

# CSS3 – selektorový jazyk, generovaný obsah, media queries

- CSS3 – modularizace (přes 50 modulů)
- velmi výrazně odlišná podpora napříč prohlížeči
  - o důvod pro používání bleeding edge (rizikové, nezdokumentované, neotestované)

## Selektorový jazyk

- **selektor** = řetězec, kterému může vyhovovat libovolné množství prvků na stránce
- při změně stránky (DOM) se mění množina prvků vyhovujících selektorům (nutno znovuvyhodnotit)
- stavební kameny – názvy značek, kombinační znaky, atributové filtry

## Základy

- název značky = všechny uzly daného typu
- čárka = sjednocení
- hvězdička = všechny uzly
- mezera = hierarchický vztah, čteno zprava

```
1. div { ... }
2. a, em { ... }
3. * { ... }
```

```
1. p a { /* odkazy v odstavci */ }
2. ul a { /* odkazy v položkách seznamu */ }
3. p * { /* vše pod všemi odstavci */ }
4. header li em { /* ? */ }
```

- **atributy**
  - o hranaté závorky = atributový filtr
  - o test na rovnost, začátek, konec, podmnožinu

```
1. input[placeholder] { /* ty vstupní prvky, které mají atribut placeholder */ }
2. input[type="text"] { /* test rovnosti hodnoty */ }
3. div[class~="test"] { /* div, jehož atribut class obsahuje mezerou oddělený řetězec test */ }
4. div[id^="article"] { /* atribut začíná na */ }
5. div[id$="article"] { /* atribut končí na */ }
6. div[id*="article"] { /* atribut obsahuje */ }
```

- **zkratky pro class a id**
  - o mřížka: `[id=]`
  - o tečka: `[class~=]`

```
1. div#id { ... }
2. #id { ... }
3. div.test { ... }
4. div.test1.test2 { ... }
```

- **kombinátory**

- o větší než = přímý potomek
- o plus = přímý následující sourozenec
- o tilda = libovolný následný sourozenec

```
1. body > * { /* značky přímo v body */ }
2. img + p { /* odstavec za obrázkem */ }
3. h1 ~ img { /* obrázky, kterým předchází nadpis */ }
```

## Pseudotřídy

- zápis pomocí dvojtečky
- slouží k popsání podmínek, které nemusí být ze stromu stránky patrné
- `:target` = právě aktivní kotva stránky
- `:hover` = při najetí myši
- `:focus` = pokud má focus

```
1. a:hover { font-weight: bold; }
2. input:focus { border: 1px solid red; }
```

- `:enabled`, `:disabled`, `:checked`, `:required` = stav vstupního prvku

```
input:checked + label { font-weight: bold; }
```

- `:nth-child()`, `:first-child`, `:last-child` = pořadí v rámci sourozenců

```
1. a:first-child { /* odkaz, pokud je prvním potomkem svého rodiče */ }
2. p:nth-child(2n+1) { /* liché odstavce */ }
3. p:nth-child(odd) { /* liché odstavce */ }
```

- `:nth-of-type()` = pořadí v rámci sourozenců stejného typu

```
1. img:nth-of-type(2n+1) { float: left; }
2. img:nth-of-type(2n+0) { float: right; }
```

- `:empty` = nemá-li potomky

```
li:empty { display: none; }
```

- `:only-child` = je-li jediný potomek

- `:not()` = negace

```
1. p:not(:target) { opacity: 0.5; }
2. input:not([required]) { color: blue; }
```

## Pseudoprvky

- `::first-line` = první řádek textu
- `::first-letter` = první písmo textu

```
p::first-letter { font-size: 200%; }
```

## Generovaný obsah

- *::before, ::after* = falešný první/poslední potomek

```
1. li.good::before {
2.     content: "✓ ";
3.     color: green;
4. }
5. li.bad::before {
6.     content: "X ";
7.     color: red;
8. }
```

✓ `<li class="good">`

**X** `<li class="bad">`

- generovat lze i obrázky a hodnoty atributů...:

```
1. a[href^="http"]::before {
2.     content: url(externi.png);
3. }
4. a[data-id]::after {
5.     content: attr(data-id);
6.     color: gray;
7. }
```

asd as http://www.seznam.cz/ isc  
asd sa as  
sdasd asd asd asd asd sasd as da c  
asd sasd as da dsd sdasd asd asd a

- ... či uvozovky:

```
1. q {
2.     quotes: "«" "»";
3. }
4. q::before {
5.     content: open-quote;
6. }
7. q::after {
8.     content: close-quote;
9. }
```

» A přece se točí «

- podpora pro čítače – reset, změna hodnoty, výpis hodnoty:

```
1. body {
2.     counter-reset: kapitola;
3. }
4.
5. h2 {
6.     counter-increment: kapitola;
7. }
8.
9. h2::before {
10.    content: counter(kapitola) ". ";
11. }
```

## Media queries

- CSS2 podporuje tzv. typy médií – možnost aplikovat stylopisy jen pro předdef. zařízení
  - o `<link rel="stylesheet" media="print" />`
  - o `@media screen { ... }`
  - o `@import url(...) projection;`
  - o neúplná podpora (9 typů médií, podporovány zpravidla jen screen a print)

- u MQ princip i syntaxe zůstává, typy rozšířeny i na parametry a logické operace

```
@media (min-width:700px) and (orientation:landscape) { ... }
```

- volitelně typ média
- následuje libovolné množství parametrizace v kulatých závorkách

```
@media [screen and] (max-device-width:640px) [and (...)] { ... }
```

- **logické operace:**

- o and = průnik
- o , = sjednocení
- o not = negace

```
@media not screen and (color), print and (color) { ... }
```

- **dostupné parametry:**

- o width, height, device-width, device-height
- o aspect-ratio, device-aspect-ratio
- o color, color-index
- o resolution, monochrome
- o min-, max- varianty předchozích
- o scan, grid, orientation

## Responsive design

- snaha navrhnout layout tak, aby nebyl příliš závislý na použitém zařízení
- používání relativních jednotek
- rozložení prvků na stránce dle dostupného místa
- sada pravidel/stylopisů pro různé media queries

- **viewport**

- o mobilní zařízení používají trik pro zpětně kompatibilní zobrazení
- o stránce se hlásí s falešnou pixelovou šířkou (typicky 960px)
- o výsledek = uživatel vidí celou stránku (zmenšenou)
- o mapování logických pixelů na fyzické lze nastavit

```
1. <meta name="viewport" content="width=640" />
2. <meta name="viewport" content="width=device-width" />
3. maximum-scale, minimum-scale, user-scalable
```

- nové jednotky - (vw, vh, vmin, vmax), procenta rozměrů průhledu

```

1  ▾ body {
2      font-size: 20vw;
3  }
4
5  ▾ body::before {
6      content: "Široký";
7      color: green;
8  }
9
10 ▾ @media all and (max-width:400px) {
11 ▾  body::before {
12     content: "Úzký";
13     color: red;
14  }
15 }
16

```

## Vendor prefix

- **problém:** výrobce chce podporovat technologii, která není plně standardizovaná
  - o implementací riskuje stížnosti při změně API
- **řešení:** vlastní prefix před názvem vlastnosti

```

1. div {
2.     -webkit-transform: rotate(45deg);
3.     -moz-transform: rotate(45deg);
4.     -ms-transform: rotate(45deg);
5.     -o-transform: rotate(45deg);
6.     transform: rotate(45deg);
7. }

```

- psaní prefixů je opruz, poslední dobou snahy o odprefixování
- preprocessing na straně serveru (generuje všechny, např. Autoprefixer)
- preprocessing na straně prohlížeče (generuje jen potřebné)

## Preprocessing

- příklad - **Less.js**

vstup:

```

1. p {
2.     @color: red;
3.     font-size: @color;
4.
5.     a {
6.         color: #fff - @color;
7.         background-color: @color;
8.     }
9. }

```

výstup:

```

1. p {
2.     font-size: red;
3. }
4. p a {
5.     color: #00ffff;
6.     background-color: red;
7. }

```