



# CSS Selektory

[dimon.work/kurs.html](http://dimon.work/kurs.html)

# Selektory w CSS

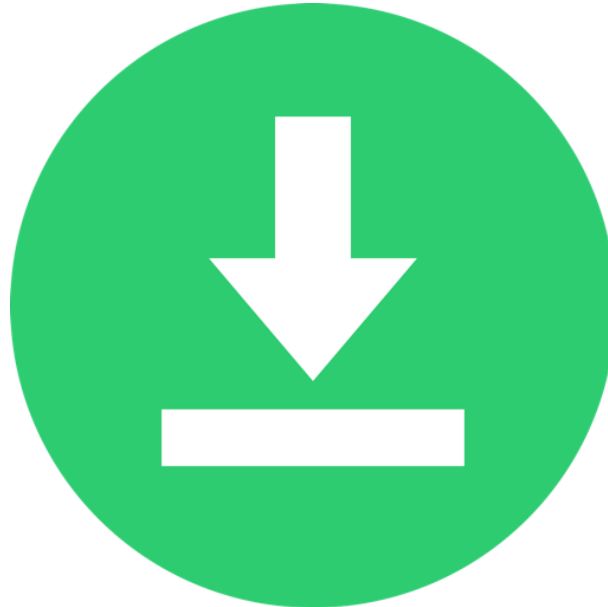
**Selektor CSS** , mówi do jakich tagów (elementów) zostanie zastosowany opisywany styl ( **selektor css** ) .

```
h3 { color: red; margin: 16px auto; }
```

**Selektor CSS** w ramach reguły informuje przeglądarkę, jakich tagów ma użyć należy zastosować zasadę stosowania, tj. ustaw warunek, na podstawie którego przeglądarka określi, czy tag nadaje się do zastosowania do niego reguły, czy nie.

# 1. Proste selektory

# Pobierz szablony



**Pobierz archiwum z potrzebnymi nam szablonami:**

**<https://github.com/filebase-xyz/css-selectors/archive/v2021.zip>**

# Trochę praktyki

1. DIV

2. DIV with class CAT;

3. DIV with id TIGER and class DOG;

4. DIV with class CAT, DOG, TIGER;

5. P with class DOG, BIRD;

6. P

7. P with class CAT, DOG;

8. P with class BIRD, HORSE;

9. ARTICLE with class HORSE;

10. P with class BIRD, DOG.

Otwórz projekt z katalogu (archiwum): [./css-selectors-master/demo-template](#)

# Selektor CSS według nazwy tagu

```
6 <style>  
7  
8     p {  
9         background: yellow;  
10    }  
11  
12 </style>
```

Styl zostanie zastosowany do wszystkich znaczników określonego typu ( nazwy ).

# Selektor według nazwy klasy

```
6 <style>
7
8     .bird {
9         background: yellow;
10    }
11
12    .cat {
13        text-decoration: line-through;
14    }
15
16 </style>
```

`<div class="bird"></div>`

. **nazwa-klasy** to selektor pozwalający wybrać tagi posiadające wymaganą klasę. Jeśli tag ma kilka klas, to ta, której szukasz, musi znajdować się wśród dostępnych.

## Selektor według atrybutu id ( znak „#” )

```
6  <style>
7
8      #tiger {
9          background: yellow;
10     }
11
12 </style>
```

`<div id = " tytiger " </div>`

**# identyfikator** – css - selektor pozwalający na wybranie tagów posiadających atrybut **id** równe danemu



# Tag i wiele zasad

Selektor definiuje zasady, według których przeglądarka określa znaczniki, do których zostaną zastosowane style. **Tag może dopasowywać selektory kilku reguł jednocześnie.**

## 2. Selektory złożone

# Złożony selektor

```
6  <style>
7
8      div.cat {
9          background: purple;
10     }
11
12     .dog.bird{
13         background: cyan;
14     }
15
16 </style>
```

Selektor złożony pozwala ustawić regułę dla tagów, które muszą pasować do kilku prostych selektorów jednocześnie, np.: *mieć dwie określone klasy* , lub *tag musi być określonego typu i mieć określoną klasę* .

## Pseudoklasa :not()

```
6  <style>
7
8      .cat:not (.dog) {
9          background: cyan;
10     }
11
12 </style>
```

Selektor negatywny zaznaczy wszystkie tagi z klasą **.cat** z wyjątkiem tych, które również mają klasę **.pies** Selektora negacji można także używać w bardziej złożonych wyrażeniach. **:nie()** akceptuje tylko prosty selektor!!!

# Pseudoklasy

W CSS natkniesz się na konstrukcję zapisaną dwukropkiem, wówczas taką konstrukcję nazywa się **pseudoklasą** . Pseudoklasy służą do wskazywania znacznika na podstawie jego pozycji w dokumencie lub stanu dynamicznego lub opiera się na innych charakterystycznych cechach.

**Selektor: pseudoklasa { . . . }**

# 3. Selektory kombinowane

# Selektor sąsiad

```
6  <style>
7
8      #tiger ~ .dog {
9          background: green;
10     }
11
12 </style>
```

To jest selektor „sąsiadujący”. Pomoże nam wybrać wszystko tagi z klasą **.dog**, które są na tym samym poziomie (bezpośrednie dzieci tego samego rodzica) i podążają za tagiem z **id** równy **tiger** .

# Selektor pierwszy sąsiad

```
6  <style>
7
8      #tiger + .dog {
9          background: green;
10     }
11
12 </style>
```

To jest sąsiedni selektor. Pomoże nam to wybrać pierwszy tag z klasą **.dog** , który jest na tym samym poziomie (bezpośredni potomek tego samego rodzica) i **od razu podążać** za tagiem z **id** równy **tiger** .



## Selektor elementu podrzędnego (tylko znak spacji) - gdy trzeba znaleźć element zagnieżdżony w innym elemencie

```
6  <style>
7
8      .dog .cat {
9          color: red;
10     }
11
12 </style>
```

Selektor mówi, że regułę należy zastosować do tagu, który ma klasę **.cat**, ale tylko wtedy, gdy wśród rodziców ma tag z klasą **.dog** (zamiast klas możesz użyć id, nazw tagów lub połączonego selektora).

## Selektor bezpośredniego elementu potomnego ( znak „>” ) – gdy trzeba znaleźć element będący bezpośrednim potomkiem

```
6  <style>
7
8      .dog > .cat {
9          color: red;
10     }
11
12 </style>
```

Selektor mówi, że regułę należy zastosować do tagu, który ma klasę **.cat** , ale tylko wtedy, gdy jest **bezpośrednim** potomkiem tagu z klasą **.dog** ( zamiast klas możesz użyć id , nazwy tagu lub połączonego selektora ).

## 4. Selektor porządkowy

### **:nth-child(n)**

# Selektor porządkowy

```
p:nth-child(odd) {  
  background: red;  
}  
p:nth-child(even) {  
  background: blue;  
}  
p:nth-child(5) {  
  background:purple;  
}  
p:first-child {  
  background: orange;  
}  
p:last-child {  
  background: #green;  
}
```

```
<div>  
  <p>tekst 1</p>  
  <p>tekst 2</p>  
  <p>tekst 3</p>  
  <p>tekst 4</p>  
  <p>tekst 5</p>  
  <p>tekst 6</p>  
  <p>tekst 7</p>  
  <p>tekst 8</p>  
  <p>tekst 9</p>  
  <p>tekst 10</p>  
</div>
```

Jeśli selektor pasuje do więcej niż jednego elementu, to pseudoklasa **:nth-child** pozwala nam wyjaśnić liczby porządkowe elementów (wśród potomków rodzica), które nas interesują. **:first-child** i **:last-child** wskazują element, jeśli jest to pierwszy lub ostatni element po jego rodzicu ( odpowiednio ).

**5. ~~Kaskadowanie~~**

**~~Specyficzność~~**

**Jak przeglądarka rozwiązuje  
konflikty?**

**Do jednego tagu można zastosować  
wiele reguł, ale co  
jeśli są one ze sobą sprzeczne?**

# W przypadku konfliktu przeglądarka preferuje jedną z właściwości stylu. Zasady mają priorytety.

## Kolejność priorytetów jest następująca:

1. style opisane bezpośrednio w tagu w atrybucie **style=„...”** ;
2. Poniżej znajdują się reguły posiadające selektor dla atrybutu **id** , tj. jak **#raport { ... }** ;
3. Po tym następują reguły z selektorami dla dowolnych innych atrybutów (w tym atrybutu **class** te. Uprzejmy **.sometype { ... }** ) ;
4. Reguły w selektorze, których wystarczy nazwa tagu, na przykład **h1 { ... }** .

*Jeżeli kilka reguł ma ten sam priorytet, wówczas stosowana jest ta, która występuje jako ostatnia (tzn. znajduje się niżej w kodzie od pozostałych).*

Komenda **!important** napisana po którejkolwiek z reguł stylu nadaje jej najwyższy priorytet, niezależnie od użytego selektora.

# 6. Trochę praktyki





### Кухонный комбайн КХ567-Т

Этот кухонный комбайн КХ567-Т оснащен мощным мотором с 2 скоростями и импульсным режимом, что обеспечивает мощность и точное управление для приготовления любых блюд.

10000 €

7499 €

- 25% !!!

Купить сейчас!

Odtwórzmy projekt komponentu w oparciu o gotowe znaczniki HTML

Style układu muszą zostać odtworzone. Bez wprowadzania zmian w pliku **index.html**

**Przyda się**

# Dowiedz się więcej o selektorach atrybutów

`[attr]{...}`

`[attr=value]{...}`

`[attr~=value]{...}`

`[attr^=value]{...}`

`[attr$=value]{...}`

`[attr*=value]{...}`

**Praca domowa**

# Ukończ grę!

Select the apple directly on a plate



<https://flukeout.github.io/>



## Jane Smith

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sed laborum inventore ratione sequi facilis aliquam quidem illum cupiditate quis aspernatur.

☆ 45

👍 67

✉ MESSAGE

# Zaimplementuj ten komponent

Jak **najbliżej układu** .

## Font Awesome i czcionki Google dla Ciebie pomoże.