



OSADZANIE STYŁÓW W DOKUMENCIE

Style wewnętrzne

Styl lokalny umieszcza się wewnątrz znacznika formatowanego elementu. Styl działa wyłącznie w elemencie, w którym się znajduje.

```
<p style="color:green; font-size:15pt;">>Treść dokumentu</p>
```

Style zewnętrzne

Zewnętrzny arkusz stylów umieszczany jest w nagłówku <head> dokumentu. Styl zewnętrzny działaniem obejmuje całą zawartość strony i może być wykorzystany również na podstronach.

```
<html>
  <head>
    <link href="arkusz.css" rel="stylesheet" type="text/css" />
  </head>
  <body>
    <p>Tekst akapitu</p>
  </body>
</html>
```

JEDNOSTKI MIAR

Jednostki względne

px — piksele

```
P {margin-left:20px;}
```

Jednostka ta opiera się na pojedynczych punktach widocznych na ekranie monitora — pikselach.

em — proporcje wysokości do czcionki danego elementu

```
P {margin-left:2em;}
```

Zasada działania tej jednostki jest stosunkowo prosta i polega na określaniu zależności pomiędzy poszczególnymi wielkościami czcionki.

ex — proporcje do wysokości litery x

```
P {margin-left:2ex;}
```

Stosowanie jednostek ex wiąże się z rodzajem użytej czcionki.

Jak zapewne wiecie, każda czcionka inaczej wygląda, a co za tym idzie, ma inną wielkość podstawową, dlatego czcionka Arial o wielkości 2ex nie będzie równa czcionce Times New Roman o wielkości 2ex.

% — procenty

```
P {font-size:10%;}
```

Procenty służą do określania wielkości względem wartości domyślnej.

Jednostki bezwzględne

in — cala

```
P {margin-left:2in;}
```

Cale wywodzą się z amerykańskiego systemu miarowego i głównie tam są wykorzystywane. W Polsce ta jednostka raczej nie będzie potrzebna.

pt — punkty

```
P {font-size:12pt;}
```

Punkty wywodzą się z typografii, gdzie są standardową jednostką miary.

W praktyce 72 punkty odpowiadają jednemu calowi, a ten z kolei

równia się 2,54 cm.

cm — centymetry

```
P {font-size:1cm;}
```

Centymetr jest miarą stosowaną w większości krajów na świecie. Jeden centymetr odpowiada 0,39 cala.

mm — milimetry

```
P {font-size:12mm;}
```

Milimetry są jednostką mniejszą niż centymetr. Występują w systemie metrycznym; na jeden centymetr składa się 10 mm.

pc — pica

```
P {font-size:12pc;}
```

Podobnie jak punkt, jest jednostką typograficzną; jeden pica równa się 12 punktom.

Jednostki czasu i jednostki kątowe

Jednostki służące do określenia czasu oraz kątów zostały wprowadzone w drugiej specyfikacji — CSS2. Mają zastosowanie przy projektowaniu stron wzbogaczonych o możliwość czytania dokumentu przez komputer oraz obsługi mediów osadzanych na stronach WWW.

Jednostki czasu to:

ms — milisekunda

```
H5 {pause-after:100ms;}
```

s — sekunda

```
H5 {pause-after:10s;}
```

Obie jednostki są używane do określania odstępu pomiędzy czytanymi fragmentami.

Jednostki kątowe to:

deg — stopnie

```
H5 {elevation:10deg;}
```

grad — gradiany

```
H5 {elevation:10grad;}
```

rad — radiany

```
H5 {elevation:10rad;}
```

Jednostki te używane są do określenia pozycji źródła dźwięku.

KOLORY

Nazwa własna koloru

Dla określania kolorów, podobnie jak w języku HTML, możemy użyć nazw kolorów, np. red, green czy blue. Niestety, nazwy zostały przypisane jedynie kilkunastu podstawowym kolorom.

Tabela I. Wybrane kolory z przypisanymi nazwami wchodzące w skład bezpiecznej palety 216 kolorów

Nazwa koloru	Zapis szesnastkowy	Fiolet	#FF00FF
Black	#000000	Green	#008000
Silver	#C0C0C0	Lime	#00FF00
Gray	#B0B0B0	Olive	#808000
White	#FFFFFF	Yellow	#FFFF00
Maroon	#800000	Navy	#000080
Red	#FF0000	Blue	#0000FF
Purple	#800080	Teal	#008080
		Aqua	#00FFFF

Wartość RGB

Zapis kolorów opierający się na modelu RGB przypisuje liczby całkowite z przedziału od 0 do 255 dla każdej z wartości składowej palety RGB (ang. red, green, blue), stąd liczb może być aż dziesięć.

Zapis w postaci RGB może przybierać następującą postać:

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

Oczywiście możemy jeszcze zapisać kolor w postaci RGB, definiując po prostu nazwy danej barwy.

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

Wartość szesnastkowa

Kolejnym sposobem zapisu kolorów jest metoda szesnastkowa, będąca tym samym, co zapis szesnastkowy doskonale znany wszystkim, którzy mieli do czynienia z komputerami.

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

SELEKTORY

W przypadku języka XHTML polecenie odpowiedzialne za podłączenie zewnętrznego arkusza stylów ma postać:

```
<?xmlstylesheet type="text/css" href="arkusz.css" ?>
```

Dla zachowania zgodności dokumentów XHTML zaleca się umieszczenie podwójnej deklaracji wypowiadającej arkusz stylów. Oczywiście oba wpisy muszą znaleźć się w nagłówku kodu witryny <head></head>.

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

Styl importowany jest pobierany z oddzielnego pliku znajdującego się pod wskazanym adresem. Podobnie jak styl zewnętrzny, działaniem obejmuje całą treść strony, na której został użyty.

```
<html>
  <head>
    <style type="text/css">
      P {
        color:green;
        font-size:15pt;
      }
    </style>
  </head>
  <body>
    <p>Tekst akapitu</p>
  </body>
</html>
```

To właśnie dzięki selektorowi możemy przypisać konkretny styl do wybranego elementu. Specyfikacji CSS przewiduje kilka rodzajów selektorów oraz identyfikatorów, klas, pseudoklas i pseudoelementów.

Selektor prosty

Selektorem prostym jest znacznik języka XHTML/HTML przypisany do elementu, który chcesz w danym momencie formatować. Styl zdefiniowany dla przykładowego selektora P będzie się odnosił wyłącznie do akapitów zamkniętych w znaczniku <p></p>.

```
P {
  color:red;
  font-size:14pt;
}
```

Selektor uniwersalny

Selektor uniwersalny pozwala na ustalenie formatowania dla wszystkich elementów strony WWW. W chwili obecnej dostępne są dwa selektory uniwersalne: BODY oraz *

```
*
```

```

  {
    color:red;
  }
  BODY {
    color:red;
  }

```

Selektor „potomka”

Selektor potomka pozwala na ustalenie formatowania dla elementu występującego wewnątrz innego znacznika. Przykład wymusza nadanie formatowania fragmentowi tekstu zamkniętego w i znajdującej się w znaczniku <h1></h1>.

```
H1 SPAN {
  font-size:10pt;
}
```

Selektor „dziecka”

Selektor dziecka jest wynikiem zależności panujących pomiędzy poszczególnymi znacznikami języka XHTML/HTML. Przykładowy styl działał w przypadku, gdy wewnątrz bloku tekstu zostanie umieszczony znacznik .

```
P > SPAN {
  font-style:italic;
  text-decoration:underline;
}
```

Selektor „rodzeństwa”

Na podstawie tego zadania selektora możemy przypisać styl elementowi bezpośrednio sąsiadującemu z innym. W warunku został spełniony, oba selektory muszą mieć wspólnego „rodzica”.

```
P > SPAN {
  font-style:italic;
  text-decoration:underline;
}
```

Selektor atrybutu

Selektor atrybutu odnosi się do wybranego znacznika XHTML/HTML mającego ścisłe określony atrybut. Przykładowy styl działał w przypadku znacznika stopnia pierwszego o określonym atrybutie title.

```
H1 [title] {
  color:red;
}
```

Grupowanie selektorów

Selektory można dowolnie grupować, jeżeli pewne wartości mają obowiązywać dla każdego z nich. Dzięki zborczemu zapisowi możemy znieść uproszczyć strukturę arkusza i pracę z kodem witryny. Poszczególne selektory wchodzące w skład grupy rozdzielamy przecinkami. Ponizej przedstawiono przykład grupowanych selektorów h1, h2 oraz p.

```
H1, H2, P {
  color:green;
}
```

Identyfikatory

Identyfikator pozwala na przypisanie formatowania do wybranego znacznika mającego unikalny id. Dzięki temu możemy zróżnicować sposoby prezentacji elementów na stronie.

```
#pochyl {
  font-style:italic;
  font-size:20pt;
}
```

Aby skorzystać z przykładowego stylu, wybrany znacznik musi mieć dodatkowy atrybut id.

```
<p id="pochyl">Treść akapitu</p>
```

Możliwy jest również inny zapis, który wymusza przypisanie identyfikatora wyłącznie do określonego znacznika.

```
H1#pochyl {
  font-style:italic;
  font-size:20pt;
}
```

Klasy

Klasa umożliwia zróżnicowanie formatowania wybranych elementów na stronie w zależności od atrybutu class umieszczonego w znaczniku XHTML/HTML. W odróżnieniu od identyfikatora klasa może być wykorzystywana wielokrotnie.

```
.moja {
  color:green;
  font-size:14pt;
}
```

Znaknik, aby skorzystać z przykładowej klasy, musi mieć zdefiniowany atrybut class z nazwą klasy.

```
<p class="moja">Treść akapitu</p>
```

Klasa może zostać również równieź powiązana z konkretnym znacznikiem XHTML/HTML.

```
P.moja {
  color:green;
  font-size:14pt;
}
```

Pseudoklasy

Specyfikacja CSS przewiduje kilka ścisłe określonych klas pozwalających na formatowanie niektórych elementów dokumentu. Do najpopularniejszych pseudoklas z całą pewnością zaliczają się pseudoklasy odzyski.

Pseudoklasa :link dotyczy formatowania odnośnika w stanie nienarzuconym.

```
:link {
  color:navy;
}
```

Pseudoklasa :visited dotyczy formatowania odnośnika, który został już odwiedzony. Pozwala na wyróżnienie go spośród innych odnośników znajdujących się na stronie.

```
:visited {
  color:green;
}
```

Pseudoklasa :hover pozwala na ustalenie formatowania elementu, nad którym w danej chwili znajduje się kurSOR myszy.

```
:hover {
  color:red;
  text-decoration:underline;
}
```

Pseudoklasa :active pozwala na ustalenie wyglądu aktywnego elementu na stronie.

```
:active {
  color:silver;
}
```

Pseudoklasa :focus pozwala na wyróżnienie aktualnie wyświetlanego elementu, np. odzyska.

```
:focus {
  color:braun;
}
```

Pseudoklasa :first-child odpowiada za formatowanie pierwszego dziecka elementu nadreżnego. Dokładniej mówiąc, jeżeli np. znacznik <div> zawiera w sobie trzy alapki określone za pomocą znacznika <div>, to formatowanie zostanie poddany tylko pierwszy z nich:

```
P:first-child {
  font-size:10pt;
  font-style:italic;
  font-weight:bold;
}
```

Pseudoklasa :lang jest wykorzystywana do określania formatowania dokumentów wielojęzycznych:

```
P:lang(p1) {
  font-weight:bold;
  color:black;
}
```

Kod XHTML/HTML, aby skorzystać z przykładowej pseudoklasy :lang zdefiniowanej powyżej, musi wyglądać w następujący sposób:

```
<p lang="pl">Tekst w języku polskim.</p>
```

Pseudoelementy

Ciekawą grupą selektorów stanowią pseudoelementy, które pozwalają nam na przypisanie kolejnych wymyslionych stylów do elementów strony WWW.

Pseudoelement :first-letter pozwala na formatowanie pierwszej litery w bloku tekstu. Przedstawiony poniżej arkusz spowoduje powiększenie pierwszej litery do wielkości 100 punktów i nadanie jej koloru czerwonego.

```
:first-letter {
  color:red;
  font-size:100pt;
}
```

Pseudoelement :first-line w przeciwieństwie do poprzednika obejmuje formatowaniem całego pierwszej linii, a nie tylko pierwszą literę.

```
:first-line {
  color:red;
  font-size:20pt;
  font-weight:bold;
}
```

Pseudoelement :after oraz :before służą do umieszczenia tekstu przed formatowanym elementem strony i za nim.

```
:before {
  content:"WAŻNA WIADOMOŚĆ";
  font-size:15pt;
  font-style:italic;
  font-weight:bold;
}
```

```
:after {
  content:"AUTOR Bartosz Danowski";
  font-size:15pt;
  font-style:italic;
  font-weight:bold;
}
```

Tablice informatyczne. CSS3

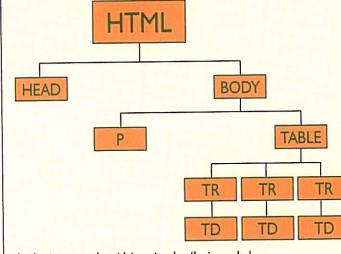
DZIEDZICZENIE I KASKADOWOŚĆ

Dziedziczenie

Zrozumienie pojęcia dziedziczenia wymaga od nas zapoznania się z hierarchią ważności poszczególnych znaczników wewnętrznych dokumentu. Arkusze stylów wprowadzają pojęcie drzewa i na jego przykładzie doskonale widać te zależności. Ponizej znajdują się przykładowy listing prostego kodu strony oraz rozrysowane dla niego drzewo.

```
<html>
  <head>
    <title>przykład drzewa</title>
  </head>
  <body>
    <p>przykład tekstu.</p>
    <table>
      <tr>
        <td></td>
      </tr>
      <tr>
        <td></td>
      </tr>
      <tr>
        <td></td>
      </tr>
    </table>
  </body>
</html>
```

Przykład drzewa dla powyższego listingu



Analizując rysunek, widzimy, że określenie np. koloru tekstu (niebieski) bezpośrednio dla pary znaczników `<body>` `</body>` spowoduje jego użycie dla każdego elementu strony znajdującej się niżej w hierarchii. Jeżeli z kolei zdefiniujemy kolor listy `` jako żółty, to tekst całego dokumentu będzie niebieski poza listą, która zostanie sformatowana kolorem żółtym.

Kaskadowość

Kolejnym bardzo ważnym pojęciem stosowanym w kaskadowych arkuszach stylów, a przy tym występującym w samej nazwie, jest kaskadowość. Funkcja ta odpowiada za określenie hierarchii stosowanych stylów w dokumencie. Wiemy już, że style do dokumentu możemy wstawiać w kilku sposobach (bezpośrednio w kodzie strony jako atrybut dowolnego znacznika, w nagłówku `<head>` lub `</head>`, globalnie dla danego dokumentu oraz przez doliczenie zewnętrznego arkusza). Mieszanie zastosowanych stylów jest możliwe

- `pre-wrap` — zachowuje dodatkowe spacje i linie wierszy tylko wtedy, gdy jest to potrzebne;
- `inherit` — pobiera wartości z elementu nadzawanego — rodzica.

Orientacja tekstu

direction

Dostępne wartości	<code>ltr</code> <code>rtl</code> <code>inherit</code>
Wartość domyślna	<code>normal</code>
Przykład	<code>p{direction:rtl;}</code>
Dziedziczenie	tak

Obsługa: Opera, Mozilla Firefox, MS Internet Explorer

- `ltr` — kierunek tekstu do lewej do prawej;
- `rtl` — kierunek tekstu od prawej do lewej;
- `inherit` — pobiera wartości z elementu nadzawanego — rodzica.

Orientacja tekstu dla elementów zagnieżdzonych

unicode-bidi

Dostępne wartości	<code>normal</code> <code>embed</code> <code>bidi-override</code> <code>inherit</code>
Wartość domyślna	<code>normal</code>
Przykład	<code>p{unicode-bidi:embed;}</code>
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer

- `normal` — nie są prowadzone zmiany w domylnych ustawieniach;
- `embed` — ustawienia dla elementów linowych pobierane ze stylem `direction`;
- `bidi-override` — pierwszeństwo ma klucz ten zdefiniowany w stylu `direction`;
- `inherit` — pobiera wartości z elementu nadzawanego — rodzica.

WŁAŚCIWOŚCI CZCIONKI

Jedną z najważniejszych funkcji oferowanych przez kaskadowe arkusze stylów jest możliwość formatowania wyglądu czcionki użytej do skomponowania strony. Dzięki CSS możemy określić:

- rodzaj czcionki,
- wielkość czcionki,
- styl czcionki,
- waga czcionki,
- wariant czcionki,
- szerokość czcionki,
- proporcje czcionki.

Waga czcionki

font-weight

Dostępne wartości	<code>normal</code> <code>bold</code> <code>100</code> <code>200</code> <code>300</code> <code>400</code> <code>500</code> <code>600</code> <code>700</code> <code>800</code> <code>900</code> <code>bolder</code> <code>lighter</code> <code>inherit</code>
Wartość domyślna	<code>normal</code>
Przykład	<code>p{font-weight:bold;}</code>
Dziedziczenie	tak
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `normal` — domylna grubość czcionki;
- `bold` — czcionka pogrubiona;
- `100 - 900` — wartość określająca grubość czcionki, od cienkiej do grubiej;
- `bolder` — czcionka grubsza niż aktualnie ustawiona;
- `lighter` — czcionka cieńsza niż aktualnie ustawiona;
- `inherit` — pobiera wartości z elementu nadzawanego — rodzica.

Styl czcionki

font-style

Dostępne wartości	<code>normal</code> <code>italic</code> <code>oblique</code> <code>inherit</code>
Wartość domyślna	<code>normal</code>
Przykład	<code>p{font-style:italic;}</code>
Dziedziczenie	tak
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `normal` — czcionka domyńska;
- `italic` — kursywa;
- `oblique` — odmiana czcionki pochyła;
- `inherit` — pobiera wartości z elementu nadzawanego — rodzica.

Warianty czcionki

font-variant

Dostępne wartości	<code>normal</code> <code>small-caps</code> <code>inherit</code>
Wartość domyślna	<code>normal</code>
Przykład	<code>p{font-variant:small-caps;}</code>
Dziedziczenie	tak
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `normal` — czcionka domyńska;
- `small-caps` — kapitałki;
- `inherit` — pobiera wartości z elementu nadzawanego — rodzica.

Zbiórowy zapis właściwości czcionek

font

Dostępne wartości	<code><font-style></code> <code><font-variant></code> <code><font-weight></code> <code><font-size></code> <code><font-height></code> <code><font-family></code>
Wartość domyślna	zob. właściwości poszczególnych stylów
Przykład	<code>p { font:italic small-caps 600 14pt/2 arial; }</code>
Dziedziczenie	tak
Obsługa	zob. właściwości poszczególnych stylów

Gdy analizuje się przykładowy zbiór stylu, od razu widać, że poszczególne wartości nie rozdzielają żadne dodatkowe znaki — przecinki, myślniki itp. Jedynym wyjątkiem są wielkość czcionki oraz odstęp lini; pomiędzy tymi wartościami występuje /.

WŁAŚCIWOŚCI LIST

Z pomocą kaskadowych arkuszy stylów można kontrolować następujące właściwości list:

- `type` lista;
- spособ wypunktowania,
- pozycje listy względem wypunktowania.

Typ listy

Dostępne wartości	<code>disc</code> <code>circle</code> <code>square</code> <code>decimal</code> <code>decimal-leading-zero</code> <code>lower-roman</code> <code>upper-roman</code> <code>lower-greek</code> <code>lower-latin</code> <code>upper-latin</code> <code>armenian</code> <code>georgian</code> <code>lower-alpha</code> <code>upper-alpha</code> <code>alpha</code> <code>none</code> <code>inherit</code>
Wartość domyślna	<code>disc</code>
Przykład	<code>ul {list-style-type:circle;}</code>
Dziedziczenie	tak
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `url` — adres pliku graficznego;
- `none` — brak wypunktowania listy;
- `inherit` — pobiera wartości z elementu nadzawanego — rodzica.

Pozycja listy względem wypunktowania

list-style-position

Dostępne wartości	<code>inside</code> <code>outside</code> <code>inherit</code>
Wartość domyślna	<code>outside</code>
Przykład	<code>ul {list-style-position:inside;}</code>
Dziedziczenie	tak
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `inside` — wypunktowanie jest schowane do trzeciego podpunktów;
- `outside` — wypunktowanie znajduje się na zewnątrz podpunktów;
- `inherit` — pobiera wartości z elementu nadzawanego — rodzica.

Zbiórzy zapis właściwości list

list-style

Dostępne wartości	<code><font-style-type></code> <code><font-style-position></code> <code><font-style-image></code>
Wartość domyślna	—
Przykład	<code>ul {list-style-type:url(punktor.gif)}</code>
Dziedziczenie	—
Obsługa	zob. właściwości poszczególnych stylów

Tablice informatyczne. CSS3

KOLOR I TŁO

Dzięki CSS możemy określić wygląd tła i koloru dla wybranych elementów. W zależności od potrzeb możemy zdefiniować tło będące wartością koloru lub obrazkiem.

Kolor elementu

color

Dostępne wartości:	<code>nazwa_tub_wartość_koloru inherit</code>
Wartość domyślna:	zależy od domylnych ustawień przeglądarki
Przykład:	<code>p { color:#FF0000; }</code>
Dziedziczenie:	tak

Obsługa: Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `nazwa_tub_wartość_koloru` – nazwa własna, zapis szesnastkowy lub RGB koloru;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Kolor tła

background-color

Dostępne wartości:	<code>nazwa_tub_wartość_koloru transparent inherit</code>
Wartość domyślna:	transparent
Przykład:	<code>p { background-color:#FF0000; }</code>
Dziedziczenie:	nie

Obsługa: Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `nazwa_tub_wartość_koloru` – nazwa własna, zapis szesnastkowy lub RGB koloru;
- `transparent` – niewidoczne tło;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Element graficzny jako tło

background-image

Dostępne wartości:	<code>url none inherit</code>
Wartość domyślna:	none
Przykład:	<code>body { background-image:url(tlo.jpg); }</code>
Dziedziczenie:	nie

Obsługa: Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `url` – adres pliku graficznego stanowiącego tło;
- `none` – brak tła z pliku graficznego;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Zatrzymanie graficznego tła

background-attachment

Dostępne wartości:	<code>scroll fixed inherit</code>
Wartość domyślna:	scroll
Przykład:	<code>body { background-attachment:fixed; }</code>
Dziedziczenie:	nie

Obsługa: Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `scroll` – ustawienie graficznego tła przy każdej krawędzi;
- `fixed` – ustawienie graficznego tła przy prawej krawędzi;
- `center` – ustawienie tła na środku;
- `top` – ustawienie tła przy górnej krawędzi;
- `bottom` – ustawienie tła przy dolnej krawędzi;
- `wartość_liczbową` – wartość bezwzględna określająca pozycję tła;
- `wartość_procentową` – wartość względna określająca pozycję tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Pudełkowy model formatowania elementów

KASKADOWE ARKUSZE STYLOWE

Kaskadowe arkusze stylów do formatowania wszystkich elementów wykorzystując model pudełka. W praktyce polega to na tym, że wszystkie elementy umieszcza się w dodatkowej przestrzeni o kształcie prostokątów.

W ramach każdego pudełka można kontrolować następujące właściwości:

- `margins` – odstęp dla całego pudełka;
- `obramowanie` – linie rozdzielające marginy i dopełnienia;
- `dopełnienia` – odstęp pomiędzy elementem a otaczającą go ramką; dopełnienia bardziej często nazywa się również marginami zewnętrznymi;
- `szerokość`;
- `wysokość`.

Oczywiście możliwe jest też określenie dla pojedynczego elementu wszystkich dostępnych właściwości.

MARGINY

Marginy to obszar występujący po zewnętrznej stronie obramowania elementu, obecny w każdym elemencie wykorzystanym na stronie WWW. Używając CSS, można kontrolować lewy, prawy, górny oraz dolny marginy.

Lewy margines

margin-left

Dostępne wartości:	<code>wartość_liczbową wartość_procentową auto inherit</code>
Wartość domyślna:	0
Przykład:	<code>p { margin-left:1cm; }</code>
Dziedziczenie:	nie

Obsługa: Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `wartość_liczbową` – wartość bezwzględna określająca rozmiar marginy;
- `wartość_procentowa` – wartość względna określająca rozmiar marginy;
- `auto` – przeglądarka automatycznie ustawi marginy;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Prawy margines

margin-right

Dostępne wartości:	<code>wartość_liczbową wartość_procentową auto inherit</code>
Wartość domyślna:	0
Przykład:	<code>p { margin-right:1cm; }</code>
Dziedziczenie:	nie

Obsługa: Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `wartość_liczbową` – wartość bezwzględna określająca rozmiar marginy;
- `wartość_procentowa` – wartość względna określająca rozmiar marginy;
- `auto` – przeglądarka automatycznie ustawi marginy;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Górny margines

margin-top

Dostępne wartości:	<code>wartość_liczbową wartość_procentową auto inherit</code>
Wartość domyślna:	0
Przykład:	<code>p { margin-top:1cm; }</code>
Dziedziczenie:	nie

Obsługa: Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `wartość_liczbową` – wartość bezwzględna określająca rozmiar marginy;
- `wartość_procentowa` – wartość względna określająca rozmiar marginy;
- `auto` – przeglądarka automatycznie ustawi marginy;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Dolny margines

margin-bottom

Dostępne wartości:	<code>wartość_liczbową wartość_procentową auto inherit</code>
Wartość domyślna:	0
Przykład:	<code>p { margin-bottom:1cm; }</code>
Dziedziczenie:	nie

Obsługa: Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `wartość_liczbową` – wartość bezwzględna określająca rozmiar marginy;
- `wartość_procentowa` – wartość względna określająca rozmiar marginy;
- `auto` – przeglądarka automatycznie ustawi marginy;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

- `scroll` – obrózek w tle przewijający się razem z tekstem;
- `fixed` – obrózek w tle jest nieruchomy podczas przewijania;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Kontrola powielania graficznego tła

background-repeat

Dostępne wartości:	<code>repeat repeat-x repeat-y no-repeat inherit</code>
Przykład:	<code>body { background-repeat:repeat; }</code>
Dziedziczenie:	nie
Obsługa:	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Dziedziczenie

Dziedziczenie:	nie
Obsługa:	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Obsługa

Obsługa:	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
Przykład:	<code>body { background-repeat:repeat; }</code>

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Wartość domyślna

Wartość domyślna:	repeat
Przykład:	<code>body { background-repeat:repeat; }</code>

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Przykład

Przykład:	<code>body { background-repeat:repeat; }</code>
Obsługa:	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Dziedziczenie

Dziedziczenie:	nie
Obsługa:	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Obsługa

Obsługa:	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
Przykład:	<code>body { background-repeat:repeat; }</code>

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Wartość domyślna

Wartość domyślna:	repeat
Przykład:	<code>body { background-repeat:repeat; }</code>

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Przykład

Przykład:	<code>body { background-repeat:repeat; }</code>
Obsługa:	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Obsługa

Obsługa:	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
Przykład:	<code>body { background-repeat:repeat; }</code>

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Wartość domyślna

Wartość domyślna:	repeat
Przykład:	<code>body { background-repeat:repeat; }</code>

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Przykład

Przykład:	<code>body { background-repeat:repeat; }</code>
Obsługa:	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Obsługa

Obsługa:	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
Przykład:	<code>body { background-repeat:repeat; }</code>

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Wartość domyślna

Wartość domyślna:	repeat
Przykład:	<code>body { background-repeat:repeat; }</code>

- `repeat` – powielanie tła w pionie i poziomie;
- `repeat-x` – powielanie tła w poziomie;
- `repeat-y` – powielanie tła w pionie;
- `no-repeat` – brak powielania tła;
- `inherit` – pobiera wartości z elementu nadzawanego — rodzica.

Przykład

Przykład:	<code>body { background-repeat:repeat; }</code>
Obsługa:	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

- `repeat` – pow

Tablice informatyczne. CSS3

Dolne obramowanie

border-bottom-width	
Dostępne wartości	none wartości liczbowe wartości procentowe thin medium thick inherit
Wartość domyślana	none
Przykład	p {border-bottom-width:5px;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• none — brak obramowania;	
• wartości liczbowe — wartość bezwzględna określająca grubość obramowania;	
• wartości procentowe — wartość względna określająca grubość obramowania;	
• thin — cienkie obramowanie;	
• medium — średnie obramowanie;	
• thick — grube obramowanie;	
• inherit — pobiera wartości z elementu nadającego — rodzica.	

Lewe obramowanie

border-left-width	
Dostępne wartości	none wartości liczbowe wartości procentowe thin medium thick inherit
Wartość domyślana	none
Przykład	p {border-left-width:5px;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• none — brak obramowania;	
• wartości liczbowe — wartość bezwzględna określająca grubość obramowania;	
• wartości procentowe — wartość względna określająca grubość obramowania;	
• thin — cienkie obramowanie;	
• medium — średnie obramowanie;	
• thick — grube obramowanie;	
• inherit — pobiera wartości z elementu nadającego — rodzica.	

Zbiorczy zapis szerokości obramowania

border-width	
Dostępne wartości	none wartości liczbowe wartości procentowe thin medium thick inherit
Wartość domyślana	none
Przykład	p {border-width:5px 2px;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

Kolor obramowania

Górne obramowanie	
border-top-color	nazwa lub wartość koloru transparent inherit
Wartość domyślana	według ustawień domyślnego koloru dla dokumentu
Przykład	p {border-top-color:red;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• nazwa lub wartość koloru — nazwa własna, zapis RGB lub wartość szesnastkowa koloru;	
• transparent — niewidoczne obramowanie;	
• inherit — pobiera wartości z elementu nadającego — rodzica.	

Prawe obramowanie

border-right-color	
Dostępne wartości	nazwa lub wartość koloru transparent inherit
Wartość domyślana	według ustawień domyślnego koloru dla dokumentu
Przykład	p {border-right-color:red;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• nazwa lub wartość koloru — nazwa własna, zapis RGB lub wartość szesnastkowa koloru;	
• transparent — niewidoczne obramowanie;	
• inherit — pobiera wartości z elementu nadającego — rodzica.	

Dolne obramowanie

border-bottom-color	
Dostępne wartości	nazwa lub wartość koloru transparent inherit
Wartość domyślana	według ustawień domyślnego koloru dla dokumentu
Przykład	p {border-bottom-color:red;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari

WYMIARY

Dzięki CSS możemy określić wymiary większości elementów na stronie. Ciekawostką jest możliwość dodatkowego zdefiniowania wymiarów minimalnych i maksymalnych.

Szerokość

width	
Dostępne wartości	wartość liczbową wartość procentową auto inherit
Wartość domyślana	auto
Przykład	p {width:30%;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• wartość liczbową — wartość bezwzględna określająca szerokość elementu;	
• wartość procentowa — wartość względna określająca szerokość elementu;	
• auto — automatyczne określenie szerokości elementu;	
• inherit — pobiera wartości z elementu nadającego — rodzica.	

Minimalna szerokość

min-width	
Dostępne wartości	wartość liczbową wartość procentową inherit
Wartość domyślana	0
Przykład	p {min-width:90px;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• wartość liczbową — wartość bezwzględna określająca minimalną szerokość elementu;	

• wartość procentowa — wartość bezwzględna określająca minimalną szerokość elementu;

• inherit — pobiera wartości z elementu nadającego — rodzica.

Wysokość

height	
Dostępne wartości	wartość liczbową wartość procentową auto inherit
Wartość domyślana	auto
Przykład	p {height:30%;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• wartość liczbową — wartość bezwzględna określająca wysokość elementu;	
• wartość procentowa — wartość względna określająca wysokość elementu;	
• auto — automatyczne określenie wysokości elementu;	
• inherit — pobiera wartości z elementu nadającego — rodzica.	

• wartość liczbową — wartość bezwzględna określająca minimalną szerokość elementu;

• wartość procentowa — wartość bezwzględna określająca minimalną szerokość elementu;

• inherit — pobiera wartości z elementu nadającego — rodzica.

Minimalna wysokość

min-height	
Dostępne wartości	wartość liczbową wartość procentową inherit
Wartość domyślana	0
Przykład	p {min-height:90px;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• wartość liczbową — wartość bezwzględna określająca minimalną wysokość elementu;	
• wartość procentowa — wartość względna określająca minimalną wysokość elementu;	
• inherit — pobiera wartości z elementu nadającego — rodzica.	

• wartość liczbową — wartość bezwzględna określająca minimalną wysokość elementu;

• wartość procentowa — wartość bezwzględna określająca minimalną wysokość elementu;

• inherit — pobiera wartości z elementu nadającego — rodzica.

Minimalna wysokość

Minimalna wysokość	
Dostępne wartości	wartość liczbową wartość procentowa none inherit
Wartość domyślana	0
Przykład	p {min-height:90px;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• wartość liczbową — wartość bezwzględna określająca minimalną wysokość elementu;	
• wartość procentowa — wartość względna określająca minimalną wysokość elementu;	
• none — brak zdefiniowanej wysokości maksymalnej elementu;	
• inherit — pobiera wartości z elementu nadającego — rodzica.	

• wartość liczbową — wartość bezwzględna określająca minimalną wysokość elementu;

• wartość procentowa — wartość względna określająca minimalną wysokość elementu;

• none — brak zdefiniowanej wysokości maksymalnej elementu;

• inherit — pobiera wartości z elementu nadającego — rodzica.

Maksymalna wysokość

max-height	
Dostępne wartości	wartość liczbową wartość procentowa none inherit
Wartość domyślana	none
Przykład	p {max-height:100px;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• wartość liczbową — wartość bezwzględna określająca maksymalną wysokość elementu;	
• wartość procentowa — wartość względna określająca maksymalną wysokość elementu;	
• none — brak zdefiniowanej wysokości maksymalnej elementu;	
• inherit — pobiera wartości z elementu nadającego — rodzica.	

Odstęp we wnętrzu tabeli

Odstęp we wnętrzu tabeli	
Dostępne wartości	wartość liczbową inherit
Wartość domyślana	0
Przykład	table {border-spacing:1cm 2cm;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• wartość liczbową — wartość określająca w sposób bezpośredni lub bezwzględny odstęp pomiędzy komórkami wewnętrz tabeli;	
• inherit — pobiera wartości z elementu nadającego — rodzica.	

Kontrola pustych komórek

Kontrola pustych komórek	
Dostępne wartości	show hide inherit
Wartość domyślana	show
Przykład	table {empty-cells:hide;}
Dziedziczenie	nie
Obsługa	Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• show — pokazuje puste komórki tabeli;	
• hide — ukrywa puste komórki tabeli;	
• inherit — pobiera wartości z elementu nadającego — rodzica.	

POZYCJONOWANIE ELEMENTÓW

pozycjonowanie elementów	
Dostępne wartości	static relative absolute fixed inherit
Wartość domyślana	static
Przykład	p {position:absolute; left:10px;}
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• static — pozycja elementu jest określana automatycznie;	
• relative — element zmienia pozycję i nie ma wpływu na pozostałe wtyry;	
• absolute — określa stałą pozycję względem boku okna przeglądarki elementu;	
• fixed — element zmienia stałą pozycję i nie zmienia jej nawet podczas przewijania dokumentu;	
• inherit — pobiera wartości z elementu nadającego — rodzica.	

• static — pozycja elementu jest określana automatycznie;

• relative — pozycja elementu zmienia się w oparciu o pozycję domyślną;

• absolute — określa stałą pozycję względem boku okna przeglądarki elementu;

• fixed — element zmienia stałą pozycję względem boku okna przeglądarki;

• inherit — pobiera wartości z elementu nadającego — rodzica.

• static — pozycja elementu jest określana automatycznie;

• relative — element zmienia pozycję i nie ma wpływu na pozostałe wtyry;

• absolute — określa stałą pozycję względem boku okna przeglądarki elementu;

• fixed — element zmienia stałą pozycję i nie zmienia jej nawet podczas przewijania dokumentu;

• inherit — pobiera wartości z elementu nadającego — rodzica.

• static — pozycja elementu jest określana automatycznie;

• relative — pozycja elementu zmienia się w oparciu o pozycję domyślną;

• absolute — określa stałą pozycję względem boku okna przeglądarki elementu;

• fixed — element zmienia stałą pozycję względem boku okna przeglądarki;

• inherit — pobiera wartości z elementu nadającego — rodzica.

• static — pozycja elementu jest określana automatycznie;

• relative — element zmienia pozycję i nie ma wpływu na pozostałe wtyry;

• absolute — określa stałą pozycję względem boku okna przeglądarki elementu;

• fixed — element zmienia stałą pozycję i nie zmienia jej nawet podczas przewijania dokumentu;

• inherit — pobiera wartości z elementu nadającego — rodzica.

• static — pozycja elementu jest określana automatycznie;

• relative — element zmienia pozycję i nie ma wpływu na pozostałe wtyry;

• absolute — określa stałą pozycję względem boku okna przeglądarki elementu;

• fixed — element zmienia stałą pozycję względem boku okna przeglądarki;

• inherit — pobiera wartości z elementu nadającego — rodzica.

• static — pozycja elementu jest określana automatycznie;

• relative — element zmienia pozycję i nie ma wpływu na pozostałe wtyry;

• absolute — określa stałą pozycję względem boku okna przeglądarki elementu;

• fixed — element zmienia stałą pozycję względem boku okna przeglądarki;

• inherit — pobiera wartości z elementu nadającego — rodzica.

• static — pozycja elementu jest określana automatycznie;

• relative — element zmienia pozycję i nie ma wpływu na pozostałe wtyry;

• absolute — określa stałą pozycję względem boku okna przeglądarki elementu;

• fixed — element zmienia stałą pozycję względem boku okna przeglądarki;

• inherit — pobiera wartości z elementu nadającego — rodzica.

• static — pozycja elementu jest określana automatycznie;

• relative — element zmienia pozycję i nie ma wpływu na pozostałe wtyry;

• absolute — określa stałą pozycję względem boku okna przeglądarki elementu;

• fixed — element zmienia stałą pozycję względem boku okna przeglądarki;

• inherit — pobiera wartości z elementu nadającego — rodzica.

• static — pozycja elementu jest określana automatycznie;

• relative — element zmienia pozycję i nie ma wpływu na pozostałe wtyry;

• absolute — określa stałą pozycję względem boku okna przeglądarki elementu;

• fixed — element zmienia stałą pozycję względem boku okna przeglądarki;

• inherit — pobiera wartości z elementu nadającego — rodzica.

• static — pozycja elementu jest określana automatycznie;

• relative — element zmienia pozycję i nie ma wpływu na pozostałe wtyry;

• absolute — określa stałą pozycję względem boku okna przeglądarki elementu;

• fixed — element zmienia stałą pozycję względem boku okna przeglądarki;

• inherit — pobiera wartości z elementu nadające

Tablice informatyczne. CSS3

EFEKTY WIZUALNE

Druża specyfikacja kaskadowych arkuszy stylów wprowadza kilka ciekawych możliwości związanych z dodawaniem efektów wizualnych na stronie. Dzięki odpowiednim stylom można kadrować i ukrywać elementy, które nie mieszczą się w zadanym obszarze, a także sterować tymi elementami.

Sterowanie wymiarowanymi elementami

overflow

Dostępne wartości	<code>visible hidden scroll auto inherit</code>
Wartość domyślna	<code>visible</code>
Przykład	<code>p {overflow:scroll;}</code>
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• <code>visible</code> — całość jest widoczna bez przycinania;	
• <code>hidden</code> — części wystające są niewidoczne;	
• <code>scroll</code> — części wystające nie są widoczne na pierwszym planie, ale można je przeglądać za pomocą wózków;	
• <code>auto</code> — przeglądarka sama zadecyduje o rozmiarze elementu wystającego;	
• <code>inherit</code> — pobiera wartości z elementu nadzawanego — rodzica.	

Kadrowanie

clip

Dostępne wartości	<code>rect (top, right, bottom, left)</code> <code>auto inherit</code>
Wartość domyślna	<code>auto</code>
Przykład	<code>p {clip:rect(10px, 150px, 150px, 10px,);}</code>
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• <code>rect</code> — pozwala na dokładne określone części widocznej (należy podać współrzędne po szczególnych narożnikach względem lewej krawędzi okna przeglądarki);	
• <code>auto</code> — cały element objęty jest działaniem stylu;	
• <code>inherit</code> — pobiera wartości z elementu nadzawanego — rodzica.	

DRUKOWANIE

Drukowanie stron WWW nigdy nie należało do łatwych i wiązało się z wieloma problemami. Na szczęście dzięki CSS możemy sporządzić specjalne arkusze, a następnie wykorzystać je do formatowania zawartości wydruku, dzięki czemu wydruk wygląda poprawnie.

Tworzenie oddzielnych stylów dla wydruku

@media

Dostępne wartości	<code>all braille embossed handheld print projection screen speech tty tv</code>
Wartość domyślna	—
Przykład	<code>@media print { }</code>
Dziedziczenie	—
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer, Chrome, Safari
• <code>all</code> — arkusz uniwersalny;	
• <code>braille</code> — arkusz dla stron przygotowanych dla osób niewidomych;	
• <code>embossed</code> — arkusz dla drukarki braila;	
• <code>handheld</code> — arkusz dla urządzeń przenośnych;	
• <code>print</code> — arkusz dla druku;	
• <code>projection</code> — arkusz dla prezentacji;	
• <code>screen</code> — arkusz do prezentacji na ekranie monitora;	
• <code>speech</code> — arkusz dla stron czytanych przez syntetyzator mowy;	
• <code>tty</code> — arkusz dla terminali i innych urządzeń o ograniczonych rozmiarach ekranu;	
• <code>tv</code> — arkusz dla stron przeglądanych na ekranie telewizora.	

Lamanie strony

Przed elementem

page-break-before

Dostępne wartości	<code>auto always avoid left right inherit</code>
Wartość domyślna	<code>auto</code>
Przykład	<code>h1 {page-break-before:always;}</code>
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, MS Internet Explorer (częściowo), Chrome, Safari
• <code>auto</code> — lamańskie strony jest realizowane automatycznie przez przeglądarkę;	
• <code>always</code> — strona jest zawsze lamanańska;	
• <code>avoid</code> — rozpoczęta nową stroną przed tym elementem;	
• <code>left</code> — lamańskie strony w przypadku druku dwustronnego;	
• <code>right</code> — lamańskie strony w przypadku druku dwustronnego;	
• <code>inherit</code> — pobiera wartości z elementu nadzawanego — rodzica.	

INTERFEJS UŻYTKOWNIKA

Druża specyfikacja kaskadowych arkuszy stylów wprowadza możliwość kontroli wyglądu elementów stanowiących część interfejsu użytkownika. Dzięki nowym stylom można kontrolować:

- wygląd kursorów;
- kontury elementów;
- ustawienia kolorów pobrane z komputera użytkownika.

Kursory

cursor

Dostępne wartości	<code>auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help progress</code>
Wartość domyślna	<code>auto</code>
Przykład	<code>a {cursor:help;}</code>

Ukrywanie elementów

visibility

Dostępne wartości	<code>visible hidden collapse inherit</code>
Wartość domyślna	<code>visible</code>
Przykład	<code>p {visibility:hidden;}</code>
Dziedziczenie	nie

- `visible` — formowany element jest widoczny;
- `hidden` — formowany element nie jest widoczny;
- `collapse` — formowany element nie jest widoczny;
- `inherit` — pobiera wartości z elementu nadzawanego — rodzica.

Kontury

Kontury są wykorzystywane do tego by lepiej podkreślić widoczność elementów na stronie. Specyfikacja CSS umożliwia kontrolę następujących właściwości konturów:

- grubość;
- stylu;
- koloru.

Grubość konturu

outline-width

Dostępne wartości	<code>none wartości liczbowe wartości procentowe thin medium thick inherit</code>
Wartość domyślna	<code>medium</code>
Przykład	<code>a {outline-width:5px;}</code>
Dziedziczenie	nie

- `none` — brak konturu;
- `wartość liczbowa` — wartość bezwzględna określająca grubość konturu;
- `wartość procentowa` — wartość względna określająca grubość konturu;
- `thin` — cienki kontur;
- `medium` — kontur o średniej grubości;
- `thick` — gruby kontur;
- `inherit` — pobiera wartości z elementu nadzawanego — rodzica.

Styl konturu

outline-style

Dostępne wartości	<code>none dotted dashed solid double groove ridge inset outset inherit</code>
Wartość domyślna	<code>none</code>
Przykład	<code>p {outline-style:dotted;}</code>

- `none` — brak oznaczenia cytat;
- `"znak"` — `"znak"` — wstawia początkowy i końcowy znak cytat;
- `inherit` — pobiera wartości z elementu nadzawanego — rodzica.

ZAWARTOŚĆ GENEROWANA I LICZNIKI

Druża specyfikacja CSS pozwala na generowanie zawartości wtryny oraz dodawanie liczników da paragrafów lub całych bloków tekstu. Niestety obsługa tych rozwiązań w chwilie obecnej jest realizowana poprawnie jedynie w przeglądarkach Opera i Mozilla Firefox.

Zawartość generowana

Do stworzenia zawartości generowanej niezbędne jest zastosowanie następujących elementów:

- pseudoelementów `:after :before;`
- reguły `content`.

Reguła content

content

Dostępne wartości	<code>normal none "dowolna treść" counter open-quote close-quote no-open-quote no-close-quote inherit</code>
Wartość domyślna	<code>normal</code>
Przykład	<code>h1:after {content:"informacja widoczna przed nagłówkiem";}</code>
Dziedziczenie	nie
Obsługa	Opera, Mozilla Firefox, Chrome, Safari
• <code>normal</code> — domyślne ustawienie;	
• <code>brak</code> — brak;	
• <code>"dowolna treść"</code> — dowolna treść zdefiniowana przez autora strony;	
• <code>counter</code> — licznik;	
• <code>open-quote close-quote</code> — wstawia początkowy i końcowy cudzysłów;	
• <code>no-open-quote no-close-quote</code> — nie wstawia początkowego i końcowego cudzysłowa;	
• <code>inherit</code> — pobiera wartości z elementu nadzawanego — rodzica.	

Cytowanie

quotes

Dostępne wartości	<code>none "znak" "znak" inherit</code>
Wartość domyślna	<code>według ustawień przeglądarki</code>

- `none` — nie zmienia wartości licznika;
- `"licznik wartość wzrostu"` — nazwa licznika, który ma zostać wygenerowany;
- `inherit` — pobiera wartości z elementu nadzawanego — rodzica.

CSS3 — PRZEGŁĄD NOWYCH FUNKCIJ

Selektory

Selektor atrybutu z możliwością dopasowania ciągu znaków

Selektor atrybutu z możliwością dopasowania ciągu znaków daje możliwość odwołania się do wybranego elementu strony na podstawie fragmentu ciągu znaków występujących jako atrybut. Dostępne są trzy sposoby odwołania się do ciągu znaków. Pierwszy z selektorów wykorzystujących ciągi znaków pozwala na podpięcie stylu tekstu, od którego zaczyna się dany ciąg znaków. Zwrót uwagę na znak `*` w zapisie selektora. Poniżej przykład dotyczy wszystkich znaczników `H1`, w których występuje atrybut `title` zaczynający się od znaków `pie`.

```
H1[title*="pie"] {
    color:red;
}
```

Drużi z selektorów wykorzystujących ciągi znaków pozwala na podpięcie stylu tekstu, który kończy się dany ciąg znaków. Zwrót uwagę na znak `$` w zapisie selektora. Poniżej przykład dotyczy wszystkich znaczników `H1`, w których występuje atrybut `title` zaczynający się od znaków `naglowek`.

```
H1[title$="naglowek"] {
    font-style:italic;
}
```

Trzeci z selektorów wykorzystujących ciągi znaków pozwala na podpięcie

stylu tekstu, który występuje w dolnym miejscu ciągu znaków.

Zwrót uwagę na znak `*` w zapisie selektora. Poniżej przykład dotyczy wszystkich znaczników `H1`, w których występuje atrybut `title` zaczynający się od znaków `napowietrza`.

```
H1[title^="napowietrza"] {
    text-decoration:underline;
}
```

Zawierająca ciąg znaków `naglowek`.

```
H1[title*= "naglowek"] {
    font-style:italic;
}
```

Selektor ogólnego „rodzeństwa”

Selektor ogólnego rodzeństwa składa się z dwóch prostych elementów selektora

rozdzielonych znakiem `:` (tylko) i pasuje do tych elementów selektora drugiego (w poniższym przykładzie jest to `P`), które są poprzedzone

elementami pierwszego selektora (w poniższym przykładzie jest to `H1`). Oba elementy muszą mieć wspólnego rodzica (w naszym przypadku

jest do `DIV`) i dodatkowo drugi element nie musi być bezpośrednio poprzedzony pierwszym elementem.

```
<div>
<h1></h1>
<p></p>
<h1></h1>
<p></p>
<h1></h1>
<p></p>
<div>
```

Przykład selektora ogólnego rodzeństwa wygląda następująco:

```
H1 ~ P
{
    color:red;
    font-weight:bold;
}
```

Pseudoklasy

Pseudoklasa :target

Stosując pseudoklasę `:target`, możemy wymusić formatowanie elementów będących kotwicami na stronie. Przykładem jest, gdy odrnioski prowadzący do kotwicy zostanie wybrany,

a adres będzie mieć postać: <http://adres.plik.html#kotwica>.

Odpowiedni arkusz stylów może mieć następującą postać:

```
*:target
{
    text-decoration:none;
    color:green;
}
```

Pseudoklasa :enabled i :disabled

Stosując pseudoklasę `:enabled` oraz `:disabled`, możemy kontrolować wygląd elementów interfejsu użytkownika (formularzy), w zależności od tego, czy są one dostępne, czy też nie.

```
input[type="text"] :enabled
{
    background:#COFFC0;
}
input[type="text"] :disabled
{
    background:#FFCOC0;
}
```

Tablice informatyczne. CSS3

Pseudoklasa :checked

Dzięki pseudoklasie `:checked` możemy kontrolować wygłąd aktywnych pól formularza typu `radio` lub `checkbox`.

```
input:checked
{
    border:1px solid red;
    color:green;
    width:25px;
    height:25px;
    background-color:red;
}
```

Pseudoklasa :root

Pseudoklasa `:root` należy do grupy pseudoklas strukturalnych, z pomocą których możemy odwołać się do struktury dokumentu XHTML/HTML; ta struktura jest poza zakresem klasycznych selektorów. Omawiana pseudoklasa odnosi się do korzenia struktury, który w przypadku dokumentów XHTML/HTML jest elementem `HTML`, i zawsze działa w całym dokumencie.

```
:root
{
    background-color:#COFFCO;
    color:navy;
}
```

Pseudoklasa :nth-child()

Dzięki pseudoklasie `:nth-child()` możemy odwołać się do elementu, który ma przed sobą określona w nawiasie liczbę rodzeństwa w strukturze dokumentu. Wartość umieszczona w nawiasie może być numerem, słowem kluczowym lub formułą.

```
p:nth-child(3)
{
    color:red;
}
UL LI:nth-child(even)
{
    color:red;
    font-style:italic;
}
OL LI:nth-child(odd)
{
    color:green;
    font-weight:bolder;
}
LI:nth-child(2n+5)
{
    color:red;
    font-style:italic;
    font-weight:bolder;
}
```

Pseudoklasa :nth-last-child()

Pseudoklasa `:nth-last-child()` działa podobnie jak opisana powyżej pseudoklasa `:nth-child()`, z tą różnicą, że liczenie elementów odwija się od końca, nie zaś od początku, jak było w pseudoklasie `:nth-child()`.

```
LI:nth-last-child(2n+5)
{
    color:red;
    font-style:italic;
    font-weight:bolder;
}
```

Pseudoklasa :nth-of-type()

Stosując pseudoklasę `:nth-of-type()`, możemy się odwołać do ścieżki określonego elementu, który ma przed sobą określona w nawiasie liczbę rodzeństwa w strukturze dokumentu. Wartość umieszczona w nawiasie może być: `numerem`, słowem kluczowym lub `formułą`. Inny słowy, `:nth-of-type()` działa podobnie jak pseudoklasa `:nth-child()`, ale odnosi się wyłącznie do elementów określonego typu, np. do akapitów.

```
P:nth-of-type(3)
{
    color:red;
}
```

Pseudoklasa :nth-last-of-type()

Pseudoklasa `:nth-last-of-type()` działa podobnie jak opisana powyżej pseudoklasa `:nth-of-type()`, z tą różnicą, że liczenie ścieżki określonych elementów odwija się od końca, nie zaś od początku, jak było w pseudoklasie `:nth-of-type()`. Oczywiście również tutaj pomiędzy nawiasami możemy stosować numer, słowa kluczowe lub formuły.

```
H1:nth-last-of-type(odd)
{
    color:red;
    font-style:italic;
    font-weight:bolder;
}
```

Pseudoklasa :last-child

Kolejną strukturalną pseudoklasą jest `:last-child`, dzięki której możemy odnieść się z formatowaniem do elementu będącego ostatnim dzieckiem swojego rodzica. Przykładowy arkusz stylów wykorzystujący pseudoklasę `:last-child` może mieć postać:

```
LI:last-child
{
    color:red;
    font-style:italic;
    font-weight:bolder;
}
```

Pseudoklasa :first-of-type

Pseudoklasa `:first-of-type` to kolejny selektor strukturalny, dzięki któremu możemy przypisać formatowanie do pierwszego ścieżki określonego elementu mającego wspólnego rodzica. Mówiąc prościej: kiedy korzystamy z prezentowanej pseudoklasy, możemy podać formatowanie na przykład pierwszą pozycję listy.

```
LI:first-of-type
{
    color:red;
    font-style:italic;
    font-weight:bolder;
}
```

Pseudoklasa :last-of-type

Pseudoklasa `:last-of-type` to następny selektor strukturalny, dzięki któremu możemy przypisać formatowanie do ostatniego ścieżki określonego elementu mającego wspólnego rodzica. Inny słowy, stojącą za prezentowanej pseudoklasy, możemy podać formatowanie na przykład ostatnią pozycję listy.

```
LI:last-of-type
{
    color:red;
    font-style:italic;
    font-weight:bolder;
}
```

Pseudoklasa :only-child

Pseudoklasa `:only-child` pozwala podać formatowanie tylko te elementy, które są pierwzym i jedynym dzieckiem swojego rodzica. Przykładowy arkusz korzystający z prezentowanej pseudoklasy może mieć postać:

```
LI:only-child
{
    color:red;
    font-style:italic;
    font-weight:bolder;
}
```

Pseudoklasa :only-of-type

Pseudoklasa `:only-of-type` działa podobnie jak opisana nieco wcześniej `:only-child`, z tą różnicą, że dotyczy tylko tych elementów, które są jedynymi dziećmi danego typu swojego rodzica.

```
P:only-of-type
{
    color:red;
    font-style:italic;
    font-weight:bolder;
}
```

Pseudoklasa :empty

Ostatnią strukturalną pseudoklasą jest `:empty`, która odpowiada za formatowanie elementów niemających swoich dzieci i niezawierających innych elementów. Dlatego z trzech poniższych przykładów tylko pierwszy element możemy uznać za pusty.

```
<div></div>
<div>Lorem ipsum dolor sit amet, consetetur adipiscing elit.</div>
<div>img src="obrazek.jpg" /></div>
```

Odpowiedni arkusz wykorzystujący omawianą pseudoklasę może mieć następującą postać.

```
DIV:empty
{
    width:80%;
    height:200px;
    background-color:silver;
    margin:auto;
}
```

Pseudoklasa :not()

Pseudoklasa `:not()` w nawiasie zawiera argument, który jest selektorem prostym, np. może to być `p`. Następnie wymusza formatowanie tych elementów, które nie są argumentem zapisanym w nawiasie.

```
:not(p)
{
    color:red;
}
```

Pseudoelementy

Pseudoelement ::selection

Pseudoelement `::selection` służy do formatowania tekstu zaznaczonego przez użytkownika strony. Wraz z tym elementem możemy używać następujących stylów:

- `color`,
- `background`,
- `cursor`,
- `outline`.

Oto przykład wykorzystania omawianego pseudoelementu.

```
P::selection
{
    color:red;
}
```

Nowe wybrane style

Pomimo że trzecia specyfikacja kaskadowych arkuszy stylów to ciągle wersja robocza, możemy w niej znaleźć wiele nowych stylów i właściwości, dzięki którym formatowanie tworzonej strony jest jeszcze wygodniejsze. W dalszej części opisalem wybrane style, które działają w nowych wersjach przeglądarki dostępnych na rynku w chwilie, gdy opracowymy te część tablic CSS.

Cieniowanie tekstu

text-shadow

Dostępne wartości: `none` | `długość` | `kolor`

Wartość domyślna: `none`

Przykład: `p {text-shadow: 2px 2px 5px red;}`

Dziedziczenie: `nie`

Obsługa: Opera, Mozilla Firefox, Safari, Chrome

• `none` — brak cienia;

• `długość` — wartość liczbowa określająca parametry cienia;

• `kolor` — definicja koloru cienia.

Obsługa: Opera, Mozilla Firefox, Safari, Chrome

• `auto` — rozmiar tła (szerokość i wysokość) będzie automatycznie dopasowywany do okna przeglądarki;

• `cover` — staje obramę przy zachowaniu proporcji boków do najmniejszych wartości, tak by dane polary temu obrazu;

• `contain` — staje obramę przy zachowaniu proporcji boków do największych wartości, tak by dane polary temu obrazu;

Kontrola styku tła i obramowania

background-clip

Dostępne wartości: `padding-box` | `border-box`

Wartość domyślna: `border-box`

Przykład: `div {background-clip: border-box;}`

Dziedziczenie: `nie`

Obsługa: Chrome

• `padding-box` — do niej wchodzi pod ramkę, jeżeli jest ona ustalona;

• `border-box` — do niej wchodzi pod ramkę, jeżeli jest ona ustalona;

Wieloelementowe tło

Jedna z ciekawych nowości dostępnych w wersji roboczej trzeciej specyfikacji CSS jest możliwość umieszczenia kilku fragmentów graficznego dla tego samego elementu. Operacja nie jest skomplikowana i może wyglądać tak:

DIV

```
{
    width:500px;
    height:500px;
    background-color:#COFFCO;
    background-image: url(kolo.jpg);
    ur1(kula.jpg);
    background-position: top left, bottom right;
    background-repeat: repeat-x, no-repeat;
}
```

Zwrót uwagę na pogrubione elementy listingu. Korzystamy z tych samych stylów, które omawiamy przy okazji wstawiania pojedynczego tła, ale po przecinku dopisujemy kolejne wartości.

Zaokrąglanie narożników — lewy górny

border-top-left-radius

Dostępne wartości: `liczba` | `percent`

Wartość domyślna: `0`

Przykład: `div {border-top-left-radius: 5px 10px;}`

Dziedziczenie: `nie`

Obsługa: Chrome

Zaokrąglanie narożników — prawy górny

border-top-right-radius

Dostępne wartości: `liczba` | `percent`

Wartość domyślna: `0`

Przykład: `div {border-top-right-radius: 5px 10px;}`

Dziedziczenie: `nie`

Obsługa: Chrome

Zaokrąglanie narożników — lewy dolny

border-bottom-left-radius

Dostępne wartości: `liczba` | `percent`

Wartość domyślna: `0`

Przykład: `div {border-bottom-left-radius: 5px 10px;}`

Dziedziczenie: `nie`

Obsługa: Chrome

Zaokrąglanie narożników — prawy dolny

border-bottom-right-radius

Dostępne wartości: `liczba` | `percent`

Wartość domyślna: `0`

Przykład: `div {border-bottom-right-radius: 5px 10px;}`

Dziedziczenie: `nie`

Obsługa: Chrome

Zaokrąglanie narożników — zapis zbiorczy

border-radius

Dostępne wartości: `wartość` | `liczba` | `percent`

Wartość domyślna: `0`

Przykład: `div {border-radius: 5px 10px;}`

Dziedziczenie: `nie`

Obsługa: Chrome

Zmiana rozmiaru wymiarowanego elementu

resize

Dostępne wartości: `none` | `both` | `horizontal` | `vertical` | `inherit`

Wartość domyślna: `none`

Przykład: `div {resize:both;}`

Dziedziczenie: `nie`

Obsługa: Chrome, Safari, Mozilla Firefox, Opera

• `none` — nie ma możliwości zmiany rozmiaru;

• `both` — użytkownik może zmienić rozmiar w pionie i poziomie;

• `horizontal` — użytkownik może zmienić rozmiar w poziomie;

• `vertical` — użytkownik może zmienić rozmiar w pionie;

• `inherit` — przejęta zostanie wartość ustawiona dla rodzica formatowanego elementu.

Offset konturu

outline-offset

Dostępne wartości: `długość` | `inherit`

Wartość domyślna: `0`

Przykład: `div {outline-offset:1px;}`

Dziedziczenie: `nie`

Obsługa: Chrome, Safari, Mozilla Firefox, Opera