



DO NOWEJ
PODSTAWY PROGRAMOWEJ

Część 1

Kwalifikacja E.14

Tworzenie stron internetowych



Podręcznik do nauki zawodu **technik informatyk**

Jolanta Pokorska



Helion Edukacja

Podręcznik dopuszczony do użytku szkolnego przez ministra właściwego do spraw oświaty i wychowania i wpisany do wykazu podręczników przeznaczonych do kształcenia w zawodzie technik informatyk, na podstawie opinii rzeczoznawców: mgr Marii Dziurzyńskiej-Ścibior, mgr Elżbiety Leszczyńskiej, dr. inż. Stanisława Szablowskiego.

Nazwa kwalifikacji: Kwalifikacja E-14. Część 1. Tworzenie stron internetowych.

Typ szkoły: technikum, szkoła policealna, kurs kwalifikacyjny.

Rok dopuszczenia 2013.

Numer ewidencyjny w wykazie: 13/2013

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

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

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

Redaktorzy prowadzący: Marcin Borecki

Projekt okładki: Maciej Pasek

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie?e14tei>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-246-5102-3

Copyright © Helion 2013

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

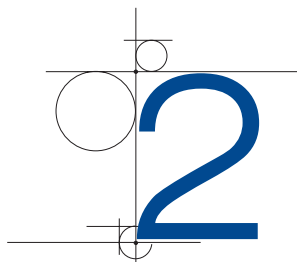
- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Wstęp	7
Część 1. Tworzenie stron internetowych	11
Rozdział 1. Język HTML	13
1.1. Wprowadzenie	13
1.2. Języki HTML, XML, XHTML	15
1.3. Kodowanie polskich znaków	20
1.4. Edytory tekstu	21
1.5. Struktura strony WWW	23
1.6. Tworzenie i formatowanie dokumentu HTML	27
Rozdział 2. Kaskadowe arkusze stylów (CSS)	76
2.1. Wstawianie stylów	77
2.2. Składnia języka CSS	84
2.3. Selektory	87
2.4. Właściwości elementów	99
2.5. Model blokowy CSS	112
2.6. Inne elementy	122
2.7. Menu w języku CSS	139
2.8. Zasady projektowania szablonu strony internetowej	145
Rozdział 3. Edytory WYSIWYG	160
3.1. Edytory stron WWW	160
3.2. Adobe Dreamweaver	162
Rozdział 4. Zasady projektowania strony internetowej	185
4.1. Projektowanie stron	185

Rozdział 5.	Grafika na stronie internetowej.	192
5.1.	Rodzaje grafiki	192
5.2.	Sposoby zapisu obrazu cyfrowego	193
5.3.	Metody pozyskiwania obrazów	195
5.4.	Modele barw	196
5.5.	Formaty plików graficznych	199
5.6.	Ogólne zasady tworzenia grafiki na potrzeby strony internetowej.	200
5.7.	Modyfikowanie grafiki na potrzeby strony internetowej	201
5.8.	Tworzenie grafiki na potrzeby strony internetowej	208
5.9.	Tło strony internetowej	214
5.10.	Ikona i logo jako elementy strony internetowej	220
5.11.	Nawigacja	222
5.12.	Mapowanie obrazu	224
5.13.	Zapisywanie obrazów na potrzeby witryny internetowej	227
Rozdział 6.	Graficzny projekt strony internetowej.	232
6.1.	Opracowanie projektu	232
6.2.	Projektowanie strony	234
6.3.	Dobór palety barw	238
Rozdział 7.	Animacja na stronie internetowej.	245
7.1.	Metody tworzenia animacji.	245
7.2.	Tworzenie animacji na potrzeby strony internetowej	246
Rozdział 8.	Dźwięk i wideo	271
8.1.	Dźwięk	271
8.2.	Edytowanie i obróbka dźwięku.	274
8.3.	Pliki wideo.	281
8.4.	Planowanie filmu.	283
8.5.	Tworzenie filmu w programie Adobe Premiere.	284

Rozdział 9.	Testowanie i publikowanie strony	293
9.1.	Sprawdzanie poprawności strony	293
9.2.	Walidacja strony	294
9.3.	Testowanie strony w przeglądarkach	297
9.4.	Publikowanie strony internetowej	298
9.5.	Testy szybkości wczytywania strony	303
9.6.	Testy funkcjonalności strony internetowej	306
9.7.	Optymalizacja strony internetowej	308
9.8.	Pozycjonowanie strony	308
Skorowidz		313



Kaskadowe arkusze stylów (CSS)

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

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

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

Cechy języka CSS:

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

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

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

2.1. Wstawianie stylów

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

W podstawowej składni kaskadowych arkuszy stylów jest kilka stałych elementów:

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

- *selektor* — określa, do jakich znaczników języka HTML odnosi się dalsza część definicji stylu. Selektorem może być dowolny znacznik.
- *właściwość* — określa, jaki atrybut znacznika będzie ustawiany.
- *wartość* — określa wartość atrybutu.

Przykład 2.1

```
body {background-color: #dfdfff; }
```

Przykład pokazuje ustawienie koloru tła strony. To samo polecenie w języku HTML miałoby postać:

```
<body bgcolor="#DFDFFF">
```

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

W dokumencie HTML istnieje kilka sposobów na dołączanie stylów CSS:

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

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

2.1.1. Styl lokalny (w linii)

Korzystając ze stylu lokalnego, można zdefiniować formatowanie pojedynczego elementu strony. Taki styl jest definiowany w tej samej linii, w której znajduje się element formatowany, dlatego nazywany jest on również stylem w linii (ang. *inline*). Umieszczany jest bezpośrednio w dokumencie HTML.

Przykład 2.2

```
<html>
<head>
<title> Kaskadowe arkusze stylów</title>
```



```

</head>
<body style="background-color: #DFDFDF;">
<p style="color: red">Definiowanie stylów</p>
</body>
</html>

```

Znacznik — rozciąganie stylu

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

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

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

Przykład 2.3

```

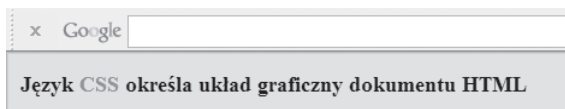
<html>
<head>
<title> Kaskadowe arkusze stylów</title>
</head>
<body style="background-color: #DFDFDF;">
<h3>Język <span style="color: red">CSS</span> określa układ graficzny
dokumentu HTML</h3>
</body>
</html>

```

Wynik interpretacji kodu został pokazany na rysunku 2.1.

Rysunek 2.1.

Zastosowanie znacznika
 do fragmentu tekstu



Znacznik div — wydzielone bloki

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

Metoda definiowania stylu dla bloków jest podobna jak w przypadku znacznika . Ustala się go za pomocą stylu lokalnego.

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

Przykład 2.4

```

<html>
<head>
<title> Bloki div</title>

```

```

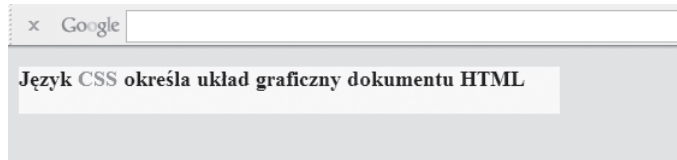
</head>
<body style="background-color: #DFDFDF;">
<div style="width: 500px; height: 50px; background-color: yellow;">
<h3>Język <span style="color: red">CSS</span> określa układ graficzny
dokumentu HTML</h3></div>
</body>
</html>

```

Wynik interpretacji kodu został pokazany na rysunku 2.2.

Rysunek 2.2.

Definiowanie
bloku <div>



2.1.2. Wewnętrzny arkusz stylów

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

Przykład 2.5

```

<html>
<head>
<title> Kaskadowe arkusze stylów</title>
<style type="text/css">
p { color: blue; }
</style>
</head>
<body>
<p>Definiowanie stylów</p>
<p>Wewnętrzny arkusz stylów</p>
</body>
</html>

```

Przykład 2.6

```

<html>
<head>
<title> Wewnętrzny arkusz stylów</title>

```

```

<style type="text/css">
body {background-color: yellow;}
h2 {font-size: 32pt;}
h3 {color: green; font-size: 24pt;}
p {color: blue; text-align: center; }
</style>
</head>
<body>
<h2> Wewnętrzne arkusze stylów</h2>
<h3>Definiowanie stylów</h3>
<p>Arkusz stylów</p>
</body>
</html>

```

Zadanie 2.1

Sprawdź działanie kodu podanego w przykładzie 2.6.

Zadanie 2.2

Używając stylu lokalnego, zdefiniuj rodzaj, kolor i rozmiar czcionki dla tekstu wyświetlanego na stronie. To samo wykonaj, używając wewnętrznego arkusza stylów. Kiedy warto zastosować styl lokalny, a kiedy wewnętrzny arkusz stylów?

2.1.3. Zewnętrzny arkusz stylów

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

Odnośnik umieszczony w dokumencie HTML powinien mieć postać:

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

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

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

Przykład 2.7

```

<html>
<head>
<title> Kaskadowe arkusze stylów</title>
<link rel="stylesheet" type="text/css" href="arkusz.css">
</head>

```

```

<body>
<p>Definiowanie stylów</p>
<p>Wewnętrzny arkusz stylów</p>
</body>
</html>

```

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

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

Zadanie 2.3

Utwórz plik z rozszerzeniem `.css`. Zdefiniuj w nim styl tekstu wyświetlanego na stronie oraz styl tła strony. Dołącz go do dokumentu HTML opisującego przykładową stronę internetową.

2.1.4. Import arkusza stylów

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

Polecenie importowania arkusza ma postać:

```

<style type="text/css">
/* <![CDATA[ */
@import url (adres importowanego arkusza stylów.css);
/* ]]> */
</style>

```

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

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

2.1.5. Kaskadowość arkuszy stylów

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

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

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

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

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

Przykład 2.8

Użycie polecenia `!important` w definicji stylu:

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

Fragment kodu:

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

Przykład 2.9

```
<html>
<head>
<title>Arkusz stylów</title>
```

```

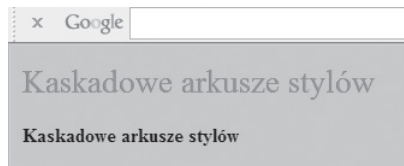
<style type="text/css">
body {background-color: #c0c0c0;}
p {font-size: 20pt !important; color: green;}
</style>
</head>
<body>
<p style= "font-size: 12pt; color: red"> Kaskadowe arkusze stylów</p>
<h2 style= "font-size: 12pt;"> Kaskadowe arkusze stylów</h2>
</body>
</html>

```

Na rysunku 2.3 został pokazany efekt działania polecenia `!important`. Dla znacznika `<p>` zastosowano kolor czcionki zdefiniowany za pomocą stylu lokalnego, natomiast rozmiar czcionki pochodzi z wewnętrznego arkusza stylów.

Rysunek 2.3.

Efekt działania
polecenia `!important`



Zadanie 2.4

W zewnętrznym arkuszu stylów został zdefiniowany znacznik `h2`:

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

W wewnętrznym arkuszu stylów również został zdefiniowany znacznik `h2`:

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

Jeżeli strona ma powiązanie z zewnętrznym arkuszem stylów, to jakie właściwości będzie miał znacznik `h2`? Sprawdź swoją odpowiedź, pisząc kod przykładowej strony internetowej.

2.1.6. Dziedziczenie

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

Przykład 2.10

```

<html>
<head>
<title> Wewnętrzny arkusz stylów</title>
<style type="text/css">
body {background-color: yellow;}
p {font-size: 20pt; color: green;}

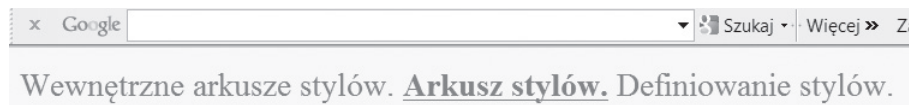
```

```

</style>
</head>
<body>
<p> Wewnętrzne arkusze stylów. <u><b>Arkusz stylów.</b></u> Definiowanie
stylów. </p>
</body>
</html>

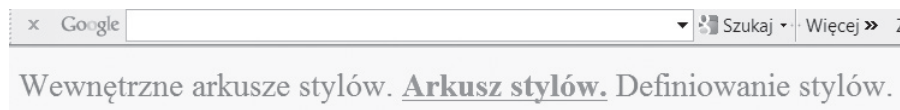
```

W podanym przykładzie dla akapitu `<p>` został zdefiniowany kolor zielony. Wewnątrz akapitu wystąpiło pogrubienie i podkreślenie, którym nie nadano żadnego stylu, więc dziedziczą go one po akapicie (rysunek 2.4).



Rysunek 2.4. Dziedziczenie stylu dla tekstu „Arkusz stylów.”

Na rysunku 2.5 widzimy efekt zdefiniowania atrybutu koloru dla fragmentu tekstu wewnątrz akapitu. Ten fragment nie odziedziczył stylu po akapicie.



Rysunek 2.5. Brak dziedziczenia stylu dla tekstu „Arkusz stylów.”

Zadanie 2.5

Kod podany w przykładzie 2.10 zmodyfikuj tak, aby uzyskać wynik pokazany na rysunku 2.5.

2.2. Składnia języka CSS

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

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

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

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

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

W języku CSS parametry formatowania definiuje się za pomocą **reguł stylów**. Każda reguła odnosi się do określonego elementu i składa się z dwóch części: **selektora**

i **deklaracji**. Selektor określa, jakiego elementu ma dotyczyć formatowanie. Deklaracja jest umieszczana w nawiasach klamrowych { . . . } i definiuje formatowanie elementu. Deklaracja musi zawierać przynajmniej jedną właściwość i przypisaną do niej wartość. Poza tym jednym ograniczeniem deklarowanych właściwości może być dowolna liczba. Deklaracja zawsze kończy się średnikiem.

Podstawowa składnia wygląda następująco:

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

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

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

W definicji stylu można grupować selektory i przypisywać im te same wartości:

```
selektor1, selektor2, selektor3 {właściwość: wartość;}
```

Przykład 2.11

```
p, h3, h4 {font-family: calibri; color: blue;}
```

Do dokumentu HTML można dołączyć arkusz stylów zapisany w pliku [arkusz.css](#) z podaną przykładową zawartością.

Przykład 2.12

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

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

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

Przykład 2.13

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

2.2.1. Wartości i jednostki

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

Mogą to być:

Liczby — całkowite i rzeczywiste. Liczba całkowita składa się z cyfr od 0 do 9. Liczba rzeczywista może zawierać część całkowitą i część ułamkową. Część ułamkowa zapisana jest po kropce.

Procenty. Wartość procentowa jest zawsze określana względem innej wartości. Może zostać użyta na przykład do określenia wielkości bloków. Podana w procentach wielkość bloku będzie się zmieniała wraz ze zmianą rozmiaru bloku, który go zawiera (rozmiar ten zwykle zależy od rozmiaru okna przeglądarki).

Słowa kluczowe. Nie zawsze wartości wyrażone za pomocą słów kluczowych są poprawnie interpretowane przez przeglądarki.

Jednostki długości. Mogą być zapisywane jako wartości bezwzględne i wartości względne.

Wartości bezwzględne: `in` (cal, 1 in = 2,54 cm), `cm` (centymetr), `mm` (milimetr), `pt` (punkty, 1 cal = 72 punkty), `pc` (pica, 1 pc = 12 pt), `px` (piksele, 1 px = 0,75 pt). Jednostki bezwzględne są rzadko stosowane. Należy ich używać wtedy, gdy chcemy, aby wybrany element miał taki sam rozmiar niezależnie od rozdzielczości monitora.

Wartości względne: `em` (1 em jest wielkością rozmiaru czcionki; wartość 0,1 em jest najmniejszą wartością, którą przeglądarki są w stanie rozpoznać), `ex` (wysokość małej litery *x* w użytej czcionce). Jednostki względne są stosowane, gdy odwołujemy się do rozmiaru innego elementu.

URL — np. `url(zdjecia/obraz.jpg)`. Błędem jest wstawienie spacji przed nawiasem otwierającym.

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

Kolory. Stosuje się je na takich samych zasadach jak w języku HTML. W wersji CSS 2.2 oprócz 16 podstawowych kolorów został zdefiniowany dodatkowy kolor podstawowy `orange` (#FFA500) — pomarańczowy. W CSS 3 zostało zdefiniowanych dodatkowo 130 kolorów rozszerzonych.

Przykłady podawania wartości kolorów w CSS 3:

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

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

2. *#RRGGBB* — podawanie wartości każdej składowej kolorów podstawowych (RR — czerwony, GG — zielony, BB — niebieski). Wszystkie liczby są dwucyfrowe i zapisane w systemie szesnastkowym, np. `#FF00CC`.

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

3. *#RGB* — jako składowe kolorów trzeba podawać liczby jednocyfrowe w systemie szesnastkowym, np. `#F0C`.

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

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

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

5. `rgb (R%, G%, B%)` — można podać oddzielnie każdą składową koloru w postaci wartości procentowej.
6. *28 kolorów systemowych* — kolory pochodzą z systemu operacyjnego (kolor pulpitu, okien, menu). Gdy użytkownik zmieni kolory w swoim systemie, zmienia się również kolory zdefiniowane tą metodą.
7. `hsl (H, S%, L%)` — odmienny od RGB sposób opisu kolorów. *H (Hue)* określa barwę, *S (Saturation)* — nasycenie, *L (Lightness)* — jasność. Kolor powstaje przez wybranie określonej barwy (*H*), a następnie sterowanie nasyceniem (*S*) i jasnością (*L*) aż do uzyskania pożądanego efektu.

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

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

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

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



2.3. Selektory

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

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

2.3.1. Selektory elementów

Składnia dla selektorów elementów to podana już podstawowa składnia języka CSS:

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

Selektor typu

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

Przykład 2.14

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

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

Selektor uniwersalny

Selektor uniwersalny (inaczej: ogólny) to selektor pasujący do wszystkich znaczników. Jest oznaczany gwiazdką `*`.

Zamiast grupować elementy, np.:

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

można użyć selektora uniwersalnego:

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

Selektor potomka

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

Jeżeli w dokumencie HTML wewnątrz znacznika znajduje się inny znacznik, np. wewnątrz znacznika `<h2>` zostały umieszczone znaczniki `<u>` oraz `<i>`:

```
<h2> Wewnętrzne arkusze stylów. <u><i>Definiowanie stylów. </i></u>Arkusz
stylów. </h2>
```

to definicja stylu dla znacznika `<i>` będzie wyglądać następująco:

```
h2 i { font-size: 20pt; color: blue; }
```

Przykład 2.15

```
<html>
<head>
<title> Selektor potomka</title>
<style type="text/css">
body {background-color: grey;}
h2 i {font-size: 20pt; color: blue;}
</style>
</head>
<body>
<h2> Wewnętrzne arkusze stylów. <u><i> Definiowanie stylów. </i></u>
Arkusz stylów.</h2>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.6.



Rysunek 2.6. Selektor potomka

Jak widać, potomek nie musi być umieszczony bezpośrednio w znaczniku, którego jest potomkiem. W powyższym przykładzie w znaczniku `<h2>` znajduje się znacznik `<u>`, a dopiero w nim znacznik `<i>`, którego dotyczy definiowany styl.

Selektor potomka jest często wykorzystywany do definiowania odsyłaczy występujących wewnątrz bloku.

Selektor dziecka

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

```
rodzic > dziecko {właściwość: wartość;}
```

gdzie symbol `>` oznacza bezpośredni związek między elementami.

Znacznik będący dzieckiem musi wystąpić bezpośrednio wewnątrz znacznika nadrzędnego.

Przykład 2.16

```
<html>
<head>
<title> Selektor dziecka</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p>u {font-size: 20pt; color: yellow;}
</style>
</head>
<body>
<p> Wewnętrzne arkusze stylów. <u><i>Definiowanie stylów.</i></u> Arkusz
stylów.</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.7.



Rysunek 2.7. Selektor dziecka

Kolor *żółty* i wielkość liter 20pt zostały nadane tylko tekstom otoczonym znacznikiem `<u>` i umieszczonym bezpośrednio w akapicie (znacznik `<p>`).

Selektor braci

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

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

gdzie symbol `+` oznacza połączenie elementów.

Przykład 2.17

```
<html>
<head>
<title> Selektor braci</title>
<style type="text/css">
```

```

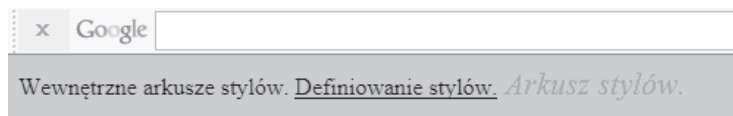
body {background-color: #c0c0c0;}
u + i {font-size: 16pt; color: red;}
</style>
</head>
<body>
<p> Wewnętrzne arkusze stylów. <u>Definiowanie stylów.</u><i> Arkusz
stylów.</i> </p>
</body>
</html>

```

Wynik interpretacji kodu został pokazany na rysunku 2.8.

Rysunek 2.8.

Selektor braci



W powyższym przykładzie znaczniki `<u>` oraz `<i>` znajdują się w tym samym rzędzie hierarchii, dlatego można dla nich ustawić selektor braci. Znacznik `<i>` występuje bezpośrednio po znaczniku `<u>` i tylko dla tekstu otoczonego tym znacznikiem jest realizowane formatowanie zdefiniowane dla selektora braci.

Zadanie 2.6

Wykorzystując arkusze stylów oraz selektory potomka, dziecka i braci, zdefiniuj style, które pozwolą na wyświetlenie tekstów sformatowanych tak jak na rysunku 2.9. Cały tekst powinien zostać zdefiniowany przy użyciu znacznika `<h2>`, natomiast wyróżnione fragmenty — za pomocą znaczników `<i>` oraz `<u>`.

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

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

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

Rysunek 2.9. Efekt użycia selektorów potomka (pierwszy wiersz), dziecka (drugi wiersz) i braci (trzeci wiersz)

2.3.2. Selektory atrybutów

W języku CSS można formatować znaczniki na podstawie posiadanych przez nie atrybutów. Na przykład jeżeli dla znacznika `<p>` (akapit) zostanie zdefiniowany atrybut `align` (wyrównanie), to inaczej będzie wyświetlany akapit wyrównany do lewej, a inaczej akapit wyśrodkowany.

Postać selektora atrybutów:

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

Przykład 2.18

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

Prosty selektor atrybutu

Prosty selektor atrybutu jest wykorzystywany dla elementów o nadanym określonym atrybucie, którego wartość nie ma znaczenia. Ma on postać:

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

lub

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

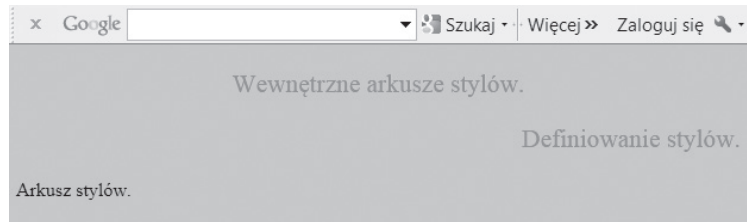
Przykład 2.19

```
<html>
<head>
<title> Selektor atrybutu</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p[align]{font-size: 16pt; color: red;}
</style>
</head>
<body>
<p align="center">Wewnętrzne arkusze stylów.</p>
<p align="right">Definiowanie stylów.</p>
<p> Arkusz stylów.</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.10.

Rysunek 2.10.

Prosty selektor atrybutu



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

Selektor atrybutu o określonej wartości

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

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

lub

```
[atrybut="wartość"] {właściwość: wartość;}
```

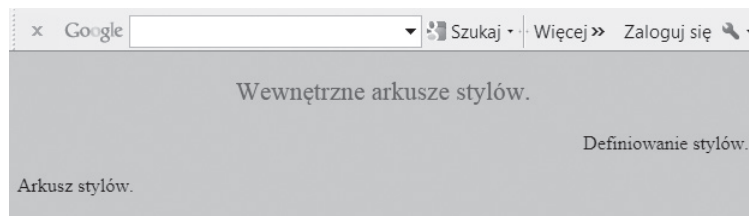
Przykład 2.20

```

<html>
<head>
<title> Selektor atrybutu</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p[align="center"]{font-size: 16pt; color: blue;}
</style>
</head>
<body>
<p align="center" > Wewnętrzne arkusze stylów.</p>
<p align="right">Definiowanie stylów.</p>
<p> Arkusz stylów.</p>
</body>
</html>

```

Wynik interpretacji kodu został pokazany na rysunku 2.11.



Rysunek 2.11. Selektor atrybutu o określonej wartości

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

Selektor atrybutu zawierającego określony wyraz

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

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

lub

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

Przykład 2.21

```

<html>
<head>
<title> Selektor atrybutu</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p[title~="jest"]{font-size: 16pt; color: green;}
</style>

```

```

</head>
<body>
<p title="To jest mój styl" > Wewnętrzne arkusze stylów.</p>
<p title="To jest mój akapit">Definiowanie stylów.</p>
<p title="Uwaga na style"> Arkusz stylów.</p>
</body>
</html>

```

Zadanie 2.7

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

2.3.3. Selektory specjalne

Selektor klasy

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

Selektor klasy ma postać:

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

gdzie *klasa* to dowolna nazwa.

Nazwa klasy musi spełniać kilka warunków:

- Musi zaczynać się od kropki.
- Mogą być w niej użyte litery, cyfry, znak podkreślenia, łącznik.
- Po kropce musi zostać wpisana litera.
- Rozróżniane są małe i wielkie litery.

Przykład 2.22

```

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

```

Odwołanie do klasy w dokumencie HTML ma postać:

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

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

Przykład 2.23

```

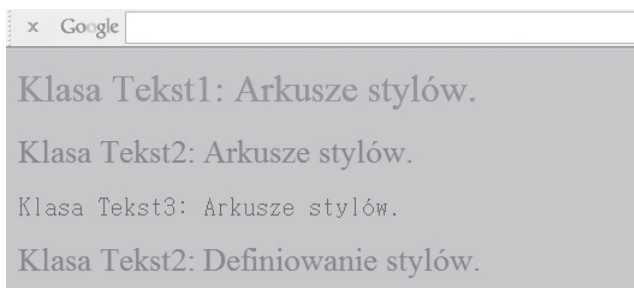
<html>
<head>
<title> Selektor atrybutu</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p.tekst1 {color: red; font-size: 24pt;}
p.tekst2 {color: green; font-size: 20pt;}
p.tekst3 {font-family: courier; color: blue; font-size: 16pt;}
</style>
</head>
<body>
<p class="tekst1"> Klasa Tekst1: Arkusze stylów.</p>
<p class="tekst2"> Klasa Tekst2: Arkusze stylów.</p>
<p class="tekst3"> Klasa Tekst3: Arkusze stylów.</p>
<p class="tekst2"> Klasa Tekst2: Definiowanie stylów.</p>
</body>
</html>

```

Wynik interpretacji kodu został pokazany na rysunku 2.12.

Rysunek 2.12.

Selektor klasy



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

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

Każdy element, któremu nadamy klasę o podanej nazwie, zostanie sformatowany zgodnie z definicją stylu dla klasy.

Przykład 2.24

```

<html>
<head>
<title> Selektor atrybutu</title>
<style type="text/css">
body {background-color: #c0c0c0;}
.nad {color: red; font-size: 20pt;}
</style>

```

```

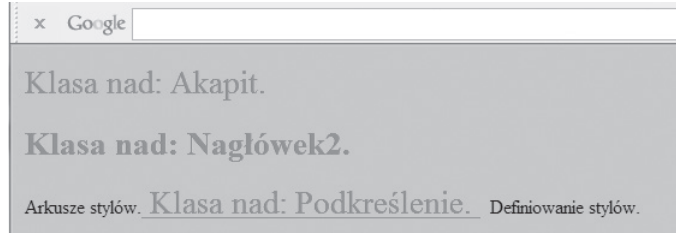
</head>
<body>
<p class="nad"> Klasa nad: Akapit.</p>
<h2 class="nad"> Klasa nad: Nagłówek2.</h2>
Arkusze stylów.<u class="nad"> Klasa nad: Podkreślenie. </u>
Definiowanie stylów.
</body>
</html>

```

Wynik interpretacji kodu został pokazany na rysunku 2.13.

Rysunek 2.13.

Efekt zastosowania klasy do formatowania różnych rodzajów znaczników



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

Selektor identyfikatora

Selektor identyfikatora jest stosowany, gdy chcemy nadać określone atrybuty formatowania elementowi, który ma przypisany jednoznaczny identyfikator. Selektor identyfikatora ma postać:

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

lub

```
#identyfikator {właściwość: wartość;}
```

Deklaracja identyfikatora musi rozpoczynać się znakiem #.

Przykład 2.25

```
h3#naglo3 {color: yellow;}
```

Odwołanie do identyfikatora w kodzie ma postać:

```
<h3 id="naglo3"> Styl nagłówkowy</h3>
```

2.3.4. Pseudoklasy

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

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

najechaniu na nie kursorem myszy, ich kliknięciu lub otwarciu. Do takiej dynamicznej zmiany stylu elementu służą **pseudoklasy**.

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

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

Przy użyciu specjalnych selektorów dla odsyłaczy można definiować style, które będą nadawane odsyłaczom w zależności od ich bieżącego stanu.

Odsyłacz podstawowy (pseudoklasa `:link`)

Nadaje atrybuty formatowania wszystkim jeszcze nieodwiedzonym odnośnikom.

```
a:link {właściwość: wartość;}
```

Przy deklarowaniu klasy można ustawić atrybuty formatowania dla wybranych odsyłaczy, które nie były jeszcze odwiedzane.

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

Przykład 2.26

```
a:link {color: green; background: yellow;}
a.nowa:link {color: orange;}
```

Odsyłacz odwiedzony (pseudoklasa `:visited`)

Nadaje atrybuty formatowania odsyłaczowi, jeśli ten był odwiedzony, a informacja o tym została umieszczona w pamięci przeglądarki.

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

Z deklaracją klasy ustawiane są atrybuty formatowania dla odsyłaczy wybranej klasy, które były otwierane.

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

Przykład 2.27

```
a:visited {color: red;}
a.po_nowa:visited {color: red;}
```

Wskazanie kursorem myszy (pseudoklasa `:hover`)

Nadaje atrybuty formatowania odsyłaczowi, gdy mysz znajduje się nad nim, ale nie aktywuje go.

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

Z deklaracją klasy ustawiane są atrybuty formatowania dla odsyłaczy wybranej klasy, które są w danej chwili wskazane myszą.

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

Przykład 2.28

```
a:hover {color: yellow;}
a.po_nowa:hover {color: yellow;}
```

UWAGA

Pseudoklasa `:hover` może być używana do definiowania stylów elementów innych niż odsyłacze.

Przykład 2.29

```
<html>
<head>
<title> Pseudoklasa :hover</title>
<style type="text/css">
body {background-color: #c0c0c0;}
.blok {height: 25px; width: 200px; background: aqua; color: black; }
.blok:hover { background: yellow; color: red; }
</style>
</head>
<body>
<div class="blok"> Strony internetowe</div>
</body>
</html>
```

Przykład pokazuje zastosowanie pseudoklas `:hover` dla znacznika `<div>`.

Odsyłacz aktywny (pseudoklasa `:active`)

Nadaje atrybuty formatowania odsyłaczowi, który w danej chwili jest aktywny, np. gdy użytkownik naciśnie i przytrzyma przycisk myszy.

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

Z deklaracją klasy ustawiane są atrybuty formatowania dla odsyłaczy wybranej klasy, które są w danej chwili aktywne.

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

Przykład 2.30

```
a:active {color: blue;}
a.po_nowa:active {color: blue;}
```

Kolejność deklarowania w arkuszu stylów pseudoklas przypisanych do odsyłaczy ma znaczenie dla ich prawidłowego działania.

Oto poprawna kolejność deklarowania pseudoklas:

```
a:link {właściwość: wartość; }
a:visited {właściwość: wartość; }
a:hover {właściwość: wartość; }
a:active {właściwość: wartość; }
```

Przykład 2.31

```
<html>
<head>
<title> Pseudoklasa odnośnik</title>
```

```

<style type="text/css">
body {background-color: #c0c0c0;}
a:link {color: green; }
a:visited {color: red; }
a:hover {color: yellow; }
a:active {color: blue; }
</style>
</head>
<body>
<p> Link do strony wydawnictwa: <a href="http://helion.pl/">HELION</a></p>
</body>
</html>

```

Pseudoklasa :focus

Nadaje atrybuty formatowania odsyłaczowi (wcześniej wybranemu) lub polu formularza, na którym został ustawiony kursor, na przykład gdy odsyłacz został wybrany za pośrednictwem klawisza *Tab* lub w polu formularza znalazł się kursor.

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

Zadanie 2.8

Wykorzystując arkusze stylów, zdefiniuj pseudoklasy, które umożliwią dynamiczną zmianę stylu linków znajdujących się na przykładowej stronie internetowej. Link podstawowy powinien mieć kolor czerwony, odwiedzany — kolor zielony, wskazany przez kursor myszy — kolor pomarańczowy, a link aktywny — kolor żółty.

2.3.5. Selektory pseudoelementów

W języku HTML nie ma mechanizmów dostępu do takich elementów strony jak pierwsza litera lub pierwsza linia akapitu bez otaczania ich znacznikami. Do formatowania tych specjalnych elementów można wykorzystać **pseudoelementy** dostępne w języku CSS.

Pierwsza linia (:first-line)

Pseudoelement `:first-line` nadaje określone formatowanie wszystkim pierwszym liniom znacznika, do którego odnosi się selektor.

```
selektor :first-line {właściwość: wartość; }
```

Selektorem może być dowolny znacznik języka HTML.

Przykład 2.32

```
p :first-line { color: red; font-size: 16pt; }
```

Pierwsza litera (:first-letter)

Pseudoelement `:first-letter` nadaje odrębne formatowanie pierwszej literze np. akapitu. Służy głównie do poprawienia wyglądu tekstu poprzez zaprojektowanie ozdobnej pierwszej litery (inicjału).

```
selektor :first-letter {właściwość: wartość; }
```

Przykład 2.33

```
p :first-letter { font-family: arial; font-size: 24pt; color: blue; }
```


960 GRID SYSTEM, 233

A

Accessibility Color Wheel,
242

ActionScript, 15, 246, 258

Adobe Dreamweaver, 161,
162, 166, 168, 169, 180

Adobe Flash, 246, 247, 262
rysowanie, 262, 265
ustawienia
predefiniowane ruchu,
267

Adobe Photoshop, *Patrz:*
Photoshop

Adobe Premiere, 283, 284

adres URL, 45, 46, 257

AJAX, 161

akapit, 31, 78

rozmiary, 119

Alligator Flash Designer,
246

animacja, 64, 179, 199,
200, 245

Flash, 179, 245, 246,
247

interaktywność, 246

mapy, 247

poklatkowa, 245, 246

skryptowa, 245, 258,
261

ustawienia predefinio-
wane ruchu, 267

z zastosowaniem klatek
kluczowych, 245

aparatus fotograficzny, 195

Areo 3W, 161

arkusz stylów kaskadowy,
Patrz: CSS

ASP, 161

ASP.NET, 161

atrybut, 15, 17, 18, 82

action, 50

align, 35, 40, 46, 65, 91

alt, 46, 49

autobuffer, 71

autocomplete, 63

autofocus, 63

autoplay, 71

autostart, 66

background-attachment,
109

background-color, 108,
111

background-image, 108

background-position,
110

background-repeat, 108,
218

bgcolor, 26, 35, 214

border, 45, 126

border-bottom-color,
115

border-bottom-width,
114

border-collapse, 126

border-color, 115

border-left-color, 115

border-left-width, 114

border-right-width, 114

border-spacing, 128

border-style, 114

border-top-color, 115

border-top-width, 114

border-width, 113

caption-side, 125

cellspacing, 41

class, 151

clear, 40, 135

color, 108

cols, 60

content, 25

controls, 66, 71

coords, 49

cursor, 138

disabled, 61

display, 136, 141

empty-cells, 126

enctype, 51

float, 132, 141

font-family, 99

font-size, 100, 104

font-style, 102, 104

font-variant, 103, 104

font-weight, 103, 104

height, 45, 65, 120

href, 48, 49, 80

hspace, 35, 41, 54

http-equiv, 25

label, 59, 60

language, 74

letter-spacing, 106

line-height, 104, 104

list-style, 125

list-style-image, 124, 125

list-style-position, 124,
125

list-style-type, 122, 125

logiczny, 19

loop, 66, 71

margin, 116

max-height, 120

maxlength, 51

max-width, 119

method, 51

min-height, 120

min-width, 119

multiple, 58

name, 25, 51, 60, 65

opacity, 129

order-right-color, 115

outline, 119

overflow, 121, 150

padding, 117

parametr, 15

playcount, 66

atrybut
 pluginspage, 67
 position, 128, 131
 preload, 71
 readonly, 61
 required, 63
 rows, 60
 rowspan, 43
 selected, 58
 shape, 49
 size, 51, 58
 src, 45, 65, 71
 start, 35
 tabindex, 62
 table-layout, 126
 target, 49
 text-align, 105
 text-decoration, 105
 text-indent, 105
 text-transform, 106
 titles, 49
 type, 35, 36, 51, 53, 54, 55, 64, 74
 usemap, 49
 valign, 35
 value, 35, 51, 58
 vertical-align, 132
 visibility, 138
 volume, 66
 vspace, 35, 41, 54
 white-space, 107, 141
 width, 45, 65, 119
 word-spacing, 106
 wrap, 61
 zdeprecjonowany, 18, 76

Audacity, 274

AVID Media Composer, 283

Avid Studio, 283

B

baner reklamowy, 246, 248
 barwa, [Patrz:](#) kolor
 Béziera krzywa, 72
 biała przestrzeń, 187
 blok, 44, 78, 119, 149, 187
 danych, 149
 krawędź, 113

rozmiary, 119
 szerokość, 139, 150
 border, [Patrz:](#) obramowanie
 bpp, 194
 brat, 89

C

Cascading Style Sheets,
[Patrz:](#) CSS
 cieniowanie, 72
 client-side, [Patrz:](#) język
 przetwarzania po stronie
 klienta
 CMYK, 196, 198, 201
 Complementary Metal
 Oxide Semiconductor,
[Patrz:](#) matryca CMOS
 CSS, 13, 18, 32, 34, 76,
 161, 168
 edytor, 84
 import, 81
 kaskadowość, 82
 media, 176
 menu, [Patrz:](#) menu
 model blokowy, 112
 pseudoelement, [Patrz:](#)
 pseudoelement
 reguła, 145
 kontekstowa, 146
 składnia, 84
 styl lokalny, 77, 82
 styl w linii, 77
 walidator, 295, 296
 wewnętrzny, 77, 79, 82
 zewnątrzny, 77, 80, 82,
 174

CSS3, 176

czas rzeczywisty, 72
 czcionka, 99, 100, 101,
 102, 103, 104, 105
 bezseryfowa, 99
 cursive, 99
 dekoracyjna, 99
 fantasy, 99
 grubość, 103
 jednostki, 101
 monospace, 99

monotypiczna, 99
 o stałej szerokości, 99
 pochyła, 99
 rozmiar, 100
 sans-serif, 99
 serif, 99
 styl, 102
 szeryfowa, 99
 wariant, 103

D

deklaracja, 85
 Document Type Definition,
[Patrz:](#) DTD
 dokument, 74
 HTML, 13, 14, 15, 22,
 23, 77, 162
 ciało, 26
 formatowanie, 27
 tworzenie, 27
 strona, [Patrz:](#) strona
 dokumentu
 struktura, 16
 DTD, 16, 17
 Frameset, 17
 wersja
 przejęciowa, [Patrz:](#)
 wersja Transitional
 ramkowa, [Patrz:](#)
 wersja Frameset
 Strict, 17
 ściśła, [Patrz:](#) wersja
 Strict
 Transitional, 17, 18
 działanie dynamiczne, 74
 dziecko, 89
 dziedziczenie, 83
 dźwięk, 19, 64, 71, 271
 edytowanie, 274, 276
 obwiednia, 280
 szum, [Patrz:](#) szum

E

edytor
 CSS, 84
 graficzny, 14
 map odnośników, 225

tekstu, 13, 14, 21
 Notepad++, [Patrz:](#)
 Notepad++
 Pico, [Patrz:](#) Pico
 Text, [Patrz:](#) Text
 wizualny, [Patrz:](#) edytory
 graficzny
 WYSIWYG, 160
 Adobe Dreamweaver,
[Patrz:](#) Adobe
 Dreamweaver
 Areo 3W, [Patrz:](#) Areo
 3W
 Visual Web Developer
 2010 Express
 Edition, [Patrz:](#) Visual
 Web Developer 2010
 Express Edition
 WYSIWYG Web
 Builder, [Patrz:](#)
 WYSIWYG Web
 Builder
 efekt
 czerwonych oczu, 207
 schodkowania, 193
 element
 graficzny, [Patrz:](#)
 grafika
 nadrzędny, 83, 88, 89
 nakładanie, 131
 obcinanie, 131
 podrzędny, 83, 88, 89
 pusty, 19
 ukrywanie, 138
 wyrównanie
 w pionie, 132
 w poziomie, 132
 wyświetlanie, 136, 138
 zastępowany, 136
 etykieta, 56
 Extensible HyperText
 Markup Language, [Patrz:](#)
 XHTML
 Extensible Markup
 Language, [Patrz:](#) XML

F

FileZilla, 301
 Flash Player, 179
 format
 aiff, 271, 272
 AIFF, 271
 AU, 271
 audio, 271, 272
 AVI, 65, 282
 BMP, 193, 199
 cdr, 192
 css, 80, 84
 DivX, 282
 flv, 179
 GIF, [Patrz:](#) GIF
 H.264, 273
 JPEG, [Patrz:](#) JPEG
 mid, 65
 MIDI, 65, 272
 mov, 65
 MP3, 65, 271
 MPEG, 65, 282
 MPEG-1 Audio Layer 3,
 271
 mpg, 65, 282
 ogg, 271, 272
 pliku graficznego, 199
 PNG, [Patrz:](#) PNG
 PSD, [Patrz:](#) PSD
 QuickTime, 282
 RealMedia, 272
 SVG, [Patrz:](#) SVG
 SWF, 246, 257
 TIFF, [Patrz:](#) TIFF
 Vorbis, 271, 272
 wav, 65, 271
 Windows Media Video,
 282
 WMA, 271, 272
 wmv, 65, 282
 formularz, 50, 51, 62, 63,
 73, 74
 grupowanie elementów,
 57
 lista rozwijana, 50
 pole

nieaktywne, 61
 tekstowe, 50, 52
 wyboru, 50
 przycisk opcji, 50, 52
 FTP, 299
 funkcja
 fillRect, 73
 strokeRect, 73

G

GIF, 44, 64, 72, 161, 193,
 199, 200, 201, 227, 228,
 229, 230
 animowany, 246
 głębia koloru, 194
 Google PageRank, 306
 GoogleMaps, 179
 gradient, 72, 205, 214, 218
 grafika, 44, 45, 160, 180
 animowana, 245, 246
 bitmapowa, [Patrz:](#)
 grafika rastrowa
 druk, 198
 na potrzeby strony
 internetowej, 200,
 208, 211, 217, 220,
 225, 227, 228, 232
 obiektowa, [Patrz:](#) grafika
 wektorowa
 rastrowa, 72, 193,
 199
 rozmiary, 119
 wektorowa, 192, 247

H

hiperłącze, [Patrz:](#) odsyłacz
 Hippo Animator, 246
 histogram, 194, 204, 205
 HSL, 87
 HSV, 196, 199
 HTML, 13, 76, 161, 245
 walidator, 295, 296
 HTML Tidy, 296
 HTML 4, 17, 18, 32
 HTML 5, 15, 19, 28, 63, 71
 HyperText Markup
 Language, [Patrz:](#) HTML

I

identyfikator, 139, 147
 ikona, 220, 236
 immediate mode rendering,
 Patrz: rendering trybu
 natychmiastowego
 indeks
 dolny, 29
 górny, 29
 inline, *Patrz:* CSS styl w linii
 iTunes, 273

J

JavaScript, 15, 64, 72, 73,
 161, 257
 język
 ActionScript, *Patrz:*
 ActionScript
 CSS, *Patrz:* CSS
 HTML, *Patrz:* HTML
 JavaScript, *Patrz:*
 JavaScript
 JScript, *Patrz:* Jscript
 Perl, *Patrz:* Perl
 PHP, *Patrz:* PHP
 przetwarzania
 po stronie
 klienta, 64
 serwera, 64
 skryptowy, 15, 73, 74,
 246
 VBScript, *Patrz:* VBScript
 XHTML, *Patrz:* XHTML
 XML, *Patrz:* XML
 znaczników, 76
 JPEG, 44, 45, 64, 72, 193,
 195, 199, 201, 227, 228,
 229
 JScript, 15
 JSP, 161

K

kadrowanie, 203
 kanał alfa, 72, 198, 200,
 254
 karta pamięci, 195

kaskadowość, *Patrz:* CSS
 kaskadowość
 klasa, 93, 95, 139, 148
 klatka kluczowa, 245
 klawiatura, 73
 klawisze skrótów, 62
 klient FTP, 161, 298, 299
 kodowanie znaków, 18
 polskich, 20, 22
 kolor, 33, 72, 86, 87, 108,
 150
 głębia, *Patrz:* głębia
 koloru
 model barw, 196
 paleta, 34, 193, 238
 24-bitowa, 45, 197
 48-bitowa, 45
 redukcja, 45
 kolumna, 135
 komentarz, 30, 85
 kompresja, 45, 199
 bezstratna, 193, 200
 dźwięku, 272
 bezstratna, 272, 273
 stratna, 271, 272
 stratna, 193, 199
 krzywa
 Béziara, 72
 drugiego stopnia, 72
 Kuler, 238
 kursor, 73

L

linia pozioma, 29
 link, *Patrz:* odsyłacz
 lista, 35, 78, 143
 atrybuty, 125
 definicji, 35, 37
 nieuporządkowana,
 139
 numerowana, 35, 122
 punktowana, 35, 36,
 122
 rozwijana, 63
 zagnieżdżona, 35, 37, 59
 logo, 220, 221, 232, 233

M

mapa
 animowana, 247
 bitowa, 193
 mapowanie obrazu, 224
 margines, 116, 139
 wewnętrzny, 117, 149
 zewewnętrzny, 118, 149
 marker, *Patrz:* punktor
 maska, 200
 matryca CMOS, 195
 menu, 139, 224, 232, 233
 animowane, 246, 258
 pionowe, 139
 poziome, 140
 zagnieżdżone, 143
 metajęzyk, 16
 model
 CMYK, *Patrz:* CMYK
 HSB, *Patrz:* HSV
 HSV, *Patrz:* HSV
 RGB, *Patrz:* RGB
 multimedia zewnętrzne, 179
 mysz, 73, 138, 246, 255
 kursor, 73, 96
 przycisk, 73, 97, 258

N

nagłówek, 23, 24, 32, 79,
 80, 187, 199, 294
 nieprzezroczystość, 129,
 Patrz też: przezroczystość
 Notepad++, 21

O

obramowanie, 45, 77, 113,
 114, 119, 149
 atrybuty, 116
 kolor, 115
 styl, 114
 szerokość, 113
 obrys, 119
 odnośnik, 49, 80, 294
 odstęp między wierszami,
 104
 odsyłacz, 48, 69, 95, 232,
 257

aktywny, 97
 graficzny, 49
 mapa, 49, 224
 odwiedzony, 96
 podstawowy, 96

P

padding, *Patrz:* margines
 wewnętrzny
 Pajączek, 225
 pasek nawigacyjny, 139
 Perl, 64
 Photoshop, 193, 200, 201,
 202, 203, 206, 207, 208,
 217, 218, 220, 228, 241
 PHP, 64, 161
 PHP5, 161
 pica, 102
 Pico, 21
 piksel, 101, 117, 193
 jasność, 194
 pingdom tools, 305
 plik
 AIFF, 271, 272
 AU, 271
 CSS, 80, 84
 index.htm, 23
 index.html, 23
 konfiguracyjny, 16
 lokalny, 180
 MP3, 271
 multimedialny, 64, 71
 Wave Form Audio
 Format, 271
 wideo, *Patrz:* wideo
 Windows Media Audio,
 271
 plug-in, *Patrz:* wtyczka
 PNG, 44, 45, 64, 72, 193,
 200, 201, 227, 229, 230
 podpis, 125
 pole
 BUTTON, 55
 INPUT, 51, 56
 SELECT, 58
 tekstowe, 73
 TEXTAREA, 60
 polecenie, 15, 23, 24, 27
 !important, 82
 @import, 81, 84
 portal
 internetowy, 14
 społecznościowy, 233
 wertykalny, *Patrz:* wortal
 internetowy
 potomek, 88
 pozycjonowanie, 128
 protokół
 FTP, 299
 sieciowy, 16
 SSH, 299
 wywoływania zdalnego
 dostępu do obiektów,
 16
 przeglądarka internetowa,
 13, 18, 65, 74, 257, 294,
 297
 Internet Explorer, 142,
 257
 obsługa błędów, 19
 przepełnienie, 121
 przeplot, 45
 przesyłanie strumieniowe,
 272
 przetwornik analogowo-
 -cyfrowy, 195
 przetwornik
 optoelektroniczny, 195
 przezroczystość, 45, 72,
 129, 198, 199, 227,
 254
 przycisk, 199, 222, 237
 animowany, 246, 252,
 255
 powiązanie
 z hiperłączem, 257
 PSD, 200
 pseudoelement, 98
 :first-letter, 98
 :first-line, 98
 pierwsza linia, 98
 pierwsza litera, 98
 pseudoklasa, 95, 96
 :active, 96

:focus, 98
 :hover, 96, 97
 :link, 96
 :visited, 96
 punkt, 101, 122, 139
 obrazek, 124
 położenie, 124

Q

QuickTime, 273

R

Radical Image Optimization
 Tool, 228, 229
 radio internetowe, 273
 ramka, 150, *Patrz też:*
 obramowanie
 Real Player, 273
 reguła stylu, 84
 rendering trybu
 natychmiastowego, 72
 RGB, 34, 86, 199, 200
 RGBA, 87, 198
 rozdzielczość, 194, 196
 ekranowa, 200
 geometryczna, 194
 radiometryczna, 194

S

Scalable Vector Graphics,
Patrz: SVG
 selektor, 77, 84, 85, 87
 atrybutu, 90
 o określonej wartości,
 91
 prosty, 91
 zawierającego
 określony wyraz, 92
 brata, 89
 dziecka, 89
 grupowanie, 128
 identyfikatora, 95
 klasy, 93
 potomka, 88
 pseudoelementu, 98
 typu, 87, 95
 uniwersalny, 87

- server-side, *Patrz:* język przetwarzania po stronie serwera
- serwer, 14, 298, 299
 - DNS, 14
 - FTP, 14
 - HTTP, 14
 - internetowy, 14
- Simple Object Access Protocol, *Patrz:* SOAP
- Site Map, *Patrz:* strona internetowa mapa
- skaner, 196
- skrypt, 73, 74
- slider, 233, 236
- SOAP, 16
- SQL, 161
- SSH, 299
- standard
 - ASCII, 20
 - ISO 8859, 20
 - UCS, 20
 - Unicode, 20
 - UTF, 20
 - UTF-8, 20, 23
 - Windows-1250, 20
- stopka, 187, 232, 233
- strona
 - dokumentu, 31
 - internetowa, 14, 234
 - animacja, *Patrz:* animacja
 - architektura, 187, 190
 - dynamiczna, 14
 - implementacja, 186
 - indeksowanie, 24, 306
 - mapa, 310
 - nagłówek, *Patrz:* nagłówek
 - nawigacja, 222
 - optymalizacja, 308
 - opuszczanie, 74
 - pozycjonowanie, 308, 309
 - projektowanie, 234
 - publikowanie, 181, 293, 298, 299
 - specyfikacja, 188
 - styczna, 14
 - szablon, 145, 150, 151, 155, 169, 232, 233
 - test funkcjonalności, 306
 - test szybkości wczytywania, 303
 - testowanie, 293, 297
 - układ, 187, 232, 233
 - użyteczność, 186, 232
 - walidacja, 294, 296
 - zasady projektowania, 185, 187, 190
 - o stałej szerokości, 150
 - o zmiennej szerokości, 155
- szum, 280
- T**
 - tabela, 39, 125, 160, 166
 - komórka, 39
 - łączenie, 42
 - odstęp, 41, 128
 - szerokość, 39
 - wyrównanie w pionie, 43
 - wyrównanie w poziomie, 43
 - krawędź, 126, *Patrz też:*
 - obramowanie
 - odstęp, 41
 - otoczona tekstem, 40
 - podpis, *Patrz:* podpis
 - rozmiary, 119
 - szerokość, 39
 - wiersz, 39
 - zagnieżdżona, 43
- tablet, 176
- tag, *Patrz:* znacznik
- Tagged Image File Format, *Patrz:* TIFF
- tapeta, 216, 217
- tekst
 - animowany, 254
 - efekty, 105, 254
 - formatowanie, 77, 99, 100, 101, 102, 103, 104, 105, 106, 164
 - wcięcie pierwszego wiersza, 105
 - zastępczy, 46, 49
- telefon komórkowy, 176
- Text, 21
- TIFF, 193, 200
- tło, 23, 26, 27, 44, 45, 76, 77, 108, 110, 112, 150, 207, 208, 211
 - formatowanie, 77
 - strony internetowej, 214, 217
- top na stronę internetową, 246
- Total Commander, 300
- V**
 - VBScript, 64, 73
 - vertical portal, *Patrz:* wortal internetowy
 - Visual Studio 2010 Express, 161
 - Visual Web Developer 2010 Express Edition, 161
- W**
 - W3C CSS Validation Service, 296
 - W3C HTML Validation Service, 296
 - W3C Link Checker, 296
 - walidator, 295
 - warstwa, 200, 208, 250
 - prezentacyjna, 76
 - semantyczna, 76

- strukturalna, 76
 - styl, 210
 - wartość, 77
 - WatchScript, 303
 - Wave Accessibility Tool, 296
 - Web Page Analyzer, 306
 - wideo, 19, 64, 71, 179, 180, 271, 273, 284, 285
 - Winamp, 273
 - Windows Media Player, 273
 - witryna internetowa, 14
 - właściwość, 77, 85, 99
 - wartość, 85, 86
 - wortal internetowy, 14
 - wtyczka, 64, 71, 179
 - wykres, 72
 - WYSIWYG Web Builder, 161
 - wyszukiwarka, 24, 309
 - indeks, 309, 310
 - wyświetlacz LCD, 195
- X**
- Xenu Link Sleuth, 297
 - XHTML, 14, 15, 17, 161
 - XHTML 1.0, 18
 - XHTML 1.1, 17, 18
 - XML, 15, 16
 - składnia, 17
- Z**
- zdarzenie, 73
 - onblur, 73
 - onchange, 73
 - onclick, 73
 - ondblclick, 73
 - onfocus, 73
 - onkeydown, 73
 - onkeypress, 73
 - onkeyup, 73
 - onload, 73
 - onmousemove, 73
 - onmouseout, 73
 - onmouseover, 73
 - onmouseup, 73
 - onreset, 73
 - onselect, 73
 - onsubmit, 73
 - onunload, 73
 - wewnętrzne, 73
 - zdjęcie
 - kadrowanie, *Patrz:* kadrowanie
 - klonowanie, 206
 - korekcja, 202, 203, 204, 206, 207
 - zmiana tła, 207
 - zmienna
 - fillStyle, 73
 - strokeStyle, 73
 - znacznik, 13, 15, 16, 23, 34
 - <*>, 149
 - <a>, 46, 48, 49, 140
 - <audio>, 71
 - , 15
 - <basefont>, 28, 34
 - <big>, 28, 34
 - <body>, 28
 -
, 15, 32, 40
 - <button>, 55
 - <canvas>, 72
 - <center>, 32, 35
 - <changeprev>, 311
 - <dd>, 37
 - <div>, 44, 78, 135, 149, 150, 151, 187
 - <dl>, 37
 - <dt>, 37
 - <embed>, 65, 67, 257
 - <fieldset>, 57
 - , 27, 28, 35
 - <form>, 50, 57
 - <head>, 23, 24, 79, 80
 - <hr>, 15, 29
 - <html>, 23
 - <i>, 28, 88
 - <iframe>, 17
 - , 45, 49
 - <input>, 51, 55, 61, 64
 - <label>, 56
 - , 15, 35, 36, 122
 - <link>, 85
 - <loc>, 311
 - <map>, 49
 - <meta>, 23, 24, 25
 - <object>, 257
 - , 35, 122
 - <optgroup>, 59
 - <option>, 58, 59
 - <p>, 31, 91
 - <priority>, 311
 - <script>, 74
 - <select>, 58, 59
 - <small>, 28
 - <spacer>, 32
 - , 78, 145, 146
 - <strike>, 29, 35
 - <style>, 79, 81
 - <sub>, 29
 - <table>, 40, 41
 - <textarea>, 60
 - <td>, 39, 40, 42, 43
 - <title>, 23, 24
 - <tr>, 39, 42
 - <u>, 29, 35, 88
 - , 36, 122, 139, 140
 - <video>, 71
 - znak
 - , 19
 - &, 19, 30
 - , 30
 - [, 19
 -], 19
 - €, 30
 - <, 19, 30
 - >, 19, 30
 - §, 30
 - ©, 30
 - ®, 30
 - kodowanie, *Patrz:* kodowanie znaków
 - specjalny, 30
 - ™, 30

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA

 **Helion SA**



Kwalifikacja E.14 Tworzenie stron internetowych

Część 1

Podręcznik do nauki zawodu **technik informatyk**

Technik informatyk nie jest zwykłym użytkownikiem komputerów. Jeśli uczeń wybiera szkołę o takim profilu, z czasem staje się prawdziwym komputerowym ekspertem.

Kwalifikacja E.14 skupia się na tworzeniu aplikacji internetowych oraz pracach związanych z bazami danych. Opracowany z myślą o przyszłych technikach informatyki podręcznik *Tworzenie stron internetowych* stanowi pierwszą z grupy książek przygotowanych pod kątem tego materiału. Podręcznik ten jest zgodny z najnowszą podstawą programową. To kompletne kompendium, zawierające teorię popartą licznymi przykładami, lecz prowadzące także do samodzielnej pracy i zdobywania praktycznych umiejętności tworzenia dobrych stron WWW czy ulepszania ich szaty graficznej. Dzięki zdobytej wiedzy każdy z uczniów zaprojektuje własną witrynę internetową, pozna zasady języka HTML oraz wykorzysta kaskadowe arkusze stylów. Dodatkowe informacje dotyczące architektury stron internetowych czy wykorzystania elementów graficznych i animacji przyczynią się do tego, że szkolne projekty osiągną profesjonalny poziom. Strony będą nie tylko estetyczne, lecz także przyjazne dla użytkownika, czytelne i zgodne z regułami walidacji.

Technik informatyk to doskonały, charakteryzujący się wysoką jakością, kompletny zestaw edukacyjny, przygotowany przez dysponującego ogromnym doświadczeniem lidera na rynku książek informatycznych – wydawnictwo Helion.

W skład zestawu *Technik informatyk* wchodzi także:

Kwalifikacja E.12. Montaż i eksploatacja komputerów osobistych oraz urządzeń peryferyjnych.
Podręcznik do nauki zawodu technik informatyk

Kwalifikacja E.13. Projektowanie lokalnych sieci komputerowych i administrowanie sieciami.
Podręcznik do nauki zawodu technik informatyk.

Kwalifikacja E.14. Część 1. Tworzenie stron internetowych. Podręcznik do nauki zawodu technik informatyk.

Kwalifikacja E.14. Część 2. Tworzenie baz danych oraz administrowanie bazami.
Podręcznik do nauki zawodu technik informatyk.

Kwalifikacja E.14. Część 3. Tworzenie aplikacji internetowych.
Podręcznik do nauki zawodu technik informatyk.

Podręczniki oraz inne pomoce naukowe należące do tej serii zostały opracowane z myślą o wykształceniu kompetentnych techników, którzy bez trudu poradzą sobie z wyzwaniami w świecie współczesnej informatyki. Według nowych przepisów, aby otrzymać dyplom potwierdzający kwalifikacje w zawodzie technik informatyk, należy zdać szereg egzaminów potwierdzających kolejne kwalifikacje w zawodzie. Książka ta powstała po to, by ułatwić to zadanie zarówno uczącym się, jak i pedagogom. Wiedza w niej zawarta pomoże zdać egzamin i zyskać wiedzę praktyczną, przydatną w przyszłej pracy.

helion.pl
księgarnia
internetowa

Nr katalogowy: 8759



Księgarnia internetowa:
<http://helion.pl>



Zamówienia telefoniczne:
0 801 339900
0 601 339900



Helion

Sprawdź najnowsze promocje:

🔗 <http://helion.pl/promocje>

Książki najchętniej czytane:

🔗 <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

🔗 <http://helion.pl/nowosci>

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel.: 32 230 98 63

e-mail: helion@helion.pl

<http://helion.pl>

sięgnij po **WIECEJ**



KOD KORZYŚCI

ISBN 978-83-246-5102-3



Informatyka w najlepszym wydaniu