

SELEKTORY CSS

Reguła

```
p {  
    font-size: 20px;  
    text-align: center;  
}
```

Wartość

```
p {  
    border: 2px solid #333;  
    margin: 10px;  
    padding: calc(10% - 5px);  
    background-image: url("grafika.jpg");  
}
```

Prefixy (przedrostki)

```
header {  
    display: -webkit-box; /* webkit - Chrome i Safari */  
    display: -ms-flexbox; /* IE */  
    display: flex;  
}
```

Starsze przeglądarki wymagały podania przedrostków dla funkcji, które wtedy były wprowadzane eksperymentalnie (jeszcze nie były oficjalnie wdrożone). Kompatybilność, ze starszymi przeglądarkami wciąż wymaga stworzenia właściwości z prefixami.

Zadanie selektora

Wybrać wszystkie elementy w strukturze DOM, które pasują do selektora.

```
div {} /* Wybrane wszystkie elementy div na stronie */
```

```
<div></div>
```

```
<main>
```

```
    <div></div>
```

```
</main>
```

Selektor elementu

Wskazujemy element, tożsamy ze znacznikiem DOM.

div {}

<div></div>

<main>

<div></div>

</main>

Kombinatory

Pomagają ustalić zależność między elementami w strukturze html (DOM).

```
header div img {} /* spacja między elementami */  
div>p {} /* znak większości */  
h1+p {} /* znak dodawania */  
div~div /* znak tyldy */
```

Kombinator potomka (spacja)

```
div p em {...} //spacja oznacza potomka
```

```
<div>
```

```
  <h1>Tytuł</h1>
```

```
  <p>Tekst<em>wyróżniony</em></p>
```

```
</div>
```

```
<p>Tekst drugi<em>też ciekawy</span></p>
```


Kombinator dziecka

`div>div {...}` //znak większości to kombinator oznaczający dziecko, czyli każdy element, który jest bezpośrednim potomkiem.

`<div>`

`<header>`

`<div>`

`<div></div>`

`</div>`

`</header>`

`</div>`

Kombinatory pierwszego młodszego brata

nav+section //znak plusa (+) oznacza pierwszego młodszego brata, czyli element równoległy występujący wprost po danym elemencie.

```
<nav>
    <ul>
        <li></li>
        </li></li>
    </ul>
</nav>
<section></section>
<section></section>
```

Kombinatory każdego młodszego brata

nav~section //znak tyldy (~) oznacza kazego młodszego brata, czyli każdy element równoległy występujący po danym elemencie.

```
<nav>
    <ul>
        <li></li>
        </li></li>
    </ul>
</nav>
<section></section>
<section></section>
```

Kombinatory i klasy w jednym selektorze

div+div>.red {}

```
<div class="red">  
    <p class="red"></p>  
</div>  
<div>  
    <p class="red"></p>  
</div>  
<div>  
    <p class="red"></p>  
</div>
```

Przecinek w selektorze

```
.article, p { font-size: 2rem;}
```

/* to to samo co poniżej w dwóch regułach */

```
.article {  
    font-size: 2rem;  
}  
p {  
    font-size: 2rem;  
}
```

Pseudoklasy

`div:hover {}`

`input:focus {}`

`a:visited {}`

`/* Pseudoklasy są nadawane przez przeglądarkę niezależnie od nas
np. otrzymują je wszystkie elementy najechane myszką */`

Pseudoelementy

`div::before {}`

`p.header::after {}`

`p::selection {}`

`p::first-letter {}`

`/* Pseudelementy, które mogą zostać utworzone w CSS przez użytkownika lub są tworzone automatycznie */`

Odwołanie do atrybutu w selektorze

```
div[data-name] {}
```

```
a [href="http://strona.pl"] {}
```

```
[title] {}
```

```
<div data-name="intro"></div>
```

/* W CSS możemy odwołać się oczywiście do atrybutów elementów, wykorzystujemy do tego nawiasy kwadratowe */

atrybuty

`img[title]` /*wszystkie elementy img z atrybutem title */

`[href="http://websamuraj.pl"]` /* wszystkie elementy z podanym hiperłączem*/

`img[alt~=' 'car"]` /* każdy obrazek zawierający w atrybucie alt słowo car (nie ważne w którym miejscu) */

`p[data-name^="web"]` /* każde p z atrybutem data-name na którego początku znajduje się "web". Daszek - co ma być na początku*/

`[src$=".png"]` /* wszystkie elementy mające atrybut src zakończony na ".png" */

`input[placeholder*="user"]` /*input mający w atrybucie placeholder w dowolnym miejscu wyrażenie "user" */

pseudoklasy (stan lub miejsce w strukturze)

Pseudoklasy są jak klasy nadawane przez przeglądarkę. Np. każdy element na który najedziemy otrzymuje pseudoklasę `hover`. Dlatego możemy napisać selektor `div:hover { }` czyli regułę braną pod uwagę wtedy gdy nad `div-em` znajduje się myszka.

`:hover`

`:focus`

`:nth-child(n)`

`:nth-of-type(n)`

pseudoelementy

Elementy w elementach, które wykraczają poza dokument stworzony na podstawie pliku html

```
::first-line /* pierwsza linia w przeglądarce */
```

```
::first-letter /* pierwsza litera */
```

```
::before
```

```
::after
```

```
::selection /* zaznaczony tekst */
```