

Kaskádové Styly

- v anglickém originále **Cascading Style Sheets** se zkratkou **CSS**, je jazyk pro popis
- způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML
- Jazyk byl navržen standardizační organizací **W3C(World Wide Web Consortium)** je mezinárodní konsorcium, jehož členové společně s veřejností vyvíjejí webové standardy pro **World Wide Web (www.----)**
- autorem prvotního návrhu byl **Håkon Wium Lie**

Dvě úrovně specifikace: **CSS1**
CSS2

- dne 7. června 2011 byla dokončena revize CSS 2.1 a pracuje se na verzi CSS3

Syntaxe

- Definice kaskádových stylů sestává z několika pravidel
- Každé pravidlo obsahuje **selektor** a **blok deklarací**
- Každý blok deklarací pak obsahuje seznam deklarací oddělených **středníky ;** a každá deklarace sestává z identifikátoru vlastnosti, následuje **dvojtečka :** a hodnota vlastnosti
- Nepovinně ještě může následovat označení **!important**, které zvýší sílu deklarace

```
body {  
    background-color: white;  
    color: black;  
    padding: 10px !important;  
}
```

Selektory

CSS definuje mnoho různých selektorů, které obvykle můžeme kombinovat. Mezi základní patří

- **body** – Tyto deklarace budou platit pro všechny výskyty elementu body
- **body p** – Tyto deklarace budou platit pro všechny elementy p, které se nachází v elementu body, v jakékoliv hloubce
- **body>div** – Tyto deklarace budou platit pro všechny elementy div, které jsou dětmi elementu body. To znamená, že pokud bychom měli element div, který se nachází v <body><blockquote><div>..., tyto deklarace by pro něj neplatily, protože tento div není přímým potomkem elementu body
- **.class** – Tyto deklarace budou platit pro všechny elementy, které mají v HTML nastavenou třídu trída. To se provádí pomocí HTML atributu class
- **#id** – Tyto deklarace budou platit pro všechny elementy, které mají v HTML nastavený identifikátor id. To se provádí pomocí HTML atributu id
- **sel1, sel2, sel3** – Selektory můžeme seskupovat pomocí čárek. Následující deklarace pak budou platit pro všechny selektory

Připojení kaskádových stylů do HTML stránky

- existuje několik možných způsobů, jak aplikovat kaskádové styly v HTML dokumentu
- v praxi se nejčastěji používá odkaz na externí soubor

a) **Přímý in-line zápis stylu pomocí atributu style.** Tato pravidla budou aplikována pouze na dotyčný element:

```
<p style="color: red; text-decoration: underline">Tento odstavec bude červený a  
podtržený.</p>
```

b) **Zápis stylů do elementu style.** Takové styly se aplikují na celou stránku podle předepsaných selektorů:

```
<style type="text/css">  
  #hlavicka{  
    width: 200px;  
    height: 450px;  
  }  
</style>
```

c) **Připojení externího souboru pomocí elementu link:**

```
<head>  
  <link rel='stylesheet' href='style.css' type='text/css'>  
</head>
```

d) **Připojení externího souboru pomocí http hlavičky link:**

```
Link: <fancy.css>; rel=stylesheet
```

Výhody CSS

- Rozsáhlejší možnosti formátování. Například pro formátování bloku textu – tj. určení vzdálenosti od jejich elementu či okraje stránky nenabízí HTML nic. CSS má vlastnosti padding a margin. V HTML by bylo potřeba vytvořit složitou konstrukci vnořených tabulek
- Jednodušší údržba webové prezentace. Pokud chceme změnit nějaký detail, jako třeba barvu nadpisu, nemusíme složitě procházet HTML kód nebo různé HTML šablony, ale můžeme změnit pouze jednu vlastnost v CSS souboru, který je přilinkován ke všem stránkám
- Oddělení struktury a stylu. V jednom (HTML) dokumentu budeme mít pouze sémanticky označen obsah a v druhém (CSS) dokumentu máme definice vzhledu stránek. Tím dosáhneme snadnějšího strojového zpracování samotného obsahu stránek, do kterého se nám nepletou prvky definující vzhled
- Cachování stylů – webový prohlížeč si může soubor se styly uložit do cache paměti, čímž může být dosaženo zrychlení načtení stránky. Na druhou stranu při použití externího CSS souboru dochází k dalšímu HTTP požadavku navíc oproti tomu, když bychom použili buď online CSS nebo přímo formátování HTML

- CSS vlastnosti jednotlivých elementů můžeme dynamicky měnit pomocí Javascriptu
- S pomocí CSS můžeme jednoduše formátovat i jakýkoliv jazyk XML.
- Mohou také existovat různé styly pro různá výstupní zařízení. Jednoduše tak můžeme nadefinovat různý styl pro tisk, projekci, mobil, PDA, běžný prohlížeč či dokonce pro každý prohlížeč jiný styl. Specifikace CSS nezapomínají ani na zrakově postižené – je možno napsat styly pro hlasový syntetizátor nebo hmatovou čtečku Braillova písma
- Koncový uživatel si může napsat svůj vlastní CSS styl pro libovolnou stránku. Většina prohlížečů nějakým způsobem podporuje uživatelské styly, takže uživatel si může například nastavit, aby měl všechny odkazy na všech webech vždy podtržené nebo aby na tomto konkrétním webu mělo písmo vždy černou barvu
- V kombinaci s Javascriptem mohou vzniknout účinné bookmarklety, které mohou různě vylepšovat vzhled stránky. Například odstranit všechny obrázky na pozadí, změnit pozadí na bílé a písmo na černé apod

Nevýhody CSS

- Hlavní nevýhodou CSS je ne vždy dostatečná podpora v majoritních prohlížečích
- Prohlížeče obsahují v implementaci CSS chyby a je někdy nesnadné napsat kód tak, aby se ve všech (resp. v běžných) prohlížečích výsledek zobrazil stejně
- Situace se poslední dobou značně zlepšuje, mj. opravami chyb v prohlížečích a nástupem nových verzí Internet Exploreru, který byl ve své šesté verzi zdrojem častých chyb v CSS

Limity CSS

- CSS selektory neposkytují přístup k rodičovským prvkům, jako to umožňují sofistikovanější jazyky jako například XPath. Nemůžete například nastylovat jen ty odstavce, které obsahují odkaz
- Horizontální kontrola prvků na stránce je intuitivní a jednoduchá, naopak vertikální stylování už působí problém. Například na obvyčejné vertikální centrování obsahu se musí používat poměrně složité kombinace a hacky, které ještě nemusí být dopředně kompatibilní
- CSS neposkytuje možnost pro symbolický zápis proměnné nebo konstanty, všechny hodnoty musí být vepsány přímo v kódu. Například pokud se na vícero místech používá stejná barva, nemůže se použít symbolický zápis barva=red; a poté už jen psát proměnnou barva, všude se musí uvádět přímo hodnota red
- CSS neumí počítat výrazy, prohlížeče například neporozumí výrazu margin-left: 10% - 3em + 4px;. Webdesigner si to vše musí spočítat sám
- CSS2 nenabízí žádnou možnost pro tvorbu kulatých rámečků nebo jiných kulatých objektů. Pracuje pouze s obdélníky
- CSS2 nenabízí žádnou možnost, jak jednomu elementu přiřadit více obrázků na pozadí

CSS Tagy

background-color:	- určuje barvu pozadí - název barvy (v Aj), transparent(průhledná)
background-image:	- určuje obrázek na pozadí - url(„cesta a název obrázku“)
font-family:	- určuje font, který se použije k vykreslení písma - jméno fontu (Arial, ...)
font-size:	- určuje velikost písma - velikost v jednotkách (px, mm, %,...) - klíčové slovo (small,...)
font-style:	- určuje normální písmo nebo kurzivu - normal (nekurzíva, normální stojaté písmo) - italic (kurzíva, sklonění písma) - oblique (skloněné písmo)
font-weight:	- určuje tučnost písma neboli duktus - normal (písmo nebude tučné) - bold (písmo bude tučné) - násobky (různé stupně tučnosti) - a další...
text-align:	- určuje zarovnání odstavce nebo jiného bloku - left, center, right, justify
position:	- určuje způsob počítání polohy prvku - absolute (prvek bude vyjmut z toku dokumentu a umístěn na určitých souřadnicích) - relative (prvek bude posunut ze svého normálního umístění) - static (prvek je umístěn normálně v dokumentu) - fixed (prvek je umístěn absolutně vzhledem k oknu - nefunguje v Exploreru 6)