

Kurs Front-End Developer
CSS3

SELECTORY CSS

Jeśli chcemy nadać wygląd paragrafowi, to tworzymy odpowiedni selektor i w nawiasach klamrowych definiujemy jego reguły.

```
p {  
    color: blue;  
    font-size: 18px;  
    background: yellow;  
}
```

Tak zdefiniowana reguła sprawi, że wszystkie paragrafy (bloki tekstu w znacznikach `<p></p>`) będą miały niebieską czcionkę wysokości 18 pikseli i cały tekst będzie na żółtym tle.

SELECTORY CSS

BEGIN NAVIGATION

SELEKTOR IDENTYFIKATORA

```
#identyfikator {  
    color: red;  
}
```

Taka reguła sprawi, że tekst w elemencie z atrybutem `id="identyfikator"` będzie czerwony.

SELECTORY CSS

BEGIN NAVIGATION

Łączenie selektorów

```
a, p, div {  
    color: red;  
}
```

Zamiast pisać osobne reguły dla “a”, “p” oraz “div” możemy je połączyć obdzielając je przecinkiem (,)

SELECTORY CSS

Selektor bezpośredniego elementu potomnego - selektory oddzielone znakiem “>”

```
li > a {  
    color: red;  
}
```

Taka reguła będzie działała tylko na “a”, które jest bezpośrednim potomkiem elementu “li”

SELECTORY CSS

Selektor elementów potomnych - selektory oddzielone spacją

```
div a {  
    color: red;  
}
```

Taka reguła będzie działała na wszystkie “a” zagnieżdżone w “div” bez względu na poziom zagnieżdżenia

SELECTORY CSS

Selektor bliskiego rodzeństwa - selektory oddzielone znakiem plus “+”

```
p + a {  
    color: red;  
}
```

Taka reguła będzie działała tylko na pierwszy “a” będący bezpośrednio po “p” na tym samym poziomie struktury dokumentu

SELECTORY CSS

Selektor bliskiego rodzeństwa - selektory oddzielone znakiem plus “+”

```
p + a {  
    color: red;  
}
```

Taka reguła będzie działała tylko na pierwszy “a” będący bezpośrednio po “p” na tym samym poziomie struktury dokumentu

SELECTORY CSS

BEGIN NAVIGATION

Selektor rodzeństwa - selektory oddzielone znakiem tyldy “~”

```
h3 ~ p {  
    color: red;  
}
```

Taka reguła będzie działała na wszystkie elementy “p” znajdujące się po “h3” na tym samym poziomie w strukturze dokumentu.

Selektor wykluczający

```
div > p:not(.last) {  
    color: red;  
}
```

Taka reguła będzie działała na wszystkie elementy “p” z wyjątkiem elementu z klasą last

SELECTORY CSS

BEGIN NAVIGATION

Selektor kolejnego elementu potomnego

```
div > p:nth-child(2) {  
    color: red;  
}
```

Taka reguła będzie działała na drugi paragraf potomny dla elementu DIV

SELECTORY CSS

Hierarchia selektorów CSS

<http://cssspecificity.com/>

Kalkulator hierarchii selektorów

<https://specificity.keegan.st/>

POZYCJA W CSS

Pozycjonowanie elementów w CSS jest jednym z zagadnień, które sprawia najwięcej problemów początkującym front-end developerom.

Pozycja, czyli właściwość **position** pozwala określać położenie danego kontenera HTML względem innych.

POZYCJA W CSS

Pozycja statyczna

`position: static;`

Domyślna pozycja nie pozwalająca na zmianę położenia elementu względem innych elementów

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quas pariatur earum laborum consequuntur nesciunt facilis, maxime similique reprehenderit eum quam quis odio accusantium assumenda sequi voluptates velit debitis ut quisquam.

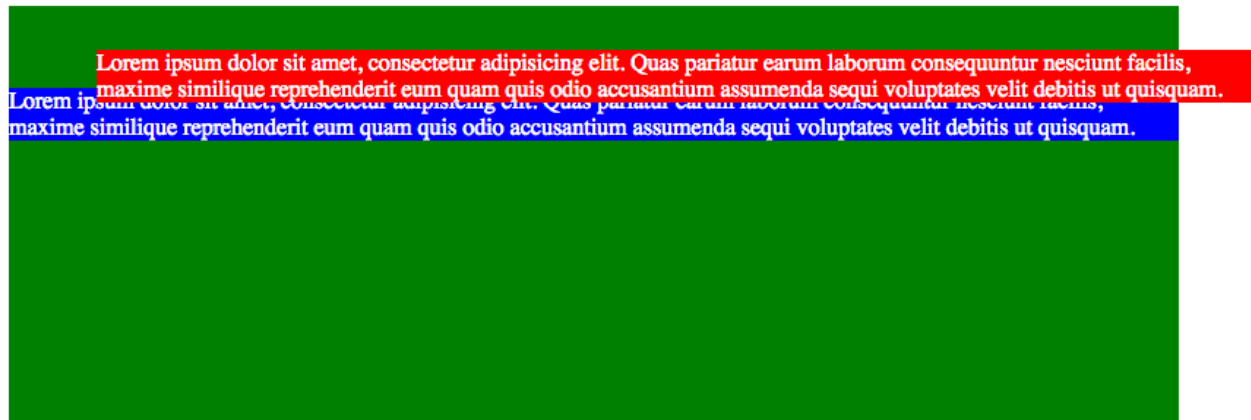
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quas pariatur earum laborum consequuntur nesciunt facilis, maxime similique reprehenderit eum quam quis odio accusantium assumenda sequi voluptates velit debitis ut quisquam.

POZYCJA W CSS

Pozycja relatywna

`position: relative;`

Pozwala na ustalenie pozycji za pomocą właściwości `top`, `bottom`, `left`, `right` względem elementu nadrzędnego. Pozycja elementu nie zmienia położenia innych elementów.



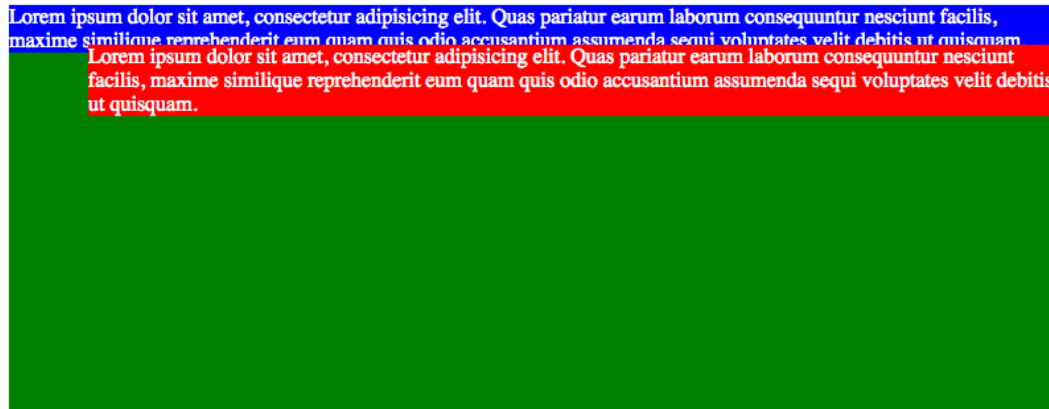
`left: 60px; top: 30px;`

POZYCJA W CSS

Pozycja absolutna

`position: absolute;`

Pozwala na ustalenie pozycji za pomocą właściwości `top`, `bottom`, `left`, `right` względem elementu nadrzędnego, który znajduje się w pozycji relatywnej. Element w pozycji absolutnej robi miejsce innym elementom w danym kontenerze



`left: 60px; top: 30px;`

POZYCJA W CSS

Pozycja ustalona

`position: fixed;`

Pozwala na ustalenie pozycji za pomocą właściwości `top`, `bottom`, `left`, `right` względem okna przeglądarki. Element ma stałą pozycję względem okna przeglądarki.

TESTOWANIE I WERYFIKACJA KODU

Kolejnym narzędziem, które może wykorzystać Front-End Developer do pracy jest <https://jsfiddle.net/>

Pozwala stworzenie w przeglądarce bloku kodu HTML, arkusza stylu CSS oraz dodanie skryptów JavaScript.

Dzięki temu możemy pokazać nasz kod innym programistom.

TESTOWANIE I WERYFIKACJA KODU

HTML

```
<div id="outer">  
  <div id="inner"></div>  
</div>
```

CSS

```
#outer {  
  position: relative;  
  width: 200px;  
  height: 100px;  
  background-color: green;  
}
```

```
#inner {  
  position: absolute;  
  width: 100px;  
  height: 100px;  
  top: 30px;  
  left: 30px;  
  background-color: blue;  
}
```

JEDNOSTKI MIARY W CSS

CSS oferuje wiele różnych jednostek miary. Dzielą się one na dwa rodzaje:

- jednostki absolutne (bezwzględne), do których należą: cm, mm, in (cale), pt (punkty), pc (pica) i px (piksele);
- jednostki relatywne (względne), do których należą: em, ex, rem, % (procenty), oraz viewport units czyli vh (viewport height), vw (viewport width), vmax (viewport maximum), vmin (viewport minimum).

W większości przypadkach osoba początkująca będzie posługiwać się jednostką px czyli pikselami.

Jednostkę miary podajemy zawsze bezpośrednio po liczbie, a wartości dziesiętne od całości oddzielamy za pomocą kropki np.

font-size: 1.5cm;

font-size: 15px;

itp.

JEDNOSTKI MIARY W CSS

Jednostki absolutne

Używa ich się wtedy gdy chcemy aby wybrany element zajmował taki sam obszar na ekranie niezależnie od wielkości i rozdzielczości monitora.

in - cale, 1 in = 2.54cm

cm - centymetry, 1 cm = 10mm

mm - milimetry, 1 mm

pt - punkty, 1 pt = 1/72in

pc - pica, 1 pc = 12pt

px - piksel, 1 px = 1/96in

JEDNOSTKI MIARY W CSS

Jednostki relatywne

Stosuje się je wtedy gdy chcemy odwołać się do rozmiaru innego elementu lub rozmiaru okna użytkownika. Jednostki względne podają długość w odniesieniu do innego rozmiaru. Używane są jeśli chcemy zachować proporcje wyświetlania różnych elementów.

em - wysokość aktualnej czcionki

ex - wysokość litery "x", czyli małej litery w aktualnej czcionce

rem - wysokość czcionki elementu korzenia w drzewie dokumentu (element `<html>`)

% - procenty - procentowa wielkość odpowiedniej własności elementu rodzica

vw - viewport width - procentowa szerokość obszaru wyświetlania

vh - viewport height - procentowa wysokość obszaru wyświetlania

vmin - viewport minimum - mniejsza z wartości vw lub vh

vmax - viewport maximum - większa z wartości vw lub vh

JEDNOSTKI MIARY W CSS

Jednostki mają istotne znaczenie przy wyświetlaniu mediów czy to na ekranie, czy na papierze (druk).

Poniższa tabela przedstawia rekomendowane użycie:

	ZALECANE	SPORADYCZNIE UŻYWANE	NIEZALECANE
EKRAN	em, px, %	ex	pt, cm, mm, in, pc
DRUK	em, cm, mm, in, pt, pc, %	px, ex	

JEDNOSTKI MIARY W CSS

Piksele (px)

Jednostka ta określa ile pikseli na ekranie naszego urządzenia ma zajmować dana właściwość CSS.

Jednostka px nie jest zdefiniowana jako stała długość, ale jako coś co zależy od typu urządzenia oraz jego zastosowania. Oznacza to że wielkość piksela zależy od odległości z jaką na niego patrzymy oraz od rozdzielczości urządzenia na jakim go wyświetlamy.

Wielkość px jest zdefiniowana jako mała, ale widoczna, tak aby pozioma linia o szerokości 1px mogła być wyświetlona wraz z ostrymi krawędziami.

W rzeczywistości standard CSS wymaga, aby 1px stanowił dokładnie 1/96 cala na każdym wyświetlanym obrazie.

JEDNOSTKI MIARY W CSS

em

Wartość 1em jest obliczana przez przeglądarkę internetową na podstawie wartości właściwości font-size elementu rodzica, np.

#HTML

```
<div>  
  <p>Tekst z czcionką 3em.</p>  
</div>
```

#CSS

```
div { font-size: 15px; }  
div p { font-size: 3em; }
```

wielkość czcionki dla elementu p - wynosi 3em, czyli 45px (ponieważ "font-size" dla rodzica tego elementu czyli dla div - wynosi 15px, a więc $3em = 3 \times 15px$, czyli 45px)

JEDNOSTKI MIARY W CSS

ex

Wartość 1ex jest obliczana przez przeglądarkę internetową na podstawie wysokości małych liter (liter bez wydłużeń czyli a, c, i, z itp.) danej czcionki elementu rodzica, przeważnie jest to wartość w granicy 51% ~ 52% całej wartości wielkości czcionki.

#HTML

```
<div>  
  <p>Tekst z czcionką 4ex.</p>  
</div>
```

#CSS

```
div { font-size: 15px; }  
div p { font-size: 4ex; }
```

wielkość czcionki dla elementu p - wynosi 4em, czyli około 30-31px (ponieważ "font-size" dla rodzica tego elementu czyli dla div - wynosi 15px)

JEDNOSTKI MIARY W CSS

rem

Jest ona zdefiniowana jako wielkość tzw. bazowej czcionki dokumentu (stąd nazwa, bazowy em od angielskiego root em).

Wartość 1rem jest obliczana przez przeglądarkę internetową na podstawie wartości właściwości font-size elementu HTML, który jest najwyżej w hierarchii dokumentu HTML, czyli tym elementem, jest element html.

```
html { font-size: 15px; }  
p { margin-left: 1.2rem; }
```

wielkość lewego marginesu dla elementu p - wynosi 1.2rem, czyli 18px (ponieważ "font-size" dla elementu "html" wynosi 15px, a więc $1.2\text{rem} = 1.2 \times 15\text{px}$, czyli 18px)

JEDNOSTKI MIARY W CSS

procenty - %

Niektóre wartości właściwości CSS możemy określić w procentach, np.

#HTML

```
<section>
  <div>
    <p>Tekst z czcionką 3em.</p>
  </div>
</section>
```

szerokość elementu div - wynosi 50% szerokości elementu section, czyli 250px

wielkość czcionki dla elementu div – wynosi 40% wielkości czcionki elementu section, czyli 8px

#CSS

```
section { width: 500px; font-size: 20px; }
div { width: 50%; font-size: 40%; }
```

JEDNOSTKI MIARY W CSS

Viewport units

Są to jednostki stosunkowo nowe. Pozwalają one zapisać reguły zależnie od wielkości okna użytkownika.

Do tych jednostek należą:

vw - viewport width

vh - viewport height






vmin - viewport minimum

vmax - viewport maximum

JEDNOSTKI MIARY W CSS

Liczby w tabeli określ pierwszą wersję przeglądarki, która w pełni obsługuje jednostkę długości.

* wsparcie w IE9 odnosi się do wspierania „vm” zamiast „vmin”

						
vw	9.0	12.0	19.0	20.0	6.0	15.0
vh	9.0	12.0	19.0	20.0	6.0	15.0
vmin	9.0*	12.0	19.0	20.0	6.0	15.0
vmax	Not supported	Not supported	19.0	26.0	7.0	15.0

Wsparcie dla Viewport Units dzięki <http://caniuse.com/>

JEDNOSTKI MIARY W CSS

Viewport width - vw

1vw jest równy 1% wartości szerokości przestrzeni w jakiej została wyświetlona nasza strona internetowa (przeważnie przestrzeni przeglądarki internetowej).

Na przykład jeżeli szerokość przeglądarki jest równa 1024px to:

$$1vw = 1\% * 1024px = 10.24px$$

$$8vw = 8\% * 1024px = 81.92px$$

```
div { font-size: 2vw; }
```

rozmiar czcionki tekstu w elemencie div - jest uzależniony od wartości szerokości okna przeglądarki internetowej w jakiej została wyświetlona strona internetowa

JEDNOSTKI MIARY W CSS

Viewport height - vh

1vh jest równy 1% wartości wysokości przestrzeni w jakiej została wyświetlona nasza strona internetowa (przeważnie przestrzeni przeglądarki internetowej).

Na przykład jeżeli wysokość przeglądarki jest równa 768px to:

$$1vh = 1\% * 768px = 7.68px$$

$$8vh = 8\% * 768px = 61.44px$$

wysokość elementu div - jest uzależniony od wartości wysokości okna przeglądarki internetowej w jakiej została wyświetlona strona internetowa

```
div { height: 10vh; }
```


JEDNOSTKI MIARY W CSS

Viewport minimum - vmin

1vmin jest równy 1vw lub 1vh, w zależności która wartość jest aktualnie mniejsza.

Na przykład jeżeli 1vw jest równe 10.24px oraz 1vh jest równe 7.68px to:

$$1\text{vmin} = 1 * \min(7.68\text{px}, 10.24\text{px}) = 7.68\text{px}$$

$$8\text{vmin} = 8 * \min(7.68\text{px}, 10.24\text{px}) = 61.44\text{px}$$

rozmiar czcionki tekstu w elemencie div - jest uzależniony od wartości szerokości lub wysokości okna przeglądarki internetowej w jakiej została wyświetlona strona internetowa

```
div { font-size: 3vmin; }
```

JEDNOSTKI MIARY W CSS

Viewport maximum - vmax

1vmax jest równy 1vw lub 1vh , w zależności która wartość jest aktualnie większa.

Na przykład jeżeli 1vw jest równe 10.24px oraz 1vh jest równe 7.68px to:

$$1\text{vmin} = 1 * \max(7.68\text{px}, 10.24\text{px}) = 10.24\text{px}$$

$$8\text{vmin} = 8 * \max(7.68\text{px}, 10.24\text{px}) = 81.92\text{px}$$

```
div { font-size: 3vmax; }
```

rozmiar czcionki tekstu w elemencie `div` - jest uzależniony od wartości szerokości lub wysokości okna przeglądarki internetowej w jakiej została wyświetlona strona internetowa

KOLORY W CSS

BEGIN NAVIGATION

W języku CSS mamy do dyspozycji następujące zapisy kolorów:

- kolory predefiniowane;
- zapis heksadecymalny;
- skrócony zapis heksadecymalny;
- zapis rgb;
- zapis rgba.

Narzędzia, które ułatwiają działanie w zakresie definiowania kolorów oraz strony poświęcone kolorom i paletą kolorów:

- www.color-hex.com
- www.colourlovers.com
- colorschemedesigner.com
- color.adobe.com

KOLORY W CSS

paleta kolorów podstawowych, predefiniowanych

Do określenia kolorów możemy posłużyć się jego angielską nazwą. Wszystkich kolorów predefiniowanych jest 140.

`background-color: red;` // tło w kolorze czerwonym

`color: blue;` // kolor czcionki niebieski

KOLORY W CSS

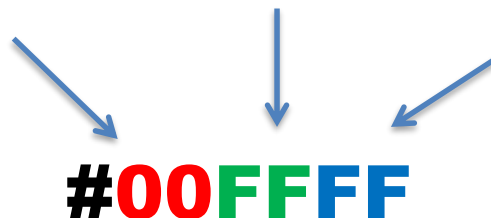
zapis heksadecymalny (szesnastkowy)

Składa się z 6 znaków poprzedzonych hashem (#).

KOLOR CZERWONY

KOLOR ZIELONY

KOLOR NIEBIESKI



background-color: #f0ffff;

color: #8a2be2;

KOLORY W CSS

Za pomocą znaków: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F nadaje się intensywność danej barwie. Litery od A do F oznaczają liczby od 10 do 15, np.

#291186

barwa zielona ma małą intensywność (11)

czerwona ma małą intensywność (29)

barwa niebieska ma najwyższą intensywność (86)

Ponieważ barwa niebieska ma największą intensywność ten kolor ma odcień niebieskiego.

KOLORY W CSS

skrócony zapis heksadecymalny

Niektóre kolory zapisane heksadecymalnie możemy zapisać skrótowo, jeżeli każda z barw będzie składała się z dwóch takich samych znaków, np.

#ff1122 - skrócony zapis to #f12

#ffffff - skrócony zapis to #fff

#1122ff - skrócony zapis to #12f

#291186 - tego zapisu nie da się uprościć

KOLORY W CSS

zapis rgb – rgb(czerwony, zielony, niebieski)

Po literach rgb, w nawiasie znajdują się trzy liczby określające intensywność kolejno barw: czerwonej, zielonej, niebieskiej. Liczby te oddzielone są od siebie przecinkami i każda z nich znajduje się w zakresie od 0 do 255. Możemy użyć też zapisu procentowego od 0% do 100%.

```
background-color: rgb(255, 0, 0); // tło w kolorze czerwonym  
background-color: rgb(100%, 0%, 0%); // tło w kolorze czerwonym
```

```
background-color: rgb(0, 255, 0); // tło w kolorze zielonym  
background-color: rgb(0%, 100%, 0%); // tło w kolorze zielonym
```

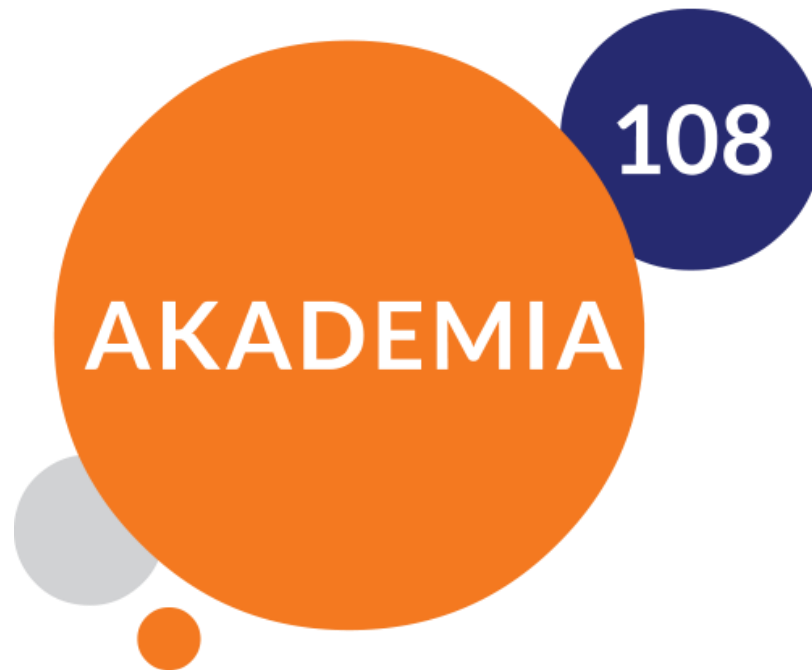

KOLORY W CSS

zapis rgba – rgba(czerwony, zielony, niebieski, krycie)

Podobny jest do zapisu rgb, tylko że wymaga on dodatkowego parametru określającego stopień przezroczystości (krycia).

Do określenia stopnia przezroczystości używa się liczb dziesiętnych z zakresu 0 - 1.0, przy czym wartość 1.0 oznacza pełne krycie, a 0 pełną przezroczystość.

`background-color: rgba(173, 216, 230, 0.5);`



Akademia 108
ul. Mostowa 6/13
31-061 Kraków