

CSS

1 Úvod – použití, výhody a nevýhody CSS

CSS (*Cascading Sytle Sheets*) neboli kaskádové styly se využívají k formátování obsahu HTML, XHTML a XML dokumentů.

První návrh normy byl zveřejněn v roce 1994, v roce 1996 byla pak vydána specifikace CSS 1, v roce 1998 CSS 2, nyní se pracuje na verzi CSS 3.

1.1 Výhody CSS

- stránky se rychleji zobrazují
- širší možnosti formátování
- menší a strukturovanější kód
- změna na jednom místě
- není třeba zasahovat do HTML
- rozdílné pořadí textu v kódu a na stránce, logický vzhled v lineárních prohlížečích
- dá se tím udělat totéž, co tabulkami
- validita

1.2 Nevýhody

hlavní nevýhodou CSS je zatím stále špatná podpora v prohlížečích

různé prohlížeče interpretují stejný CSS kód jinak a je někdy velmi obtížné jej napsat tak, aby se na všech (resp. na několika vybraných) prohlížečích výsledek zobrazil stejně

2 Využití, princip CSS a definice stylu v dokumentu

2.1 Kdy používat CSS

- při správě většího webu s mnoha stránkami, které by měli vypadat podobně
- často píšete texty určené pro Internet a nechcete ztrácet čas složitým formátováním
- pomocí CSS můžete například nastavit libovolnou a přesnou velikost písma udělat odsazení prvního řádku odstavce, zvětšit řádkování, zrušit nebo zvětšit prázdný prostor po odstavci automaticky formátovat nadpisy zvýrazňovat odkazy po přejetí myši udělat automaticky grafické odrážky, nebo spoustu dalších věcí
- zabýváte se skriptováním, zejména Javascriptem
- v kombinaci se skripty je dnes CSS nejmocnější zbraň pro "rozhýbání" stránek

2.2 Princip CSS

obecný zápis kaskádového stylu

selektor {[deklarace](#)}

selektor:

- selektor typu (type)
- jméno selektoru je odvozeno od jména html elementu (tagu)
- takto nastavený styl se projeví u všech elementů daného druhu v dokumentu

o `h1 {text-align: center; color: blue;}`

- selektor třídy (class)
- jméno selektoru definuje autor tak, že zapíše tečku, a za ni uvede jméno třídy
- v HTML dokumentu může použít tuto třídu jako hodnotu atributu class nějakého elementu
- počet elementů příslušné třídy v jednom dokumentu není omezen

- o `p.novinka {letter-spacing: 0.3em; background-color: green;}`
- o `.novinka {background-image: url(new.png);}`

- selektor id
- jméno tohoto selektoru definuje autor tak, že zapíše znak hash (#), a za něj uvede jméno id
- v HTML dokumentu může použít id jako hodnotu atributu id libovolného elementu
- v celém dokumentu může být příslušné id použito jen u jednoho elementu
 - o `#kapitola3 {font-variant: small-caps;}`
 - o `#navigacni_schema {font-size: 12pt; color: lime;}`
- selektor pseudotřídy (pseudo-class)
- použitím selektorů pseudo-třídy zajišťujeme, v závislosti na aktuálním stavu dokumentu, jeho struktuře nebo prohlížení, přiřazení vlastnosti elementům
- selektor pseudo-třídy zapíše autor tak, že uvede znak dvojtečka, za který napíše název pseudo-třídy
- mezi pseudotřídy patří: first-child, link, visited, hover, active, focus, lang
- `a:link {color: green;}`
- `a:hover {color: red;}`
- selektor pseudo-elementu (pseudo-element)
- pseudo-elementy nám umožní přiřadit vlastnosti elementům v závislosti na stavu dokumentu
- selektor pseudo-elementu zapíše autor tak, že uvede znak dvojtečka, za který napíše název pseudo-elementu
- mezi pseudo-elementy patří: first-line, first-letter, before, after
- `p.abstrakt:first-line{color: blue;}`
- `p.ukol:before{font-size:xx-small;}`
- univerzální selektor
- v jazyce CSS byl definován i univerzální selektor
- ten je zapsán pomocí znaku hvězdička *
- použitím tohoto selektoru specifikujeme, že vlastnosti u tohoto selektoru uvedené, budou aplikovány na všechny elementy dokumentu
- `* {color:blue;}`

deklarace: deklarace obsahuje seznam vlastností nastavených pro daný selector. Deklarace může vypadat třeba takto:

```
{vlastnost1:hodnota; vlastnost2:hodnota; ... ;vlastnostn:hodnota}
```

2.3 Definice stylu

Styl se může nadeklarovat třemi způsoby:

- Přímý zápis
- přímo v textu zdroje u formátovaného elementu pomocí atributu `style="..."`
- nešikovné, ale občas se používá
- Ukázka:
 - `<p style="color: red">Tento odstavec bude červený.</p>`
- Pomocí "stylopisu" ("stylesheet") v hlavičce stránky
- stylopis je jakýsi seznam stylů
- je v něm obecně napsáno, co má být jak zformátováno, například že nadpisy mají být zelené
- do stránky se stylopis píše mezi tagy `<style>` a `</style>`
- Ukázka:
 - Do hlavičky se napíše:


```
<style>
p {color: red}
</style>
```
 - a do těla stránky:


```
<p>Tento odstavec bude červený. </p>
```
- V těle dokumentu budou všechny odstavce červené.
- Použitím externího stylopisu
- soubor *.css, na který stránka odkazuje tagem `<link>`

- v souboru je umístěný styl
- hlavní výhodou je v tom, že na jeden takový soubor se dá nalinkovat mnoho stránek, takže pak všechny vypadají podobně
- často používané
- Ukázka: Vytvoří se soubor, který se pojmenuje třeba `style.css`. V něm bude pouze tento text:
- `p {color: red}`
- Do hlavičky html dokumentu, který má být stylem ovlivněn, se musí napsat odkaz na tento soubor:
- `<link rel="stylesheet" type="text/css" href="style.css">`
- V těle dokumentu pak budou opět všechny odstavce červené.
- Je nutné všimnout si, kde se používají uvozovky, dvojtečky, složené závorky, středníky a čárky.
- Příklad správného zápisu:

```
h2 {color: green; background-color: yellow}
```

```
Komentáře ve stylech se dělají /* komentář */
```

3 Vlastnosti stylů – dědičnost, priorita použití

3.1 Dědičnost

Je vlastnost, kdy vnořený element přijímá vlastnost od nadřazeného.

Ovšem ne každá vlastnost se dá dědit. Pokud chceme vědět, jestli se daná vlastnost dědí, poté nám nezbyvá nic jiného než se podívat na specifikaci CSS na W3C, kde pod položkou Inherited (dědičnost) je uvedeno yes/no.

Příklad: jednoduchá XHTML stránka:

```
<html>
<head>
<title>Titul</title>
</head>
<body>
<h3>Nadpis</h3>
<p>odstavec</p>
<div>
<h4>Nadpis</h4>
<p>odstavec</p>
</div>
</body>
</html>
```

Nadpis `h3` a `p` jsou potomky elementu **body** a proto budou vykresleny červenou barvou písma. Naproti tomu nadpis `h4` a odstavec `p` umístěné v elementu **div** jsou jeho přímými potomky, a protože jsme v souladu se zásadami specifičnosti změnili definici, budou všechny texty v elementech uvnitř `div` vykresleny černou barvou písma.

Vidíme tedy, že elementy mimo `div` zdědily barvu písma definovanou pro element `body`. Naopak elementy umístěné v bloku `div` zdědily barvu písma definovanou pro něj definovanou.

3.2 Priorita použití

1. zápis stylu v externím souboru
 - a. tento zápis má nejnižší prioritu, tzn., že platí u elementů, které jsou stylvány v externím souboru a nikde jinde
2. zápis v hlavičce dokumentu
 - a. tento zápis má střední prioritu, tzn., že pokud v externím souboru stylujeme element např. `h1` modře a zde v hlavičce dokumentu červeně, pak bude platit barva červená.
 - b. pozor: musíme si uvědomit, že externí soubory nám většinou platí pro celý náš web, ale styl v hlavičce dokumentu platí pouze pro ten samý dokument, kde je uveden
3. zápis přímo u elementu

- a. tento zápis má prioritu nejvyšší, tzn., že pokud je prvek h1 stylován ve výše zmíněných možnostech a přímo u elementu je stylován černě, pak bude platit černá deklarace
 - b. dále si musíme uvědomit, že takto můžeme nebo musíme stylovat každý element
4. Další možností kde využíváme prioritu, je skládání více vlastností za sebe.

Příklad:

```
<span style="color: black"><span style="color: red"> text </span></span>
```

V tomto případě se bude brát v úvahu vlastnost bližší obsahu elementu, tzn., že v tomto případě bude text červený.

4 problémy s definicí šířky (width) a výšky (height) elementů (objektů, prvků) v různých prohlížečích

Internet Explorer poněkud chybně interpretuje vlastnost šířky objektu width, když do ní zahrnuje i padding a border.

Chybná interpretace vlastnosti width
v Internet Exploreru

