstawowym protokołem przeznaczonym do przesyłania plików między komputerami. Wykorzystuje on architekturę klient-serwer, tzn. jeden komputer pełni funkcję serwera, a drugi ma zainstalowane odpowiednie oprogramowanie klienta. Oprogramowanie klienta umożliwia wysyłanie plików na serwer i odbieranie ich z niego.

Oprogramowanie zainstalowane na komputerze klienta często jest nazywane **klientem FTP**. Najprostsze programy FTP są dostępne w systemie operacyjnym; nowsze wersje przeglądarek internetowych wyposażają w funkcje klienta FTP. Do popularniejszych programów obsługujących protokół FTP należą Total Commander i FileZilla. Główną wadą protokołu FTP jest brak szyfrowania danych. Obecnie ten protokół wychodzi z użycia zastępowany przez protokół FTPS.

## Protokół SSH

SSH (ang. *Secure Shell*) to protokół komunikacyjny służący głównie do zdalnej pracy w sieci. Może zostać wykorzystany również do przesyłania plików. Od protokołu FTP różni się tym, że transfer danych jest szyfrowany. Nazwy SSH używa się także dla całej rodziny protokołów, które oferują podobne usługi. Przesyłanie plików odbywa się za pomocą opartego na SSH protokołu SCP (ang. *Secure CoPy* — bezpieczne kopiowanie plików). Inne usługi to zdalna kontrola i tunelowanie (tworzenie dodatkowego zaszyfrowanego kanału).

### 9.4.3. Umieszczanie strony na serwerze

Korzystanie z klienta FTP jest bardzo proste. Po uruchomieniu programu należy zalogować się na konto, a następnie przeciągnąć pliki lub katalogi z zawartością strony z okna klienta do okna serwera.

#### Total Commander

**Total Commander** jest popularnym menedżerem plików. Ma również wbudowanego klienta FTP, w związku z tym nadaje się do przesyłania plików zaprojektowanej witryny internetowej na serwer. Aby korzystać z tego narzędzia, trzeba mieć na serwerze konto oraz znać do niego login i hasło.

Po uruchomieniu programu Total Commander (rysunek 9.2) należy wybrać z menu *Net/FTP Connect* lub kliknąć ikonę *Connect to FTP*. Zostanie otwarte okno z listą dostępnych połączeń (rysunek 9.3). Po kliknięciu przycisku *New connection* zostanie wyświetcone kolejne okno (rysunek 9.4), w którym należy skonfigurować połączenie z serwerem.

Wpisane dane dotyczące serwera trzeba zatwierdzić.

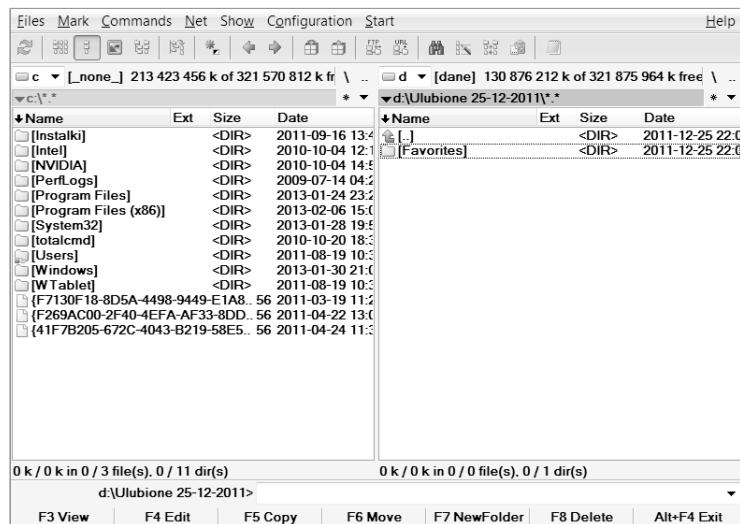
Teraz na liście połączeń powinno się znaleźć zdefiniowane połączenie. Aby połączyć się z serwerem, wystarczy wybrać to połączenie z listy i kliknąć przycisk *Connect*.

Na serwer należy skopiować całą strukturę witryny, zachowując układ katalogów i plików. Główny plik zawierający stronę startową utworzonej witryny powinien mieć nazwę *index.html*, *index.htm* lub *index.php* (dla serwerów obsługujących język skryptowy PHP).

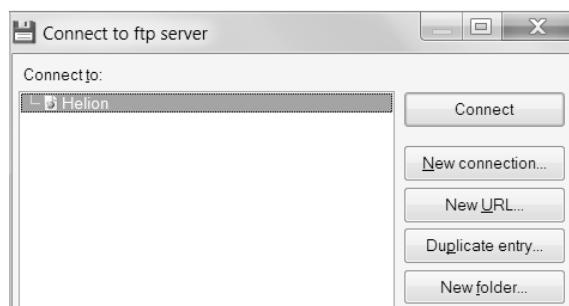
Struktura katalogów na serwerze może być różna. Zależy to od systemu operacyjnego serwera, oprogramowania obsługującego serwer oraz od administratora. Najczęściej główny katalog serwisu internetowego, w którym należy umieścić witrynę, ma nazwę *public\_html*.

**Rysunek 9.2.**

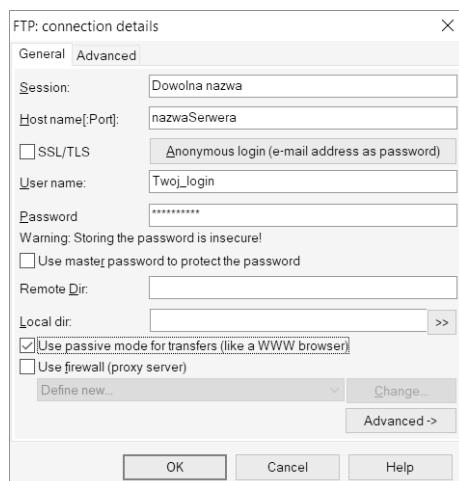
Okno programu  
Total Commander

**Rysunek 9.3.**

Okno Klienta FTP

**Rysunek 9.4.**

Nowe połączenie



## FileZilla

FileZilla jest darmowym klientem FTP, dostępnym na zasadach open source. Obsługuje protokoły: FTP, FTPS i SFTP oraz współpracuje z protokołem IPv6 i serwerami proxy. Program jest łatwy w obsłudze, a także dostępny w wielu wersjach językowych.

Aby połączyć się z serwerem FTP, po uruchomieniu programu należy w polu *Serwer*: wpisać adres serwera, podać nazwę użytkownika i hasło (jeżeli jest wymagane), podać numer portu (jeśli stosowany port nie jest portem domyślnym) i kliknąć przycisk *Szybkie łączenie* (rysunek 9.5).

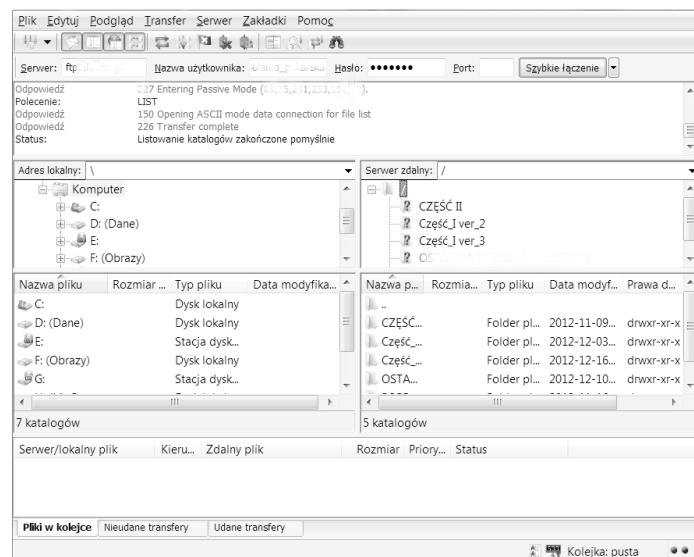


**Rysunek 9.5.** Łączenie się z serwerem FTP

Po połączeniu się z serwerem w panelu *Serwer zdalny*: wyświetlna zostanie nazwa dostępnego na serwerze folderu (wraz z drzewem podfolderów). W panelu poniżej wyświetlana jest jego zawartość (rysunek 9.6).

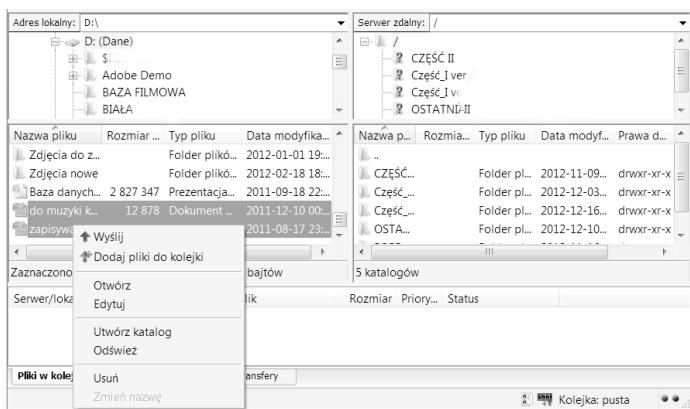
### Rysunek 9.6.

Połączenie z serwerem



Jednym ze sposobów pobierania plików z serwera lub przesyłania ich na serwer jest ich zaznaczenie i dwukrotne kliknięcie. Zostaną one dodane do kolejki transferu (dolny panel okna). Transfer rozpoczęcie się automatycznie.

Innym sposobem jest kliknięcie wybranych plików prawym przyciskiem myszy i wybór z menu kontekstowego opcji *Wyślij* (jeśli pliki mają zostać wysłane na serwer) lub *Pobierz* (jeśli pliki mają zostać z serwera pobrane — rysunek 9.7). Można również przeciągnąć pliki z jednego panelu do drugiego.



**Rysunek 9.7.** Transfer plików

Jeżeli pliki mają zostać przekazane w późniejszym czasie, można po ich zaznaczeniu kliknąć prawym przyciskiem myszy i z menu kontekstowego wybrać opcję *Dodaj pliki do kolejki*. Po wybraniu z menu *Transfer/Rozpocznij (Ctrl+P)* rozpocznie się transfer plików.

### Zadanie 9.3

Wybierz serwer, na którym zostanie opublikowana przygotowana przez Ciebie strona internetowa. Korzystając z protokołu FTP oraz poznanych narzędzi, umieść swoją witrynę internetową na serwerze.

## 9.5. Testy szybkości wczytywania strony

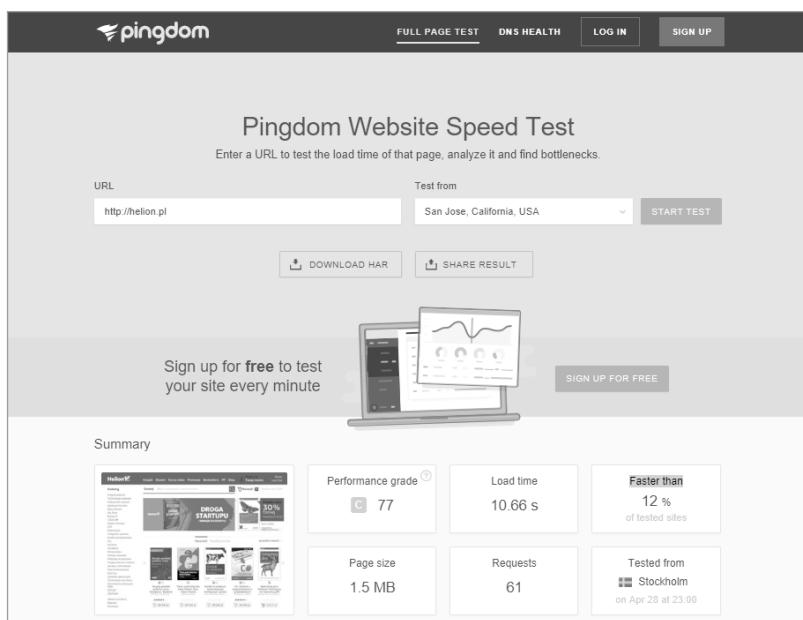
To, jak szybko witryna jest wczytywana, stanowi jeden z najważniejszych elementów wpływających na jej atrakcyjność. Szybkość wczytywania zależy od wielu czynników, m.in. od szybkości serwera, na którym została umieszczona strona, elementów multimedialnych znajdujących się na stronie czy błędów w kodzie strony.

W internecie istnieje wiele narzędzi pomagających ocenić szybkość ładowania witryny internetowej oraz narzędzi znajdujących elementy witryny, które powodują największe opóźnienia w jej ładowaniu.

### 9.5.1. Narzędzia do testowania szybkości wczytywania strony

#### Pingdom

Innym narzędziem do testowania szybkości ładowania strony jest **Pingdom**. Narzędzie to jest dostępne pod adresem <http://tools.pingdom.com/fpt/>. Prezentuje parametry poszczególnych elementów witryny wczytywanych w czasie ładowania. Pokazuje wielkość ładowanych elementów witryny, ich umiejscowienie oraz wynikowy czas ładowania całego serwisu (rysunek 9.8).



**Rysunek 9.8.** Test szybkości ładowania przykładowej strony

## Web Page Analyzer

Pod adresem [www.websiteoptimization.com](http://www.websiteoptimization.com) znajdziemy wiele narzędzi do optymalizowania strony internetowej. Między innymi udostępniane są usługi pozycjonowania stron internetowych, optymalizowania witryny, a także darmowy Web Page Analyzer, służący do testowania szybkości ładowania strony internetowej. Po przetestowaniu narzędzie to generuje raport na temat poszczególnych elementów strony internetowej. Podaje czas ładowania bloków witryny, jak również czas ładowania witryny przy różnych szybkościach łącza internetowego.

### Zadanie 9.4

Korzystając z poznanych narzędzi, przetestuj szybkość ładowania swojej strony internetowej. Przeanalizuj otrzymane wyniki. Określ, które elementy strony internetowej spowalniają jej ładowanie. Zoptymalizuj te elementy.

## 9.6. Testy funkcjonalności strony internetowej

Profesjonalne testy witryny internetowej pozwalają gruntownie ocenić jej funkcjonalność. Testy te badają wybraną witrynę internetową za pośrednictwem specjalnych narzędzi, które indeksują stronę główną wraz z jej podstronami. Wymagane jest podanie adresu witryny oraz parametrów testu.

Najczęściej w ramach tych testów oceniane są wszystkie elementy serwisu, pogrupowane w kilku kategoriach.

Kategorie dla całego serwisu:

- wartość (liczbową) strony według wskaźnika Google PageRank,
- liczba stron zindeksowanych w wyszukiwarkach Google i Yahoo!,
- liczba linków prowadzących do serwisu poprzez Google i Yahoo!,
- liczba osiągalnych podstron na podstawie zindeksowanej struktury serwisu.

Kategorie dla struktury serwisu:

- liczba linków prowadzących do poszczególnych podstron,
- ślepe odnośniki — błędne linki przekierowujące do nieistniejących stron,
- połączenia RSS i HTTPS.

Kategorie dla podstron:

- wielkość pliku HTML,
- czas ładowania strony,
- walidacja HTML,
- ocena elementów graficznych na stronie pod względem obecności atrybutów alt i title,
- ocena linków pod względem obecności atrybutu title,
- stosunek treści strony do kodu.

Kategorie dla SEO:

- analiza metatagów,
- analiza optymalizacji strony pod kątem kryteriów title, description i keywords,
- ranking słów najczęściej występujących na stronie, w linkach oraz treściach atrybutu alt i znacznika <title>,
- analiza pod kątem unikatowości treści.

Po przeprowadzonych badaniach zostaje automatycznie wygenerowany raport. Najczęściej zawiera on oceny testowanych elementów strony oraz ich porównanie ze średnimi wynikami uzyskiwanymi w testach.

### UWAGA

SEO (ang. *Search Engine Optimization*) to optymalizacja witryny dla wyszukiwarek internetowych, zwana także pozycjonowaniem. Polega na działaniach prowadzących do uzyskania przez witrynę jak najwyższej pozycji w wynikach wyszukiwania.

Można również przeprowadzać testy funkcjonalności witryny internetowej, korzystając z bezpłatnych narzędzi dostępnych w interecie. Jednym z nich jest [webstatsdomain.com](http://www.webstatsdomain.com/) i można je znaleźć pod adresem <http://www.webstatsdomain.com/>. Umożliwia przeprowadzenie automatycznego testu stron internetowych w ramach pozycjonowania WWW i optymalizacji SEO. Po podaniu adresu witryny otrzymamy analizę struktury strony, pozycję strony w wyszukiwarkach i analizę serwisu według rankingu Alexa.com (rysunek 9.9).

The screenshot shows the webstatsdomain.com interface for the domain helion.pl. At the top, there's a search bar with 'helion.pl' and buttons for 'analyze' and 'compare'. Below the header, it says 'Helion.pl is UP and reachable, 52 seconds ago'. It provides a detailed report: Helion.pl is ranked 69,328th in the world, gets 94.3% of traffic from Poland, and has a value of \$23143. It has 353 backlinks and a 43% SEO score. The 'Safety and Legitimacy' section shows a green status for safety and legitimacy. The 'Social Signals' section shows 19 signals. The 'Links' section lists links to the website, Twitter, Google+, and other platforms. The 'Website Topics' section includes 'books, informatyka + 3 more...'. The 'Top Keywords from Search Engines' section includes 'architektura informacji + 4 more...'. At the bottom, there are navigation links for Safety and Legitimacy, Social Monitoring, Site Info, SEO, Visitors, Website Data, and Contact & support.

**Rysunek 9.9.** Wynik testowania strony przez webstatsdomain.com

### Zadanie 9.5

Korzystając z poznanych narzędzi, przetestuj swoją stronę internetową pod względem funkcjonalności. Przeanalizuj otrzymane wyniki. Określ, które elementy strony powinny zostać poprawione.

## 9.7. Optymalizacja strony internetowej

Optymalizacja strony internetowej to wszystkie działania dotyczące jej budowy, treści i struktury, prowadzące do tego, że znajdzie się ona na wysokiej pozycji w wynikach wyszukiwarek internetowych.

Optymalizując treści wchodzące w skład strony, należy zwrócić uwagę na:

- umieszczenie odpowiednich treści w znaczniku <title>,
- odpowiednie umieszczenie słów kluczowych w teksthach,
- ustalenie hierarchii nagłówków,
- odpowiednie zaprojektowanie menu,
- odpowiednie zaprojektowanie elementów tworzących linki wewnętrzne witryny,
- ustalenie tekstu zastępczego dla elementów graficznych.

Optymalizując strukturę strony internetowej, należy zwrócić uwagę na:

- stosowanie się na stronie do standardów W3C,

- oddzielenie warstwy logicznej struktury strony od warstwy jej prezentacji (stosowanie zewnętrznych arkuszy CSS),
- zmodyfikowanie elementów strony w celu skrócenia czasu jej ładowania,
- skrócenie długich linków.

### Zadanie 9.6

Sprawdź, czy wszystkie elementy utworzonej strony internetowej spełniają wymagania związane z jej optymalizacją.



## 9.8. Pozycjonowanie strony

Pozycjonowanie strony internetowej to działania, których celem jest umieszczenie strony jak najwyżej w wynikach wyszukiwania. Istnieje wiele technik pozycjonowania strony. Dzielimy je na dwie grupy:

- pozycjonowanie naturalne,
- techniki manipulacyjne (spammerskie).

**Pozycjonowanie naturalne** polega na podnoszeniu atrakcyjności strony poprzez zamieszczanie ciekawych treści, rozbudowę zawartości strony i dostosowywanie się do zasad działania wyszukiwarek. Ważne jest dobieranie odpowiednich słów i fraz kluczowych, istotna jest też popularność strony.

**Techniki manipulacyjne** polegają na próbie obejścia mechanizmów wyszukiwania w wyszukiwarkach internetowych. Manipulacja może polegać m.in. na zbudowaniu innej zawartości strony dla użytkowników i innej dla automatów wyszukujących. Takie działania mogą doprowadzić do poważnych konsekwencji dla właściciela strony, włącznie z jej usunięciem z indeksu wyszukiwarki.

### 9.8.1. Działanie automatów wyszukiwarek

Aby strona internetowa była dostępna dla wyszukiwarki internetowej, musi się znaleźć w indeksie wyszukiwarki. Każda wyszukiwarka jest wyposażona w specjalny program zwany **robotem** (inne określenia to **bot**, **pająk**), którego zadaniem jest ciągłe przeszukiwanie internetu. Robot odczytuje zawartość strony oraz odnajduje na niej odnośniki do innych stron i odczytuje zawartość stron, do których te odnośniki prowadzą.

Zebrane w ten sposób informacje są indeksowane, katalogowane i umieszczane w bazach danych wyszukiwarki. W bazach tych umieszczane są również kopie stron, co pozwala na wyświetlanie strony z pamięci podręcznej wyszukiwarki. Są wyszukiwarki, które indeksują zawartość całej strony internetowej, oraz takie, które indeksują tylko stronę główną. Roboty w regularnych odstępach czasowych, ustalonych przez algorytm wyszukiwarki, odczytują wszystkie strony, aby zaktualizować zawartość indeksów wyszukiwarek. Strony, których zawartość jest często zmieniana, są aktualizowane częściej. Na przykład strony serwisów informacyjnych mogą być odwiedzane przez roboty

wyszukiwarek nawet co kilka lub kilkanaście minut. Natomiast serwisy, których treść rzadko się zmienia, są odwiedzane co kilka dni. Wejście robota na stronę jest określone terminem *crawl*.

Informacje, które po wejściu na stronę zebrał robot, są poddawane analizie według specjalnych algorytmów, a wyniki analiz są zwracane w wyszukiwarkach internetowych. Wyliczenia dotyczące ważności strony są dokonywane na podstawie liczby odnośników prowadzących do danej strony. Analizowana jest również zależność między wyrazami znajdującymi się na stronie głównej i na podstronach. Dla oceny strony ważne są również: częstotliwość jej aktualizacji, wiek witryny oraz jej charakter.

Głównymi czynnikami, które wpływają na pozycję strony, są:

- liczba i jakość linków prowadzących do strony,
- budowa kodu strony,
- treść strony,
- częstość odwiedzania strony.

Wyszukiwarki stosują różne algorytmy do wyszukiwania stron. Sposób działania robotów w każdej z wyszukiwarek jest inny. Z tych powodów wyniki wyszukiwania dla tych samych kryteriów w zależności od wybranej wyszukiwarki mogą się znacznie różnić.

Należy również zdawać sobie sprawę z tego, że wiele stron, które ładnie wyglądają w oknie przeglądarki, jest zupełnie nieczytelnych dla robota wyszukiwarki. Roboty wyszukiwarki „nie widzą” tekstów zawartych w elementach graficznych, nie rozpoznają menu graficznego. Niektóre z nich traktują Flash jako grafikę, w związku z czym nie potrafią odczytać tekstu z takich stron.

### Zadanie 9.7

Sprawdź, czy Twoja strona internetowa jest znajdowana przez różne wyszukiwarki internetowe. Zastanów się, co zrobić, aby podnieść pozycję strony w różnych wyszukiwarkach.

## 9.8.2. Dodawanie strony do indeksu wyszukiwarki

Jedną z metod dodawania strony do indeksu wyszukiwarki jest wypełnienie specjalnego formularza na stronie wybranej przeglądarki. Nie jest to najlepszy sposób rejestracji strony, ponieważ strona może zostać dodana do bazy dopiero po paru tygodniach od daty rejestracji.

Lepszą metodą jest umieszczenie łącza do nowej strony na innej, działającej już stronie lub umieszczenie jej w katalogu stron, tak aby robot wyszukiwarki trafił na nową stronę. Najlepiej, aby łącze do nowej strony znalazło się na stronie, która jest popularna i często odwiedzana.

Ponieważ roboty niezbyt dobrze radzą sobie z linkami w skryptach JavaScript i ze stronami dynamicznymi, korzystnie jest zamieszczać linki w postaci znacznika `<a href...>`.

Niektóre wyszukiwarki mają narzędzia do sprawdzania, czy strona jest obecnie indeksowana przez wyszukiwarkę. Można też w polu zapytań wyszukiwarki umieścić polecenie `site:adres_strony`.

### Zadanie 9.8

Dodaj swoją stronę internetową do wyszukiwarki Google. Aby to zrobić, wypełnij specjalny formularz dostępny na tej stronie.

## 9.8.3. Google Search Console

**Google Search Console** to bezpłatna usługa Google, przeznaczona dla administratorów stron internetowych. Pomaga monitorować i kontrolować pozycję strony internetowej przez wyszukiwarkę Google. Umożliwia wysyłanie i sprawdzanie statusu mapy strony internetowej, przeglądanie statystyk indeksowania strony przez roboty Google oraz analizę jej widoczności w wyszukiwarce Google. Pozwala także na przesyłanie nowych treści do zaindeksowania i usuwanie określonej zawartości z wyników wyszukiwania.

Udostępniając stronę internetową w Google Search Console, dajemy dostęp do jej wszystkich treści, takich jak: teksty, obrazy, linki i przekierowania. Wszystkie te treści będą indeksowane, a Google Search Console po analizie przekaże sugestie, jakie zmiany powinny zostać wprowadzone na stronie, aby była ona bardziej przyjazna algorytmom wyszukiwarki Google. Dodanie strony do usługi pozwoli na monitorowanie i rozwiązywanie problemów ze spamem oraz złośliwym oprogramowaniem na stronie. Otrzymamy również informację o tym, które zewnętrzne strony i linki prowadzą do naszej strony oraz jak wersja mobilna strony wypada przy wyszukiwaniu na urządzeniach mobilnych.

## 9.8.4. Mapy strony

Istotnym elementem witryny jest umieszczenie na niej mapy strony (ang. *Sitemaps*). Mapa strony to specjalny dokument XML zawierający strukturę utworzonej witryny internetowej. Zawiera ona w hierarchicznej formie listę podstron znajdujących się w witrynie.

Mapy strony pozwalają przekazać użytkownikowi informacje o stronach, które są trudne do odnalezienia dla robotów wyszukujących. Są szczególnie przydatne, gdy witryna:

- jest nowa;
- zawiera treści generowane dynamicznie;
- zawiera grafikę;
- zawiera podstrony, które nie są dobrze połączone linkami.

Mapę strony można utworzyć samodzielnie, korzystając z prostego edytora tekstu.

## Przykład 9.1

```
<?xml version="1.0" encoding="UTF-8" ?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">

<url>
    <loc>http://www.moje_podroze.pl/index.html</loc>
    <changefreq>monthly</changefreq>
    <priority>0.7</priority>
</url>

<url>
    <loc>http://www.moje_podroze.pl/morze.html</loc>
    <changefreq>weekly</changefreq>
    <priority>0.6</priority>
</url>

<url>
    <loc>http://www.moje_podroze.pl/gory.html</loc>
    <changefreq>daily</changefreq>
    <priority>0.5</priority>
</url>

</urlset>
```

Znacznik `<loc>` określa plik, który ma zostać uwzględniony w mapie strony.

Znacznik `<priority>` określa priorytet danego adresu w stosunku do innych adresów witryny. Priorytetem jest wartość od 0.0 do 1.0. Najwyższy priorytet powinna mieć strona główna serwisu, a wartość domyślna to 0.5.

Znacznik `<changefreq>` określa częstotliwość zmian na stronie i może przyjmować wartości:

- `always` — zawsze (przy każdym otwarciu strony),
- `hourly` — raz na godzinę,
- `daily` — raz dziennie,
- `weekly` — co tydzień,
- `monthly` — raz w miesiącu,
- `yearly` — raz do roku,
- `never` — nigdy.

Podana częstotliwość jest tylko wskazówką dla robota. Nie można wymusić na nim określonej częstotliwości indeksowania strony.

W internecie można znaleźć gotowe narzędzia do tworzenia mapy strony (np. <http://www.xml-sitemaps.com/>). Również w serwisie Google można znaleźć kreator generowania mapy strony, który przeprowadzi użytkownika przez proces jej tworzenia. Efektem działania takich narzędzi jest powstanie pliku *sitemap.xml*, który zawiera mapę strony i powinien zostać umieszczony w katalogu głównym tworzonej witryny.

Jeżeli tworzymy stronę za pomocą CMS WordPress, to możemy opracować mapę strony, korzystając z gotowych wtyczek, np. *WordPress SEO by Yoast* lub *Better WordPress Google XML Sitemaps*.

*WordPress SEO by Yoast*, poza tym, że generuje mapy strony, znaczaco poprawia jej pozycjonowanie. Pozwala dostosować tytuły i metatagi oraz zmodyfikować wygląd odnośników.

*Better WordPress Google XML Sitemaps* to modyfikacja wtyczki *Google XML Sitemap*, która generuje wyłącznie mapę strony internetowej.

## 9.9. Pytania i zadania

### 9.9.1. Pytania

1. Na czym polega sprawdzenie poprawności strony internetowej?
2. Do czego służą validatory strony internetowej?
3. Jakie narzędzia można stosować do walidacji strony internetowej?
4. Dlaczego utworzoną stronę internetową należy testować w różnych przeglądarkach?
5. Jakie znasz metody publikowania strony internetowej?
6. Od czego zależy szybkość wczytywania strony internetowej?
7. Na czym polegają testy funkcjonalności strony internetowej?
8. Na czym polega optymalizacja strony internetowej?
9. Jakie są zasady działania automatów wyszukiwarek internetowych?
10. Na czym polega pozycjonowanie strony internetowej?
11. W jakim celu tworzy się mapy strony internetowej?

### Pytania egzaminacyjne

#### Zadanie 1.

Która z reguł walidacji strony internetowej jest błędem?

- A. Jeżeli w instrukcji używa się kilku atrybutów, ich kolejność powinna być zgodna z alfabetem, np. .

- B.** Wyłączanie znaczników musi następować w odwrotnej kolejności, niż były one włączane, np. <p> ... <big> ... </big></p>.
- C.** Znaczniki, oprócz zamykających się, działają do momentu ich wyłączenia znakiem "/" , np. <p> ... </p>.
- D.** W znacznikach nie są rozróżniane wielkie i małe litery, np. <p> i <P> to ten sam znacznik.

### Zadanie 2.

Aby przenieść witrynę na serwer, można skorzystać z oprogramowania:

- A.** Bugzilla.
- B.** Go!Zilla.
- C.** FileZilla.
- D.** CloneZilla.

### Zadanie 3.

Za pomocą którego protokołu należy wysłać pliki na serwer WWW?

- A.** DHCP.
- B.** POP3.
- C.** DNS.
- D.** FTP.

### Zadanie 4.

Proces walidacji strony internetowej to:

- A.** zespół działań mających na celu zwiększenie oglądalności.
- B.** sprawdzenie jej w celu wyeliminowania błędów.
- C.** publikowanie w sieci.
- D.** promocja strony.

### Zadanie 5.

W procesie walidacji stron internetowych nie bada się:

- A.** działania linków.
- B.** błędów składni kodu.
- C.** zgodności z przeglądarkami.
- D.** źródła pochodzenia narzędzi edytorskich.

### Zadanie 6.

W kodzie źródłowym zapisanym w języku HTML wskaż błąd walidacji dotyczący tego fragmentu:

```
<h6>CSS</h6>  
<p>Kaskadowe arkusze stylów (<b>ang. <i>Cascading Style Sheets</b></i>)<br>to  
język służący...</p>
```

- A.** Nieznany znacznik h6.
- B.** Znacznik br nie został poprawnie zamknięty.
- C.** Znacznik br nie może występować wewnątrz znacznika p.
- D.** Znacznik zamykający /b niezgodny z zasadą zagnieżdzania.

## 9.9.2. Zadania

### Zadanie 1.

Po utworzeniu witryny internetowej według wcześniej przygotowanego projektu sprawdź jej poprawność za pomocą validatora strony internetowej, a następnie umieść ją na wybranym serwerze. Przeprowadź walidację witryny i przetestuj jej działanie w różnych przeglądarkach. Przetestuj szybkość ładowania witryny i zoptymalizuj jej działanie.

### Zadanie 2.

Korzystając z narzędzi dostępnych w internecie, utwórz mapę swojej witryny i dołącz ją do zaprojektowanej strony internetowej.



# Bibliografia

## Literatura

- [1] Jolanta Pokorska, *Kwalifikacja EE.09. Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych. Część 1. Tworzenie stron internetowych. Podręcznik do nauki zawodu technik informatyk*, Helion, Gliwice 2018.
- [2] Brian P. Hogan, *HTML5 i CSS3. Standardy przyszłości*, Helion, Gliwice 2011.
- [3] Matthew MacDonald, *HTML5. Nieoficjalny podręcznik*, Helion, Gliwice 2012.

## Źródła internetowe

- [1] <https://www.w3schools.com/html/default.asp>
- [2] <https://www.w3schools.com/css/default.asp>
- [3] <https://inkscape.org/learn/>
- [4] <https://www.w3.org/TR/WCAG20/>
- [5] <https://www.joomla.pl/>

# Skorowidz

## A

Accessibility Color Wheel, 324  
ActionScript, 333  
Adobe Animate, 334  
Adobe Premiere, 375  
adres bazowy, 25  
akapit, 27  
animacje, 333

- ustawianie parametrów, 338
- wstawianie napisu, 337
- animation, 349
- CSS3, 334
- HTML5, 334
- transition, 344

artykuł, 415

- dodawanie grafiki, 422

atrybut

- autocomplete, 87
- autofocus, 87
- background-attachment, 154
- background-position, 156
- background-repeat, 153
- border, 174
- border-bottom-color, 162
- border-bottom-style, 161
- border-bottom-width, 160
- border-color, 161
- border-left-color, 162
- border-left-style, 161
- border-left-width, 160
- border-right-color, 162
- border-right-style, 161
- border-right-width, 160
- border-spacing, 176
- border-style, 160
- border-top-color, 162
- border-top-style, 161
- border-top-width, 160
- border-width, 159
- caption-side, 174
- clear, 190, 195
- display, 192–195, 199, 202
- empty-cells, 174
- even, 253
- float, 188, 190, 204, 223
- height, 167
- href, 66
- list-style, 173
- list-style-image, 172
- list-style-position, 173
- list-style-type, 171
- margin, 164
- max-height, 168
- max-width, 167
- min-height, 167

min-width, 167  
odd, 253  
opacity, 179  
outline, 166  
overflow-x, 170, 251  
overflow-y, 170  
pattern, 87  
position, 178, 184, 186  
required, 87  
table-layout, 175  
type, 73–77  
vertical-align, 187  
visibility, 197  
white-space, 151  
width, 167

atrybuty

- artykułów, 415
- czcionki, 148
- listy, 173
- obramowania, 163
- obrazu, 59
- tła, 158

Audacity, 364, 369  
Avid Media Composer, 374  
Avid Studio, 374

## B

baner animowany, 335  
białe znaki, 151  
biblioteki LAME, 366  
blog, 394  
blokada grafiki, 154  
bloki danych, 219  
border, *Patrz* Obramowanie

## C

ciało dokumentu, 26  
CMS, Content Management System, 385  
CSS, Cascading Style Sheets, 107

- menu, 198
- model blokowy, 159
- reguły, 214
- składnia, 108, 118
- transformacje 2D, 340
- tworzenie animacji, 339

czcionki, 140, 325, 328

## D

definiowanie

- reguł, 214
- sekwencji, 376

dobór

czcionki, 325  
palety barw, 321  
Drupal, 386  
DTD, Document Type Definition, 12  
dynamiczna strona internetowa, 10  
dziedziczenie, 117  
dźwięk, 361

- edytowanie, 364
- modyfikowanie, 366
- nagrywanie, 369
- obróbka, 364

## E

edytory

- graficzne, 10
- stron WWW, 17

edytowanie dźwięku, 364  
efekt

- Cross Dissolve, 382
- Dip to Black, 381
- Dip to White, 382

efekty przejść wideo, 381  
element, *Patrz* także znacznik

- blokowy, 27, 193
- wbudowany, 193

elementy

- menu, 206, 208
- właściwości, 140

etykieta, 78

## F

FileZilla, 440

- łączenie się z serwerem, 440
- transfer plików, 441

film, 375

- montaż, 377
- planowanie, 375
- tworzenie, 375

format

- AIFF, 362
- AVI, 373
- BMP, 260, 265
- CDR, 259
- DivX, 373
- DXF, 259
- GIF, 58, 260, 265, 335
- JPEG, 260, 265
- JPG, 59
- MIDI, 362
- MP3, 361
- MPEG, 373
- Ogg Vorbis, 362
- PNG, 59, 260, 265

**f**ormat

- PSD, 266
  - QuickTime, 373
  - RAW, 266
  - RealMedia, 362
  - SVG, 259, 335
  - SWF, 259
  - TIFF, 260, 266
  - WAVE, 361
  - Windows Media Video, 373
  - WMA, 362
- f**ormaty plików
- dźwiękowych, 361
  - multimedialnych, 91
  - wideo, 373
- f**ormularz, 70
- grupowanie elementów, 79
  - nieaktywne pola, 84
  - rejestracyjny, 88
- f**ormy animacji, 335
- f**TP, File Transfer Protocol, 437

**G**

- GIF animowany, 334, 336
- G**IMP, 271
- modyfikowanie obrazu, 275
  - okno Przybornika, 271
  - retusz zdjęcia, 283
  - skalowanie obrazu, 277
  - tworzenie animacji, 336
  - właściwości obrazu, 277
  - zaznaczanie, 272
  - zaznaczanie zaawansowane, 279
- głębina koloru, 262
- G**oogle
- Search Console, 447
  - Web Designer, 334
- gradient, 279, 296
- g**raficzny projekt strony internetowej, 315
- grafika, 422, 432
- jako tło, 153
  - maski, 271
  - rastrowa, 260
  - warstwy, 270
  - wektorowa, 257
- zasady tworzenia, 269
- g**rupowanie
- elementów formularza, 79
  - selektorów, 140

**H**

- HDTV, High Definition TV, 372
- hiperłącza, 66
- histogram obrazu cyfrowego, 262
- HSV, Hue Saturation Value, 269
- HTML, HyperText Markup Language, 9, 11
- tworzenie animacji, 339
- HTML5, 15, 21

**I**

- identyfikatory, 216
- ikony, 299
- import arkusza stylów, 114
- indeks wyszukiwarki, 446
- informacje
- o obrazie, 277
  - o utworze, 369
- Inkscape, 303
- iTunes, 363

**J**

- jednostki, 119
- j**ęzyk
- ActionScript, 333
  - CSS, 107
  - HTML, 11
  - HTML5, 15, 21
  - XHTML, 13
  - XML, 12
- Joomla!, 386, 406, 410, 428

**K**

- kadrowanie, 288, 318
- kanał alfa, 268
- kaskadowe arkusze stylów, CSS, 107
- kaskadowość arkuszy stylów, 115
- k**ategorie, 412
- dla całego serwisu, 443
  - dla podstron, 443
  - dla SEO, 443
  - dla struktury serwisu, 443
- klasy, 217
- klawisze skrótów, 85
- k**lient FTP
- FileZilla, 440
  - Total Commander, 438
- klonowanie, 283
- kodowanie polskich znaków, 16, 18
- kolor, 120, 152, 220
- obramowania, 161
  - tekstu, 325
  - tf, 152, 325
- kolumny na stronie, 190
- komentarze, 30, 397
- kompresja dźwięku, 362
- bezstratna, 363
  - stratna, 363
- konfiguracja
- artykułów, 415
  - witryny, 421
- korekcja
- zdjęć, 279, 288
  - jasności i kontrastu, 278
- krawędzie tabeli, 52
- krój czcionki, 327
- kształty, 305
- Kuler, 322
- kursywa, 29

**L**

- listy, 44, 171
- definicji, 47
  - kontaktów, 427
  - menu, 198
  - nieuporządkowane, 45
  - numerowane, 44, 171
  - punktowane, 171
  - styl, 171
  - uporządkowane, 44
  - wypunktowane, 45
- wyswietlanie elementów w linii, 202
- zagnieżdżone, 46, 82
- lupa, 276, 287, 367

**L**

- łatka, 285
- łączenie
- komórek, 54
  - plików muzycznych, 371

**M**

- mapy
- odsyłaczy, 68
  - strony, 447
- marginesy
- wewnętrzne, 165
  - zewnętrzne, 164
- maski, 271
- menedżer plików
- kategorii, 413
  - Total Commander, 438
- menu, 198, 208
- animowane, 354, 335
  - lista stron, 400
  - moduł, 425
- modyfikowanie, 418
- modyfikowanie modułu, 424
- pionowe, 198
- poziome, 202, 208
- struktura, 400
- typy pozycji, 418
- tworzenie, 417
- wybór położenia, 401
- zagnieżdżone, 210, 213
- metoda
- matrix(), 342
  - rotate(), 341
  - scale(), 341
  - skew(), 341
  - translate(), 341
- model blokowy CSS, 159
- modele przestrzeni barw, 266
- CMYK, 268
  - HSV, 269
  - RGB, 267
- moduły systemu Joomla!, 426
- modyfikowanie
- dźwięku, 366
  - menu, 418
- modułu menu, 424

- obrazu, 275  
zdjęć, 288  
montaż filmu, 377  
motywów, 395  
multimedia, 91
- N**  
nagłówek, 23, 28, 395  
nagrywanie dźwięku, 369  
nakładanie elementów, 184  
nawigacja, 300  
Notepad++, 17  
nowa linia, 28
- O**  
obcinanie elementu, 186  
obramowanie, 159, 174  
  atrybuty, 163  
  kolor, 161  
  styl, 160  
  szerokość, 159  
obraz, 58  
  gęstość koloru, 262  
  histogram, 262  
  rozmiar, 261  
obrazki, 246  
obrys, 166  
odstęp między  
  komórkami, 176  
  literami, 150  
  wyrazami, 150  
  wierszami, 327  
odsyłacz, 58, 66  
  aktywny, 138  
  graficzny, 67  
  odwiedzony, 136  
  podstawowy, 136  
odsyłacze do plików, 91  
odtwarzanie  
  audio, 363  
  wideo, 373  
opcje  
  kodowania znaków, 18  
  publikacji artykułu, 416  
optymalizacja  
  plików graficznych, 310  
  strony internetowej, 444
- P**  
padding, 165  
PAL, Phase Alternating Line, 372  
paleta  
  Analogous, 324  
  kolorów, 121, 321  
  Monochromatic, 323  
parametry  
  animacji, 338  
  wyświetlania, 420  
Photoshop, 271, 286  
phpMyAdmin
- okno programu, 389  
pierwsza  
  linia, 139  
  litera, 140  
Pingdom, 441  
planowanie filmu, 375  
pliki  
  animowane, 335  
  audio, 93, 361  
  wideo, 92, 372  
  z animacją, 337  
  zapisywanie, 19  
pogrubienie, 29  
pole  
  BUTTON, 78  
  DATALIST, 86  
  SELECT, 80  
  TEXTAREA, 83  
polskie litery, 16  
portal internetowy, 10  
pozioma linia, 30  
pozycja tła, 156  
pozycjonowanie  
  elementów, 178  
  strony, 445  
programy  
  do edytowania wideo, 374  
  do walidacji stron, 434  
projekt  
  graficzny strony, 315, 317  
  szablonu, 214  
protokół  
  FTP, 437  
  SSH, 438  
przeglądarka internetowa, 432  
przepełnienie, 169  
przepłot, 337  
przejrzystość, 179  
przycinanie sekwencji wideo, 378  
przyciski  
  animowane, 335  
  nawigacyjnych, 300  
pseudoelement, 139, 140  
pseudoklasa, 135  
publikowanie strony, 431, 437  
punktör, 172
- Q**  
QuickTime, 363
- R**  
ręczka, 287  
RealPlayer, 364  
reguła, 214  
  @keyframes, 349, 351  
  @media, 236, 242  
reguły kontekstowe, 215  
relatywny rozmiar czcionki, 326  
responsywny układ strony, 236, 316  
restart widoku, 288  
retusz zdjęcia, 283
- rozciąganie stylu, 109  
rozdzielcość obrazu, 261, 372  
rozmiar  
  czcionki, 325  
  elementu, 166  
  obrazu, 261  
rozмещение, 175  
róźdżka, 274  
RWD, Responsive Web Design, 316
- S**  
scenopis, 379  
schemat kategorii, 411  
selektory, 122 – 124, 126, 131  
  grupowanie, 140  
  specjalne, 131  
serwer internetowy, 10  
skalowanie obrazu, 277  
skaner, 264  
składnia języka CSS, 108, 118  
skrypty, 100  
slider, 321  
Sony Vegas Movie Studio, 374  
SSH, Secure Shell, 438  
standard ISO 8859-2, 16  
statyczna strona internetowa, 10  
stempel, 289  
strona internetowa, 10  
struktura strony internetowej, 321  
styl  
  czcionki, 145  
  listy, 171  
  lokalny, 109  
  obramowania, 160  
  przycisku, 302  
system zarządzania treścią, CMS, 385, 386  
szablon, 428  
  GRID SYSTEM, 316  
   płynny strony, 221, 235  
  z dwiema kolumnami, 231  
  z trzema kolumnami, 221, 226  
szerokość  
  elementu, 167  
  maksymalna, 167  
  minimalna, 167  
  obramowania, 159  
szybka maska, 282
- T**  
tabela, 51, 174, 251  
  responsywna, 252  
szerokość, 176  
wysokość, 176  
testowanie  
  funkcjonalności strony, 442  
  strony w przeglądarkach, 431, 436  
  szybkości wczytywania strony, 441  
tło strony, 152  
  atrybuty, 158

tło strony  
 grafika, 153  
 kolor, 152  
 powtarzanie grafiki, 153  
 pozycja, 156  
 usuwanie, 158  
 zdjęcie, 404  
 Total Commander, 438  
 okno klienta FTP, 439  
 transformacje  
 2D, 340  
 CSS, 340  
 tekstu, 150  
 Tumult Hype, 334  
 tworzenie  
 animacji, 333  
 GIF, 336  
 w programie GIMP, 336  
 za pomocą CSS i HTML, 339  
 animowanego menu, 354  
 artykułów, 414  
 bloga, 394  
 filmu, 375  
 grafiki, 269  
 ikon, 299  
 kategorii, 412  
 kształtów, 305  
 menu, 417  
 przycisków nawigacyjnych, 300  
 typy  
 pozycji menu, 418  
 pól, 72  
 tytuł dokumentu, 23

**U**

ukrywanie elementów, 197  
 ułatwienia dostępu, 328  
 umieszcianie  
 strony na serwerze, 438  
 tekstu, 295  
 Unicode, 16  
 urządzenia mobilne, 433  
 ustawienia strony głównej, 401  
 ustawienie w poziomie, 188  
 usuwanie  
 szumów, 370  
 tła, 158  
 UTF-8, 16  
 użyteczność strony, 315

**W**

W3C CSS Validation Service, 435  
 W3C HTML Validation Service, 435  
 W3C Link Checker, 435  
 validator.org, 433  
 CSS, 434  
 HTML, 434  
 warstwy, 184, 270, 291  
 wartości, 119

wartość inline-block, 195  
 Wave Accessibility Tool, 434  
 Web Page Analyzer, 442  
 wewnętrzny arkusz stylów, 111  
 wideo, 372  
 Windows Media Player, 363  
 witryna internetowa, 10, 409  
 WordPress, 386  
 wortal internetowy, 10  
 wstawianie  
 obrazu, 59  
 stylów, 108  
 tekstu, 285  
 wtyczka, 96  
 Elementor, 402  
 wycinanie pojedynczych kolumn, 298  
 wyciszenie, 367  
 wymiary strony, 220  
 wyrównanie w pionie, 187  
 wysokość, 167  
 maksymalna, 168  
 minimalna, 167  
 tabeli, 176  
 wyszukiwarki internetowe  
 działanie automatów, 445  
 wyświetlanie elementów, 192  
 listy, 202

**X**

XHTML, Extensible HyperText Markup Language, 13  
 XML, Extensible Markup Language, 12

**Z**

zakładki, 67  
 zapisywanie  
 obrazów, 261, 310  
 plików, 19  
 zapytania o media, 236  
 zarządzanie postami, 395  
 zastosowanie grafiki wektorowej, 258  
 zaznaczanie, 272  
 zaawansowane, 279  
 prostokątne, 273  
 zdarzenia, 99  
 wewnętrzne, 99  
 zdjęcia fotograficzne, 264  
 jako tło, 404  
 zewnętrzny arkusz stylów, 113  
 zmiana tła zdjęcia, 292  
 znacznik, 12, 21  
 \*, 219  
 <a>, 62  
 <area>, 68  
 <article>, 38  
 <aside>, 39  
 <audio>, 92, 93  
 <b>, 29  
 <base>, 25  
 <body>, 22, 26  
 <br>, 28  
 <button>, 78  
 <canvas>, 94  
 <caption>, 56  
 <div>, 31, 219, 345  
 <dl>, 47  
 <dt>, 47  
 <embed>, 96  
 <fieldset>, 79  
 <figcaption>, 62  
 <figure>, 62  
 <footer>, 39, 219  
 <form>, 70  
 <head>, 22  
 <header>, 36, 219  
 <hr>, 30  
 <html>, 21  
 <i>, 29  
 <img>, 59, 248  
 <input>, 84, 86  
 <label>, 78  
 <legend>, 79  
 <li>, 44, 45, 171  
 <link>, 25  
 <map>, 68  
 <meta>, 23, 236  
 <nav>, 37, 219  
 <object>, 96, 99  
 <ol>, 44, 171  
 <optgroup>, 82  
 <option>, 81, 82, 86  
 <p>, 128  
 <picture>, 247  
 <section>, 38, 219  
 <select>, 81, 82  
 <source>, 248  
 <span>, 109, 125, 214  
 <style>, 111  
 <table>, 51, 56  
 <td>, 51  
 <textarea>, 83  
 <th>, 51  
 <title>, 23  
 <tr>, 51  
 <ul>, 45, 171  
 <video>, 92

znaczniki  
 grupujące, 62  
 semantyczne, 40  
 semantyczne podziału strony, 36  
 wydzielonych bloków, 110  
 znak niedrukowany, 17  
 znaki specjalne, 30

# PROGRAM PARTNERSKI

— GRUPY HELION —



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!  
<http://program-partnerski.helion.pl>



## KOMPLEKSOWO SZKOLIMY NOWOCZESNY BIZNES



IT



BIZNES



PROJEKTY



PROCESY

NASZE SZKOLENIA SĄ PROWADZONE  
ZGODNIE Z METODĄ

## BLENDDED LEARNING

modelem kształcenia, który łączy tradycyjne szkolenie z dostępem do nowoczesnych narzędzi - videokursów, e-booków i audiobooków

T: 609 850 372 E: SZKOLENIA@HELION.PL

[WWW.HELIONSZKOLENIA.PL](http://WWW.HELIONSZKOLENIA.PL)