

OSADZANIE STYLÓW W DOKUMENCIE

Style wewnętrzne

Styl lokalny umieszcza się w wnętrzu znacznika formatowanego elementu. Działanie stylu obejmuje wyłącznie element, w którym się znajduje.

```
<p style="color:green;font-size:15pt;">Treść dokumentu</p>
Styl zagnieźdzony umieszcza się w nagłówku strony. Swym działaniem może obejmować cały dokument w zależności od konstrukcji.
<html>
<head>
<style type="text/css">
p
{
color:green;
font-size:15pt;
}
</style>
</head>
<body>
<p>Tekst akapitu</p>
</body>
</html>
```

Style zewnętrzne

Zewnętrzny arkusz stylów umieszcza się w nagłówku <head> dokumentu. Działaniem styl zewnętrzny obejmuje całą zawartość strony i może być wykorzystany również na innych 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. Oczywiście oba wpisy muszą znaleźć się w nagłówku kodu witryny <head></head>.
<link rel="stylesheet" type="text/css"
href="arkusz.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");
p
{
color:green;
}
</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 */
p
{
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 określeniu zależności pomiędzy poszczególnymi wielkościami.

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 o wielkości 2ex.

% — procenty
P {font-size:10%;}
Procenty służą do określenia 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 naszym przypadku 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;}
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 wzbogaconych o możliwość czytania dokumentu przez komputer oraz obłąki mediów.

Jednostki czasu to:
ms — milisekunda
H5 {pause-after:100ms;}
s — sekunda
H5 {pause-after:10s;}
Obie jednostki są używane do określenia odstępu pomiędzy czytanyimi fragmentami.

Jednostki kątowe to:
deg — stopnie
H5 {elevation:10deg;}
grad — gradiany
H5 {elevation:10grad;}
rad — radiany
H5 {elevation:10rad;}
Jednostki te są używane do określenia pozycji źródła dźwięku.

KOLORY

Nazwa własna koloru

Do określenia kolorów, podobnie jak w języku HTML, możemy użyć nazw kolorów, np. red, green czy blue.
DIV {color:red;}
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	Fuchsja	#FF00FF
Black	#000000	Green	#008000
Silver	#C0C0C0	Lime	#00FF00
Gray	#808080	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 (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 dokładnie tym samym, co zapis szesnastkowy doskonale znany wszystkim, którzy mieli do czynienia z komputerami.
DIV {color:#FF0000;}

Pobieranie koloru użytkownika

Specyfikacja CSS2 dopuszcza dodatkowo zastosowanie odwołań do kolorów użytych przez osobę oglądającą stronę. Przykładowo można stworzyć stronę mającą to identyczne jak kolor pulpit lub okna.

Tabela 2. Podstawowe kolory systemowe, z których możemy skorzystać w CSS2

Wartość	Opis i przykład
Background	Pobiera kolor pulpit: BODY {background-color:background;}
Window	Pobiera kolor tła z okna: BODY {background-color:window;}
Windowframe	Pobiera kolor ramki okna: BODY {background-color:windowframe;}
Windowtext	Pobiera kolor tekstu z okna: P {color:windowtext;}
ThreeDLightShadow	Pobiera kolor jasnego cienia z elementów 3D: BODY {background-color:ThreeDLightShadow;}
ThreeDarkShadow	Pobiera kolor ciemnego cienia z elementów 3D: BODY {background-color:ThreeDarkShadow;}
ThreeDFace	Pobiera kolor powierzchni z elementów 3D: BODY {background-color:ThreeDFace;}
ThreeDHighlight	Pobiera kolor podświetlenia z elementów 3D: BODY {background-color:ThreeDHighlight;}
ActiveCaption	Pobiera kolor belki tytułowej aktywnego okna: P {color:ActiveCaption;}
InactiveCaption	Pobiera kolor belki tytułowej nieaktywnego okna: P {color:InactiveCaption;}
ActiveBorder	Pobiera kolor obramowania głównego okna: P {color:ActiveBorder;}
AppWorkspace	Pobiera kolor tła z aplikacji pracującej na wielu oknach: P {color:AppWorkspace;}
ButtonFace	Pobiera kolor z powierzchni przycisków: P {color:ButtonFace;}
ButtonHighlight	Pobiera kolor z obramowania przycisków: P {color:ButtonHighlight;}
ButtonShadow	Pobiera kolor z cienia przycisku: P {color:ButtonShadow;}
ButtonText	Pobiera kolor z tekstu przycisku: P {color:ButtonText;}
CaptionText	Pobiera tekst z nagłówków: P {color:CaptionText;}
GrayText	Pobiera kolor z nieaktywnego tekstu: P {color:GrayText;}
Highlight	Pobiera kolor z wybranego elementu: P {color:Highlight;}
HighlightText	Pobiera kolor z tekstu w wybranych elementach: P {color:HighlightText;}
InfoBackground	Pobiera kolor z tła podpowiedzi: P {color:InfoBackground;}
InfoText	Pobiera kolor z tekstu podpowiedzi: P {color:InfoText;}
Menu	Pobiera kolor z tła menu: P {color:Menu;}
MenuItem	Pobiera kolor z tekstu znajdującego się w menu: P {color:MenuItem;}
Scrollbar	Pobiera kolor tła z belki przewijania: P {color:Scrollbar;}
ThreeDarkShadow	Pobiera kolor z ciemnego cienia belki przewijania: P {color:ThreeDarkShadow;}

SELEKTORY

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

Selektor prosty

Selektorem prostym jest znacznik języka HTML przypisany do elementu, który chcesz w danym momencie formatować. Styl zdefiniowany dla przykładowego selektora P będzie odnosił się wyłącznie dla akapitów zamkniętych w znacznik <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 ustawienie formatowania dla elementu występującego w wnętrzu innego znacznika. Przykład wymusza nadanie formatowania fragmentowi tekstu zamkniętego w i identyfikację się w znaczniku <h1></h1>.

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

Selektor „dziecka”

Selektory dziecka są wynikiem zależności panujących pomiędzy poszczególnymi znacznikami języka HTML. Przykładowy styl zadziała w przypadku, gdy wewnątrz bloku tekstu zostanie umieszczony znacznik .

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

Selektor „rodzeństwa”

Na podstawie tego rodzaju selektora możemy przypisać styl elementowi bezpośrednio sąsiadującemu z innymi. 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 HTML posiadającego ściśle określony atrybut. Przykładowy styl zadziała dla nagłówka stopnia pierwszego posiadającego określony atrybut title.

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

Grupowanie sektorów

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 przecinkami. Poniżej przedstawiono przykład zgrupowanych selektorów h1, h2 oraz p.

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

Identyfikatory

Identyfikator pozwala na przypisanie formatowania do wybranego znacznika posiadającego 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 posiadać dodatkowy atrybut id.
<p id="pochyly">Treść akapitu</p>
Możliwy 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 HTML. W odróżnieniu od identyfikatora klasa może być wykorzystywana wielokrotnie.

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

Aby skorzystać z przykładowej klasy, znacznik musi mieć zdefiniowany atrybut class z nazwą klasy.
<p class="moja">Treść akapitu</p>
Klasa może zostać również powiązana z konkretnym znacznikiem HTML.

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

