7 Składnia języka HTML

ZAGADNIENIA

- Co to jest HTML?
- Co to są znaczniki?
- Co to jest komentarz?
- Jak poprawnie wprowadzać znaczniki?
- Jak wprowadzić komentarz?

HTML jest hipertekstowym językiem znaczników służącym do tworzenia stron internetowych. Jego początki sięgają 1991 roku, kiedy to w internecie pojawiła się pierwsza specyfikacja tego języka pod nazwą HTML Tags. W roku 2008 W3C (The World Wide Web Consortium) opublikowało najnowszą wersję tego języka – HTML 5.

Na bazie języka HTML, w postaci XML, skonstruowano nowy język XHTML. Po dziś dzień trwają dyskusje, który z języków jest bardziej odpowiedni do tworzenia stron internetowych. Język XHTML, dzięki możliwości łączenia z innymi językami zgodnymi z XML, daje dużo więcej możliwości. Niestety, nie wszystkie wersje przeglądarek obsługują takie rozszerzenia. Powoduje to, że wielu programistów zdecydowanie woli standardowy język HTML. Przyczynia się do tego również fakt, że XHTML bezwzględnie wymaga poprawnej składni.

Podstawowym elementem składni języka HTML jest znacznik (ang. *tag*). Jest to specjalny tekst otoczony nawiasami ostrymi, np. **** (znacznik odpowiadający za formatowanie czcionki). Znaczniki mają bezpośredni wpływ na strukturę oraz wygląd witryny. Formatując dany fragment strony, należy umieścić tekst (tytuł strony itp.) pomiędzy znacznikiem otwierającym (np. ****) a znacznikiem zamykającym, który dodatkowo zawiera znak „/", czyli ukośnik (np. ****). Znaczniki nie są widoczne na stronie i odpowiadają jedynie za wprowadzane na niej zmiany.

Działanie znacznika można rozszerzyć, przypisując mu odpowiedni atrybut. Każdy atrybut posiada zakres wartości (wprowadzany w cudzysłowie – " ") odpowiadający za zmiany dokonywane przez znacznik. Na przykład atrybut **size** wprowadzony w znaczniku otwierającym **font** przyjmuje wartości od 1 do 7 i wpływa na zmianę rozmiaru czcionki - **** ... ****.

Budowa znaczników oraz specyficzna budowa języka HTML spowodowały wprowadzenie dodatkowych znaków specjalnych. Łatwo to zrozumieć, gdy chcemy na stronie wyświetlić znak mniejszości < lub znak większości >, które w języku HTML są częścią znacznika. Znaki specjalne wprowadzamy, zaczynając od znaku ampersand "&", a kończymy znakiem średnika ";". Istnieją trzy sposoby zapisu takich znaków. Pierwszy to podanie nazwy znaku (< - **<**). Drugi związany jest z podaniem kodu dziesiętnego znaku poprzedzonego znakiem "#" (< - **<**), a trzeci z podaniem kodu szesnastkowego znaku poprzedzonego "#x" (< - **<**). Kilka najczęściej stosowanych znaków specjalnych prezentuje tabela 1.1.

Tabela 1.1. Znaki specjalne

Przykład	Opis
<	Znak mniejs
>	Znak większ
"	Cudzysłów
&	Ampersand
^	Daszek
~	Tylda
	Twarda spac
†	Krzyżyk
‰	Promil
€	Znak euro
©	Copyright
®	Znak handl
→	Strzałka

Często zdarza się, że
ne tylko dla jego twórcy
Budowa komentarza ró
<!-- tekst komenta
Komentarza można r
pamiętać, że komentarz
znaczników.

SPRAWDŹ S

1. Co to jest hiperteksto
2. Do czego stosujemy
3. Co to jest atrybut zna
4. Do czego służą znaki

Tabela 1.1. Znaki specjalne

Przykład	Opis	Nazwa	Kod dziesiętny	Kod szesnastkowy
<	Znak mniejszości	&lt;	&#60;	&#x3C;
>	Znak większości	&gt;	&#62;	&#x3E;
"	Cudzysłów	&quot;	&#34;	&#x22;
&	Ampersand	&amp;	&#38;	&#x26;
^	Daszek	&circ;	&#710;	&#x2C6;
~	Tylda	&tilde;	&#732;	&#x2DC;
	Twarda spacja			
†	Krzyżyk	&dagger;	&#8224;	&#x2020;
‰	Promil	&permil;	&#8240;	&#x2030;
€	Znak euro	&euro;	&#8364;	&#x20AC;
©	Copyright	&copy;	&#169;	&#xA9;
®	Znak handlowy	&reg;	&#174;	&#xAE;
→	Strzałka	&rarr;	&#8594;	&#x2192;

Często zdarza się, że chcemy umieścić w kodzie strony informacje, które będą widoczne tylko dla jego twórcy (np. opis fragmentów strony). Stosujemy do tego komentarze. Budowa komentarza również wykorzystuje nawiasy ostre:

```
<!-- tekst komentarza -->
```

Komentarza można również użyć do ukrycia jakiegoś fragmentu strony. Należy jednak pamiętać, że komentarze nie mogą być zagnieżdżane ani umieszczane wewnątrz innych znaczników.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Co to jest hipertekstowy język znaczników?
2. Do czego stosujemy znaczniki?
3. Co to jest atrybut znacznika?
4. Do czego służą znaki specjalne?

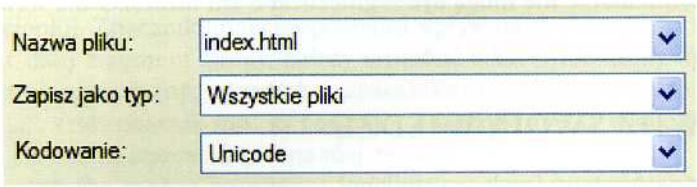
2 Odpowiednie narzędzia

ZAGADNIENIA

- **Jakie edytory stron internetowych są najpopularniejsze?**
- **Na co warto zwrócić uwagę, dobierając edytor?**
- **Jak wykorzystać Notatnik (edytor tekstu) do tworzenia strony WWW?**

Przystępując do tworzenia strony internetowej, warto zastanowić się nad doborem odpowiedniego narzędzia. Ponieważ dokument HTML jest plikiem tekstowym, może być tworzony w dowolnym edytorze tekstu.

Podstawowym edytorem tekstu dostępnym w systemie Windows jest Notatnik. Za jego pomocą można stworzyć różne strony WWW. Pierwszym krokiem jest wprowadzenie w edytorze dowolnej treści opatrzonej odpowiednimi znacznikami. Następnie w menu **Plik** należy wybrać opcję **Zapisz jako...**. Przy tworzeniu strony, która będzie rozpoznawana przez serwer internetowy jako pierwsza, należy pamiętać o wprowadzeniu odpowiedniej nazwy – **index.html** (index.htm). W oknie **Zapisz jako...** wprowadzamy nazwę pliku z rozszerzeniem .html (.htm), typ zapisu ustawiamy na **Wszystkie pliki** oraz wybieramy rodzaj kodowania (rys. 2.1). Po zapisaniu w wybranym folderze pojawi się ikona odpowiadająca zainstalowanej na komputerze przeglądarce. Po otwarciu pojawi się okno przeglądarki wyświetlające stworzoną stronę.



Rys. 2.1. Fragment okna **Zapisz jako...** odpowiadającego za zapisanie strony tworzonej w Notatniku

Tworzenie stron w Notatniku pozwala na bardzo dobre opanowanie składni języka HTML, ponieważ nie udostępnia on żadnych narzędzi wspomagających pracę webmastera. Można tu polegać wyłącznie na swojej wiedzy.

Dostępne na rynku edytory HTML mają za zadanie dbać o poprawność składniową i edytować informacje nagłówkowe (meta). znajduje się w nich opcja kolorowania składni, proponują podgląd dokumentu oraz udostępniają gotowe przykłady i szablony. Pomocnym elementem jest również możliwość bezpośredniego przesyłania danych na serwer FTP.

Bardzo popularnym płatnym edytorem HTML jest Pajączek NxG. Edytor ten jest jednym z najbardziej rozbudowanych narzędzi dostępnych w języku polskim. Posiada ponad 40 różnych schematów kolorowania składni, synchronizację serwisów oraz zaawansowa-

ne narzędzia edycji dla r
witrynami. W zależności
skich edytorów, które zy
edHTML, Zajączek, ezH

Pamiętaj, że stronę m
sposób będziesz tworzyć
widualne projekty.

SPRAWDŹ SW

1. Wymień znane Ci edy
2. Jakie możliwości dają
3. Jak stworzyć stronę in
4. Podaj przykłady serwi
   ce edycję stron intern

ne narzędzia edycji dla różnych języków (PHP, XHTML, XML, JavaScript) i zarządzania witrynami. W zależności od potrzebnych funkcji można wybrać jeden z darmowych polskich edytorów, które zyskują coraz większą popularność. Warte uwagi są: Notepad++, edHTML, Zajączek, ezHTML, MiniPad i wiele innych.

Pamiętaj, że stronę możesz pisać w dowolnym edytorze. Tylko od Ciebie zależy, w jaki sposób będziesz tworzyć przykładowe strony zawarte w tym podręczniku oraz swoje indywidualne projekty.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Wymień znane Ci edytory stron internetowych.
2. Jakie możliwości dają rozbudowane edytory?
3. Jak stworzyć stronę internetową, wykorzystując Notatnik?
4. Podaj przykłady serwisów internetowych udostępniających oprogramowanie ułatwiające edycję stron internetowych.

3 Podstawowa struktura dokumentu

ZAGADNIENIA

- Zadeklarowanie języka HTML
- Definicja nagłówka (element HEAD)
- Definicja treści dokumentu (element BODY)
- Jak poprawnie zbudować podstawową strukturę dokumentu?
- Jak utworzyć tytuł strony?
- Jak wyświetlić tekst na stronie?

Budowę dokumentu hipertekstowego rozpoczyna się od zadeklarowania, w jakim języku tworzona będzie strona internetowa. Popularnym hipertekstowym językiem znaczników jest HTML. Odpowiada on za właściwe przedstawienie informacji zawartych w treści dokumentu oraz wprowadzenie dodatkowych elementów multimedialnych lub plików. Ramy dokumentu hipertekstowego stanowią znaczniki określające język, które umieszczone są w nawiasach ostrych:

```
<html>
<!--zawartość dokumentu hipertekstowego -->
</html>
```

Po zadeklarowaniu języka, w jakim tworzona będzie strona, można przystąpić do wprowadzenia nagłówka dokumentu **<head>**. Nagłówek odpowiedzialny jest za przechowywanie informacji dotyczących całego dokumentu. Można wprowadzić tu takie elementy jak: tytuł strony, kodowanie polskich znaków, słowa kluczowe, podstawowy opis strony, style formatujące dokument czy dane autora. Ogólny zarys nagłówka dokumentu przedstawia się następująco:

```
<head>
<!--nagłówek dokumentu-->
</head>
```

Elementy przedstawione powyżej nie są widoczne dla użytkownika przeglądającego daną stronę w internecie. Informacje te są zrozumiałe jedynie dla przeglądarki. Treści bezpośrednio uwidocznione na stronie umieszczone są w treści dokumentu **<body>**. Jest to miejsce przeznaczone do wprowadzania m.in. tekstów i obrazów. Ogólny zarys treści dokumentu przedstawia się następująco:

```
<body>
<!--treść dokumentu (ciało)-->
</body>
```

Łącząc ze sobą w odpowiednim porządku powyższe elementy, otrzymamy **podstawową strukturę dokumentu** zawierającą deklarację języka HTML, nagłówek oraz treść dokumentu.

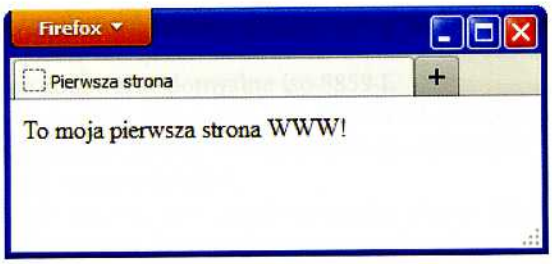
```
<html>
  <head>
  <!--nagłówek dokumentu-->
  </head>
  <body>
  <!--treść dokumentu (ciało)-->
  </body>
</html>
```

Opierając się na poznanej strukturze, zbudujemy naszą pierwszą stronę internetową. W nagłówku dokumentu wprowadzimy dodatkowy znacznik **<title>**, odpowiedzialny za wyświetlenie **tytułu strony** (Pierwsza strona), który wyświetla się w lewym górnym rogu na pasku tytułu przeglądarki. Następnie wprowadzimy tekst w treści dokumentu (To moja pierwsza strona WWW!).

Listing 3.1

```
<html>
  <head>
      <title>
      Pierwsza strona
      </title>
  </head>
  <body>
  To moja pierwsza strona WWW!
  </body>
</html>
```

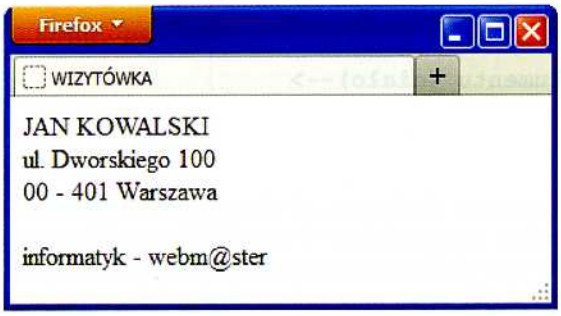
Efekt naszej pracy prezentuje obrazek (rys. 3.1):



Rys. 3.1. Pierwsza strona utworzona na podstawie list. 3.1

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj stronę internetową na podstawnie poniższego rysunku. Pamiętaj o wprowadzeniu znacznika **
**, który odpowiada za przejście do nowej linii. Znacznik **
** nie ma znacznika zamykającego.



Rys. 3.2. Strona WWW prezentująca prostą wizytówkę

SPRAWDŹ SWOJĄ WIEDZĘ

1. Za co odpowiada znacznik **<title>**?
2. Jakie informacje może przechowywać nagłówek dokumentu?
3. Jaka jest podstawowa struktura dokumentu HTML?
4. Dlaczego ważne jest zachowanie odpowiedniej kolejności elementów wchodzących w skład struktury dokumentu?

4

ZAGADNIENIA

- Co to są znaki diakrytycz
- Co to jest kodowanie str
- Jakiego kodowania używ
- Jak poprawnie wyświetla
- Jak określić język strony?
- Jak stworzyć pusty szabl

Znaki diakrytyczne (litery wielkie odpowiedniki), w wą, należy zadbać, aby w celu należy wprowadzić ne sposoby kodowania p

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

Kodowanie polskich z dowanie znaków dla utf-

```
<head>
<meta http-equiv="
charset=utf-8">
</head>
```

Należy pamiętać, że w dokumencie tylko jed ment HTML zastosuje

Obowiązującym stan Normalizacyjny. W przy kach, warto wprowadzi

Kolejnym ważnym e kę, w jakim języku poda francuskim – fr, włoski dza się do odpowiedniej strony jako język polski

```
<head>
<meta http-equiv=
</head>
```

rysunku. Pamiętaj o wprowa-
o nowej linii. Znacznik

ntu?

ści elementów wchodzących

4 Znaki diakrytyczne i definiowanie języka dokumentu

ZAGADNIENIA

- Co to są znaki diakrytyczne?
- Co to jest kodowanie strony?
- Jakiego kodowania używać?
- Jak poprawnie wyświetlać polskie znaki na stronie WWW?
- Jak określić język strony?
- Jak stworzyć pusty szablon polskiej strony WWW?

Znaki diakrytyczne (litery diakrytyzowane) są to takie litery jak: ą ć ę ł ń ó ś ź ż (oraz ich wielkie odpowiedniki), występujące tylko w polskim alfabecie. Tworząc stronę internetową, należy zadbać, aby wszystkie takie znaki były na niej prawidłowo wyświetlane. W tym celu należy wprowadzić w kodzie odpowiednie oznaczenia. Obecnie najbardziej popularne sposoby kodowania polskich znaków to:

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

Kodowanie polskich znaków wprowadzane jest przez znacznik **meta**. Przykładowo, kodowanie znaków dla utf-8 prezentuje się następująco:

```
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
</head>
```

Należy pamiętać, że element **meta** odpowiadający za kodowanie może pojawić się w dokumencie tylko jeden raz. Jeżeli kodowanie znaków nie zostanie określone, dokument HTML zastosuje kodowanie domyślne iso-8859-1.

Obowiązującym standardem kodowania jest iso-8859-2, przyjęte przez Polski Komitet Normalizacyjny. W przypadku, gdy na stronie pojawiają się również teksty w innych językach, warto wprowadzić kodowanie utf-8.

Kolejnym ważnym elementem jest określenie języka strony. Informuje on przeglądarkę, w jakim języku podawane są dane umieszczone na stronie, np. w języku polskim – pl, francuskim – fr, włoskim – it , rosyjskim – ru itp. Jeden z wymienionych skrótów wprowadza się do odpowiedniego znacznika **meta** w sekcji **head**. Poniższy przykład określa język strony jako język polski:

```
<head>
<meta http-equiv="Content-Language" content="pl">
</head>
```

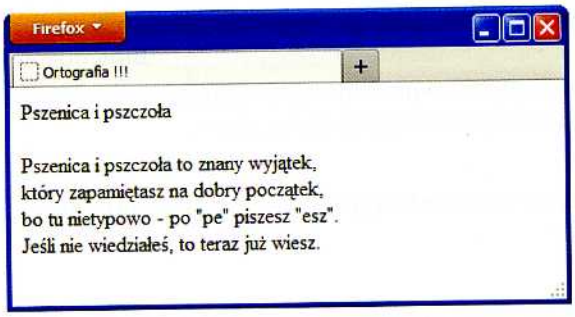
Wykorzystując dotychczasowo poznane elementy, możemy zbudować szablon strony WWW (list. 4.1) i zastosować go do budowy dowolnej strony w języku polskim.

Listing 4.1

```
<html>
  <head>
  <title>
  Strona w języku polskim
  </title>
  <meta http-equiv="Content-Type" content="text/html; charset=
iso-8859-2">
  <meta http-equiv="Content-Language" content="pl">
  </head>
  <body>
  Szablon strony!
  </body>
</html>
```

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj stronę internetową na podstawnie poniższego rysunku. Pamiętaj o zastosowaniu odpowiedniego kodowania dla prawidłowego wyświetlenia polskich znaków.



Rys. 4.1. Strona WWW prezentująca polskie znaki diakrytyczne

SPRAWDŹ SWOJĄ WIEDZĘ

1. Za co odpowiada deklaracja języka strony?
2. Co to są znaki diakrytyczne? W jaki sposób prawidłowo wyświetlać znaki w dokumencie HTML?

5

ZAGADNIENIA

- Co to są metainformacje?
- Jakie znaczenie mają słow
- Co to są roboty?
- Jak umieścić metainform

Elementy **meta** odpowi witryny. Informacje taki strony (**<head>** ... **</h** Argument **name** identyf z argumentem **http-e** Argument **lang** określa

Pierwsze opisane w p oraz określenie języka st

Kolejny element meta wadzić istotne informac szukiwarki internetowe, niezbyt długi i zachęcał

```
<meta name="descri
```

Innym ważnym elem to wyrazy, które najlepie wadzić kilka słów kluczo

```
<meta name="keywo
```

Metainformacje prze praw autorskich, oprogr osoby odpowiedzialnej składnię elementów me

```
<meta name="autho
<meta name="compa
<meta name="publi
<meta name="copyr
<meta name="gener
<meta name="reply
```

Roboty internetowe t brane informacje na ter

5 Elementy meta

ZAGADNIENIA

- Co to są metainformacje?
- Jakie znaczenie mają słowa kluczowe?
- Co to są roboty?
- Jak umieścić metainformacje w witrynie?

Elementy **meta** odpowiadają za przekazanie dodatkowych informacji na temat danej witryny. Informacje takie nazywamy metainformacjami. Umieszczamy je w nagłówku strony (**<head>** ... **</head>**). Elementy meta posiadają charakterystyczne argumenty. Argument **name** identyfikuje rodzaj metainformacji. Jest on stosowany naprzemiennie z argumentem **http-equiv**. Argument **content** zawiera konkretne metainformacje. Argument **lang** określa język strony, zaś argument **dir** kierunek przetwarzania tekstu.

Pierwsze opisane w podręczniku elementy meta odpowiadały za kodowanie znaków oraz określenie języka strony.

Kolejny element meta związany jest z przedstawieniem opisu strony. Należy tu wprowadzić istotne informacje dotyczące witryny. Informacje te są wykorzystywane przez wyszukiwarki internetowe, dlatego należy pamiętać, aby wprowadzony tekst był interesujący, niezbyt długi i zachęcał do odwiedzenia strony. Składnia tego elementu jest następująca:

```
<meta name="description" content="opis mojej strony">
```

Innym ważnym elementem, z którego korzystają wyszukiwarki, są słowa kluczowe. Są to wyrazy, które najlepiej opisują zawartość i tematykę strony internetowej. Można wprowadzić kilka słów kluczowych, oddzielając je od siebie przecinkiem:

```
<meta name="keywords" content="wyraz1, wyraz2, ...">
```

Metainformacje przekazują również dane dotyczące autora witryny, firmy, wydawcy, praw autorskich, oprogramowania zastosowanego do stworzenia strony oraz adres poczty osoby odpowiedzialnej za zarządzanie stroną. Stosując powyższą kolejność, deklarujemy składnię elementów meta:

```
<meta name="author" content="imie i nazwisko autora">
<meta name="company" content="nazwa firmy">
<meta name="publisher" content="nazwa wydawcy">
<meta name="copyright" content="prawa autorskie">
<meta name="generator" content="nazwa oprogramowania">
<meta name="reply-to" content="adres@poczta.pl">
```

Roboty internetowe to programy, które współpracują z wyszukiwarką, przekazując wybrane informacje na temat witryny do bazy danych wyszukiwarki. Zadaniem robotów jest

przeanalizowanie kodu HTML witryny i zaindeksowanie go, czyli przekazanie do bazy danych wyszukiwarki niezbędnych informacji pomocnych w znalezieniu interesującej nas witryny. Wprowadzając odpowiednie elementy **meta**, kierujemy pracą robotów. Jeżeli chcemy zaindeksować całą stronę, wystarczy wprowadzić jeden z następujących elementów:

```
<meta name="robots" content="all">
```

lub

```
<meta name="robots" content="index">
```

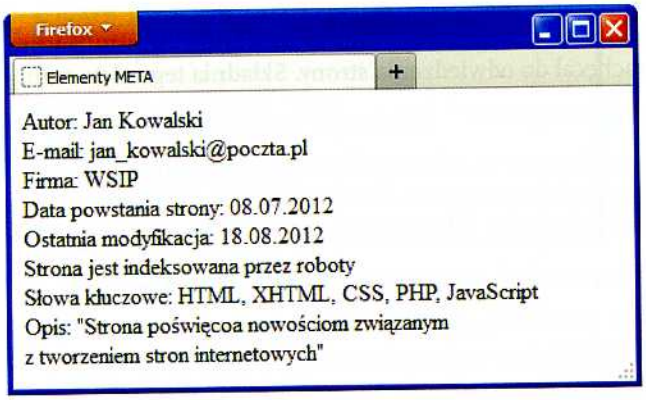
W wypadku, gdy nie chcemy, żeby wyszukiwarki obsługiwały naszą stronę, wystarczy dla argumentu **content** ustawić wartość **none** lub **noindex**. Niestety nie wszystkie roboty interpretują to polecenie, więc może się zdarzyć, że nasza strona będzie mimo to rozpoznawana przez wyszukiwarkę.

Dodatkowymi informacjami są dane związane z datą powstania strony, datą ważności strony oraz datą ostatniej modyfikacji strony.

```
<meta name="creation-date" content="data powstania">
<meta name="expires" content="data ważności">
<meta name="last-modified" content="data modyfikacji">
```

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj stronę internetową na podstawnie rysunku. Wprowadź elementy meta według informacji wyświetlanych na stronie.



Rys. 5.1. Strona WWW wyświetlająca informacje, które mogą być wprowadzane dla elementów **meta**

SPRAWDŹ SWOJĄ WIEDZĘ

1. Co to są metainformacje?
2. Jaką funkcję pełnią słowa kluczowe?
3. Co to są roboty internetowe?
4. Pominięcie jakich metainformacji może spowodować błędne funkcjonowanie strony?

6

ZAGADNIENIA

- Co to jest akapit?
- Co to jest blok?
- Jakie cechy przyjmuje tytu
- Jak tworzyć akapity, bloki

Akapit stanowi fragment
odpowiednie rozmieszcz
treść nie zlewa się w jed
w ten sam sposób co teks
świetlanie jedynie pojed
dzenie kilku akapitów je

Dla przykładu, stwórz
Każda sentencja umiesz

Listing 6.1

```
<p> Nie dla szkoły
<p> Czegokolwiek s
<p> Powtarzanie je
```

Rys. 6.1. Strona WWW stw

Kolejnym elementem
** ** ... ** **. Ele
jak akapity, tytuły czy i
nowej linii.

Tym razem stworzon
blokach **div**.

6 Elementy podstawowe

ZAGADNIENIA

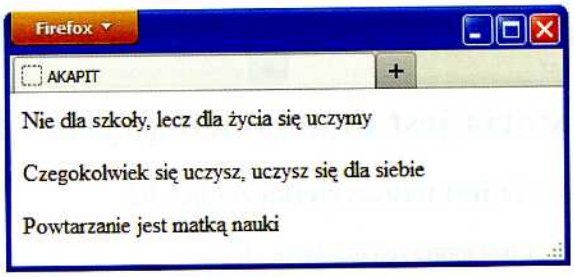
- **Co to jest akapit?**
- **Co to jest blok?**
- **Jakie cechy przyjmuje tytuł?**
- **Jak tworzyć akapity, bloki i tytuły?**

Akapit stanowi fragment tekstu ograniczonego znacznikami ** ** ... ** **. Pozwala on na odpowiednie rozmieszczenie tekstów na tworzonej stronie. Dzięki niemu wprowadzana treść nie zlewa się w jeden ciąg znaków. Tekst znajdujący się w akapicie wyświetlany jest w ten sam sposób co tekst znajdujący się poza akapitem. Chodzi tu w szczególności o wyświetlanie jedynie pojedynczych białych znaków oraz sposób łamania wierszy. Wprowadzenie kilku akapitów jeden pod drugim wygeneruje między nimi wolny wiersz.

Dla przykładu, stwórzmy stronę internetową przedstawiającą trzy sentencje łacińskie. Każda sentencja umieszczona będzie w osobnym akapicie.

Listing 6.1

```
<p> Nie dla szkoły, lecz dla życia się uczymy </p>
<p> Czegokolwiek się uczysz, uczysz się dla siebie </p>
<p> Powtarzanie jest matką nauki </p>
```



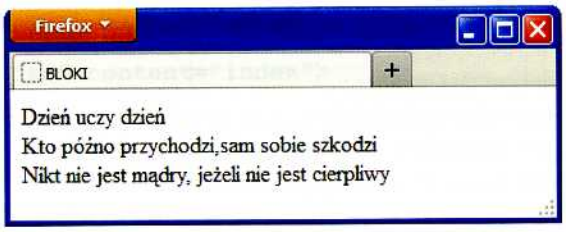
Rys. 6.1. Strona WWW stworzona na podstawie list. 6.1

Kolejnym elementem odpowiedzialnym za wydzielenie pewnej części strony jest blok ** ** ... ** **. Element **div** może zawierać w sobie wiele innych elementów, takich jak akapity, tytuły czy inne bloki. Kolejne bloki są oddzielone od siebie jedynie znakiem nowej linii.

Tym razem stworzona strona będzie wyświetlać trzy sentencje zawarte w pojedynczych blokach **div**.

Listing 6.2

```
<div> Dzień uczy dzień. </div>
<div> Kto późno przychodzi, sam sobie szkodzi. </div>
<div> Nikt nie jest mądry, jeżeli nie jest cierpliwy. </div>
```

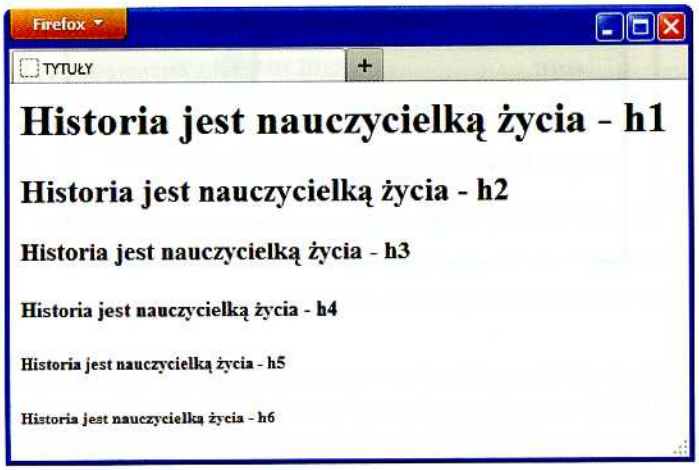


Rys. 6.2. Stronna WWW stworzona na podstawie list. 6.2

Tytuł, zwany również nagłówkiem, odpowiada za wydzielenie pewnej części tekstu (rozdziału, podrozdziału), a co za tym idzie – za definiowanie struktury dokumentu. Nadanie tytułu wybranemu fragmentowi odbywa się za pomocą znaczników `<h1> ... </h1>`, `<h2> ... </h2>`, ... , do `<h6> ... </h6>`. Elementy zawarte w nagłówku h1 wyświetlane są największą czcionką, a w nagłówku h6 – najmniejszą. Stopniowanie wielkości czcionek w zależności od nagłówka przedstawia poniższy przykład:

Listing 6.3

```
<h1> Historia jest nauczycielką życia - h1 </h1>
<h2> Historia jest nauczycielką życia - h2 </h2>
<h3> Historia jest nauczycielką życia - h3 </h3>
<h4> Historia jest nauczycielką życia - h4 </h4>
<h5> Historia jest nauczycielką życia - h5 </h5>
<h6> Historia jest nauczycielką życia - h6 </h6>
```



Rys. 6.3. Stronna WWW stworzona na podstawie list. 6.3

SPRAWDŹ SW

1. Wykonaj stronę interne

NIC DWA RAZY – tytu

Wisława Szymborska –

Nic dwa razy się nie zd
i nie zdarzy. Z tej przyc
zrodziliśmy się bez wp
i pomrzemy bez rutyny

Choćbyśmy uczniami b
najtępszymi w szkole ś
nie będziemy repetowa
żadnej zimy ani lata. –

SPRAWDŹ SW

1. Co to jest akapit?
2. Czym różni się akapit
3. Ile rozmiarów (wielkoś
4. W jakim celu stosowa
czy tytuł?

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj stronę internetową na podstawie schematu:

   NIC DWA RAZY – **tytuł h1**

   Wisława Szymborska – **tytuł h3**

   Nic dwa razy się nie zdarza
   i nie zdarzy. Z tej przyczyny
   zrodziliśmy się bez wprawy
   i pomrzemy bez rutyny. – **cała zwrotka jako akapit**

   Choćbyśmy uczniami byli
   najtępszymi w szkole świata,
   nie będziemy repetować
   żadnej zimy ani lata. – **cała zwrotka jako akapit**

SPRAWDŹ SWOJĄ WIEDZĘ

1. Co to jest akapit?
2. Czym różni się akapit od bloku?
3. Ile rozmiarów (wielkości czcionek) możemy zastosować w nagłówkach (tytułach)?
4. W jakim celu stosowane są na stronach internetowych elementy takie jak akapit, blok czy tytuł?

elenie pewnej części tekstu (roz-
struktury dokumentu. Nadanie
znaczników <h1> ... </h1>,
arte w nagłówku h1 wyświetlane
Stopniowanie wielkości czcionek

7 Tekst

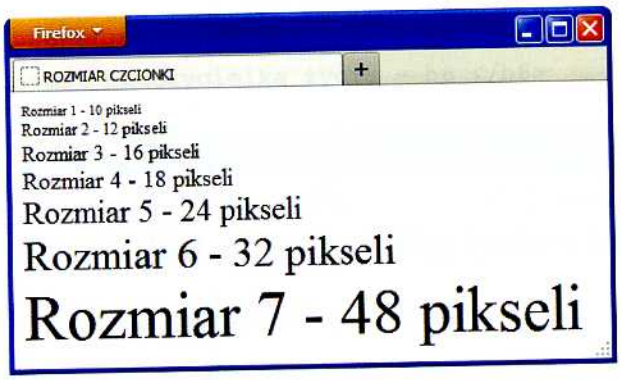
ZAGADNIENIA

- Jakiemu formatowaniu można poddać tekst?
- Jakie cechy przyjmuje tekst preformatowany?
- Jak zmienić kolor, rozmiar i krój czcionki?
- Jak pogrubić, pochylić i podkreślić tekst?
- Jak wyświetlać indeksy (górny, dolny) na stronie?

Za zmianę wyglądu czcionki odpowiada znacznik `<font>` ... `</font>`. W zależności od przyjętego atrybutu może formatować rozmiar, kolor i krój czcionki.

```
<font size="rozmiar"> ... </font>
<font color="kolor"> ... </font>
<font face="krój"> ... </font>
```

Najłatwiej pokazać to w praktyce. Zacznijmy od rozmiaru czcionki. Może on przyjmować wartości od 1 do 7 (1 – najmniejsza, 3 – domyślna, 7 – największa). Poniższy przykład (list. 7.1) prezentuje stronę (rys. 7.1) ukazującą wszystkie rozmiary czcionki.



Rys. 7.1 Stronna WWW stworzona na podstawie list. 7.1

Listing 7.1

```
<font size="1">Rozmiar 1 - 10 pikseli</font><br>
<font size="2">Rozmiar 2 - 12 pikseli</font><br>
<font size="3">Rozmiar 3 - 16 pikseli</font><br>
<font size="4">Rozmiar 4 - 18 pikseli</font><br>
```

```
<font size="5">Roz
<font size="6">Roz
<font size="7">Roz
```

Zmianie możemy po
dostępnych predefiniow
wadzenia dowolnej barw
#A52A2A. W tym przypa
lony 2A (42 w zapisie dz
dziesiętne podanych ba
kombinację trzech kolor
lony, B – *blue* – niebiesk
palety barw. Wartości d
programów graficznych

Tabela 7.1. Podstawowe na

Nazwa argumentu	
Black	C
White	Bi
Silver	Sr
Gray	Sz
Navy	G
Blue	N
Yellow	Ż
Red	C

Praktyczny przykład
stawu przedstawionego

Listing 7.2

```
<font color="oliv
<font color="lime
<font color="purp
<font color="fuch
<font color="maro
<font color="aqua
<font color="teal
<font color="#FF7
```

```
<font size="5">Rozmiar 5 - 24 piksele</font><br>
<font size="6">Rozmiar 6 - 32 piksele</font><br>
<font size="7">Rozmiar 7 - 48 pikseli</font><br>
```

Zmianie możemy poddać również kolor czcionki. HTML standardowo korzysta z 16 dostępnych predefiniowanych kolorów (tab. 7.1). Dodatkowo, istnieje możliwość wprowadzenia dowolnej barwy za pomocą liczby szesnastkowej, np. dla koloru brązowego – #A52A2A. W tym przypadku czerwony ma nasycenie A5 (165 w zapisie dziesiętnym), zielony 2A (42 w zapisie dziesiętnym) oraz niebieski 2A (42 w zapisie dziesiętnym). Wartości dziesiętne podanych barw przekładają się na model przestrzeni barw RGB. Stanowi on kombinację trzech kolorów stanowiących jego nazwę (R – *red* – czerwony, G – *green* – zielony, B – *blue* – niebieski) o różnym stopniu nasycenia. Pozwala to na dobranie szerokiej palety barw. Wartości dziesiętne dla kodu RGB można ustalić, korzystając z dostępnych programów graficznych (np. Paint).

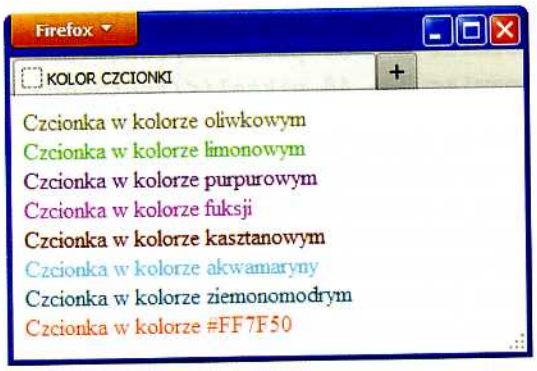
Tabela 7.1. Podstawowe nazwy kolorów wykorzystywane w języku HTML

Nazwa argumentu	Kolor	Nazwa argumentu	Kolor
Black	Czarny	Green	Zielony
White	Biały	Olive	Oliwkowy
Silver	Srebrny	Lime	Limonkowy
Gray	Szary	Purple	Purpurowy
Navy	Granatowy	Fuchsia	Fuksja
Blue	Niebieski	Maroon	Kasztanowy
Yellow	Żółty	Aqua	Akwamaryna
Red	Czerwony	Teal	Turkusowy

Praktyczny przykład (list. 7.2, rys. 7.2) pokazuje kilka wybranych kolorów czcionek z zestawu przedstawionego w tab. 7.1.

Listing 7.2

```
<font color="olive"> Czcionka w kolorze oliwkowym </font><br>
<font color="lime"> Czcionka w kolorze limonkowym </font><br>
<font color="purple"> Czcionka w kolorze purpurowym </font><br>
<font color="fuchsia"> Czcionka w kolorze fuksji </font><br>
<font color="maroon"> Czcionka w kolorze kasztanowym </font><br>
<font color="aqua"> Czcionka w kolorze akwamaryny </font><br>
<font color="teal"> Czcionka w kolorze turkusowym </font><br>
<font color="#FF7F50"> Czcionka w kolorze #FF7F50</font><br>
```



Rys. 7.2. Strona WWW stworzona na podstawie list. 7.2

Atrybut **face**, kolejny atrybut znacznika **font**, pozwala na zmianę kroju czcionki. Dobierając odpowiedni krój, najlepiej wybrać taki, który jest rozpoznawany przez wszystkie przeglądarki. Do zestawu takich krojów należą m.in.: Arial, Times New Roman, Verdana, Courier New, Tahoma, Georgia i Trebuchet MS. Należy pamiętać, że kilkuczłonowe nazwy krojów umieszczamy w pojedynczych apostrofach.

Atrybut **face** może zawierać kilka wartości oddzielonych przecinkami. Jest to bardzo pomocne w przypadku, gdy przeglądarka nie zna pierwszego z wprowadzonych w tym atrybucie krojów. W takim przypadku tekst przyjmuje krój kolejny z rzędu. Jeżeli została wprowadzona tylko jedna wartość nierozpoznawalna przez przeglądarkę, tekst przyjmie krój domyślny.

Standardowy zestaw krojów prezentuje strona na rys. 7.3 stworzona w oparciu o list. 7.3.

Listing 7.3

```
<font face="Arial"> Czcionka - Arial </font><br>
<font face='Times New Roman'>Czcionka - Times New Roman</font><br>
<font face="Verdana"> Czcionka - Verdana </font><br>
<font face='Courier New'> Czcionka - Courier new </font><br>
<font face="Tahoma"> Czcionka - Tahoma </font><br>
<font face="Georgia"> Czcionka - Georgia </font><br>
<font face='Trebuchet MS'> Czcionka - Trebuchet MS </font><br>
```

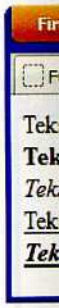
Rys. 7.3. Stronna WWW stworzona na podstawie list 7.3

HTML udostępnia z
 ... odpowia
tekstu, zaś znacznik <u
wać pojedynczo lub w
pamiętać, aby znacznik
list. 7.4 (rys. 7.4).

Listing 7.4

```
Tekst bez formato
<b> Tekst pogrubi
<i> Tekst pochylo
<u> Tekst podkreś
<b><i><u> Tekst p
```

Znacznik <strike>
ny. Kolejny znacznik <b
jednak rozpoznawany p
HTML umożliwia też w
</sup> oraz indeksu do



Rys. 7.4. Strona WWW stw

Ciekawym przykłade
formatowany. Ma on ki
przypomina tekst pisan
ścia do nowej linii oraz
Działanie wymienior
wie list. 7.5.

Listing 7.5

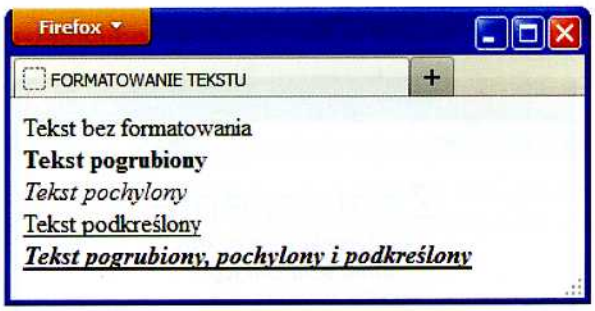
```
<strike> Tekst pr
<blink> Tekst mig
Indeks <sup> górn
Indeks <sub> doln
<pre> Tekst prefo
wszystkie puste l

oraz przejścia do
```

HTML udostępnia zestaw kilku znaczników zmieniających format tekstu. Znacznik **** ... **** odpowiada za pogrubienie tekstu, znacznik **** ... **** za pochylenie tekstu, zaś znacznik **<u>** ... **</u>** podkreśla dany tekst. Znaczników tych można używać pojedynczo lub w grupie. Wprowadzając więcej niż jedno formatowanie, należy pamiętać, aby znaczniki się nie przecinały. Prawidłowy zapis prezentuje ostatnia linia list. 7.4 (rys. 7.4).

Listing 7.4

```
Tekst bez formatowania <br>
<b> Tekst pogrubiony </b><br>
<i> Tekst pochylony </i><br>
<u> Tekst podkreślony </u><br>
<b><i><u> Tekst pogrubiony, pochylony i podkreślony </u></i></b>
```

Znacznik **<strike>** ... **</strike>** powoduje, że wyświetlany tekst jest przekreślony. Kolejny znacznik **<blink>** ... **</blink>** odpowiada za migotanie tekstu, nie jest on jednak rozpoznawany przez wszystkie przeglądarki, dlatego nie zaleca się jego stosowania. HTML umożliwia też wprowadzenie na stronie tekstu w postaci indeksu górnego ^{...} oraz indeksu dolnego _{...}.



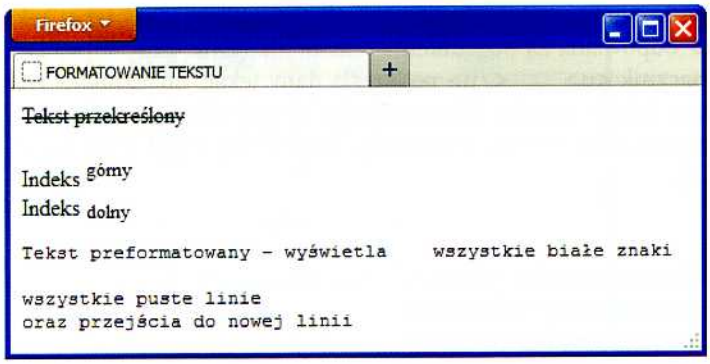
Rys. 7.4. Strona WWW stworzona na podstawie list. 7.4

Ciekawym przykładem formatowania tekstu jest znacznik <pre> ... </pre> – **tekst preformatowany**. Ma on kilka cech charakterystycznych. Przede wszystkim swoim wyglądem przypomina tekst pisany na maszynie. Jako jedyny wyświetla wszystkie białe znaki, przejścia do nowej linii oraz linie puste, które wprowadzono pomiędzy znacznikami.

Działanie wymienionych elementów przedstawia strona (rys. 7.5) stworzona na podstawie list. 7.5.

Listing 7.5

```
<strike> Tekst przekreślony </strike><br>
<blink> Tekst migający </blink><br>
Indeks <sup> górny </sup><br>
Indeks <sub> dolny </sub><br>
<pre> Tekst preformatowany - wyświetla wszystkie białe znaki
wszystkie puste linie

oraz przejścia do nowej linii </pre>
```



Rys. 7.5. Strona WWW stworzona na podstawie list. 7.5

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj stronę internetową na podstawie poniższego rysunku. W celu wyśrodkowania treści zaproszenia zastosuj znacznik `<center>` ... `</center>`. Odpowiada on za wyrównanie wszystkich treści znajdujących się pomiędzy znacznikiem otwierającym i zamykającym.

Rys. 7.6. Strona WWW prezentująca zaproszenie wykonane za pomocą odpowiednich znaczników

SPRAWDŹ SWOJĄ WIEDZĘ

1. Jakiemu formatowaniu można poddać czcionkę?
2. Jak można zmienić kolor czcionki?
3. Jakie cechy ma tekst preformatowany?
4. Porównaj sposoby zmiany koloru tekstu na stronie internetowej.

8

ZAGADNIENIA

- Jakiego rodzaju listy wystę
- W jakich przypadkach nale
- Jak tworzyć listę punktową
- Jak tworzyć listę numerow
- Jak tworzyć listy zagnieżdż

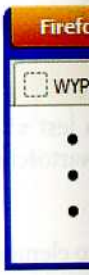
Lista, nazywana również grup informacji w postac rządkowanych (lista punkt mentów konstrukcyjnych,

Listę punktową (nienu Pierwszy znacznik `<ul>` odpowiada znacznik `<li>`

Przykład (list. 8.1, rys. z trzech elementów. Nale kający.

Listing 8.1

```
<ul>
  <li> Punkt pierw
  <li> Punkt drugi
  <li> Punkt trzec
</ul>
```



Rys. 8.1. Strona WWW stwo

8 Listy

ZAGADNIENIA

- Jakiego rodzaju listy występują w języku HTML?
- W jakich przypadkach należy wykorzystywać listy?
- Jak tworzyć listę punktową?
- Jak tworzyć listę numerowaną?
- Jak tworzyć listy zagnieżdżone?

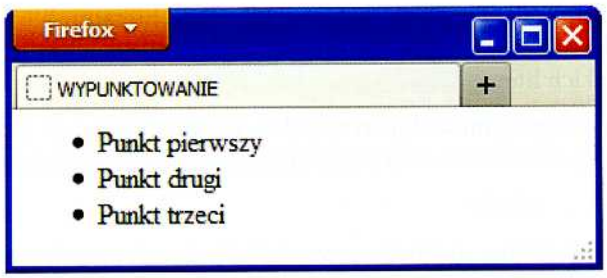
Lista, nazywana również często wykazem, daje możliwość tworzenia uszeregowanych grup informacji w postaci punktów uporządkowanych (lista numerowana) lub nieuporządkowanych (lista punktowa, nienumerowana). Jest to jeden z częściej stosowanych elementów konstrukcyjnych, używany na przykład do budowy menu strony internetowej.

Listę punktową (nienumerowaną) tworzymy, wykorzystując dwa rodzaje znaczników. Pierwszy znacznik `<ul>` ... `</ul>` tworzy zewnętrzną ramę listy. Za każdy element listy odpowiada znacznik `<li>` ... `</li>`. Lista może składać się z dowolnej liczby elementów.

Przykład (list. 8.1, rys. 8.1) przedstawia podstawową listę punktową składającą się z trzech elementów. Należy pamiętać, że każdy element listy musi mieć znacznik zamykający.

Listing 8.1

```
<ul>
   <li> Punkt pierwszy </li>
   <li> Punkt drugi </li>
   <li> Punkt trzeci </li>
</ul>
```



Rys. 8.1. Strona WWW stworzona na podstawie list. 8.1

Domyślnym graficznym punktatorem jest wypełnione koło. Istnieje możliwość zmiany tej grafiki przez wprowadzenie dla znacznika **ul** atrybutu **type**, który przyjmuje jeden z trzech wariantów:

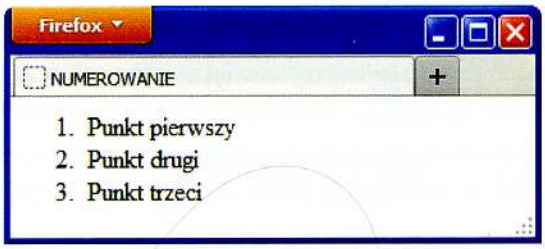
- **disc** (domyślny) – wypełnione koło,
- **circle** – okrąg,
- **square** – wypełniony kwadrat.

Lista numerowana (uporządkowana) wykorzystywana jest przede wszystkim do tworzenia wykazów o ustalonej kolejności. Znacznik **
** ... **
** odpowiada za tworzenie ram listy, a znacznik **
** ... **
** za każdy pojedynczy element listy.

Przykład (list. 8.2, rys. 8.2) przedstawia podstawową listę uporządkowaną składającą się z trzech elementów. Należy pamiętać, że każdy element listy musi mieć znacznik zamykający.

Listing 8.2

```
<ol>
   <li> Punkt pierwszy </li>
   <li> Punkt drugi </li>
   <li> Punkt trzeci </li>
</ol>
```



Rys. 8.2. Strona WWW stworzona na podstawie list. 8.2

Domyślnym numerowaniem dla listy uporządkowanej są cyfry arabskie. Istnieje możliwość zmiany wyświetlanych cyfr przez wprowadzenie dla znacznika **ol** atrybutu **type**, który przyjmuje jeden z pięciu wariantów:

- 1 (domyślny) – numeracja według liczb arabskich,
- I – według liczb rzymskich,
- i – według małych liczb rzymskich zapisanych małymi literami,
- a – według małych liter,
- A – według wielkich liter.

Znacznik **ol** może przyjmować jeszcze jeden atrybut, jakim jest **start**. Atrybut ten umożliwia rozpoczęcie numerowania w zależności od podanej wartości **n**:

```
<ol start="n"> ... </ol>
```

Zmianę numeracji można również wprowadzić dla dowolnego elementu listy. Umożliwia to atrybut **value** przypisany do znacznika **li**. Wartość **n** ustawiona dla atrybutu **value** spowoduje zmianę kolejności numerowania również dla kolejnych elementów listy:

```
<li value="n"> ... </li>
```

Istnieje możliwość zagn
rządkowanych. Przykład (
z trzech poziomów. Należy
ków **ol**, **ul** i **li** (znaczniki

Listing 8.3

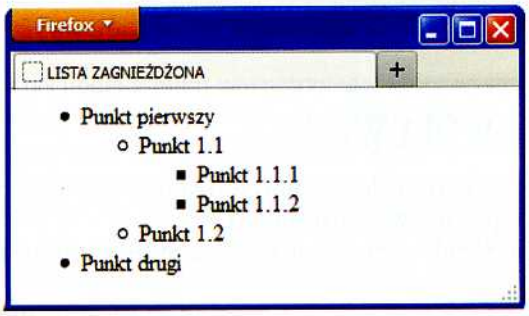
```
<ul>
   <li> Punkt pierw
      <ul>
         <li> Punkt
            <ul>
               <li>
               <li>
            </ul>
         </li>
         <li> Punkt
      </ul>
   </li>
   <li> Punkt drugi
</ul>
```

Rys. 8.3. Strona WWW stwor

Istnieje możliwość zagnieżdżenia wykazów, zarówno uporządkowanych, jak i nieuporządkowanych. Przykład (list. 8.3, rys. 8.3) przedstawia listę zagnieżdżoną składającą się z trzech poziomów. Należy zwrócić uwagę na miejsca zamykania odpowiednich znaczników **ol**, **ul** i **li**(znaczniki te wytłuszczono).

Listing 8.3

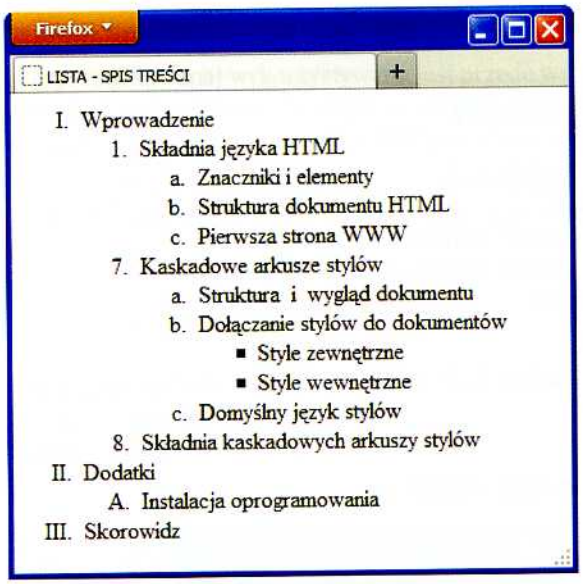
```
<ul>
   <li> Punkt pierwszy
      <ul>
         <li> Punkt 1.1
            <ul>
               <li> Punkt 1.1.1 </li>
               <li> Punkt 1.1.2 </li>
            </ul>
         </li>
         <li> Punkt 1.2 </li>
      </ul>
   </li>
   <li> Punkt drugi </li>
</ul>
```



Rys. 8.3. Strona WWW stworzona na podstawie list. 8.3

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj stronę internetową na podstawie rysunku.



Rys. 8.4. Strona WWW prezentująca rozbudowaną listę zagnieżdżoną

SPRAWDŹ SWOJĄ WIEDZĘ

1. Jakie są rodzaje list? O czym należy pamiętać, tworząc listy zagnieżdżone?
2. Jakie atrybuty może przyjmować znacznik `ol`?
3. Podaj praktyczne przykłady zastosowania listy na stronie internetowej.

9

ZAGADNIENIA

- Jakie formaty obrazów st
- Jak zmienić tło strony W
- Jakie multimedia wprow
- Jak wstawiać obrazek na
- Jak umieszczać przesuw
- Jak osadzać elementy m

Grafika jest jednym z
plików graficznych, któ
Za wyświetlenie pojedyn
znacznika zamykającego

```
<img src="adres pl
```

Wyświetlany obraz po
stroną internetową. Atr
z jego nazwą i rozszerze
go, widocznego po najed

Dodatkowymi atrybu
podawane w pikselach
rozmiaru obrazka, jedna
wprowadzony na stronę
rozmiaru plikowego, a c
atrybutów jest dobrym n
wczytany. Wówczas zajr
elementy na stronie nie

Rys. 9.1. Strona WWW stw

9 Obrazy i multimedia

ZAGADNIENIA

- Jakie formaty obrazów stosuje się na stronach internetowych?
- Jak zmienić tło strony WWW?
- Jakie multimedia wprowadzić na stronę?
- Jak wstawiać obrazek na stronę WWW?
- Jak umieszczać przesuwający się tekst na stronie WWW?
- Jak osadzać elementy multimedialne, m.in. film, muzykę?

Grafika jest jednym z elementów nadających charakter stronie internetowej. Formaty plików graficznych, które powinny być stosowane na stronach, to JPEG, GIF lub PNG. Za wyświetlenie pojedynczego obrazu na stronie odpowiada znacznik **img**, który nie ma znacznika zamykającego:

```
<img src="adres pliku graficznego" alt ="tekst alternatywny">
```

Wyświetlany obraz pochodzi z pliku zewnętrznego, który występuje w zestawieniu ze stroną internetową. Atrybut **src** odpowiada za podanie ścieżki dostępu do pliku wraz z jego nazwą i rozszerzeniem. Atrybut **alt** umożliwia wyświetlenie tekstu alternatywnego, widocznego po najechaniu na obrazek.

Dodatkowymi atrybutami są **width** – szerokość oraz **height** – wysokość obrazka podawane w pikselach lub procentach (list 9.1, rys. 9.1). Pozwalają one na modyfikację rozmiaru obrazka, jednak takie działanie jest niezalecane. Warto pamiętać, żeby obrazek wprowadzony na stronę miał już dostosowane rozmiary. Powoduje to zmniejszenie jego rozmiaru plikowego, a co za tym idzie – szybsze ładowanie się strony. Wprowadzenie obu atrybutów jest dobrym nawykiem. Pomaga to w przypadku, gdy dany obrazek nie zostanie wczytany. Wówczas zajmie on obszar określający jego wysokość i szerokość, a pozostałe elementy na stronie nie ulegną przemieszczeniu.

Rys. 9.1. Strona WWW stworzona na podstawie list. 9.1

Listing 9.1

```
<p align="center">
<img src="html.png" alt="HTML" width="195" height="90">
</p>
```

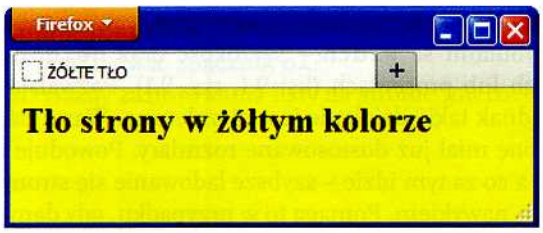
Umieszczając obrazek na stronie, decydujemy również, jak zostanie on ustawiony względem znajdującego się na stronie tekstu. Umożliwia to argument **align** umieszczony wewnątrz znacznika **img**. Może on przyjmować następujące wartości:

- **left** – obrazek ustawiony po lewej stronie względem tekstu,
- **right** – obrazek ustawiony po prawej stronie względem tekstu,
- **top** – tekst ustawiony przy górnej części obrazka,
- **middle** – tekst ustawiony w środkowej części obrazka,
- **bottom** – tekst ustawiony na dole obrazka (domyślnie).

Grafika na stronie to nie tylko obrazki, to również tło strony. Tło strony można modyfikować na dwa sposoby. Oba polegają na wprowadzeniu dodatkowego atrybutu wewnątrz znacznika **body**. Pierwszy sposób pozwala na wprowadzenie tła w określonym kolorze (list. 9.2, rys. 9.2) za pomocą atrybutu **bgcolor**, który przyjmuje jako wartość nazwę wybranego koloru.

Listing 9.2

```
<body bgcolor="yellow">
   <h2>
      Tło strony w żółtym kolorze
   </h2>
</body>
```



Rys. 9.2. Strona WWW stworzona na podstawie list. 9.2

Drugi sposób pozwala na wprowadzenie tła obrazkowego (list. 9.3, rys. 9.3) za pomocą atrybutu **background**, który przyjmuje jako wartość ścieżkę dostępu do pliku graficznego wraz z jego nazwą i rozszerzeniem.

Listing 9.3

```
<body background="tlo.jpg">
   <h2>
      Tło obrazkowe strony
   </h2>
</body>
```

Rys. 9.3. Strona WWW stwor

Przeglądarka Internet E
ce animację tekstu. Znaczn
ale rozpoznają go również
wających na animację:

- **behavior** – tryb przes
  prawej do lewej, **slide**
  nate – od prawej do lew
- **bgcolor** – kolor tła;
- **direction** – kierunek
  – w górę, **down** – w dół)
- **loop** – liczba powtórze
- **scrollamount** – skok
- **scrolldelay** – opóźn

Kolejnym rozszerzenie
bgsound. Polecenie to po
wprowadzone. Znacznik **b**

- **src** – ścieżka dostępu d
- **loop** – liczba powtórze
- **volume** – poziom głośn
- **balance** – balans międ

Rys. 9.4. Strona WWW stwor

Rys. 9.3. Strona WWW stworzona na podstawie list. 9.3

Przeglądarka Internet Explorer (IE) wprowadziła dodatkowe rozszerzenie umożliwiające animację tekstu. Znacznik **<marquee>** ... **</marquee>** nie jest częścią języka HTML, ale rozpoznają go również przeglądarki. Znacznik ma do dyspozycji kilka atrybutów wpływających na animację:

- **behavior** – tryb przesuwania się tekstu na ekranie (**scroll** – tekst przesuwa się od prawej do lewej, **slide** – od prawej do lewej, a następnie odbija się i powraca, **alternate** – od prawej do lewej tylko raz, a później się zatrzymuje i pozostaje nieruchomy);
- **bgcolor** – kolor tła;
- **direction** – kierunek przesuwania się tekstu (**left** – w lewo, **right** – w prawo, **up** – w górę, **down** – w dół);
- **loop** – liczba powtórzeń;
- **scrollamount** – skok o wskazaną liczbę pikseli;
- **scrolldelay** – opóźnienie tekstu, wartość podawana w milisekundach.

Kolejnym rozszerzeniem IE (działa również w przeglądarce Opera) jest polecenie **bgsound**. Polecenie to pozwala na odtwarzanie dźwięku w tle strony, na której zostało wprowadzone. Znacznik **bgsound** wykorzystuje następujące atrybuty:

- **src** – ścieżka dostępu do pliku dźwiękowego wraz z nazwą i rozszerzeniem,
- **loop** – liczba powtórzeń (**infinite** – nieskończoność),
- **volume** – poziom głośności (od -10000 do 0),
- **balance** – balans między głośnikami (od -10000 do +10000).

Rys. 9.4. Strona WWW stworzona na podstawie list. 9.4

Poleceniem związanym z umieszczaniem na stronie różnych plików multimedialnych, niewchodzącym w skład specyfikacji HTML, jest **embed**. Obsługuje ono takie formaty plików jak avi, mpeg, mp3, mid, wav, mov, asf i inne. Jego zaletą jest współpraca z różnymi wtyczkami (specjalnymi niewielkimi programami, które można doinstalować do przeglądarki, zwiększając jej możliwości), co zwiększa pulę odtwarzanych przez nie plików.

Dla przykładu (list. 9.4) umieśćmy plik dźwiękowy blus.wma. Na stronie automatycznie pojawia się odtwarzacz multimedialny, co prezentuje rys. 9.4.

Listing 9.4

```
<embed src="blus.wma">
```

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj stronę internetową (elektroniczną kartkę świąteczną) na podstawnie poniższego rysunku. Obrazek, tło oraz treść życzeń możesz dobrać dowolne, lecz zachowaj układ elementów. Wprowadź jako tło dźwiękowe dowolną kolędę.

Rys. 9.5. Strona WWW prezentująca elektroniczną kartkę świąteczną

SPRAWDŹ SWOJĄ WIEDZĘ

1. Jakie rozszerzenia plików graficznych należy stosować na stronie WWW?
2. Za co odpowiada znacznik **marquee**?
3. Jak wprowadzamy tło dźwiękowe?
4. Jak ustawić tło strony?

10

T

ZAGADNIENIA

- **W jakim celu należy stosow**
- **Jakie są sposoby na zmianę**
- **Jak wstawiać tabelę na stro**
- **Jak łączyć komórki tabeli?**
- **Jak zmieniać wygląd tabeli?**

Tabele stanowią obecnie je
korzystywane do tworzenia

Tabela jest strukturą sk
ramę tabeli stanowi znacz
<tr> ... **</tr>**, a wewnąt

Podstawową strukturę t
dla znacznika **table** wpro
bości obramowania. Jeżeli
wyświetlone, widoczna będ

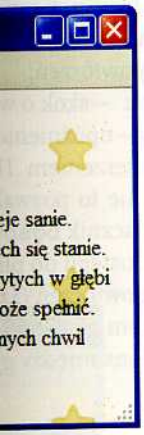
Listing 10.1

```
<table border="1">
   <tr>
      <td>komórka 1<
      <td>komórka 2<
   </tr>
   <tr>
      <td>komórka 3<
      <td>komórka 4<
   </tr>
</table>
```

Znacznik **table** ma kil
Odpowiada on za wprowa
komórkach tabeli. Wartość
pomocny w odpowiednim
czytelność tabeli jest **cells**
mi komórkami (w pikselac
fragment kodu, który zosta
stawiona na rys. 10.2.

nych plików multimedialnych,
bsługuje ono takie formaty pli-
ietą jest współpraca z różnymi
ożna doinstalować do przegląrzanych przez nie plików.
yma. Na stronie automatycznie
.4.

ąteczną) na podstawnie poniżdobrać dowolne, lecz zachowaj
ną kolędę.



eczną

na stronie WWW?

10 Tabele

ZAGADNIENIA

- **W jakim celu należy stosować tabele na stronie WWW?**
- **Jakie są sposoby na zmianę wyglądu tabeli?**
- **Jak wstawiać tabelę na stronę WWW?**
- **Jak łączyć komórki tabeli?**
- **Jak zmieniać wygląd tabeli?**

Tabele stanowią obecnie jeden ze sposobów prezentowania danych. Początkowo były wykorzystywane do tworzenia układów stron, które teraz opierają się na elementach **div**.

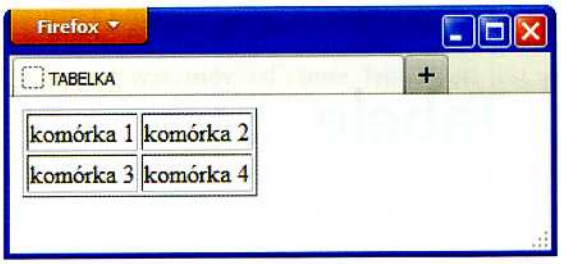
Tabela jest strukturą składającą się z wierszy podzielonych na komórki. Zewnętrzną ramę tabeli stanowi znacznik **<table>** ...**</table>**. Wewnątrz umieszczamy wiersze **<tr>** ... **</tr>**, a wewnątrz nich – poszczególne komórki **<td>** ... **</td>**.

Podstawową strukturę tabeli 2x2 prezentuje przykład (list. 10.1, rys. 10.1). Dodatkowo dla znacznika **table** wprowadzono argument **border**, którego wartość odpowiada grubości obramowania. Jeżeli nie zostanie on wprowadzony, obramowanie tabeli nie zostanie wyświetlone, widoczna będzie jedynie jej zawartość.

Listing 10.1

```
<table border="1">
   <tr>
      <td>komórka 1</td>
      <td>komórka 2</td>
   </tr>
   <tr>
      <td>komórka 3</td>
      <td>komórka 4</td>
   </tr>
</table>
```

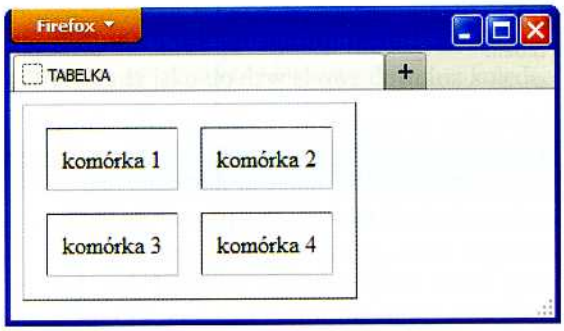
Znacznik **table** ma kilka przydatnych atrybutów. Jednym z nich jest **cellpadding**. Odpowiada on za wprowadzenie marginesów (pionowych i poziomych) we wszystkich komórkach tabeli. Wartość marginesów mierzona jest w pikselach. Atrybut ten jest bardzo pomocny w odpowiednim ustaleniu wyglądu tabeli. Innym atrybutem mającym wpływ na czytelność tabeli jest **cellspacing**, który zmienia rozmiar odstępów pomiędzy wszystkimi komórkami (w pikselach). Działanie obu atrybutów ilustruje list. 10.2, który prezentuje fragment kodu, który został zmieniony w list. 10.1. Efektem tych zmian jest tabela przedstawiona na rys. 10.2.



Rys. 10.1. Strona WWW stworzona na podstawie list. 10.1

Listing 10.2

```
<table border="1" cellpadding="10" cellspacing="15">
```



Rys. 10.2. Strona WWW stworzona na podstawie list. 10.2

Poza zwykłymi komórkami do tabeli możemy również wprowadzić komórki nagłówkowe **<th>** ... **</th>**, zarówno poziome, jak i pionowe. Wstawia się je podobnie jak zwykłe komórki. Jedyna różnica polega na ich wyglądzie. Tekst wyświetlany w komórkach nagłówkowych jest pogrubiony i wyśrodkowany.

Każda tabela powinna mieć tytuł. Wprowadza się go za pomocą znacznika **<caption>** ... **</caption>** umieszczonego zaraz za znacznikiem **table**. Posiada on również atrybut **align** odpowiadający na wyrównanie tytułu (**left**, **right**, **center**, **top**, **bottom**):

```
<table>
<caption align="wyrówanie"> Tytuł tabeli </caption>
<!-- pozostała część tabeli-->
</table>
```

Atrybuty **width** (szerokość) oraz **height** (wysokość) można zastosować do ustawienia wymiarów tabeli. Można je wprowadzić wewnątrz znacznika **table** dla zmiany rozmiaru całej tabeli lub wewnątrz znacznika **td** dla określenia wymiarów konkretnej komórki. Oba atrybuty przyjmują konkretne wartości podane w pikselach lub wartości procentowe (w stosunku do wielkości strony):

```
<table width="200" height="100">
```

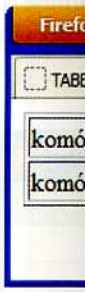
lub

```
<td width="40%" height="20%">
```

Tworzona tabela może r
tak, ponieważ język HTM
scalone w wierszu przez w
mującego wartość odpowi
pamiętać, że przy scalaniu
szach (dwie scalone w pier

Listing 10.3

```
<table border="1">
<tr>
<td colspan="2">komó
</tr>
<tr>
<td>komórka 3</td>
<td>komórka 4</td>
</tr>
</table>
```



Rys. 10.3. Strona WWW stwor

Komórki mogą również
nika **td** atrybutu **rowspar**
mórek (list. 10.4, rys. 10.4).
wać ich liczbę (dwie scalon
w drugim wierszu).

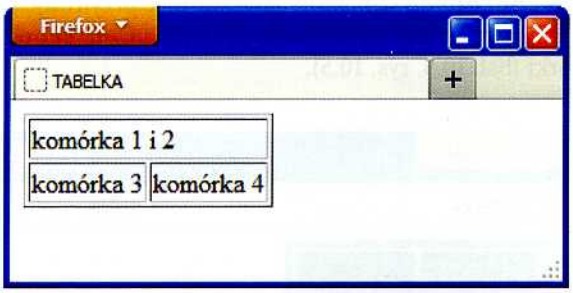
Listing 10.4

```
<table border="1">
<tr>
<td rowspan="2">komó
<td>komórka 2</td>
</tr>
<tr>
<td>komórka 4</td>
</tr>
</table>
```

Tworzona tabela może mieć różną liczbę komórek w wierszach i kolumnach. Dzieje się tak, ponieważ język HTML pozwala na scalanie komórek w tabeli. Komórki mogą zostać scalone w wierszu przez wprowadzenie wewnątrz znacznika **td** atrybutu **colspan**, przyjmującego wartość odpowiadającą liczbie scalonych komórek (list. 10.3, rys. 10.3). Należy pamiętać, że przy scalaniu komórek musimy dostosować ich liczbę w odpowiednich wierszach (dwie scalone w pierwszym wierszu, dwie pojedyncze w drugim wierszu).

Listing 10.3

```
<table border="1">
<tr>
<td colspan="2">komórka 1 i 2</td>
</tr>
<tr>
<td>komórka 3</td>
<td>komórka 4</td>
</tr>
</table>
```

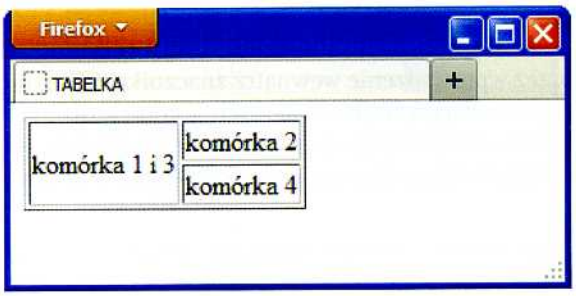


Rys. 10.3. Strona WWW stworzona na podstawie list. 10.3

Komórki mogą również być scalone w kolumnie przez wprowadzenie wewnątrz znacznika **td** atrybutu **rowspan**, przyjmującego wartość odpowiadającą liczbie scalonych komórek (list. 10.4, rys. 10.4). Należy pamiętać, że przy scalaniu komórek musimy dostosować ich liczbę (dwie scalone w pierwszym wierszu i pojedyncza komórka, jedna komórka w drugim wierszu).

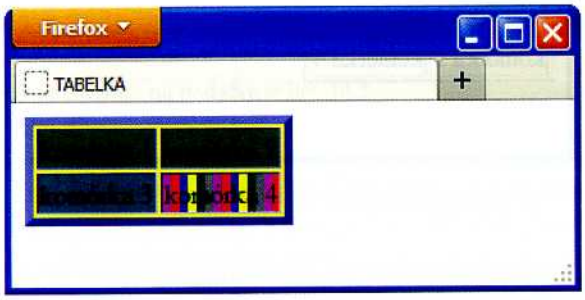
Listing 10.4

```
<table border="1">
<tr>
<td rowspan="2">komórki 1 i 3</td>
<td>komórka 2</td>
</tr>
<tr>
<td>komórka 4</td>
</tr>
</table>
```



Rys. 10.4. Strona WWW stworzona na podstawie list. 10.4

Pozostało jeszcze dodać tabeli nieco koloru. HTML daje nam kilka możliwości. Pierwszą z nich jest wprowadzenie jednolitego koloru tła. Umożliwia to atrybut **bgcolor**, poznany już wcześniej przy ustawianiu tła całej strony. Aby ustawić odpowiedni kolor, wystarczy podać jego nazwę. Atrybut **background**, odwołujący się do odpowiedniego pliku graficznego (ścieżka dostępu wraz z nazwą i rozszerzeniem), pozwala wprowadzić tło graficzne. Ostatnim elementem, którego kolor można zmienić, jest obramowanie. Odpowiednim do tego atrybutem jest **bordercolor**, który jako wartość przyjmuje również nazwę koloru. Wszystkie te atrybuty możemy wprowadzić dla całej tabeli, dla pojedynczego wiersza lub dla jednej komórki (list. 10.5, rys. 10.5).



Rys. 10.5. Strona WWW stworzona na podstawie list. 10.5

Listing 10.5

```
<table border="5" bgcolor="yellow" bordercolor="blue">
<tr bgcolor="green">
<td>komórka 1</td>
<td>komórka 2</td>
</tr>
<tr>
<td bgcolor="teal">komórka 3</td>
<td background="tlo.png">komórka 4</td>
</tr>
</table>
```

SPRAWDŹ SW

1. Wykonaj poniższe tabel
pomysłu, wykorzystując

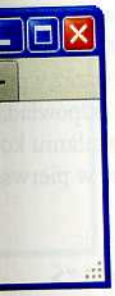
Tabela 1

Tabela2

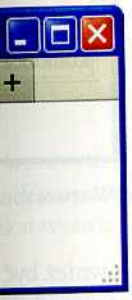
Tabela 3

SPRAWDŹ SW

1. Do czego może być wyko
2. Jak można zmienić wygl
3. W jakim celu stosuje się



am kilka możliwości. Pierwszą
to atrybut **bgcolor**, poznany
ć odpowiedni kolor, wystarczy
odpowiedniego pliku graficz-
vala wprowadzić tło graficzne.
obramowanie. Odpowiednim
rzyjmuje również nazwę kolo-
beli, dla pojedynczego wiersza



```
olor="blue">
```

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj poniższe tabelki na stronie WWW. Każdą z nich sformatuj według własnego pomysłu, wykorzystując poznane atrybuty (zmiana koloru, tła, rozmiaru).

Tabela 1

Tabela2

Tabela 3

SPRAWDŹ SWOJĄ WIEDZĘ

1. Do czego może być wykorzystywana tabela?
2. Jak można zmienić wygląd tabeli?
3. W jakim celu stosuje się komórki nagłówkowe?

11 Odsyłacze

ZAGADNIENIA

- Co to jest odsyłacz (link)?
- Jakie są rodzaje linków?
- Jak zmienić kolor linku?
- Jak tworzyć przejścia między stronami?
- Jak budować menu strony internetowej?
- Jak wstawiać adres e-mail na stronie?

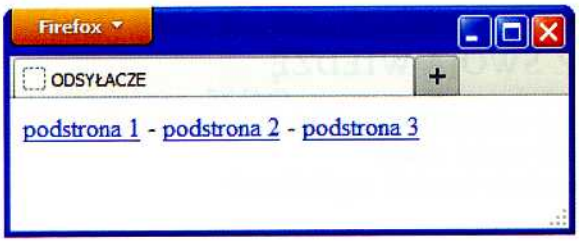
Zadaniem odsyłaczy (linków, hiperłączy) jest utworzenie połączenia umożliwiającego przejście do innej strony internetowej, podstrony lub pliku. Tworząc stronę WWW, zakładamy, że będzie się ona składała z określonej liczby podstron, do których umożliwimy dostęp za pomocą odsyłaczy. Znacznikiem odpowiadającym za wprowadzenie linku jest **a**. Posiada on atrybut **href** określający adres (podstrony, strony pliku), do którego zostaniemy przeniesieni po kliknięciu w link. Tekst znajdujący się wewnątrz znaczników **a** jest nazwą wyświetlaną na stronie internetowej:

```
<a href="adres"> nazwa wyświetlana </a>
```

Można wyróżnić kilka rodzajów odsyłaczy. Jest to związane z ich ścieżką docelową. Przy budowie menu strony korzystamy z odsyłaczy do podstron. Podstrona jest kolejnym dokumentem HTML znajdującym się w tej samej lokalizacji co strona główna lub w jednym z folderów na serwerze, na którym strona zostanie zapisana. Wówczas dla atrybutu **href** wystarczy wprowadzić nazwę podstrony wraz z rozszerzeniem. Gdy podstrona znajduje się w innej lokalizacji niż strona główna, należy również ją uwzględnić.

Listing 11.1

```
<a href="1.html">podstrona 1</a> -
<a href="2.html">podstrona 2</a> -
<a href="nowyfolder/3.html">podstrona 3</a>
```



Rys. 11.1. Strona WWW stworzona na podstawie list. 11.1

HTML pozwala na tworz
wej. Są to tzw. odsyłacze w
kietach) oznaczających mie
wowaniu odsyłacza. Pierws
strony:

```
<a name="nazwa_etyki
```

Do etykiety skieruje nas od
dzoną znakiem #:

```
<a href="#nazwa_etyk
```

Odsyłacze mogą prowa
Wówczas do atrybutu **href**
wraz z **http://**. Jeżeli ch
oknie, wystarczy wprowadz
ścią **_blank**.

Listing 11.2

```
<p> Ciekawe portale:
<ul>
<li><a href="http://
<li><a href="http://
<li><a href="http://
<li><a href="http://
</ul>
```

Rys. 11.2. Strona WWW stwo

Innym rodzajem odnoś
otwarcie domyślnego prog
ka (np. Outlook). Pole z ad
jedynie wpisać treść listu i

```
<a href="mailto: ap
```

Odsyłacze mogą równie
Mogą to być pliki o dowol
przeglądarki po kliknięciu

HTML pozwala na tworzenie odsyłaczy operujących wewnątrz jednej strony internetowej. Są to tzw. **odsyłacze wewnątrzstronicowe**. Ich działanie opiera się na kotwicach (etykietach) oznaczających miejsce na stronie, do którego zostaniemy przeniesieni po aktywowaniu odsyłacza. Pierwszym krokiem jest zdefiniowanie etykiety w wybranym miejscu strony:

```
<a name="nazwa_etykiety"> nazwa wyświetlana </a>
```

Do etykiety skieruje nas odsyłacz, który zamiast adresu przyjmuje nazwę etykiety poprzedzoną znakiem #:

```
<a href="#nazwa_etykiety"> nazwa wyświetlana </a>
```

Odsyłacze mogą prowadzić również bezpośrednio do innych stron internetowych. Wówczas do atrybutu **href** należy przypisać dokładny adres wybranej strony internetowej wraz z **http://**. Jeżeli chcemy, aby po kliknięciu w link strona otwierała się w nowym oknie, wystarczy wprowadzić wewnątrz znacznika **a** dodatkowy atrybut **target** z wartością **_blank**.

Listing 11.2

```
<p> Ciekawe portale: </p>
<ul>
<li><a href="http://www.onet.pl"> onet.pl </a></li>
<li><a href="http://www.wp.pl" target="_blank"> wp.pl </a></li>
<li><a href="http://www.o2.pl"> o2.pl </a></li>
<li><a href="http://www.interia.pl"> interia.pl </a></li>
</ul>
```

Rys. 11.2. Strona WWW stworzona na podstawie list. 11.2

Innym rodzajem odnośnika jest **odsyłacz pocztowy**. Po kliknięciu w taki link nastąpi otwarcie domyślnego programu pocztowego zainstalowanego na komputerze użytkownika (np. Outlook). Pole z adresem nadawcy zostanie automatycznie uzupełnione, wystarczy jedynie wpisać treść listu i wysłać.

```
<a href="mailto: aplikacje@poczta.pl"> napisz do mnie</a>
```

Odsyłacze mogą również wskazywać na inne zasoby, nie tylko dokumenty internetowe. Mogą to być pliki o dowolnym rozszerzeniu: .pdf, .zip, .doc, .txt, .mp3, .exe. Zachowanie przeglądarki po kliknięciu użytkownika w taki link zależy od rozszerzenia pliku oraz po-

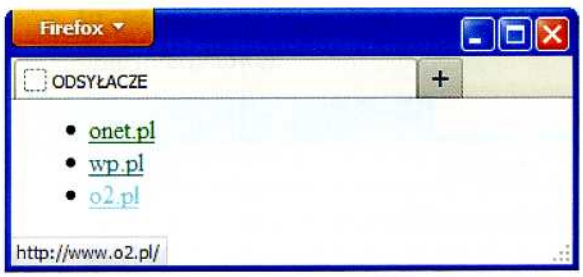
siadanego oprogramowania. Zawartość pliku (np. plików tekstowych) może zostać wyświetlona w oknie przeglądarki lub rozpocznie się procedura pobierania danego pliku.

```
<a href="raport.pdf">raport</a>
<a href="dane.zip">dane - plik archiwalny</a>
<a href="firefox.exe">firefox - plik instalacyjny</a>
```

Jak łatwo zauważyć, wygląd linków jest dość charakterystyczny. Wszystkie linki są podkreślone oraz używają trzech kolorów. Niebieski – odpowiada linkom nieodwiedzonym, fioletowy – linkom odwiedzonym, natomiast czerwony pojawia się w chwili kliknięcia linku. Każdy z wariantów możemy zmienić. Odpowiadają za to atrybuty umieszczane wewnątrz znacznika **body**. Są to: **link** – odsyłacz nieodwiedzony, **vlink** – odsyłacz odwiedzony oraz **alink** – odsyłacz aktywny (przy kliknięciu). Każdy z nich jako wartość przyjmuje nazwę wybranego koloru.

Listing 11.3

```
<body link="green" vlink="teal" alink="aqua">
<ul>
<li> <a href="http://www.onet.pl"> onet.pl </a> </li>
<li> <a href="http://www.wp.pl"> wp.pl </a> </li>
<li> <a href="http://www.o2.pl"> o2.pl </a> </li>
</ul>
</body>
```



Rys. 11.3. Strona WWW stworzona na podstawie list. 11.3

Powyżej przedstawiono szereg odsyłaczy tekstowych. Alternatywą takich odsyłaczy mogą być odsyłacze graficzne. W tym przypadku zamiast wyświetlanego fragmentu tekstu pojawia się obrazek (np. przycisk), który po kliknięciu zachowuje się identycznie jak odsyłacz tekstowy. Jedyna różnica w budowie odsyłacza polega na wprowadzeniu znacznika osadzenia obrazka (**img**) w miejsce nazwy wyświetlanej:

```
<a href="adres"><img src="adres pliku graficznego" alt="tekst alternatywny"></a>
```

Rozszerzoną wersją odsyłacza obrazkowego jest mapa odsyłaczy. Jej zadaniem jest stworzenie (mapowanie) odsyłacza z fragmentu wybranego obrazka. Definicja mapowania przedstawia się następująco:

```
<img src="adres pliku graficznego" alt="tekst alternatywny" usemap="#nazwa_mapy">
```

```
<map id="nazwa_mapy"
  <area    shape="ksz
  alt="tekst alterna
</map>
```

Mapowany obrazek otrzy
wskazuje na parametry map
nik map zawiera identyfika
pomocą atrybutów kształt (
od wybranego kształtu zmie

Tabela 11.1. Właściwości atrybu

shape	
rect – prostokąt	„x x1 y1 x2 y2
poly – wielokąt	„x xn wie
circle – koło	„x x, r –

SPRAWDŹ SWO

1. Wykonaj minisłownik ar
stron otwierających się w
Prowadzi do strony inter
cym znaczenie linku.

SPRAWDŹ SWO

1. Co to jest odsyłacz?
2. Jakie znasz rodzaje odsył
3. Jak zmienić wygląd odsył

```
<map id="nazwa_mapy" name="nazwa_mapy">
  <area    shape="kształt"    coords="współrzędne"    href="adres"
  alt="tekst alternatywny">
</map>
```

Mapowany obrazek otrzymuje dla znacznika **img** dodatkowy atrybut **usemap**, który wskazuje na parametry mapowania poprzez podanie nazwy identyfikującej mapę. Znacznik map zawiera identyfikator **id** określający nazwę mapy. Znacznik **area** określa za pomocą atrybutów kształt (**shape**) oraz współrzędne (**coords**) odsyłacza. W zależności od wybranego kształtu zmieniają się współrzędne, co prezentuje tabela 11.1.

Tabela 11.1. Właściwości atrybutów znacznika **area**aws

shape	coords
rect – prostokąt	„**x1,y1, x2,y2**" **x1** – współrzędna pozioma lewego górnego wierzchołka prostokąta **y1** – współrzędna pionowa lewego górnego wierzchołka **x2** – współrzędna pozioma prawego dolnego wierzchołka prostokąta **y2** - współrzędna pionowa prawego dolnego wierzchołka
poly – wielokąt	„**x1,y1, x2,y2, x3,y3 ...**" **xn,yn** – współrzędne poziome i pionowe kolejnych wierzchołków wielokąta
circle – koło	„**x, y, r**" **x, y** – współrzędne środka **r** – długość promienia

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj minisłownik angielsko-polski. Strona główna ma cztery odnośniki do podstron otwierających się w nowych oknach. Przykładowy odnośnik to np.: **dog** (pl. pies). Prowadzi do strony internetowej z obrazkiem psa i odpowiednim podpisem tłumaczącym znaczenie linku.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Co to jest odsyłacz?
2. Jakie znasz rodzaje odsyłaczy?
3. Jak zmienić wygląd odsyłaczy?

12 Formularze

ZAGADNIENIA

- Jak zbudowany jest formularz?
- Jakie pola można umieścić w formularzu?
- Jak wstawiać formularz na stronę WWW?
- Jak wstawiać różne elementy formularza?

Formularz jest pewnego rodzaju narzędziem. Może być wykorzystany do zbierania informacji (ankieta), logowania, przesyłania poczty, złożenia zamówienia w sklepie internetowym itp. Przetwarzanie danych zawartych w formularzu realizowane jest za pomocą języka skryptowego (np. PHP) po stronie serwera. Język HTML odpowiada jedynie za wprowadzenie odpowiednich elementów w formularzu.

Znacznikiem tworzącym ramy formularza jest **<form>** ... **</form>**. Wewnątrz ramy można umieszczać różne elementy tworzące formularz. Większość z nich tworzona jest za pomocą znacznika **input**.

Najczęściej stosowanym elementem jest pole tekstowe (wiersz wprowadzania danych). Ma ono wysokość jednej linii tekstu oraz określoną długość. Powstaje w wyniku zastosowania w znaczniku **input** typu **text** oraz określenia *nazwy* charakteryzującej dane pole (np. skrócona treść pytania). Ważne jest, aby *nazwa* nie była zbyt długa oraz by była unikatowa dla danego formularza.

```
<input type="text" name="nazwa">
```

Znacznik **input** może przyjmować kilka atrybutów:

- **value** – treść stanowiąca odpowiedź domyślną, która zostanie dołączona do formularza przy wysyłaniu, jeżeli użytkownik nie wprowadzi własnej treści;
- **size** – długość pola podawana w liczbie znaków;
- **maxlength** – maksymalna dopuszczalna liczba znaków, jaką można wprowadzić;
- **readonly** – tylko do odczytu – tekst wprowadzony w tym polu jako domyślny, nie może zostać zmodyfikowany (przyjmuje wartość – **readonly="readonly"**);
- **disabled** – blokada pola (przyjmuje wartość – **disabled="disabled"**).

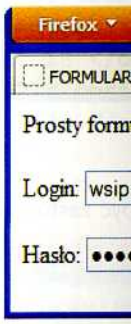
Elementem podobnym do pola tekstowego jest pole hasła. Na pierwszy rzut oka nie widać między nimi różnicy, ale gdy zaczniemy wpisywać do niego treść, zamiast znaków pojawią się gwiazdki (*).

```
<input type="password" name="nazwa">
```

Pole takie spotykamy często w formularzach logowania. Jest wprowadzane po to, aby ktoś stojący w pobliżu nie mógł zobaczyć wpisywanego hasła (list. 12.1, rys. 12.1).

Listing 12.1

```
<form>
Prosty formularz l
Login: <input type
Hasło: <input type
</form>
```



Rys. 12.1. Strona WWW stworz

Pole wyboru (pole typu **ch** zaznaczać i odznaczać. Pozw

```
<input type="checkbo
```

Tworząc pole, należy pam tania, oraz wartości – **valu** zostaną przesłane wraz z fo

Przykładowy formularz z pytającej o zainteresowania

Listing 12.2

```
<form>
Podaj swoje zaintere
<input type="checkbo
<input type="checkbo
<input type="checkbo
<input type="checkbo
</form>
```

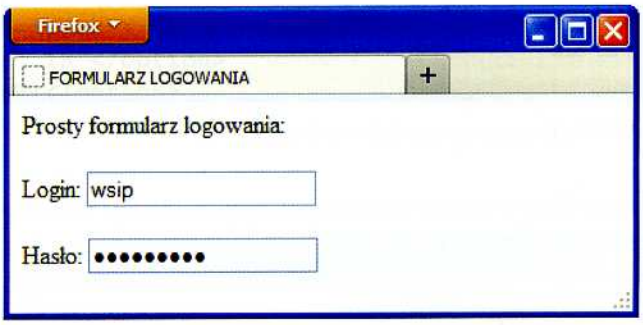
Pole opcji (pole typu **rad** stwie do pola wyboru możn

```
<input type="radio"
```

Tworząc pole, należy pam tania, oraz wartości – **valu** zostaną przesłane wraz z fo

Listing 12.1

```
<form>
Prosty formularz logowania:<br><br>
Login: <input type="text" name="login"><br><br>
Hasło: <input type="password" name="haslo">
</form>
```



Rys. 12.1. Strona WWW stworzona na podstawie list. 12.1

Pole wyboru (pole typu **checkbox**) to mały kwadracik, który za pomocą myszki można zaznaczać i odznaczać. Pozwala to na wybranie kilku odpowiedzi z wszystkich możliwych.

```
<input type="checkbox" name="nazwa" value="wartość">
```

Tworząc pole, należy pamiętać o wprowadzeniu nazwy – **name**, związanej z treścią pytania, oraz wartości – **value**, odpowiadającej określonej odpowiedzi. Oba te parametry zostaną przesłane wraz z formularzem.

Przykładowy formularz z zastosowaniem pól wyboru może być zastosowany w ankiecie pytającej o zainteresowania (list. 12.2, rys. 12.2).

Listing 12.2

```
<form>
Podaj swoje zainteresowania: <br><br>
<input type="checkbox" name="hobby" value="muzyka"> muzyka <br>
<input type="checkbox" name="hobby" value="sport"> sport <br>
<input type="checkbox" name="hobby" value="taniec"> taniec <br>
<input type="checkbox" name="hobby" value="film"> film <br>
</form>
```

Pole opcji (pole typu **radio**) to małe kółeczko, które można zaznaczyć. W przeciwieństwie do pola wyboru można w tym przypadku zaznaczyć tylko jedną odpowiedź.

```
<input type="radio" name="nazwa" value="wartość">
```

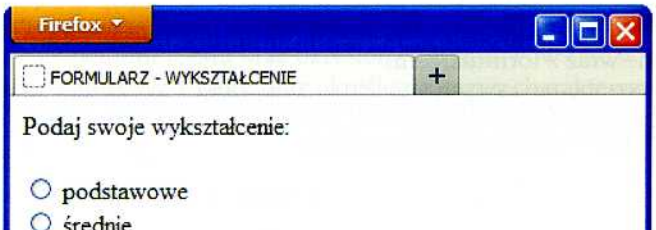
Tworząc pole, należy pamiętać o wprowadzeniu nazwy – **name**, związanej z treścią pytania, oraz wartości – **value**, odpowiadającej określonej odpowiedzi. Oba te parametry zostaną przesłane wraz z formularzem.

Rys. 12.2. Strona WWW stworzona na podstawie list. 12.2

Przykładowy formularz z zastosowaniem pól opcji może być zastosowany w ankiecie pytającej o wykształcenie (list. 12.3, rys. 12.3).

Listing 12.3

```
<form>
Podaj swoje wykształcenie: <br><br>
<input type="radio" name="wyksz" value="podst"> podstawowe <br>
<input type="radio" name="wyksz" value="sr"> średnie <br>
<input type="radio" name="wyksz" value="wyz"> wyższe
</form>
```



Rys. 12.3. Strona WWW stworzona na podstawie list. 12.3

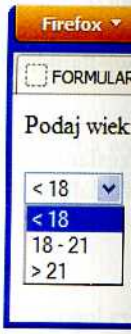
Lista rozwijana również umożliwia wybór tylko jednej odpowiedzi. Jej budowa różni się od pozostałych elementów. Składa się z dwóch znaczników. Znacznik **<select>** ... **</select>** stanowi ramę listy. Każda z możliwych odpowiedzi znajduje się wewnątrz znaczników **<option>** ... **</option>**. Liczba odpowiedzi jest dowolna:

```
<select name="nazwa">
<option> odpowiedz 1 </option>
<option> odpowiedz 2 </option>
...
</select>
```

Należy pamiętać o nada
która będzie przesłana wra
Przykładowy formularz
kiecie pytającej o przedział

Listing 12.4

```
<form>
Podaj wiek: <br><br>
<select name="wiek">
<option> &lt; 18 </o
<option> 18-21</opti
<option> &gt; 21 </o
</select>
</form>
```



Rys. 12.4. Strona WWW stwor

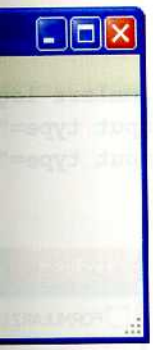
Obszar tekstowy jest elem
wolnej długości tekstu. Za p
area wraz z dodatkowymi a
szerokość mierzoną liczbą
wprowadzony pomiędzy zn
szarze tekstowym (list. 12.5,

Listing 12.5

```
<form>
Opisz siebie...<br><
<textarea name="nazwa
Maksymalnie 250 znak
</textarea>
</form>
```

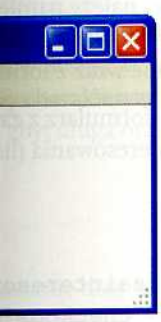
Ostatnim elementem, jak
typu file). Jego zadaniem je
mularzu prezentuje się jako
okno umożliwiające przegląd

```
<input type="file" na
```



e być zastosowany w ankiecie

```
st"> podstawowe <br>
> średnie <br>
"> wyższe
```



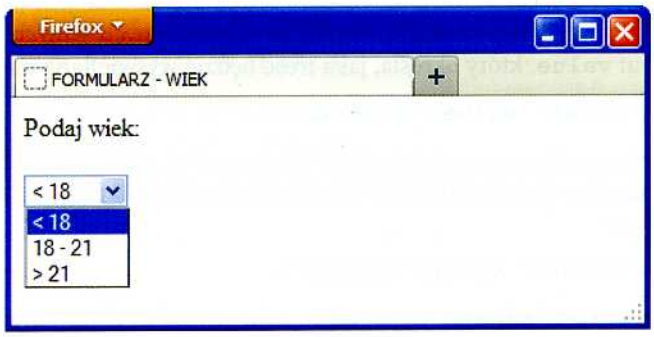
odpowiedzi. Jej budowa różni
ików. Znacznik **<select>** ...
wiedzi znajduje się wewnątrz
i jest dowolna:

Należy pamiętać o nadaniu odpowiedniej nazwy – **name** wewnątrz znacznika **select**, która będzie przesłana wraz z formularzem.

Przykładowy formularz z zastosowaniem listy rozwijanej może być zastosowany w ankiecie pytającej o przedział wiekowy (list. 12.4, rys. 12.4).

Listing 12.4

```
<form>
Podaj wiek: <br><br>
<select name="wiek">
<option> &lt; 18 </option>
<option> 18-21</option>
<option> &gt; 21 </option>
</select>
</form>
```



Rys. 12.4. Strona WWW stworzona na podstawie list. 12.4

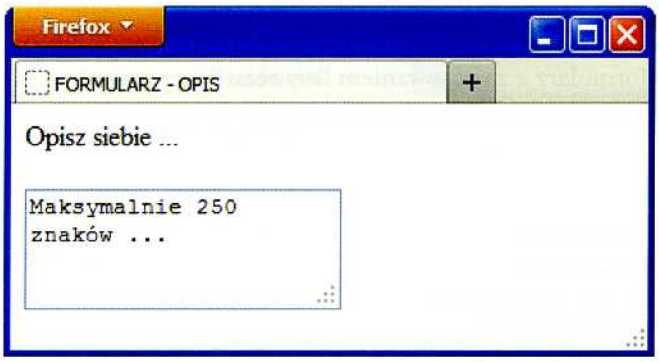
Obszar tekstowy jest elementem pozwalającym na wprowadzenie przez użytkownika dowolnej długości tekstu. Za pojawienie się obszaru tekstowego odpowiada znacznik **textarea** wraz z dodatkowymi atrybutami określającymi wymiary pola. Atrybut **rows** określa szerokość mierzoną liczbą wierszy, **cols** określa długość mierzoną liczbą kolumn. Tekst wprowadzony pomiędzy znacznikami **textarea** jest wyświetlany jako domyślny w obszarze tekstowym (list. 12.5, rys.12.5).

Listing 12.5

```
<form>
Opisz siebie...<br><br>
<textarea name="nazwa" cols="20" rows="3">
Maksymalnie 250 znaków...
</textarea>
</form>
```

Ostatnim elementem, jaki można spotkać w formularzu, jest pole wyboru pliku (pole typu file). Jego zadaniem jest przekazanie pliku na serwer za pomocą formularza. W formularzu prezentuje się jako przycisk z napisem *Przeglądaj...* Po jego wybraniu ukazuje się okno umożliwiające przeglądanie plików dostępnych na dysku użytkownika.

```
<input type="file" name="nazwa">
```



Rys. 12.5. Strona WWW stworzona na podstawie list. 12.5

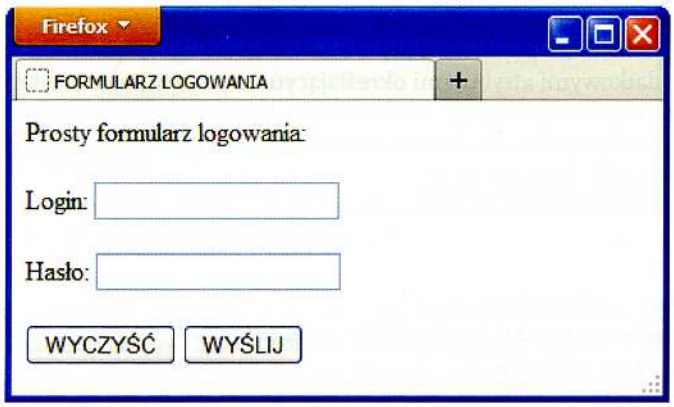
W przypadku, gdy formularz zawiera większą liczbę pól, dobrze jest wprowadzić przycisk typu **reset**, umożliwiający wyczyszczenie wprowadzonych danych. Pojawia się tu dodatkowy atrybut **value**, który określa, jaka treść będzie wyświetlana na przycisku:

```
<input type="reset" value="WYCZYŚĆ">
```

Na koniec brakuje jedynie przycisku odpowiadającego za przesłanie formularza. Jest to przycisk typu **submit** z dodatkowym atrybutem **value**, odpowiadającym za treść wyświetlaną na przycisku.

```
<input type="submit" value="WYŚLIJ">
```

Oba przyciski, **reset** i **submit**, wprowadzono do formularza logowania z listingu 12.1. Efekt końcowy przedstawia rys. 12.6.



Rys. 12.6. Formularz logowania z przyciskami

SPRAWDŹ SW

1. Wykonaj formularz ele
   Podaj imię – pole tekst
   Podaj nazwisko – pole
   PESEL – pole tekstowe,
   Kod pocztowy – dwa po
   2 znaki, długość drugie
   Płeć: kobieta, mężczyzn
   Wybierz województwo
   Zainteresowania – wpro
   Wyczyść, Wyślij – przyc

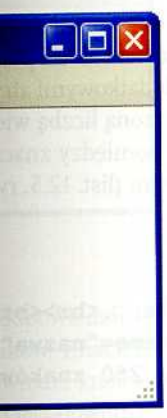
SPRAWDŹ SW

1. Jakie pola można umieś
2. W jaki sposób można w
3. Podaj praktyczne przykł

dobrze jest wprowadzić przy-
onych danych. Pojawia się tu
yświetlana na przycisku:

przesłanie formularza. Jest to
owiadającym za treść wyświe-

larza logowania z listingu 12.1.



SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj formularz elektroniczny według instrukcji.

   Podaj imię – pole tekstowe.

   Podaj nazwisko – pole tekstowe.

   PESEL – pole tekstowe, maksymalna liczba znaków – 11.

   Kod pocztowy – dwa pola tekstowe rozdzielone dywizem (-), długość pierwszego pola – 2 znaki, długość drugiego pola – 3 znaki.

   Płeć: kobieta, mężczyzna – pole opcji.

   Wybierz województwo – wypisz wszystkie województwa w liście rozwijanej.

   Zainteresowania – wprowadź 8 możliwości, pole wyboru.

   Wyczyść, Wyślij – przyciski.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Jakie pola można umieścić w formularzu?
2. W jaki sposób można wyczyścić formularz?
3. Podaj praktyczne przykłady zastosowania formularzy.

13 Ramki

ZAGADNIENIA

- Co to jest ramka?
- Jakie atrybuty przyjmuje ramka?
- Jak wczytać stronę do ramki?
- Jak podzielić stronę na kilka ramek?
- Jak stworzyć menu odnoszące się do wybranej ramki?

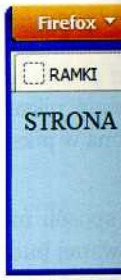
Ramki odpowiadają za podział obszaru strony na charakterystyczne bloki. Każdy z nich może wyświetlać inną stronę internetową lub inny obraz. Kiedyś stanowiły idealny sposób na dokonanie podziału na stronie (logo, menu, treść, stopka). Lecz z czasem okazało się, że przynoszą więcej szkody niż pożytku. Związane jest to m.in. z niedokładnym indeksowaniem takiej strony przez wyszukiwarki, gdyż składa się ona z wielu dokumentów HTML oraz jednego pliku opisującego budowę ramki. Wyszukiwarki, błędnie interpretując składnię dokumentów, odnosiłyzą się tylko do pojedynczych stron, a nie skupiają na całości. Mimo to na wielu stronach można spotkać się z taką budową dokumentu.

Cechą charakterystyczną takich dokumentów jest brak znacznika **body**. Zastępuje go znacznik **<frameset>** ... **/frameset>** odpowiadający za sposób podziału strony. Można ją podzielić na kolumny – **cols** oraz na wiersze – **rows**. Oba te atrybuty mogą przyjmować wartości na trzy różne sposoby: w pikselach (np. 120), procentach (np. 30%) oraz tak, by pozostały dostępny obszar był oznaczany jako *. Po dokonaniu podziału wprowadzamy odpowiednią liczbę ramek za pomocą znacznika **frame**. Ma on dodatkowy atrybut **src**, który określa, jaki dokument HTML zostanie wyświetlony w danej ramce.

Poniższy przykład przedstawia stronę podzieloną na trzy kolumny. Pierwsza ma szerokość 100 pikseli, druga 140 pikseli, a trzecia zajmuje cały pozostały obszar (*). W każdej ramce wyświetla się podstrona mająca inny kolor tła.

Listing 13.1

```
<frameset cols="100,140,*">
   <frame src="strona1.html">
   <frame src="strona2.html">
   <frame src="strona3.html">
<noframes>
   <body>
   Strona dla przeglądarek nieobsługujących ramek
   </body>
</noframes>
</frameset>
```



Rys. 13.1. Strona WWW stworz

Tworząc stronę opartą na
... **</noframes>** zawiera
w **body** w przeglądarkach n

Kolejny przykład przedst
kość 30%, drugi zajmuje ca
na mająca inny kolor tła (w
utworzyć trzy strony różniąc
na2.html, strona3.html).

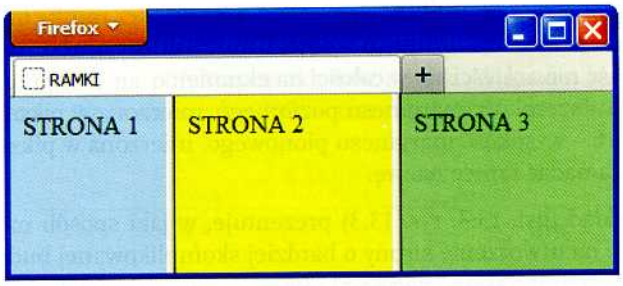
Listing 13.2

```
<frameset rows="30%,
   <frame src="strona
   <frame src="strona
<noframes>
   <body>
   Strona dla przegl
   </body>
</noframes>
</frameset>
```

Rys. 13.2. Strona WWW stwor

Znacznik **frame** może p

- **noresize** – pozwala na
- **frameborder** – określ
mowanie widoczne, war



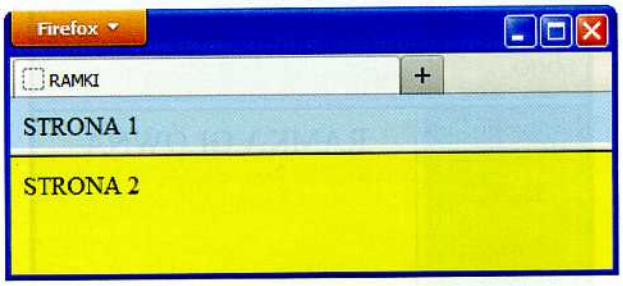
Rys. 13.1. Strona WWW stworzona na podstawie list. 13.1

Tworząc stronę opartą na ramkach, dodatkowo należy umieścić znacznik **<noframes>** ... **</noframes>** zawierający sekcję **body**. Pozwala to na wyświetlenie treści zawartej w **body** w przeglądarkach nieobsługujących ramek.

Kolejny przykład przedstawia stronę podzieloną na dwa wiersze. Pierwszy ma szerokość 30%, drugi zajmuje cały pozostały obszar (*). W każdej ramce wyświetla się podstrona mająca inny kolor tła (w celu przetestowania poniższych listingów należy dodatkowo utworzyć trzy strony różniące się kolorem tła o odpowiednich nazwach strona1.html, strona2.html, strona3.html).

Listing 13.2

```
<frameset rows="30%,*">
   <frame src="strona1.html">
   <frame src="strona2.html">
<noframes>
   <body>
   Strona dla przeglądarek nieobsługujących ramek
   </body>
</noframes>
</frameset>
```



Rys. 13.2. Strona WWW stworzona na podstawie list. 13.2

Znacznik **frame** może przyjmować kilka dodatkowych atrybutów:

- **noresize** – pozwala na zablokowanie zmiany rozmiaru ramki;
- **frameborder** – określa, czy obramowanie ramki będzie widoczne (wartość **1** – obramowanie widoczne, wartość **0** – obramowanie niewidoczne);

- **scrolling** – określa, czy suwaki przewijania zawartości ramki mają być widoczne (wartość **yes** – suwak widoczny, **no** – suwak niewidoczny, **auto** – suwak pojawi się, gdy jej zawartość nie zmieści się w całości na ekranie);
- **marginwidth** – szerokość marginesu poziomego, mierzona w pikselach;
- **marginheight** – wysokość marginesu pionowego, mierzona w pikselach;
- **name** – pozwala nadać ramce nazwę.

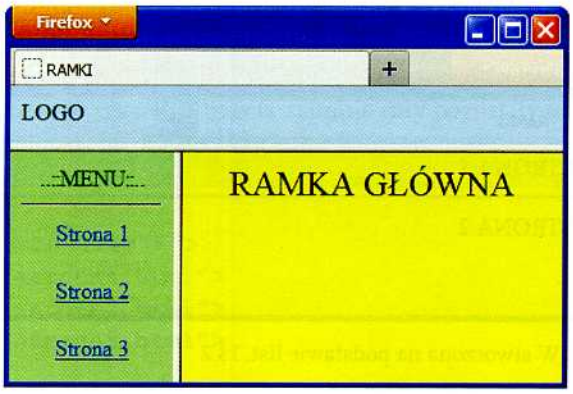
Następny przykład (list. 13.3, rys. 13.3) prezentuje, w jaki sposób można zagnieździć ramki. Pozwala to na utworzenie strony o bardziej skomplikowanej budowie. Dodatkowo ramka trzecia ma atrybut **name="ramkaglowna"**. Została ona użyta dodatkowo jako wartość atrybutu **target** dla znacznika **a** na stronie menu.html:

```
<a href="strona1.html" target="ramkaglowna">Strona 1</a>
```

W takim przypadku po kliknięciu linku Strona 1 zmieni się zawartość ramki trzeciej, w której aktualnie wyświetlana jest strona **start.html**. Po zmianie w tej ramce pokaże się **strona1.html**.

Listing 13.3

```
<frameset rows="40,*">
   <frame src="logo.html">
<frameset cols="30%,*">
   <frame src="menu.html">
   <frame src="start.html" name="ramkaglowna">
</frameset>
<noframes>
   <body>
   Strona dla przeglądarek nieobsługujących ramek
   </body>
</noframes>
</frameset>
```



Rys. 13.3. Strona WWW stworzona na podstawie list. 13.3

SPRAWDŹ SWO

1. Wykonaj stronę opartą n
stronę internetową. Pierw
ki analogicznie.

30%, 50%

30%, 50%

SPRAWDŹ SWO

1. Co daje umieszczenie ra
2. Za co odpowiada atrybut
3. Jak zablokować ramkę?
4. Podaj przykładowe strony

ci ramki mają być widoczne
ny, **auto** – suwak pojawi się,

zona w pikselach;
zona w pikselach;

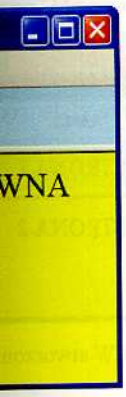
ki sposób można zagnieździć
kowanej budowie. Dodatkowo
ona użyta dodatkowo jako war-
l:

```
>Strona 1</a>
```

i się zawartość ramki trzeciej,
o zmianie w tej ramce pokaże

```
a">
```

ramek



SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wykonaj stronę opartą na poniższym układzie ramek. W każdej ramce wczytaj inną stronę internetową. Pierwsza ramka: 30% – szerokość, 50% – wysokość, pozostałe ramki analogicznie.

<table>
<tr><td rowspan="2">30%, 50%</td><td rowspan="3">*,*</td><td>30%,*</td></tr>
<tr><td>30%,50%</td></tr>
<tr><td>30%, 50%</td><td>30%,*</td></tr>
</table>

SPRAWDŹ SWOJĄ WIEDZĘ

1. Co daje umieszczenie ramek na stronie?
2. Za co odpowiada atrybut **target**?
3. Jak zablokować ramkę?
4. Podaj przykładowe strony, na których zastosowano ramki. Dlaczego jest ich tak niewiele?

16 Składnia kaskadowych arkuszy stylów

ZAGADNIENIA

- Co to jest selektor?
- Co to jest grupowanie selektorów?
- Co to są reguły stylów?
- Jak budować style wpisane?
- Jak budować style osadzone i zewnętrzne?

Za zmianę wyglądu witryny w kaskadowych arkuszach stylów odpowiadają **reguły stylów**. Każda z reguł powiązana jest z konkretnym elementem (znacznikiem) występującym w dokumencie HTML. Składa się ona z dwóch części: selektora i deklaracji. **Selektor** określa, jaki element (znacznik) ma zostać sformatowany, natomiast **deklaracja** składa się z jednej lub kilku cech, którym przypisane są konkretne wartości. Cecha i wartość deklaracji rozdzielone są dwukropkiem i umieszczone w nawiasie klamrowym. Jeżeli wartość ma postać wielowyrazową, umieszczamy ją w cudzysłowie, np. `"Times New Roman"`.

selektor {cecha: wartość}

Selektor może być dowolnym znacznikiem występującym w dokumencie hipertekstowym, np. **p** (akapit), **table** (tabela), **body** (ciało dokumentu) czy **ul** (wypunktowanie). Po przypisaniu odpowiednich cech do selektora wszystkie elementy znajdujące się pomiędzy formatowanym znacznikiem ulegną zmianie. Jeżeli będziemy chcieli zmienić tło strony na kolor zielony, użyjemy selektora body:

```
body {background-color: green}
```

Wybrany selektor może posiadać kilka cech oddzielonych od siebie średnikiem. Na przykład akapit **p** przyjmuje kolor czcionki niebieski, krój czcionki Courier New oraz pogrubienie tekstu:

```
p {
   color: blue;
   font-family: "Courier New";
   font-weight: bold;
}
```

Arkusze stylów pozwalają na nadanie tych samych cech kilku selektorom jednocześnie. Takie działanie określa się mianem **grupowania selektorów**. Dla przykładu, można nadać efekt podkreślenia dla całej grupy tytułów od h1 do h6, wystarczy kolejne znaczniki oddzielić od siebie przecinkiem:

```
h1, h2, h3, h4, h5, h6 {text-decoration: underline}
```

Sposób definiowania wy
stylów zewnętrznych oraz s
stylów wpisanych. Styl wpi
kumentu HTML. Na przyk
kowania i zmiany koloru c

```
<p style="text-align
Akapit ze stylami wp
</p>
```

Efektem działania takieg
glądającego napisu:

Należy pamiętać, że sty
acjach, gdy istnieje konie
elementu. Zalecane jest st
globalne formatowanie zaw
dzanie zmian.

SPRAWDŹ SW

1. Co to jest selektor?
2. Z czego składa się dekla

Sposób definiowania wyglądu witryny za pomocą selektorów jest charakterystyczny dla stylów zewnętrznych oraz stylów osadzonych. Sytuacja wygląda nieco inaczej w przypadku stylów wpisanych. Styl wpisany wprowadzamy bezpośrednio w wybranym znaczniku dokumentu HTML. Na przykład wprowadzimy zmianę dla akapitu **p** w postaci jego wyśrodkowania i zmiany koloru czcionki na turkusowy:

```
<p style="text-align: center; color: teal">
Akapit ze stylami wpisanymi
</p>
```

Efektem działania takiego stylu będzie wyświetlenie na ekranie strony następująco wyglądającego napisu:

Akapit ze stylami wpisanymi

Należy pamiętać, że styl wpisany powinien być stosowany tylko w wyjątkowych sytuacjach, gdy istnieje konieczność wprowadzenia specyficznego stylu dla pojedynczego elementu. Zalecane jest stosowanie głównie stylów zewnętrznych, odpowiadających za globalne formatowanie zawartości witryny. Pozwala to na szybsze i sprawniejsze wprowadzanie zmian.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Co to jest selektor?
2. Z czego składa się deklaracja stylów?

17 Właściwości CSS

ZAGADNIENIA

- **Jakie możliwości daje CSS?**
- **Jakie cechy i wartości udostępnia CSS?**
- **Jak formatować stronę za pomocą stylów?**

Wiemy już, że selektorom możemy przypisać pewne cechy o charakterystycznych własnościach (wartościach). Cechy te odpowiadają za zmianę między innymi: czcionki, tła, dekoracji tekstu, wyglądu list, kursora. Dla ułatwienia wszystkie te informacje umieszczono w tabelach odpowiadających różnym grupom cech.

Tabela 17.1. Właściwości CSS – czcionka

Cecha	Opis	Możliwe wartości
font-family	deklaracja kroju czcionki	• główne kroje czcionek: Arial, Verdana, Georgia, Times New Roman, Courier New, Helvetica, Tahoma • główne rodziny czcionek: serif, Sans-serif
font-size	deklaracja rozmiaru czcionki	wysokość podawana w wybranej jednostce (1 cm, 10 pt, 15 px, 70%) lub jako predefiniowana wartość: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger
font-style	deklaracja stylu czcionki	normal, italic, oblique
font-variant	deklaracja typu czcionki	normal, small-caps
font-weight	deklaracja grubości czcionki	lighter, normal, bold, bolder, 100, 200, 300, 400, 500, 600, 700, 800, 900
font	skrócona forma deklarowania wszystkich właściwości czcionki w jednym miejscu	oddzielone spacjami wartości w następującej kolejności: style, variant, weight, size, family

PRZYKŁAD 17.1

Formatowanie akapi
– 500, rozmiar – 20 p

```
p {
   font-family: "Couri
   font-style: italic;
   font-weight: 500;
   font-size: 20px;
}
```

lub forma skrócona:

```
p{font: italic 500 20p
```

Tabela 17.2. Właściwości CSS

Cecha	
color	deklaracja k
line-height	deklaracja o
letter-spacing	deklaracja o mi tekstu
text-align	deklaracja p nia tekstu
text-decoration	deklaracja w racja czcion
text-indent	deklaracja w wiersza teks
text-transform	deklaracja z w tekście
word-spacing	deklaracja o wyrazami w

PRZYKŁAD 17.2

Formatowanie tytułu
mi – 10px, wyśrodko

```
h1 {
   text-decoration: ove
   word-spacing: 10px
   text-align: center;
   color: green;
}
```

PRZYKŁAD 17.1

Formatowanie akapitu **p**: krój czcionki – Courier New, styl – pochylenie, grubość – 500, rozmiar – 20 px.

```
p {
   font-family: "Courier New";
   font-style: italic;
   font-weight: 500;
   font-size: 20px;
}
```

lub forma skrócona:

```
p{font: italic 500 20px "Courier New"}
```

Tabela 17.2. Właściwości CSS – tekst

Cecha	Opis	Możliwe wartości
color	deklaracja koloru tekstu	nazwy kolorów RGB lub ich oznaczenia szesnastkowe
line-height	deklaracja odstępu między liniami	wysokość podawana w wybranej jednostce (1 cm, 10 pt, 15 px, 70%)
letter-spacing	deklaracja odstępu między znakami tekstu	odstęp podawany w wybranej jednostce
text-align	deklaracja poziomego wyrównania tekstu	left, right, center, justify
text-decoration	deklaracja wyróżnienia lub deklaracja czcionki	none, underline, overline, line-through, blink
text-indent	deklaracja wcięcia pierwszego wiersza tekstu	długość podawana w wybranej jednostce
text-transform	deklaracja zmiany wielkości liter w tekście	none, capitalize, uppercase, lowercase
word-spacing	deklaracja odległości między wyrazami w tekście	długość podawana w wybranej jednostce

PRZYKŁAD 17.2

Formatowanie tytułu **h1**: dekoracja tekstu – nadkreślenie, odstępy między wyrazami – 10px, wyśrodkowanie, kolor tekstu – zielony:

```
h1 {
   text-decoration: overline;
   word-spacing: 10px;
   text-align: center;
   color: green;
}
```

Tabela 17.3. Właściwości CSS – tło

Cecha	Opis	Możliwe wartości
background-attachment	definiuje, czy obrazek tła przewija się razem ze stroną, czy jest nieruchomy	fixed, scroll
background-color	deklaracja koloru tła	nazwa koloru RGB lub zapis szesnastkowy
background-image	deklaracja obrazka jako tła strony	adres URL
background-position	deklaracja pozycji obrazka tła	współrzędne podawane w wybranych jednostkach (10 px 30 px), procentach lub wartości predefiniowane: top, left, right, bottom, center
background-repeat	deklaracja sposobu powtarzania obrazka w tle strony	repeat, repeat-x, repeat-y, no-repeat
background	skrócona forma deklarowania wszystkich właściwości tła w jednym miejscu	oddzielone spacjami wartości w następującej kolejności: color, image, repeat, attachment, position

PRZYKŁAD 17.3

Ustawienie tła strony: tło obrazkowe (/img/tlo.jpg), nieruchome, bez powtarzania:

```
body {
    background-image: url(/img/tlo.jpg);
    background-attachment: fixed;
    background-repeat: no-repeat
}
```

lub forma skrócona:

```
body{background:url(/img/tlo.jpg) no-repeat fixed}
```

Tabela 17.4. Właściwości CSS – obramowanie

Cecha	Opis	Możliwe wartości
border-color border-top-color border-left-color border-right-color border-bottom-color	deklaracja koloru wszystkich krawędzi lub wybranej krawędzi (górna, lewa, prawa, dolna)	nazwa koloru RGB lub zapis szesnastkowy (domyślnie przeźroczyste)
border-style	deklaracja stylu wszystkich krawędzi lub wybranej krawędzi (górna, lewa, prawa, dolna)	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset
border-width	deklaracja grubości wszystkich krawędzi lub wybranej krawędzi (górna, lewa, prawa, dolna)	grubość określona w pikselach lub wartość predefiniowana: thin, medium, thick
border	skrócona forma deklarowania wszystkich właściwości obramowania w jednym miejscu lub wybranej krawędzi (górna, lewa, prawa, dolna)	oddzielone spacjami wartości w następującej kolejności: width, style, color

PRZYKŁAD 17.4

Ustawienie obramo
bość – średnia.

```
div {
    border-style: dotte
    border-color: red;
    border-width: med
}
```

lub forma skrócona:

```
div{border:medium d
```

Tabela 17.5. Właściwości CS

Selektor:pseudoklasa
a:link
a:visited
a:hover
a:active

PRZYKŁAD 17.5

Zmiana wyglądu lin
– brak, tło – szare,
niu myszką: kolor –
kreskowana.

```
a:link, a:visited {
    color: black;
    text-decoration: n
    background-color:
    border-style: solid;
}
a:active, a:hover {
    color: silver;
    text-decoration: n
    background-color:
    border-style: dashe
}
```

Możliwe wartości
ru RGB lub zapis szesnastkowy
ne podawane w wybranych h (10 px 30 px), procentach lub redefiniowane: top, left, right, nter
eat-x, repeat-y, no-repeat
e spacjami wartości w następ- lejności: color, image, repeat, t, position

eruchome, bez powtarzania:

Możliwe wartości
a koloru RGB lub zapis szesnast- (domyślnie przeźroczyste)
, hidden, dotted, dashed, solid, le, groove, ridge, inset, outset
ość określona w pikselach lub war- predefiniowana: thin, medium,
ielone spacjami wartości w nastę- cej kolejności: width, style, color

PRZYKŁAD 17.4

Ustawienie obramowania dla bloku **div**: styl – kropkowane, kolor – czerwony, grubość – średnia.

```
div {
    border-style: dotted;
    border-color: red;
    border-width: medium;
}
```

lub forma skrócona:

```
div{border:medium dotted red}
```

Tabela 17.5. Właściwości CSS – odsyłacze (linki)

Selektor:pseudoklasa	Opis
a:link	deklaracja odsyłacza na stronie
a:visited	deklaracja linku odwiedzonego
a:hover	deklaracja linku po najechaniu myszką
a:active	deklaracja linku przy kliknięciu

PRZYKŁAD 17.5

Zmiana wyglądu linków. Odsyłacz zwykły i odwiedzony: kolor – czarny, dekoracja – brak, tło – szare, obramowanie – linia ciągła. Odsyłacz aktywny i przy najechaniu myszką: kolor – srebrny, dekoracja – brak, tło – czarne, obramowanie – linia kreskowana.

```
a:link, a:visited {
    color: black;
    text-decoration: none;
    background-color: gray;
    border-style: solid;
}
a:active, a:hover {
    color: silver;
    text-decoration: none;
    background-color: black;
    border-style: dashed;
}
```

Tabela 17.6. Właściwości CSS – listy

Cecha	Opis	Możliwe wartości
list-style-type	deklaracja rodzaju markera w listach wyliczeniowych	disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, hebrew, armenian, georgian, cjk-ideographic, hiragana, ...
list-style-position	deklaracja pozycji znacznika wypunktowania listy	inside, outside
list-style-image	deklaracja obrazka zastępującego marker w listach	adres URL
list-style	skrócona forma deklarowania wszystkich właściwości list	oddzielone spacjami wartości w następującej kolejności: type, position, image

PRZYKŁAD 17.6

Ustawienie obrazka (gwiazdka.gif) zastępującego markera dla listy `ol`:

```
ol {
    list-style-image: url(gwiazdka.gif);
}
```

Tabela 17.7. Właściwości CSS – kursor

Cecha	Opis	Możliwe wartości
coursor	deklaracja typu kursora	adres URL pliku z kursorem lub właściwość predefiniowana: auto, crosshair, default, pointer, move, e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize, text, wait, help

PRZYKŁAD 17.7

Zmiana wyglądu kursora na krzyżyk po najechaniu na tabelę:

```
table, tr, td {
    cursor: crosshair;
}
```

Tabela 17.8. Właściwości CSS – rozmiary

Cecha	Opis	Możliwe wartości
height	deklaracja wysokości elementu	wysokość podawana w wybranej jednostce (1 cm, 10 pt, 15 px, 70%)
width	deklaracja szerokości elementu	szerokość podawana w wybranej jednostce (1 cm, 10 pt, 15 px, 70%)

PRZYKŁAD 17.8

Ustawienie wysoko

```
div {
    height: 95%;
    width: 80%;
}
```

Tabela 17.9. Właściwości CSS

Cecha	
margin-top margin-left margin-right margin-bottom	deklaracja prawego l
margin	skrócona f nia wszyst jednocześ

PRZYKŁAD 17.9

Ustawienie margines

```
body {
    margin: 0px 10px
}
```

SPRAWDŹ SW

1. Wprowadź następujący s
   rozmiar czcionki – śred
   (`<p> Akapit formato`
2. Wprowadź następujący s
   rozmiar czcionki – 18 px,
   liniami tekstu – 20 px, ku

SPRAWDŹ SW

1. Wymień cechy i wartości
2. Wymień cechy i wartości
3. Wymień cechy i wartości
4. Wymień przykładowe cec
5. Wymień przykładowe cec
6. Wymień przykładowe cec

Możliwe wartości
quare, decimal, lower-roman, , lower-alpha, upper-alpha, lower-latin, upper-latin, hebrew, eorgian, cjk-ideographic,
de
spacjami wartości w następującej ype, position, image

kera dla listy **ol**:

Możliwe wartości
kursorem lub właściwość prede- rosshair, default, pointer, move, , nw-resize, n-resize, se-resize, e, w-resize, text, wait, help

tabelę:

Możliwe wartości
awana w wybranej jednostce 5 px, 70%)
awana w wybranej jednostce 5 px, 70%)

PRZYKŁAD 17.8

Ustawienie wysokości (95%) i szerokości (80%) dla elementu **div**:

```
div {
    height: 95%;
    width: 80%;
}
```

Tabela 17.9. Właściwości CSS – marginesy

Cecha	Opis	Możliwe wartości
margin-top margin-left margin-right margin-bottom	deklaracja górnego, lewego, prawego lub dolnego marginesu	odległość podawana w wybranej jednostce (1 cm, 10 pt, 15 px, 70%)
margin	skrócona forma deklarowania wszystkich marginesów jednocześnie	wartości oddzielone spacjami w kolejności: top, right, bottom, left; podanie jednej wartości spowoduje zmianę dla wszystkich marginesów; podanie dwóch wartości spowoduje odpowiednio zmianę dla górnego i dolnego marginesu, a następnie dla lewego i prawego; trzy wartości to zmiana dla marginesu górnego, prawego i lewego, dolnego (trzecia wartość)

PRZYKŁAD 17.9

Ustawienie marginesów dla strony: górny 0 px, prawy i lewy 10 px, dolny 5 px:

```
body {
    margin: 0px 10px 5px;
}
```

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Wprowadź następujący styl dla akapitu **p**: kolor teksu – czerwony, krój czcionki – Arial, rozmiar czcionki – średni, tło – zielone, obramowanie – kreskowane, zielone, 10 px (** Akapit formatowany w stylach zewnętrznych **).
2. Wprowadź następujący styl dla bloku **div**: kolor teksu – biały, krój czcionki – Verdana, rozmiar czcionki – 18 px, tło – szare, obramowanie – podwójne, czarne, odstępy między liniami tekstu – 20 px, kursor – klepsydra, marginesy – 10 px.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Wymień cechy i wartości zmieniające czcionkę.
2. Wymień cechy i wartości formatujące obramowanie.
3. Wymień cechy i wartości zmieniające tło strony.
4. Wymień przykładowe cechy i wartości zmieniające czcionkę.
5. Wymień przykładowe cechy i wartości formatujące obramowanie.
6. Wymień przykładowe cechy i wartości zmieniające tło strony.

18 Klasy i identyfikatory

ZAGADNIENIA

- Co to jest klasa?
- Co to jest identyfikator?
- Jak określić specjalny wygląd wybranego znacznika HTML?
- Jak korzystać z klas i identyfikatorów?

Kaskadowe arkusze stylów CSS pozwalają na formatowanie każdego ze znaczników dokumentu HTML na kilkanaście różnych sposobów. W tym celu stosuje się klasy i identyfikatory.

Klasa jest atrybutem znaczników dokumentu HTML i odnosi się do selektora klasy lub klasy uniwersalnej. Atrybut **class** dla dowolnego znacznika prezentuje się następująco:

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

Selektor klasy jest nazwą klasy połączoną z innym selektorem za pomocą kropki:

selektor.nazwa_klasy{cecha: wartość}

Nazwa klasy może zawierać małe i wielkie litery, cyfry, dywizy (-) i znaki podkreślenia (_). Należy pamiętać, że klasa nie może zaczynać się od cyfry ani od dywizu oraz że nie powinno się używać w nazwie polskich znaków.

Znając już zasadę tworzenia klas, zbudujemy za ich pomocą trzy akapity **p** o różnym formatowaniu:

Listing 18.1 (HTML)

```
<p class="czerwony"> akapit - czerwony, courier new </p>
<p class="zielony"> akapit - zielony, verdana </p>
<p class="niebieski"> akapit - niebieski, garamond </p>
```

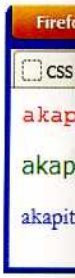
Listing 18.2. (CSS)

```
p.czerwony {color:red; font-family:"courier new"}
p.zielony {color:green; font-family:verdana}
p.niebieski {color:blue; font-family:garamond}
```

Definiując klasę w stylach CSS, można pominąć nazwę selektora. Wówczas powstanie tzw. **klasa uniwersalna**, którą można stosować dla wszystkich znaczników dokumentu HTML.

.klasa_uniwersalna {cecha:wartość}

Stwórzmy klasę uniwersalną wprowadzającą dla wybranych znaczników obramowanie zielone, kreskowane:



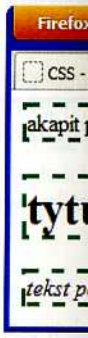
Rys. 18.1. Stronna WWW st

Listing 18.3 (CSS)

.rameczka {border: dash

Listing 18.4 (HTML)

```
<p class="rameczka"
<h1 class="rameczka
<i class="rameczka"
```



Rys. 18.2. Strona WWW stwo

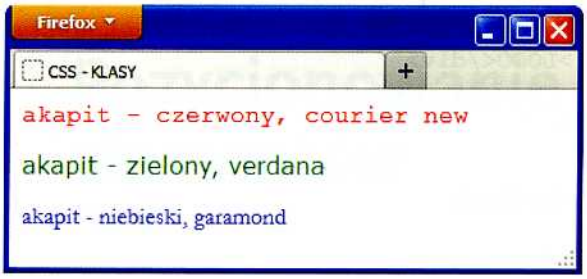
Innym narzędziem do
dobną rolę jak klasy z tą ró
dany identyfikator. Najczę
takich jak: nagłówek (logo
Identyfikator wprowadzan
podlega takim samym zas

```
<znacznik id="nazwa
```

Selektor identyfikatora jes
krzyżyka ("#"):

selektor#nazwa_identyfi

Znając już zasadę twor
w znacznikach **div**:



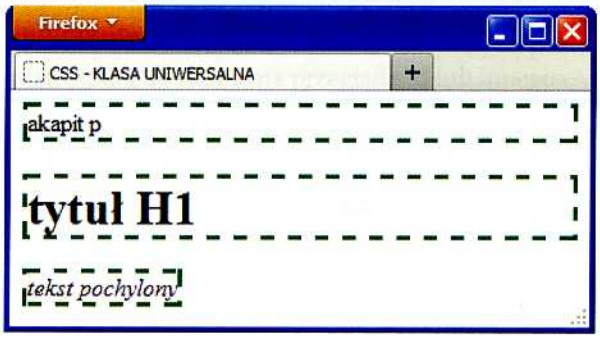
Rys. 18.1. Stronna WWW stworzona na podstawie list. 18.1 (HTML) i list. 18.2 (CSS)

Listing 18.3 (CSS)

```
.rameczka {border: dashed green}
```

Listing 18.4 (HTML)

```
<p class="rameczka"> akapit p </p>
<h1 class="rameczka"> tytuł H1 </h1>
<i class="rameczka"> tekst pochylony </i>
```



Rys. 18.2. Strona WWW stworzona na podstawie list. 18.4 (HTML) i list. 18.3 (CSS)

Innym narzędziem do zmiany elementów strony są **identyfikatory**. Spełniają one podobną rolę jak klasy z tą różnicą, że tylko jeden element w dokumencie HTML może mieć dany identyfikator. Najczęściej stosuje się je do określania podstawowych sekcji strony, takich jak: nagłówek (logo), menu nawigacyjne, blok zawierający treść strony oraz stopka. Identyfikator wprowadzamy, stosując atrybut **id** dla wybranego znacznika, którego nazwa podlega takim samym zasadom jak nazwa klasy:

```
<znacznik id="nazwa_identyfiktora">
```

Selektor identyfikatora jest nazwą atrybutu **id** połączoną z innym selektorem za pomocą krzyżyka ("#"):

selektor#nazwa_identyfikatora{cecha: wartość}

Znając już zasadę tworzenia identyfikatorów, zbudujemy za ich pomocą logo zawarte w znacznikach **div**:

Listing 18.5 (HTML)

```
<div id="logo">LOGO</div>
```

Listing 18.6 (CSS)

```
div#logo {
    background-color: black;
    color: white;
    font: 50px arial;
    text-align: center;
    text-shadow: silver 4px 4px 8px;
    border: double silver;
    letter-spacing: 15px;
}
```

Rys. 18.3. Stronna WWW stworzona na podstawie list. 18.5 (HTML) i list. 18.6 (CSS)

Jeżeli chcemy zastosować atrybut **id** dla dowolnego znacznika, wystarczy utworzyć **identyfikator uniwersalny**, pomijając definicję selektora:

```
#identyfikator_uniwersalny
```

Wywołanie identyfikatora uniwersalnego wygląda tak samo jak w poprzednim przypadku, z tą różnicą, że można go przypisać do dowolnie wybranego znacznika w dokumencie.

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Zbuduj stronę, na której umieścisz swój ulubiony wiersz. W celu sformatowania poszczególnych elementów wiersza (tytuł, autor, treść) wykorzystaj identyfikatory.
2. Zbuduj stronę, na której umieścisz cztery przysłowia. Utwórz trzy klasy: przymiotnik, rzeczownik i czasownik, z których każda będzie mieć inne formatowanie (np. przymiotnik – czerwony, pochylenie). Zastosuj klasy dla odpowiednich części mowy w zaproponowanych przysłowiach.

SPRAWDŹ SWOJĄ WIEDZĘ

1. Czym są klasa i identyfikator?
2. Jak wprowadzić klasę uniwersalną?
3. Jak wprowadzić identyfikator uniwersalny?

19

ZAGADNIENIA

- Co to jest pozycjonowanie
- Jakie są rodzaje pozycjono
- Jak umieszczać wybrany e
- Jak nakładać elementy na

Za pomocą kaskadowych
elementów oraz mieć wpł
wiada za umiejscowienie
właściwości *position*. Rozr
ne (ang. *relative*) oraz pozy

Pozycjonowanie bezwzg
łożenia elementu w stosu
go. Wykorzystuje się w tyn
którym należy przypisać k
odpowiadające odległości
elementu nadrzędnego.

```
selector {
    position: absolute;
    left: wartość;
    right: wartość;
    top: wartość;
    bottom: wartość;
}
```

Pozycjonować można d
pozycjonować obrazek (cs
katora, a następnie wprow

Listing 19.1 (HTML)

```
<img src="css.png"
```

Dla lepszego efektu wa
ściwości *margin* i *padding* n
strony (*kratka.gif*), pozwala
li). Obrazek otrzymał nast
pozycjonowanie absolutne
krawędzi okna. Końcowy e

) i list. 18.6 (CSS)

acznika, wystarczy utworzyć

o jak w poprzednim przypad-
ego znacznika w dokumencie.

sz. W celu sformatowania po-
orzystaj identyfikatory.
twórz trzy klasy: przymiotnik,
e formatowanie (np. przymiot-
dnich części mowy w zapropo-

19 Pozycjonowanie elementów

ZAGADNIENIA

- **Co to jest pozycjonowanie?**
- **Jakie są rodzaje pozycjonowania?**
- **Jak umieszczać wybrany element w dowolnym miejscu na stronie?**
- **Jak nakładać elementy na siebie?**

Za pomocą kaskadowych arkuszy stylów można manipulować wyglądem poszczególnych elementów oraz mieć wpływ na ich dokładne położenie na stronie. **Pozycjonowanie** odpowiada za umiejscowienie elementu w dowolnie wybranym miejscu przez zastosowanie właściwości *position*. Rozróżnia się dwa rodzaje pozycjonowania: pozycjonowanie względne (ang. *relative*) oraz pozycjonowanie bezwzględne (ang. *absolute*).

Pozycjonowanie bezwzględne, nazywane również absolutnym, pozwala na ustalenie położenia elementu w stosunku do ram okna przeglądarki lub innego elementu nadrzędnego. Wykorzystuje się w tym celu dodatkowo cztery właściwości (ang. *left, right, top, bottom*), którym należy przypisać konkretne wartości (w pikselach lub innych jednostkach miary) odpowiadające odległości od lewej, prawej, dolnej lub górnej krawędzi okna lub innego elementu nadrzędnego.

```
selector {
    position: absolute;
    left: wartość;
    right: wartość;
    top: wartość;
    bottom: wartość;
}
```

Pozycjonować można dowolnie wybrany element na stronie. Jeśli na przykład chcemy pozycjonować obrazek (`css.png`), należy nadać mu określoną nazwę klasy lub identyfikatora, a następnie wprowadzić odpowiednie właściwości w kaskadowym arkuszu stylów.

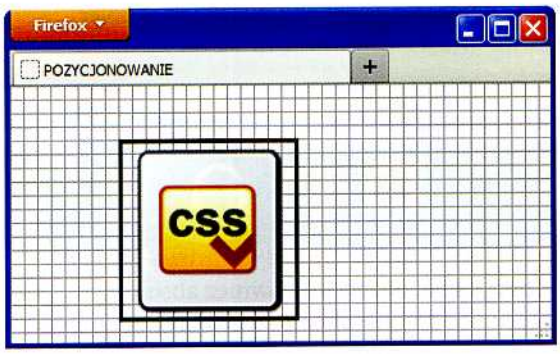
Listing 19.1 (HTML)

```
<img src="css.png" id="obrazek_css">
```

Dla lepszego efektu warto usunąć domyślne marginesy i dopełnienia, ustawiając właściwości *margin* i *padding* na *0* dla selektora *body*. Dodatkowo wprowadzono obrazkowe tło strony (*kratka.gif*), pozwalające na zobrazowanie współrzędnych (jedna kratka = 10 pikseli). Obrazek otrzymał następujące właściwości w stylach: obramowanie ciągłe czarne oraz pozycjonowanie absolutne w odległości 80 px od lewej krawędzi okna i 40 px od górnej krawędzi okna. Końcowy efekt prezentuje rys. 19.2.

Listing 19.2 (CSS)

```
body {
    margin: 0;
    padding: 0;
    background-image: url(kratka.gif);
}
#obrazek_css {
    border: solid black;
    position: absolute;
    left: 80px;
    top: 40px;
}
```



Rys. 19.1. Pozycjonowanie bezwzględne w stosunku do krawędzi okna (list. 19.1, list. 19.2)

Standardowo wyświetlany element domyślnie posiada właściwość *position* o wartości *static*. Jeżeli wartość jego właściwości *position* ulegnie zmianie na *absolute*, to staje się on elementem nadrzędnym dla znajdujących się w nim elementów. Pozycjonowane wewnątrz niego elementy będą ustawiane względem jego krawędzi.

Przykład (list. 19.3) przedstawia dwa elementy: blok **div** oraz znajdujący się wewnątrz niego obrazek. Oba elementy mają identyfikatory pozwalające na ich modyfikację za pomocą styli CSS.

Listing 19.3 (HTML)

```
<div id="kwadrat">
  <img src="css.png" id="obrazek_css">
</div>
```

Blok **div** ma następujące właściwości: wysokość 290 px, szerokość 170 px, obramowanie – czarna linia ciągła, tło obrazkowe (*kratka.gif*) oraz pozycjonowanie absolutne w odległości 10 px od lewej krawędzi okna i 10 px od górnej krawędzi okna. Natomiast obrazek ma obramowanie ciągłe czarne oraz pozycjonowanie absolutne w odległości 80 px od lewej krawędzi bloku **div** i 20 px od górnej krawędzi bloku **div**.

Listing 19.4 (CSS)

```
body {
    margin: 0;
    padding: 0;
}
div#kwadrat {
    width: 290px;
    height: 170px;
    border: solid black;
    background-image: url
    position: absolute;
    left: 10px;
    top: 10px;
}
#obrazek_css {
    border: solid black;
    position: absolute;
    left: 80px;
    top: 20px;
}
```

Rys. 19.2. Pozycjonowanie bezw

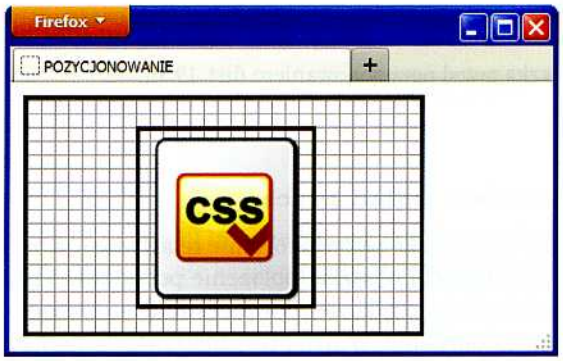
Pozycjonowanie względne

położenia elementu w stosu
celu dodatkowo cztery właś
kretne wartości (w pikselach
w lewo, prawo, w dół lub do

```
selector {
    position: relative;
    left: wartość;
    right: wartość;
    top: wartość;
    bottom: wartość;
}
```

Przykład (list. 19.5, rys. 1
W pierwszej komórce znajdu
pozycjonowania relatywnego
wanie ma czarne ciągłe obra

```
    padding: 0;
}
div#kwadrat {
    width: 290px;
    height: 170px;
    border: solid black;
    background-image: url(kratka.gif);
    position: absolute;
    left: 10px;
    top: 10px;
}
#obrazek_css {
    border: solid black;
    position: absolute;
    left: 80px;
    top: 20px;
}
```



Rys. 19.2. Pozycjonowanie bezwzględne w stosunku do krawędzi bloku **div** (list. 19.3, list. 19.4)

Pozycjonowanie względne (*relative*), nazywane również relatywnym, pozwala na zmianę położenia elementu w stosunku do jego położenia początkowego. Wykorzystuje się w tym celu dodatkowo cztery właściwości (*left, right, top, bottom*), którym należy przypisać konkretne wartości (w pikselach lub innych jednostkach miary) odpowiadające przesunięciu w lewo, prawo, w dół lub do góry.

```
selector {
    position: relative;
    left: wartość;
    right: wartość;
    top: wartość;
    bottom: wartość;
}
```

Przykład (list. 19.5, rys. 19.3) przedstawia tabelę, która składa się z dwóch komórek. W pierwszej komórce znajduje się tekst informujący, że jest to przykład bez zastosowania pozycjonowania relatywnego. W drugiej komórce jest obrazek, który jako jedyne formatowanie ma czarne ciągłe obramowanie.

Listing 19.5 (HTML)

```
<table border>
<tr>
<td width="150"> brak pozycjonowania relatywnego </td>
<td width="150"><img src="css.png" id="obrazek_css"></td>
</tr>
</table>
```

Rys. 19.3. Przykład obrazka przed pozycjonowaniem (list. 19.5)

Listing 19.6 (CSS)

```
#obrazek_xss{border: solid black;
```

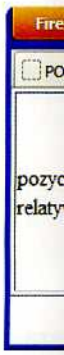
Kiedy wprowadzimy dla obrazka pozycjonowanie relatywne (*css.png*), zostaje on przesunięty o 20 px od prawej krawędzi obrazka (położenie początkowe) oraz o 20 px od górnej krawędzi obrazka (położenie początkowe). Po przesunięciu obrazek przysłoni wszystkie elementy rozmieszczone domyślnie. W tym przypadku przysłonięty został fragment tabeli.

Listing 19.7 (HTML)

```
<table border>
<tr>
<td width="150"> pozycjonowanie relatywne </td>
<td width="150"><img src="css.png" id="obrazek_css"></td>
</tr>
</table>
```

Listing 19.8 (CSS)

```
body {
   margin: 0;
   padding: 0;
}
#obrazek_css {
   border: solid black;
   position: relative;
   right: 20px;
   top: 20px;
}
```



Rys. 19.4. Przykład obrazka

Jeśli zmienimy położe
siebie nachodzić, a nawet
w jakiej kolejności elemer
jąca wartości liczbowe po
spodzie. Element o najwy

```
selektor {
   position: rodzaj;
   parametry;
   z-index: numer;
}
```

Przykład (list. 19.9, list 1
zostanie wyświetlony dwu

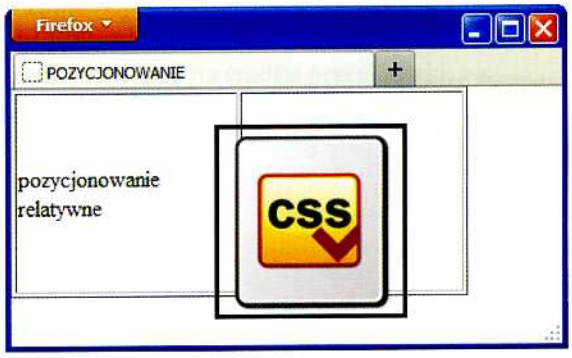
Listing 19.9 (HTML)

```
<p id="akapit"> A
<img src="css.png
<img src="css.png
```

Na samym spodzie (
zek_css oraz czarne ob
i lewej krawędzi okna pr
identyfikator **obrazek2**
od górnej i lewej krawęd
napis AKAPIT P w kolor
od lewej krawędzi o 70 p

Listing 19.10 (CSS)

```
body {
   margin: 0;
   padding: 0;
}
#obrazek_css {
   border: solid black;
   position: absolute;
```



Rys. 19.4. Przykład obrazka z pozycjonowaniem względnym (list. 19.7, list. 19.8)

Jeśli zmienimy położenie kilku elementów na stronie, może się zdarzyć, że będą one na siebie nachodzić, a nawet jeden element przysłoni całkowicie inny. Wówczas warto ustalić, w jakiej kolejności elementy będą się nakładać. Służy do tego właściwość *z-index*, przyjmująca wartości liczbowe począwszy od 1, która oznacza element znajdujący się na samym spodzie. Element o najwyższej wartości *z-index* będzie zakrywał wszystkie pozostałe.

```
selektor {
   position: rodzaj;
   parametry;
   z-index: numer;
}
```

Przykład (list. 19.9, list 19.10, rys. 19.10) przedstawia akapit oraz obrazek (**css.png**), który zostanie wyświetlony dwukrotnie, ale każdemu z tych dwu przypisano inny identyfikator.

Listing 19.9 (HTML)

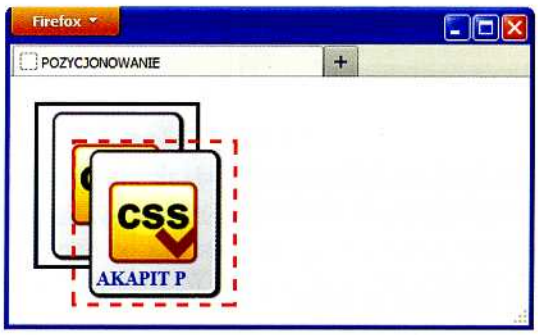
```
<p id="akapit"> AKAPIT P </p>
<img src="css.png" id="obrazek_css">
<img src="css.png" id="obrazek2_css">
```

Na samym spodzie (*z-index:1*) znajduje się obrazek mający identyfikator **obrazek_css** oraz czarne obramowanie. Został on umieszczony w odległości 20 px od górnej i lewej krawędzi okna przeglądarki. Nałożono na niego obrazek (*z-index:2*) posiadający identyfikator **obrazek2_css** oraz obramowanie czerwone kreskowane. Oddalony jest od górnej i lewej krawędzi okna o 50 px. Na samym wierzchu (*z-index:3*) umieszczono napis AKAPIT P w kolorze niebieskim, oddalony od górnej krawędzi okna o 150 px oraz od lewej krawędezi o 70 px.

Listing 19.10 (CSS)

```
body {
   margin: 0;
   padding: 0;
}
#obrazek_css {
   border: solid black;
   position: absolute;
```

```
    left: 20px;
    top: 20px;
    z-index: 1;
}
#obrazek2_css {
    border: dashed red;
    position: absolute;
    left: 50px;
    top: 50px;
    z-index: 2;
}
#akapit {
    color: blue;
    font-weight: 800;
    position: absolute;
    left: 70px;
    top: 150px;
    z-index: 3;
}
```



Rys. 19.5. Przykład nakładania elementów (list. 19.9, list. 19.10)

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Korzystając z list. 19.11 i list 19.12, utwórz czerwony kwadracik.

Listing 19.11 (HTML)

```
<div class="czerwony"> czerwony </div>
```

Listing 19.12 (CSS)

```
div.czerwony {
border-style:solid;
background-color:red;
width:100px;
height:100px;
}
```

Wykorzystując dowolny
pujących układach:
a) Umieść kwadracik ce

Rys. 19.6. Pozycjonowanie 1

b) Wprowadź cztery dod
szy układ.

Rys. 19.7. Pozycjonowanie 2

c) Zmień rozmiar kwadr

Rys. 19.8. Pozycjonowanie 3

SPRAWDŹ SW

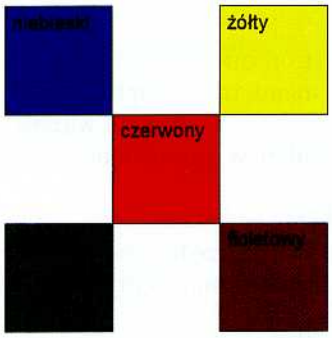
1. Co to jest pozycjonowani
2. Co to jest pozycjonowani
3. Jak ukryć dowolny eleme

Wykorzystując dowolny rodzaj pozycjonowania, umieść utworzony kwadracik w następujących układach:

a) Umieść kwadracik centralnie na środku przeglądarki.

Rys. 19.6. Pozycjonowanie 1 – kwadrat wyśrodkowany względem okna

b) Wprowadź cztery dodatkowe kwadraciki w odpowiednich kolorach i wykonaj poniższy układ.



Rys. 19.7. Pozycjonowanie 2 – kwadraty stykające się narożnikami

c) Zmień rozmiar kwadratów i umieść jeden na drugim w odpowiedniej kolejności.

Rys. 19.8. Pozycjonowanie 3 – kwadraty nachodzące na siebie

SPRAWDŹ SWOJĄ WIEDZĘ

1. Co to jest pozycjonowanie absolutne?
2. Co to jest pozycjonowanie relatywne?
3. Jak ukryć dowolny element na stronie?

20 Przykładowe układy stron

ZAGADNIENIA

- Na co zwrócić uwagę, tworząc szablon oparty na elementach `div`?
- Jak formatować elementy blokowe strony?
- Jak stworzyć układ strony o stałej szerokości?
- Jak stworzyć układ strony o zmiennej szerokości?

Kolejnym krokiem tworzenia stron internetowych jest przejście do rozmieszczenia elementów głównych (nagłówek, menu, treść, stopka) za pomocą bloków **div**. Jest to obecnie najpopularniejszy sposób tworzenia szkieletu witryny. Dla ułatwienia warto przygotować sobie kilka prostych szablonów strony, które można wykorzystać do przyszłych projektów.

Blok **div** w dokumencie HTML wykorzystywany jest do podziału strony w poziomie. Style CSS dostarczają odpowiednich narzędzi (ang. *float, clear, width, height*), które pozwalają na stworzenie dowolnego układu strony. Witryna taka jest prosta w budowie, przejrzysta i można ją szybko zmodyfikować.

Pierwszym etapem budowy jest stworzenie zagnieżdżonego układu znaczników **div**. Główną rolę pełni zewnętrzny znacznik **div**, tworzący coś w rodzaju pudełka, do którego wrzucamy wszystkie pozostałe elementy (logo, menu, treść). Każdy ze znaczników **div** otrzymuje unikalny identyfikator **id**, który charakteryzuje przydzielony mu fragment strony. Po określeniu, z ilu bloków będzie składał się dany układ, można przystąpić do ustalenia ich szerokości, wysokości oraz innych cech formatowania.

Przykład pierwszy (rys. 20.1) stanowi jeden z najprostszych układów strony. Składa się z zewnętrznego pudełka, w którym umieszczono bloki: logo, menu, treść strony oraz stopkę. Wszystkie elementy występujące w pudełku zachowują układ poziomy i mają stałe wymiary.

Listing 20.1

```
<div id="pudelko">
<div id="logo">LOGO</div>
<div id="menu">
<a href="">link</a>
<a href="">link</a>
<a href="">link</a>
<a href="">link</a>
</div>
<div id="tresc">
treść strony
</div>
```

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

Tworząc style, należy
szczególnych elementów
okna przeglądarki czy z
marginesy i dopełnienie
glądarki, nie pozostawia
wymiary dla poszczególn
dzono różne odcienie sz

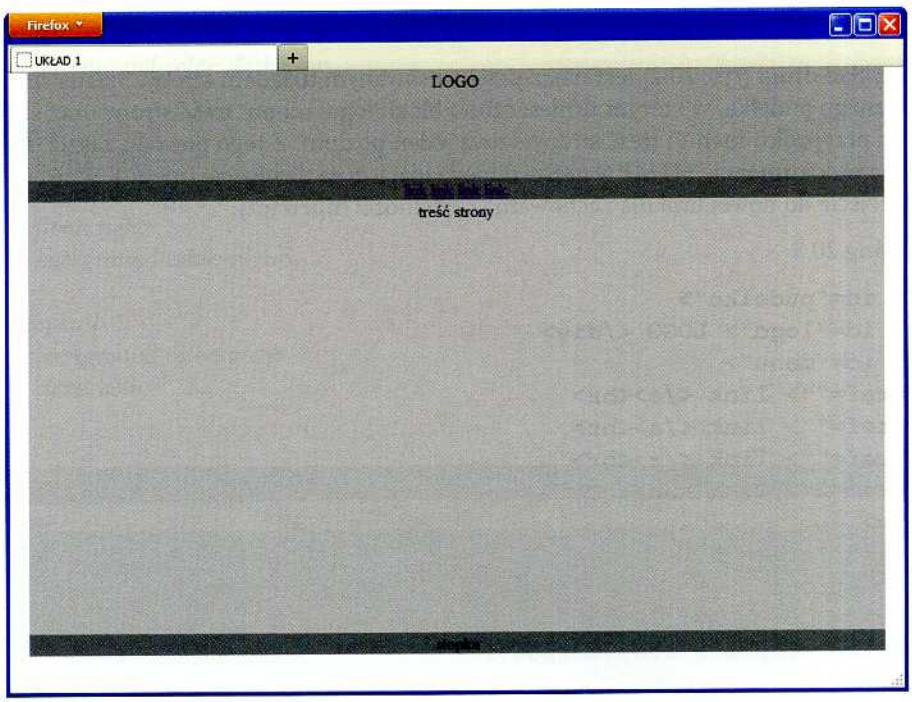
Rys. 20.1. Układ 1

Listing 20.2

```
body {
    margin:0;
    padding:0;
    text-align: center;
}
#pudelko {
    margin: auto;
    width: 800px;
}
#logo {
    height: 100px;
```

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

Tworząc style, należy pamiętać, że jest to układ o stałej szerokości, czyli szerokość poszczególnych elementów układu nie zależy od rozdzielczości ekranu monitora, wielkości okna przeglądarki czy zawartości. W przypadku każdego układu usuwane są domyślne marginesy i dopełnienie dla **body**. Powoduje to, że strona przylega do krawędzi przeglądarki, nie pozostawiając wolnej przestrzeni. Następnie wystarczy ustalić odpowiednie wymiary dla poszczególnych elementów. Dla lepszego zaprezentowania układu wprowadzono różne odcienie szarości tła elementów.



Rys. 20.1. Układ 1

Listing 20.2

```
body {
    margin:0;
    padding:0;
    text-align: center;
}
#pudelko {
    margin: auto;
    width: 800px;
}
#logo {
    height: 100px;
```

```
    background-color: darkgray;
}
#menu {
    background-color: gray;
}
#tresc {
    height:400px;
    background-color: silver;
}
#stopka {
    background-color: gray;
}
```

Przykład drugi (rys. 20.2) jest najczęściej stosowanym układem strony. Składa się z zewnętrznego pudełka, w którym umieszczono bloki: logo, menu, treść strony oraz stopkę. W tym przypadku menu i treść strony dzielą jeden poziom. Z tego powodu zaszła drobna zmiana w dokumencie HTML w części odpowiadającej za nawigację (menu). Wprowadzono przejście do nowej linii dla zachowania czytelności hiperłączy.

Listing 20.3

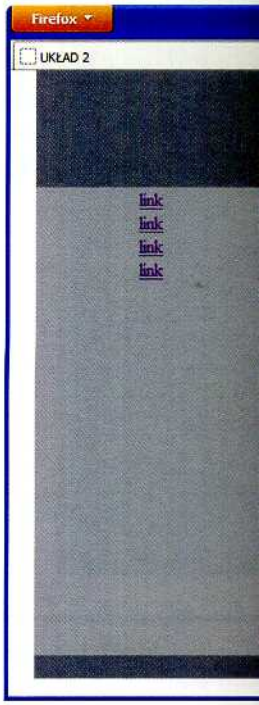
```
<div id="pudelko">
<div id="logo"> LOGO </div>
<div id="menu">
<a href=""> link </a><br>
<a href=""> link </a><br>
<a href=""> link </a><br>
<a href=""> link </a><br>
</div>
<div id="tresc">
treść strony
</div>
<div id="stopka"> stopka </div>
</div>
```

Układ ma stałą szerokość. Ze względu na to, że menu i treść strony znajdują się na jednym poziomie, dosunięto je odpowiednio do lewej i prawej krawędzi pudełka, w którym się znajdują. Służy do tego właściwość *float* o wartościach *left* i *right*. Elementy, które mają cechę *float*, nazywamy **pływającymi**. Zastosowanie takich elementów wymusza wprowadzenie właściwości *clear*. Pozwala ona ustalić pozycje elementów sąsiadujących względem elementu pływającego. Określa, które krawędzie elementu **następującego** po elemencie pływającym nie będą do niego przylegać. Wartość *bath* powoduje, że żaden bok nie przylega do poprzedzającego elementu pływającego.

Listing 20.4

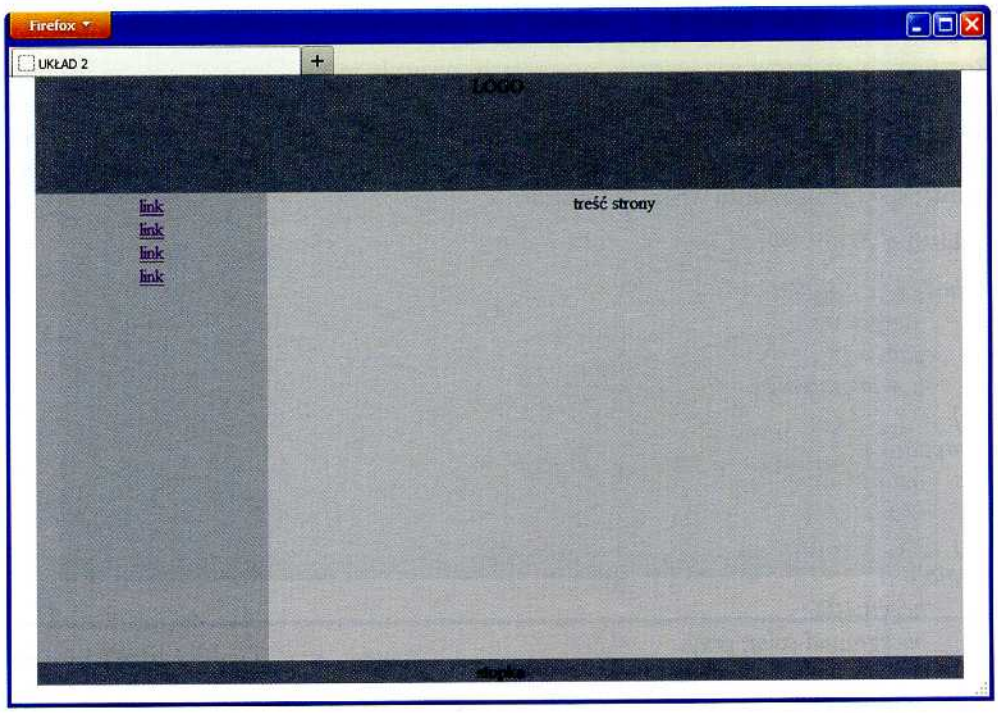
```
body {
    margin: 0;
    padding: 0;
    text-align: center;
}
```

```
#pudelko {
    margin: auto;
    width: 800px;
}
#logo {
    height: 100px;
    background-color:
}
#menu {
    width: 200px;
    height: 400px;
    float: left;
    background-color:
}
#tresc {
    width: 600px;
    height: 400px;
    float: right;
    background-color:
}
#stopka {
    background-color:
    clear: both;
}
```



Rys. 20.2. Układ 2

```
#pudelko {
    margin: auto;
    width: 800px;
}
#logo {
    height: 100px;
    background-color: gray;
}
#menu {
    width: 200px;
    height: 400px;
    float: left;
    background-color: darkgray;
}
#tresc {
    width: 600px;
    height: 400px;
    float: right;
    background-color: silver;
}
#stopka {
    background-color: gray;
    clear: both;
}
```



Rys. 20.2. Układ 2

m strony. Składa się z ze-
treść strony oraz stopkę.
go powodu zaszła drobna
ację (menu). Wprowadzo-
y.

strony znajdują się na jed-
awędzi pudełka, w którym
ight. Elementy, które mają
nentów wymusza wprowa-
w sąsiadujących względem
stępującego po elemencie
uje, że żaden bok nie przy-

Przykład trzeci (rys. 20.3) składa się z zewnętrznego pudełka, w którym umieszczono bloki: logo, menu lewe, treść strony, menu prawe oraz stopkę. W tym przypadku wprowadzono dodatkowe menu po prawej stronie. Menu prawe, treść strony oraz menu lewe znajdują się na jednym poziomie.

Listing 20.5

```
<div id="pudelko">
<div id="logo"> LOGO </div>
<div id="menulewe">
<a href=""> link </a><br>
<a href=""> link </a><br>
<a href=""> link </a><br>
<a href=""> link </a><br>
</div>
<div id="tresc">
treść strony
</div>
<div id="menuprawe">
<a href=""> link </a><br>
<a href=""> link </a><br>
<a href=""> link </a><br>
<a href=""> link </a><br>
</div>
<div id="stopka"> stopka </div>
</div>
```

Zastosowano tu układ zmiennej szerokości, zwany również układem pływającym. Rozmiary elementów wyrażono w procentach, pozwala to na dostosowanie się elementów do bieżącego rozmiaru okna. Ze względu na to, że trzy elementy znajdują się na jednym poziomie, menu lewe i treść strony dosunięto do lewej krawędzi pudełka (ang. *float: left*), a menu prawe – do krawędzi prawej (ang. *float: right*). W takim przypadku należy pamiętać o zastosowaniu właściwości *clear*.

Listing 20.6

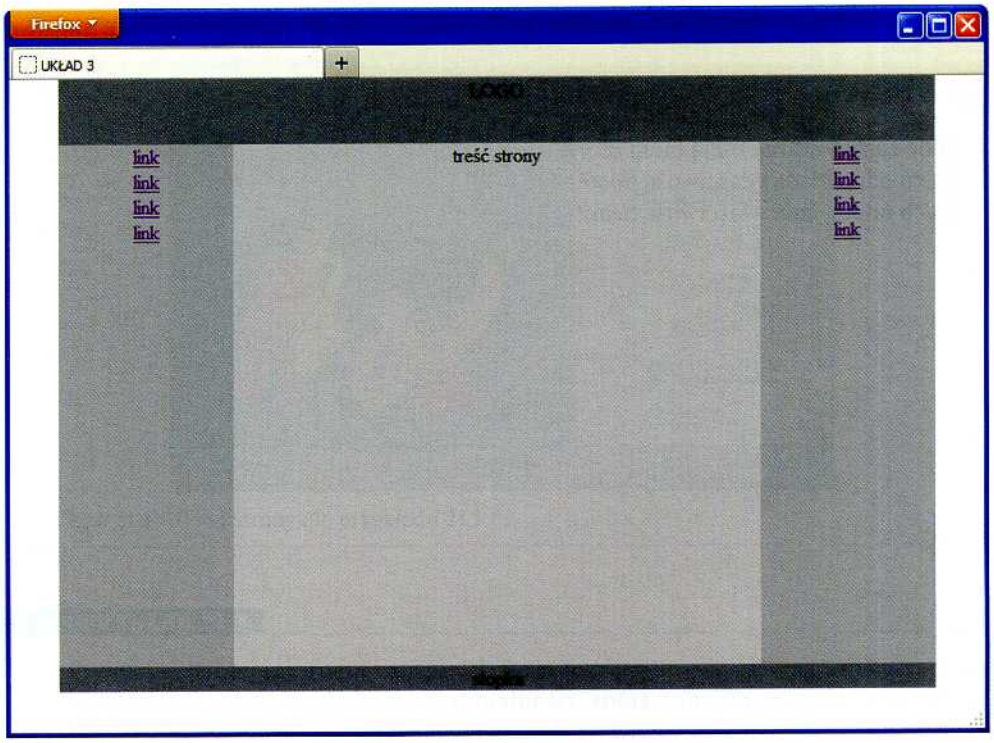
```
body {
    margin:0;
    padding:0;
    text-align:center;
}
#pudelko {
    margin:auto;
    width:90%;
}
#logo {
    height:10%;
    background-color: gray;
}
#menulewe {
    width: 20%;
    height: 80%;
    float: left;
    background-color:
}
#tresc {
    width: 60%;
    height: 80%;
    float: left;
    background-color:
}
#menuprawe {
    width: 20%;
    height: 80%;
    float: right;
    background-color:
}
#stopka {
    background-color:
    clear: both;
}
```

Rys. 20.3. Układ 3

w którym umieszczono
W tym przypadku wpro-
strony oraz menu lewe

eż układem pływającym.
dostosowanie się elemen-
menty znajdują się na jed-
wędzi pudełka (ang. *float:*
takim przypadku należy

```
    height: 80%;
    float: left;
    background-color: darkgray;
}
#tresc {
    width: 60%;
    height: 80%;
    float: left;
    background-color: silver;
}
#menuprawe {
    width: 20%;
    height: 80%;
    float: right;
    background-color: darkgray;
}
#stopka {
    background-color: gray;
    clear: both;
}
```

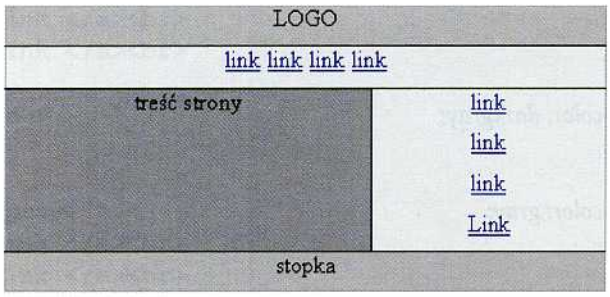


Rys. 20.3. Układ 3

Stosowanie układów stron zbudowanych na blokach **div** daje wiele korzyści. Przede wszystkim struktura kodu jest bardzo przejrzysta, a co za tym idzie – można ją w prosty i szybki sposób modyfikować. Użycie znaczników jest znacznie mniejsze, co ma wpływ na rozmiar strony.

SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Zbuduj poniższy układ opierający się na elementach **div** (rys. 20.4). Zwróć uwagę na dodatkowe obramowania i tło.



Rys. 20.4. Układ strony do zadania

SPRAWDŹ SWOJĄ WIEDZĘ

1. Jakie zastosowanie mają bloki **div**?
2. Za co odpowiada właściwość *float*?
3. Za co odpowiada właściwość *clear*?

21

Po zapoznaniu się z ele
przystąpić do tworzeni
wych. Staraj się wykorz
dy stron opierające się
dostosowaniu szaty gra

PRZYKŁAD 21.1

Przygotuj stronę
zycznemu. Strona
stopka. Każda poc
suj kolorystykę li
ne (film – teledys

Rys. 21.1. Układ stro

PRZYKŁAD 21.2

Przygotuj stronę
lu. Portal.pl skład
Przykładowe info
zachować zapreze
zujące portal i do
kład, i zrealizuj sv