

CSS Selektoren

simple

1. Nach Tag-Art, zB: - h1
2. .class
3. #id

Beistriche weisen den Block allen zutreffenden zu! Kann kombiniert werden mit ",", zb `span, li {}`

Am brauchbarsten die Klassen!

1. `div.red` nur divs mit klasse "red"
2. `div.red.green` divs, die beide Klassen haben.

nested

`ul li {}` alle `` inside ``, egal wie tief drinnen.

> direct child

`li.red ~ li` all siblings that come after `li.red`

`li.red + li` **only** the next sibling

pseudo-Klassen

Nach User-Interaktion

`li:hover {}` `input:focus {}`

`input:required {}` für requireden input `input:checked {}` für angeklickte boxen `:disabled`

Nach Kindern

- `:first-child`
- `:last-child`
- `:nth-child()` bzw. `nth-last-child`
 - (3)
 - (2n -1)
- `:only-child` (ist einziges Kind)
- `li:first-of-type` bzw. `li:last-of-type` nimmt nur siblings dieses Typs
- `li:nth-of-type(2)` oder `li:nth-list-of-type(2)`
- `li:only-of-type`

jeweils innerhalb deren Container!

not Selektor

- `li:not(.green)` - alle `` die nicht die Klasse `green` haben

pseudo Elements

```
div.red::after {  
  content: "After";  
  background: red;  
}
```

Ebenso: `::before`

custom attributes

Match, wenn Attribut *vorhanden*

```
[data-red] {  
  background: blue;  
}
```

```
<div data-red>My Div</div>
```

Match, wenn Attribut vorhanden und Value ident:

```
[data-red="true"] {  
  background: blue;  
}
```

```
<div data-red="true">My Div</div>
```

Match, wenn Value beginnt mit:

```
[data-red^="tr"] {  
  background: blue;  
}
```

danach:

```
[data-red$="tr"] {  
  background: blue;  
}
```

match irgendwo:

```
[data-red*="tr"] {  
  background: blue;  
}
```