

Wstęp do tworzenia stron WWW

Lekcja 10. Podstawy stylów CSS. Formatowanie tekstu.

1. Rodzaje selektorów

Selektor typu (selektor pojedynczego elementu)

Selektor typu pozwala wybrać pojedynczy element dokumentu (X)HTML zdefiniowany przez określony znacznik i nadać mu definicję (za pomocą *cechy* oraz *wartości*), która modyfikuje jego wygląd lub sposób zachowania. Przykłady:

`p {font-size: 24 pt;}` – modyfikacja rozmiaru czcionki dla akapitu

`div {background-color:yellow;}` – modyfikacja koloru tła dla warstwy

Selektor uniwersalny

Selektor uniwersalny pozwala ustalić określone atrybuty dla wszystkich elementów strony, a więc dla różnych selektorów typu. Można za jego pomocą nadać to samo formatowanie dla wszystkich elementów na całej stronie, niezależnie od ich typu (p, div, li itd.). Schemat definicji ma następującą postać:

```
*{  
    definicje  
}
```

Przykładowo, dodając poniższą definicję do sekcji <head>:

```
<style type="text/css">  
*{  
    font-size:22pt  
}  
</style>
```

to wielkość czcionki dla każdego elementu strony (p, div, li itd.) będzie wynosić 22 pt.

Selektor klasy

Selektor klasy umożliwia zdefiniować zestaw reguł, które będzie można później przypisać wybranym elementom strony WWW. Taki zestaw reguł określany jest mianem klasy. Jest to bardzo pomocny typ selektora, ponieważ pozwala zmienić właściwości wybranych elementów witryny niezależnie od znaczników jakie zostały użyte do ich stworzenia.

```
selektor.nazwa_klasy {  
    definicje  
}
```

Jeżeli dana klasa ma odnosić się do wszystkich elementów (znaczników), należy pominąć określenie elementu:

```
.nazwa_klasy {  
    definicje  
}
```

W celu odniesienia się do reguł zawartych w klasie, należy użyć atrybutu *class*:

```
<znacznik class="nazwa klasy" pozostałe_atrybuty>  
treść znacznika  
</znacznik>
```

Jako nazwę klasy należy podać dowolny pojedynczy wyraz, który nie może zawierać znaków: spacji, kropki, przecinka, dwukropka, pytajnika, nawiasów, znaku równości, plusa itp. Może natomiast zawierać litery (A-Z, a-z), cyfry (0-9), myślniki ("-") i podkreślniki ("_"). Lepiej nie używać polskich liter. Nie może się on również rozpoczynać cyfrą ani myślnikiem. Jeśli użycie "zakazanych" znaków jest konieczne, należy w deklaracji klasy poprzedzić je odwróconym ukośnikiem "\\"

Przykład wykorzystania selektora klas przedstawiono poniżej.

Listing 1. Przykład ogólnej klasy.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html lang="pl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Moja strona WWW</title>
<style type="text/css">

.niebieski {
color:blue;
font-size:18pt
    }

</style>
</head>
<body>
<p class="niebieski">
Akapit korzystający z klasy niebieski.</p>
<p>Ten akapit nie odnosi się do klasy, ale została ona użyta
w <span class="niebieski">wyróżnieniu liniowym</span>.</p>
</body>
</html>
```

Rysunek 1. Przykład ogólnej klasy.



Selektor identyfikatora

Ten typ selektora pozwala na nadanie określonych atrybutów formatowania dla elementu, który ma jednoznaczny identyfikator (id), czyli występuje tylko raz w całym dokumencie (w odróżnieniu od klasy). Schemat definicji:



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

„Era Entera – e-learning dla młodzieży”

Projekt współfinansowany
ze środków Unii Europejskiej
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



```
#identyfikator {  
    definicje  
}
```

Przykład definicji:

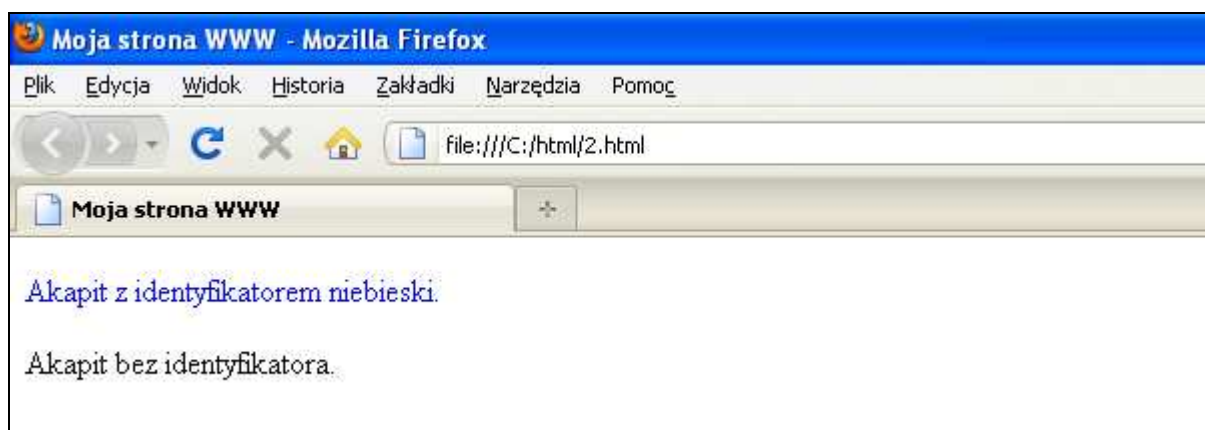
```
<style type="text/css">  
  
    #niebieski {  
        color:blue;  
    }  
</style>
```

Element witryny, któremu zostanie przypisany identyfikator o nazwie niebieski będzie wyświetlany w kolorze niebieskim.

Listing 2. Selektor identyfikatora.

```
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
<title>Moja strona WWW</title>  
<style type="text/css">  
  
    #niebieski {  
        color:blue;  
    }  
  
</style>  
</head>  
<body>  
<p id="niebieski">  
Akapit z identyfikatorem niebieski.</p>  
<p>Akapit bez identyfikatora.</p>  
</body>
```

Rysunek 2. Selektor identyfikatora.



Jako nazwę klasy należy podać dowolny pojedynczy wyraz, który nie może zawierać znaków: spacji, kropki, przecinka, dwukropka, pyłajnika, nawiasów, znaku równości, plusa itp. Może natomiast zawierać litery (A-Z, a-z), cyfry (0-9), myślniki ("-") i podkreślniki ("_"). Lepiej nie używać polskich liter. Nie może się on również rozpoczynać cyfrą ani myślnikiem. Jeśli użycie "zakazanych" znaków jest konieczne, należy w deklaracji poprzedzić je odwróconym ukośnikiem "\\"

Selektor pseudoklasy

Selektor pseudoklasy nadaje określone formatowanie elementowi witryny w zależności od jego stanu bądź właściwości. Jest dynamiczny w tym sensie, że element "nabywa" lub "traci" pseudoklasę w trakcie interakcji z użytkownikiem. Przykładem jest podświetlenie elementu po wskazaniu go myszką przez użytkownika.

```
selektor:pseudo_klasa {  
    definicje  
}
```

Można stosować kombinację klas i pseudoklas:

```
selektor.klasa:pseudo_klasa{  
    definicje  
}
```

Pseudoklasy są predefiniowane, a listę dostępnych pseudoklas przedstawia tabela 1.

Tabela 1. Lista pseudoklas.

Nazwa	Opis
:active	Element w stanie aktywnym
:first-child	Element będący pierwszym elementem podrzędnym
:focus	Element, który ma fokus
:hover	Element, na którym znajduje się kursor myszy
:lang	Element z przypisanym atrybutem Lang o specyficznej wartości
:link	Nieodwiedzony odnośnik
:visited	Odwiedzony odnośnik

Przykład użycia pseudoklas zawarto poniżej:

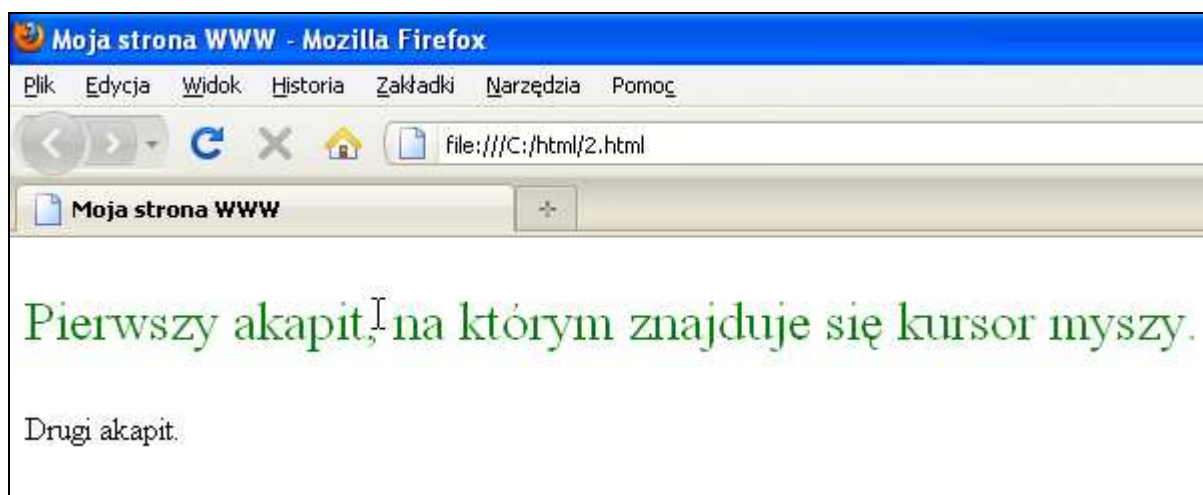
Listing 3. Użycie selektora pseudoklasy.

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Moja strona WWW</title>
<style type="text/css">

    p:hover {
        color:green;
        font-size:20pt;
    }

</style>
</head>
<body>
    <p>Pierwszy akapit, na którym znajduje się kursor myszy.</p>
    <p>Drugi akapit.</p>
</body>
```

Rysunek 3. Użycie selektora pseudoklasy.



Selektory elementów potomnych

Selektor tego typu pozwala nadać atrybuty elementom, które leżą niżej w hierarchii drzewa dokumentu (zawierają się w innych zewnętrznych znacznikach). Dzięki temu możemy zmienić typ formatowania tylko dla określonych elementów, które są podrzędne w stosunku do innych (przodków). Potomek nie musi leżeć bezpośrednio wewnątrz znacznika przodka. Może być zawarty jeszcze w innych znacznikach, które z kolei zawierają się w rodzicu. Nie jest wtedy konieczne podawanie w deklaracji wszystkich rodziców, a jedynie przodka i potomka. Ogólna postać selektora:

```
selektor1 selektor2 selektorN {  
    definicje  
}
```

Przykładowa reguła:

Listing 4. Przykład użycia selektora elementów potomnych.

```
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
<title>Moja strona WWW</title>  
<style type="text/css">  
  
div p span {  
    color: red;  
}
```



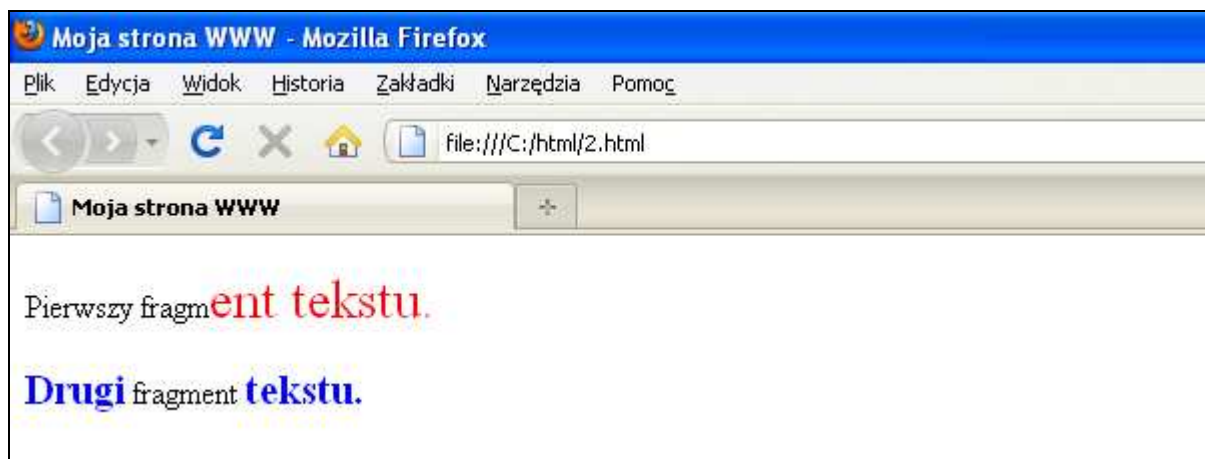

```

        font-size:20pt;
    }

    div p.bold span {
        font-weight:600;
        color:blue;
        font-size:16pt;
    }
</style>
</head>
<body>
<div><p>Pierwszy fragm<span>ent tekstu.</span></p></div>
<div><p class="bold"><span>Drugi</span> fragment
<span>tekstu.</span></p></div>
</body>

```

Rysunek 4. Przykład użycia selektora elementów potomnych.



Selektor wieloelementowy

Jeżeli chcemy nadać taką samą regułę wielu elementom witryny to należy skorzystać z selektora wieloelementowego. Taka deklaracja stylu pozwala nadać te same wartości atrybutów kilku różnym elementom jednocześnie.

```

selektor1, selektor2, selektorN {
    definicje
}

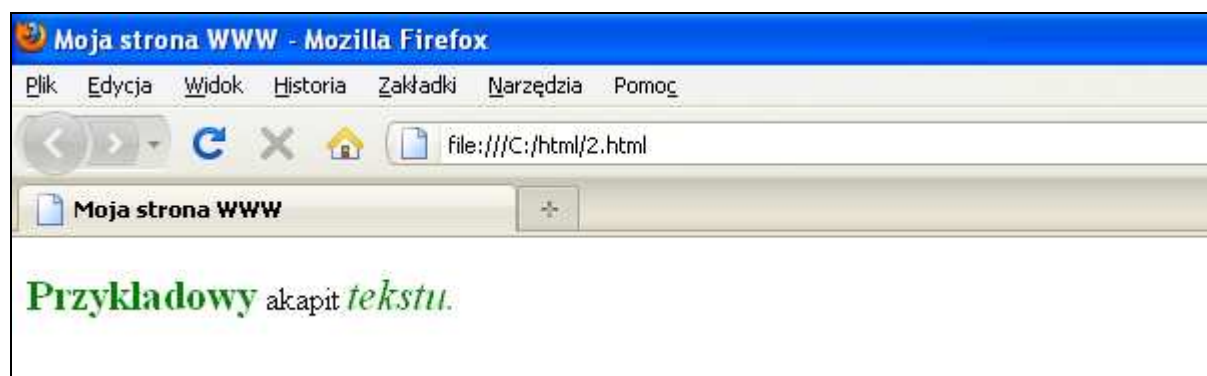
```

Listing 5. Użycie selektora wieloelementowego.

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Moja strona WWW</title>
<style type="text/css">

i, b {
    color: green;
    font-size:16pt;
}
</style>
</head>
<body>
<p><b>Przykładowy</b> akapit <i>tekstu</i>.</p>
</body>
```

Rysunek 5. Użycie selektora wieloelementowego.



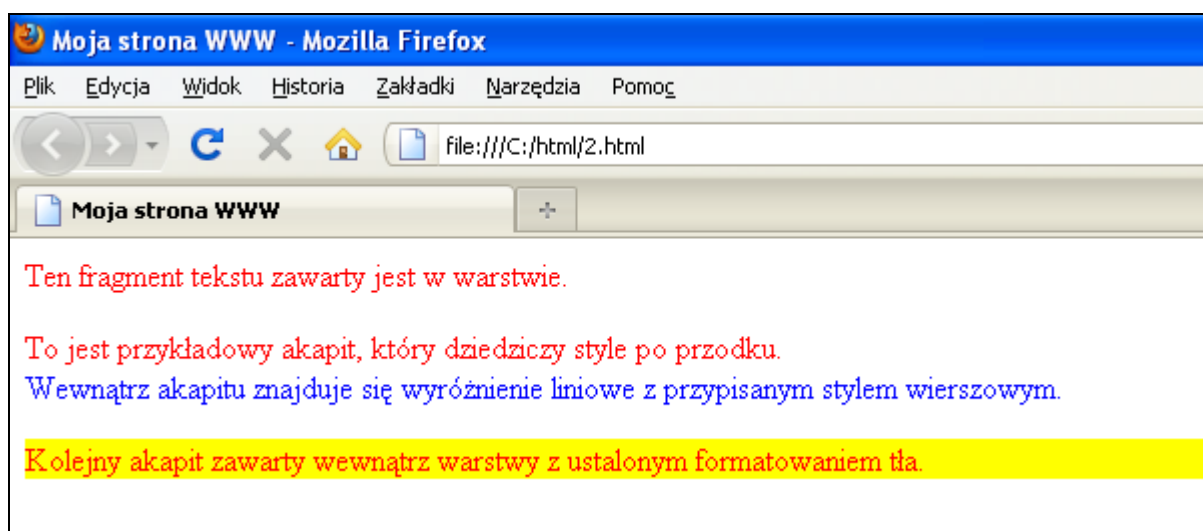
Dziedziczenie stylów

Polega na tym, że elementy leżące niżej w hierarchii (potomkowie), jeśli nie zaznaczymy inaczej, dziedziczą (przejmują) atrybuty formatowania, które zostały nadane ich przodkom. Niestety w niektórych przeglądarkach internetowych zdarza się błędna interpretacja dziedziczenia stylów, dlatego zawsze należy sprawdzić w praktyce zastosowanie tej własności.

Listing 6. Przykład dziedziczenia stylów.

```
<body>
<div style="color:red;">
Ten fragment tekstu zawarty jest w warstwie.
<p>To jest przykładowy akapit, który dziedziczy style po przodku.
<br />
<span style="color:blue;">Wewnątrz akapitu znajduje się wyróżnienie
liniowe z przypisanym stylem wierszowym.</span>
</p><p style="background-color:yellow">Kolejny akapit zawarty
wewnątrz warstwy z ustalonym formatowaniem tła.</p>
</div>
</body>
```

Rysunek 6. Przykład dziedziczenia stylów.



Kaskadowość stylów (reguły nakładania styli)

Kaskadowość stylów określa pierwszeństwo w oddziaływaniu na te same elementy strony stylów z różnych źródeł. Pierwszeństwo mają te, które znajdują się "bliżej" formatowanego elementu. Ważna jest kolejność nakładania styli, która jest ściśle określona. Obowiązywał będzie styl nałożony jako ostatni. Kolejność jest następująca:

- styl pochodzący z zewnętrznego arkusza stylów,
- styl pochodzący z wewnętrznego arkusza stylów,
- styl wierszowy (inline).

Oddziaływanie stylów z arkuszy zewnętrznych może być modyfikowane przez style zdefiniowane w nagłówku dokumentu, to z kolei mogą być zmieniane przez style zdefiniowane bezpośrednio w ciele dokumentu (inline).

Kolejność nakładania stylów jest również uzależniona od rodzaju zastosowanego selektora. Ogólna zasada jest następująca: najpierw nakładane są style bardziej ogólne, a później bardziej szczegółowe. Obowiązywał będzie styl nałożony jako ostatni. Kolejność jest następująca:

- selektory ogólne,
- selektory elementu,
- selektory klas i pseudoklas,
- selektory identyfikatorów.

2. Formatowanie tekstu

Style CSS pozwalają na swobodne kontrolowanie sposobu wyświetlania tekstu na stronie WWW.

Wyrównywanie poziome

Wyrównywanie tekstu w poziomie określa cecha *text-align*, która pozwala wybrać jeden z możliwych sposobów wyrównania tekstu, czyli jego ułożenia na ekranie.

```
selektor { text-align: wyrównanie }
```

Selektorem może być dowolny znacznik wyświetlany w bloku.

Wartości cechy *text-align*:

- left - wyrównanie tekstu do lewego marginesu (domyślnie),
- right - wyrównanie do prawego marginesu,
- center - do środka (wyśrodkowanie),
- justify - do obu marginesów jednocześnie (justowanie).



Wyrównanie w pionie

Zdefiniowanie wyrównania tekstu w pionie jest możliwe dzięki właściwości *vertical-align*,

```
selektor { vertical-align: wyrównanie }
```

Selektorem może być dowolny znacznik wyświetlany w bloku.

Cecha przyjmuje poniższe wartości:

1. Wartości predefiniowane:

- baseline – wyrównanie do linii bazowej (domyślne),
- sub – indeks dolny,
- super – indeks górny,
- top - górna linia elementu jest wyrównywana do najwyższej linii w wierszu,
- text-top - górna linia elementu jest wyrównywana do górnej linii czcionki elementu nadrzędnego,
- middle - wyrównanie do linii środkowej,
- bottom - dolna linia elementu jest wyrównywana do najniższej linii w wierszu,
- text-bottom - dolna linia elementu jest wyrównywana do dolnej linii czcionki elementu nadrzędnego.

2. Jednostka miary – względne, bezwzględne oraz wartość procentowa.

UWAGA!

Podane sposoby wyrównywania w poziomie i pionie w rzeczywistości mają zastosowanie do wszystkich elementów (np. obrazów) zawartych w znaczniku, którym przypisano styl formatujący wyrównanie.

Odstępy

CSS pozwala określić w tekście odstępy pionowe (między wierszami) oraz poziome (między wyrazami oraz znakami)

Odstęp między wierszami

Cecha *line-height* pozwala ustalić odstęp pomiędzy liniami bazowymi wszystkich wierszy w bloku. Dzięki temu linijki mogą być bardziej ścieśnione lub rozciągnięte.

```
selektor { line-height: odstęp }
```

Selektorem może być dowolny znacznik, w którym można wpisywać tekst.

Wartość cechy może być określona następująco:

- wartość wraz z jednostką – odstępy będą określone za pomocą wskazanej jednostki długości,
- wartość bez jednostki – wartość będzie traktowana jako wielokrotność bieżącej wysokości czcionki,
- wartość procentowa - wartość będzie traktowana jako procent bieżącej wysokości czcionki.

Odstęp między wyrazami

Cecha *word-spacing* pozwala ustalić odstęp pomiędzy wszystkimi poszczególnymi wyrazami w bloku. Dzięki temu wyrazy w liniach mogą być bardziej ścieśnione lub rozciągnięte. Wartość atrybutu to wybrana jednostka długości, dzięki czemu odstęp będzie określony za pomocą wskazanej jednostki miary.

```
selektor { word-spacing: odstęp }
```

Selektorem może być dowolny znacznik, w którym można wpisywać tekst.

Odstęp między literami

Do określenia odległości między literami używana jest cecha *letter-spacing*. Wartość atrybutu to jednostka długości, dzięki czemu odstęp będzie określony za pomocą wskazanej jednostki miary.

UWAGA!

W celu zmniejszenia odstępów poziomych (zarówno pomiędzy wierszami i literami) należy użyć wartości ujemnych.

Wcięcia

Wcięcia dotyczą blokowych elementów tekstowych i definiują przesunięcie pierwszego wiersza. Aby dodać wcięcie należy zastosować cechę *text-indent*, która przyjmuje wartości określone jednostką długości lub określone procentowo.

```
selektor { text-indent: wcięcie }
```

Selektorem może być każdy znacznik wyświetlany w bloku.

Wcięcie przesuwa pierwszy wiersz w prawo jeżeli wartość jest dodatnia, natomiast w przypadku podania wartości ujemnych przesunięcie jest w lewo.

Dekoracje

Cecha *text-decoration* umożliwia wybór określonej dekoracji tekstu. Wartość atrybutu określa rodzaj dekoracji jaką tekst zostanie wyróżniony.

```
selektor { text-decoration: dekoracja }
```

Selektorem może być dowolny znacznik, w którym można wpisywać tekst.

Wartości cechy:

- none - bez zmian
- underline - podkreślenie
- line-through - przekreślenie
- overline - nadkreślenie
- blink

Aby dodać więcej niż jedną dekorację należy podać kilka wartości oddzielając je od siebie znakiem spacji. Migotanie może nie być interpretowane przez wszystkie przeglądarki m.in. Internet Explorer.

Transformacje

Stosując cechę *text-transformation* można zmienić wielkość liter. Niezależnie od tego, jak tekst został zapisany oryginalnie w pliku z kodem HTML, litery mogą zostać zamienione na małe bądź wielkie.

```
selektor { text-transform: transformacja }
```

Selektorem może być dowolny znacznik, w którym można wpisywać tekst.

Wartości jakie przyjmuje atrybut:

- none - bez zmian
- capitalize - zamiana pierwszych liter wszystkich wyrazów na wielkie
- uppercase - zamiana wszystkich liter na wielkie
- lowercase - zamiana wszystkich liter na małe

Białe znaki

```
selektor { white-space: sposób }
```

Selektorem może być każdy znacznik wyświetlany w bloku.

Wartości cechy:

- normal - zamiana kilku sąsiadujących ze sobą spacji na jedną; wiersze są przełamywane automatycznie na końcu szerokości strony (domyślnie),
- pre - umożliwia wyświetlenie wielu spacji obok siebie; wiersze są przełamywane po napotkaniu znaku nowej linii - klawisz Enter (tekst preformatowany),
- nowrap - tryb identyczny jak "normal" z tym, że nie pozwala na przełamanie wiersza (przejście do następnej linii), aż do napotkania znacznika
,



- pre-wrap - (CSS 2.1) pozwala wyświetlić kilka sąsiadujących spacji; wiersze są przełamywane po napotkaniu znaku nowej linii - klawisz Enter - oraz automatycznie na końcu szerokości strony (Opera 7.5, Firefox 3, Chrome, Konqueror),
- pre-line - (CSS 2.1) jak wyżej z tym, że sąsiadujące spacje są zamieniane na jedną.

Polecenie pozwala sterować wyświetlaniem spacji na ekranie. Domyślnie nie jest możliwe postawienie obok siebie kilku spacji, ponieważ zostaną one automatycznie zamienione na jedną. Nie można również postawić spacji na początku wiersza. Dzięki użyciu komendy *white-space: pre*, możemy pozbyć się tych niedogodności. Polecenie działa podobnie jak znacznik `<pre>...</pre>`, lecz tekst jest napisany zwykłą czcionką. Ponadto, definicja *white-space: nowrap* uniemożliwia przeglądarce przełamanie wiersza i przejście do następnej linii. Stosuje się je, gdy koniecznie zależy nam, aby podany tekst, był wpisany w jednej linii. Jeśli nie zmieści się on na ekranie, pojawi się suwak do przewijania zawartości strony. Wiersz taki możemy przełamać tylko poprzez zastosowanie znacznika `
`. Komenda działa podobnie jak `<nobr>...</nobr>`.

