

CSS

Selektorer

sandra.larsson@jenseneducation.se



Selektorer

Att välja det man vill styla!

- Är väldigt anpassningsbara
- Så enkelt som möjligt använd **class**!
- Lär känna de mest vanliga - ej krångla till i onödan

CSS Syntax

```
selector {  
    property-1: value;  
    property-2: value;  
    property-3: value;  
}
```

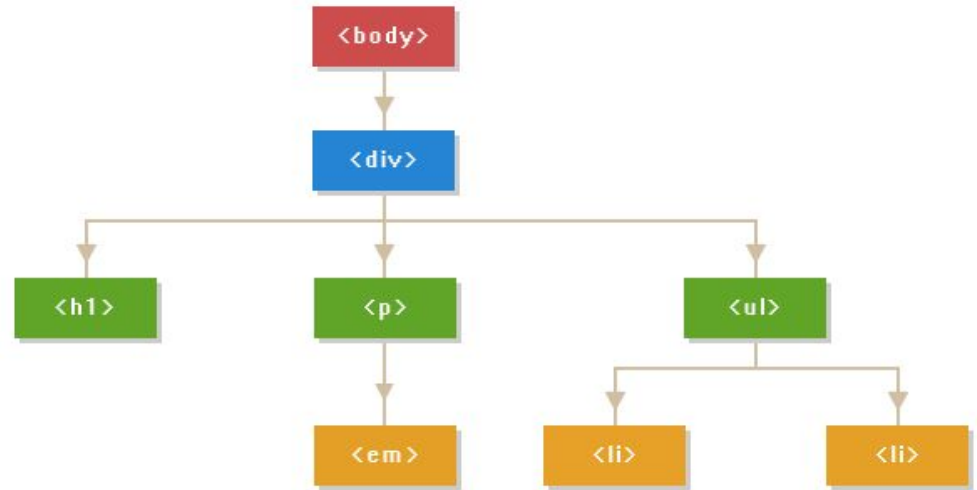
CSS Syntax

```
h1 {  
    font-family: 'Arial';  
    font-size: 18px;  
    color: hotpink;  
}
```

CSS Selectors

Fungerar som en markering i DOM-trädet.

Egenskaperna ska gälla den markeringen.



Selektorer är väldigt anpassningsbara

```
section.articles > article:first-of-type > h1 + h2
```

```
input.validate + label:nth-of-type (2n+1)
```

```
#search-form input[type="text"], #login-form input[type="text"]
```

Elementselektorn X

väljer alla element med namnet X

```
h1 {  
    color: hotpink;  
}
```

Gruppselektorn X, Y

väljer alla element med namnet X och Y

```
h1, p {  
    color: hotpink;  
}
```


Arvselektorn (Descendant) X Y

väljer alla element Y som "bor" i X

```
li a {  
    color: hotpink;  
}
```

Använda en class

.X

väljer alla element med klassnamnet X

Punkt

```
.main-nav {  
    color: hotpink;  
}
```

Använda ett id

#X

väljer elementet med id X

Hashtag!

```
#container {  
    color: hotpink;  
}
```

Skillnad class och id

- ***id*** - är unikt och kan endast användas en gång
 - ***class*** - ingen begränsning
 - Använd **class** i så stor utsträckning som möjligt
 - Bra att känna till **id**, men använd endast om nödvändigt
- !

Universalselektorn *

väljer alla element på sidan - använd sparsamt!

```
* {  
    box-sizing: border-box;  
}
```

Pseudoclass

X:hover

ger ett tillstånd (musen förs över) ett element X

```
p:hover {  
    color: hotpink;  
}
```

Pseudoclass

X:visited

ger ett tillstånd (alla besökta länkar) till element X

```
li a:visited {  
    color: hotpink;  
}
```

Attributselektorn

X[Y]

väljer alla element X som har attributet Y

```
a[title] {  
    color: hotpink;  
}
```


Attributselektorn

X[Y="foo"]

väljer alla element X med attributet Y och värde "foo"

```
input[type="text"] {  
    color: hotpink;  
}
```

Direct child selector

X > Y

väljer alla element Y som är direkta barn till X

```
.container > ul {  
    color: hotpink;  
}
```

```
<div class="container">  
  <ul>  
    <li> List Item  
      <ul>  
        <li> Child </li>  
      </ul>  
    </li>  
    <li> List Item </li>  
    <li> List Item </li>  
    <li> List Item </li>  
  </ul>  
</div>
```

X:first-child

Det första barnet i ordningen till element X

```
ul li:first-child {  
    color: hotpink;  
}
```

X:last-child

Det sista barnet i ordningen till element X

```
ul li:last-child {  
    color: hotpink;  
}
```

Adjacent sibling selector

X + Y

väljer alla element Y som kommer direkt efter X

```
ul + p {  
    color: hotpink;  
}
```

```
<div class="container">  
  <ul>  
    <li> List Item </li>  
    <li> List Item </li>  
    <li> List Item </li>  
  </ul>  
  <p>Some text</p>  
  <h2>Heading 2</h2>  
  <p>More text..</p>  
</div>
```

General sibling selector

X ~ Y

väljer alla element Y som kommer efter X

```
ul ~ p {  
    color: hotpink;  
}
```

```
<div class="container">  
  <ul>  
    <li> List Item </li>  
    <li> List Item </li>  
    <li> List Item </li>  
  </ul>  
  <p>Some text</p>  
  <h2>Heading 2</h2>  
  <p>More text..</p>  
</div>
```

X:nth-child(n)

väljer n:te element som är barn till X

```
section:nth-child(3) {  
    color: hotpink;  
}
```

X:nth-of-type(n)

väljer n:te element av typ X

```
section ul:nth-of-type(3) {  
    color: hotpink;  
}
```


A person with glasses and a mustache, wearing a jacket, is holding a vintage computer keyboard. They are sitting at a desk with two computer monitors in the background. The entire image is covered with a semi-transparent green overlay.

Lets code!

CSS Selektorer

<https://github.com/zocom-sandra-larsson/CSS-basics---Selectors>