

OSADZANIE STYLÓW 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>

Styl zagnieżdżony umieszcza się w nagłówku strony. Swym działaniem może obejmować cały dokument w zależności od konstrukcji stylu.

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

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>
```

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

```
<?xml-stylesheet type="text/css"
href="arkusz.css" ?>
```

Dla zachowania zgodności dokumentów XHTML zaleca się umieszczenie podwójnej deklaracji wywołującej arkusz stylów.

```
<?xml-stylesheet type="text/css" />
<?xml-stylesheet 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">
@import url("arkusz.css");
</style>
</head>
<body>
<p>Tekst akapitu</p>
</body>
</html>
```

Zewnętrzny arkusz jest zwyczajnym plikiem tekstowym, a jego zawartość może wyglądać np. tak jak na poniższym przykładzie. Ważne jest tylko to, aby plik miał rozszerzenie `.css`, np. `arkusz.css`.

```
/* To jest przykład zewnętrznego arkusza stylów */
{
color:green;
font-size:15pt;
}
```

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 jednostki `ex` wiąże się z rodzajem użytej czcionki.

Jak zapewne wiesz, 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 — cale

```
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 równa 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;}
```

Milimetr 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 punktów.

Jednostki czasu i jednostki kątowe

Jednostki służące do określania 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 czytanimi fragmentami.

Jednostki kątowe to:

deg — stopnie

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

grad — gradiany

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

rad — radiany

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

Jednostki te używane są do określania pozycji źródła dźwięku.

KOLORY

Nazwa własna koloru

Do 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 1. Wybrane kolory z przypisanymi nazwami wchodzące w skład bezpiecznej palety 216 kolorów

Nazwa koloru	Zapis szesnastkowy
Black	#000000
Silver	#C0C0C0
Gray	#808080
White	#FFFFFF
Maroon	#800000
Red	#FF0000
Purple	#800080
Fuchsia	#FF00FF
Green	#008000
Lime	#00FF00
Olive	#808000
Yellow	#FFFF00
Navy	#000080
Blue	#0000FF
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 dziewięć.

Zapis w postaci RGB może przybierać następującą postać:

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

Oczywiście możemy jeszcze zapisać kolor w postaci RGB, definiując procent nasycenia 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

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

Selektor prosty

Selektorem prostym jest znacznik języka XHTML/HTML przypisany do elementu, który chcemy 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 `<span></span>` i znajdującego 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 zadziała w przypadku, gdy wewnątrz bloku tekstu zostanie umieszczony znacznik `<span></span>`.

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

Selektor „rodzeństwa”

Na podstawie tego rodzaju selektora możemy przypisać styl elementowi bezpośrednio sąsiadującemu z innym. By warunek 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 ściśle określony atrybut. Przykładowy styl zadziała w przypadku nagłówka stopnia pierwszego o określonym atrybucie `title`.

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

Grupowanie selektorów

Selektory można dowolnie grupować, jeżeli pewne wartości mają obowiązywać dla każdego z nich. Dzięki zbiorczemu zapisowi możemy znacznie uprościć strukturę arkusza i pracę z kodem witryny. Poszczególne selektory wchodzące w skład grupy rozdzielamy przeciskami. Poniżej przedstawiono przykład zgrupowanych 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 sposób prezentacji elementów na stronie.

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

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

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

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

```
H1#pochyly
{
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;
}
```

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

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

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

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

Pseudoklasy

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

Pseudoklasa `:link` dotyczy formatowania odnośnika w stanie nienaruszonym.

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

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

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

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

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

Pseudoklasa `:active` odpowiada za ustalenie wyglądu aktywnego elementu na stronie.

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

Pseudoklasa `:focus` pozwala na wyróżnienie aktualnie wyświetlanego elementu, np. *odśladcy*.

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

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

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

Pseudoklasa `:lang` jest wykorzystywana do określania formatowania dokumentów wielojęzycznych:

```
P:lang(pl)
{
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 grupę selektorów stanowią pseudoelementy, które pozwalają nam na przypisanie kolejnych wyśmienitych 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.

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

Pseudoelement `:first-line` w przeciwieństwie do poprzednika obejmuje formatowaniem całą pierwszą linię, a nie tylko pierwszą literę.

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

Pseudoelementy `:after` oraz `:before` służą do umieszczania tekstu przed formatowanym elementem strony i za nim.

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

P:after

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

*Dalsza część książki dostępna w wersji
pełnej.*

