

CSS - 1. část

+ git

Ing. Martin Dostal, Ph.D.
madostal@kiv.zcu.cz
github.com/madostal/kiv-web

Obsah

- Základy CSS
- Git a github
- Diskuze

Web - technologie

JavaScript

JavaScript - interaktivita s uživatelem

CSS

CSS - prezentace

HTML

HTML - obsah včetně struktury

CSS

- Co to je?
 - CSS = Cascading Style Sheets = kaskádové styly
- K čemu to je?
 - popisuje vzhled webové stránky
 - jak mají html značky vypadat na obrazovce, na tabletu, při tisku ...
- Proč?
 - oddělení obsahu (html) od jeho vzhledu (css)
 - 1 css soubor může popisovat vzhled celého webu

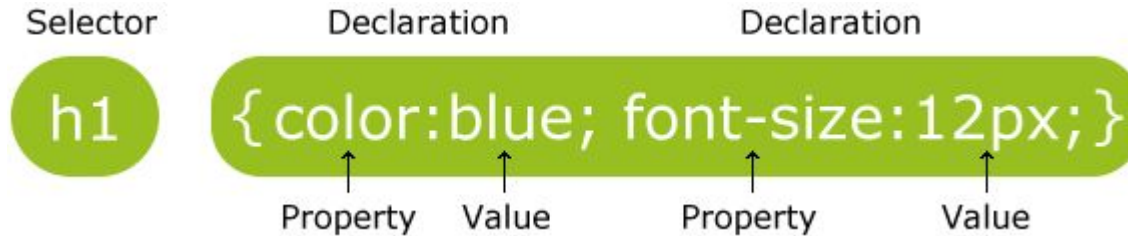
CSS - příklad

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: red;
  text-align: center;
}
</style>
</head>
<body>

<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>

</body>
</html>
```

CSS syntaxe



- selektor - co se bude formátovat
 - **element** - h1, p
 - **id** - #obsah {} - právě 1 element s tímto id na 1 stránce !!!
 - pozor: ID nesmí začínat číslem ~~#4article~~
 - **class** - .red {} - více elementů s touto class
- deklarace - pravidla ve formátu vlastnost: hodnota;

CSS selektory

- Definují to, co má být formátováno
- Jeden prvek: `p {...}`
- Seskupení prvků: `h1, h2 {...}`
- Kontextový selektor: `ul li {...}` - vnoření prvků
- Selektor třídy: `.dulezite {color: red}`
 - použití u prvku `<p class="dulezite">`
 - prvek ve více třídách: `<p class="trida1 trida2">`
- Třída pro jeden typ prvků
`p.dulezite {...}` - aplikuje se pouze na `<p class="dulezite">`
- Selektor id: `#html {...}`, u značky `<p id="html">`
 - Id musí být v dokumentu jedinečné a nesmí začínat číslem!!!

CSS - proč kaskádové styly?

- Kaskádové styly - styly se kaskádovitě přepisují
 - obecná pravidla lze přepsat přesnějšími pravidly

```
body {font-size: 12px;}
```

```
p {font-size: 10px;}
```


Pseudotřídy - pro speciální stavy elementu

- Odkazy
 - a:link - nenavštívený odkaz
 - a:visited - navštívený odkaz
 - a:active - vybraný odkaz
 - a:hover - po najetí myši
- Formuláře
 - input:focus { color: red; } - aktuálně vybraný, píšu do něj
- Text
 - p:first-line - první řádek
 - p:first-letter - první písmeno

CSS komentáře

```
p {  
    color: red;  
    /* This is a single-line comment */  
    text-align: center;  
}
```

CSS - vložení do stránky

3 možnosti:

1. Externí css soubor - jediná správná varianta
2. Interní css pravidla v <head>
3. Inline styl

ad 1)

<head>

`<link rel="stylesheet" type="text/css" ref="mystyle.css">`

</head>

Interní css pravidla

ad 2)

```
<head>
<style>
body {
    background-color: linen;
}

h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
```

Inline style - pouze pro případy nouze

ad 3)

```
<h1 style="color:blue;margin-left:30px;">
```

This is a heading.

```
</h1>
```

- pouze pro prototypy, dočasné html stránky
- možné použití v rámci příspěvků v cizích systémech, když je to nezbytně nutné

CSS - poznámky

Nepřidávejte mezeru mezi hodnotu a jednotku:

špatně: `margin-left: 20 px;`

správně: `margin-left: 20px;`

CSS - formátování textu

- text-align: left | center | right | justify
- text-decoration: none | underline | blink | ...
- vertical-align: baseline | sub | text-bottom | ...
- word-spacing: 1em (letter-spacing)
- text-decoration: none | underline | overline | line-through | blink

CSS - barvy

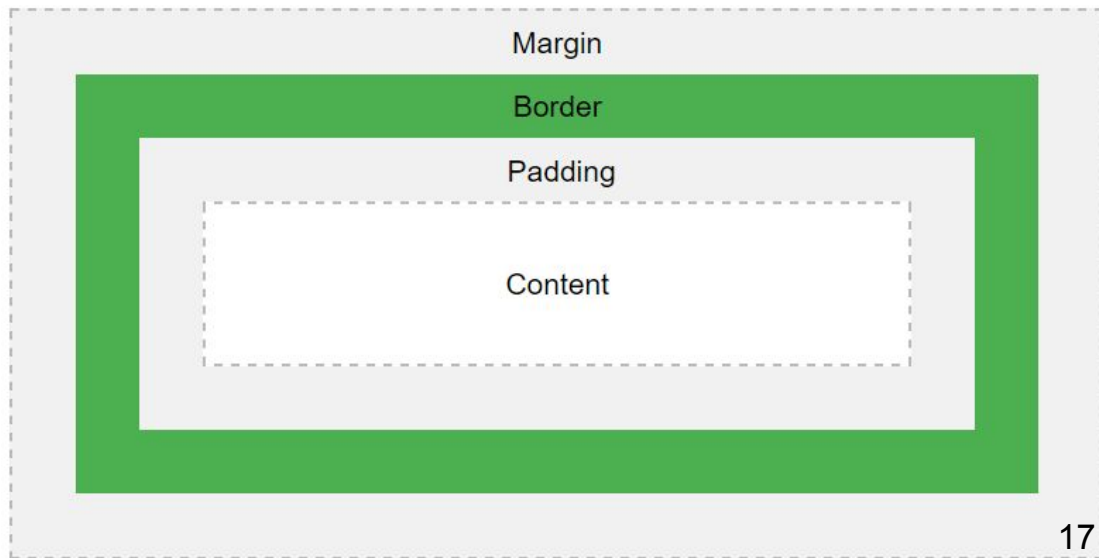
3 možnosti definice barvy:

- název barvy - např. red
- RGP - rgb(255, 0, 0)
- HEX hodnota - #000000 - asi nejrychlejší pro zápis

Všechny 3 možnosti jsou v praxi používané.

Box model

- box model je používán v souvislosti s designem a layoutem
- padding - prázdná oblast kolem obsahu, je vždy průhledná
- margin - prázdná oblast kolem ohraničení, je průhledná
- border - ohraničení, lze zadat barvu a styl
- content = obsah



CSS overflow

CSS overflow specifikuje, co udělat s obsahem, když se text nevejde do cílové oblasti (např. divu):

- visible - defaultní, text se zobrazí mimo ohraničenou oblast
- hidden - text mimo oblast je oříznut
- scroll - text je oříznut, ale je přidán scrollbar pro zobrazení zbytku obsahu
- auto - pokud je obsah oříznut, scrollbar bude přidán automaticky

CSS float a clear

- Obtékání
 - float: none | left | right
 - clear: none | left | right | both - ukončit obtékání
- Absolutní x Relativní pozicování
 - position: absolute | relative | static
 - left, top, bottom, right

Proč? Pokud chceme zobrazit 2 divy vedle sebe.

```
#div1 { float: left;  
        width: 100px; }
```

```
#div2 { float: left;  
        width: 100px;  
        margin-left: 50px; }
```

Kaskáda a dědičnost (1)

- Obsah daného prvku dědí vlastnosti daného prvku
 - např. font-size relativně k nadřazenému prvku
- Některé vlastnosti se nedědí
 - background, background image, margins, borders nebo padding
- Dědičnost:
 - Je hodnota někde určena? -> kaskáda
 - Není-li určena, může se zdědit?
 - Použij výchozí hodnotu
- Kaskáda: prvek může mít určitou vlastnost definovanu vícekrát.

Která definice se použije?

Kaskáda a dědičnost (2)

- Ovlivnění kaskády s pomocí příkazu !important, ale raději se tomu vyhněte
- Kaskáda - postup:
 - Najdi všechny definice pro daný prvek
 - Seřaď je podle váhy
 - Prohlížeč < uživatel < web < web s !important < uživatel s important
 - Pokud stejná váha – seřaď podle specifičnosti (specifičtější selektory mají přednost před obecnějšími)
 - Seřazení podle pořadí definování
- Příklady specifity:
 - LI specificity = 1
 - UL OL LI specificity = 3
 - LI.red specificity = 11
 - OL LI.red specificity = 12
 - #x34y specificity = 100
 - style="..." specificity = 1000

Verze CSS - 1, 2 a 3

- CCS 1 - publikována 1996
 - font, color, margin, padding ...
- CSS 2 - publikována 1998
 - pozicování, text LTR a RTL - funguje angličtina i arabština
 - box model
 - stíny a další rozšíření CSS 1
- CSS 3 - publikována 2012, ale další moduly až do 2014
 - zaoblené rohy
 - animace
 - podpora různých typů médií - screen, print, projection ...

Další selektory v CSS 2

- Univerzální selektor (*) = libovolný element
- div * p – p je potomek divu
- děti (>) a sourozenci (+)
- p > em – <p><i> nevyhovuje
- h1 + h2 – <h2> po <h1>
- Atributy ([...])
 - p[attr] – vyhovuje p s nastaveným attr
 - p[attr=val] – totéž, pokud attr="val"
 - p[attr~=val] – totéž, pokud <p attr="x y val z">
- Přidané pseudotřídy
 - p:first-child – matches first child only
 - p:lang(en) – matches if P's lang is "en"

Generování obsahu

- p:before – značí začátek prvku
- p:after – značí konec prvku
- content – vložený obsah

Příklad:

```
.poznámka: before {  
    content: „Poznámka: “;  
}
```


GIT

Verzování

- je uchovávání historie všech změn v datech
- využívá se pro sledování změn ve zdrojových kódech
- eviduje kdo, kdy a co změnil
- umožňuje spolupráci velkého množství programátorů
- 2 základní typy:
 - centralizované
 - distribuované

Proč to děláme? (5x Proč?)

Verzovací systémy

- Centralizované
 - CVS (Concurrent Version System) - verzuje se každý soubor zvlášť, dnes již zastaralé
 - SVN - subversion - client-server, data uložena na serveru, nelze provádět commity bez Internetu
- Decentralizované
 - Git - každý vývojář má vlastní lokální repozitář a otestované funkční celky odesílá na sdílený repozitář (origin)

Úvod do GITu

- Git je distribuovaný systém správy verzí
- je navržen jako sada nástrojů

Postup práce s Githubem:

1. Vytvoření hlavního repozitáře (např. Github) - origin
2. Clone repozitáře,
 - a. pokud již existuje, tak Pull = stažení aktualizací
3. Provedení změn nebo přidání nových souborů
4. Commit - potvrzení změn + okomentování, commit získá identifikační kód
5. Push - odeslání na server

Git - příkazy

- `git init` - vytvoření repozitáře v daném adresáři
- `git status` - status repa
 - dostanu např. untracked files
- `git add` - přidání souboru
 - `git add '*.txt'` - je možné používat wildcards
- `git commit -m "Comment"` - commit změn
- `git log` - seznam commitů s komentáři
- `git remote add origin https://.../try_git.git` - přidání remote repozitáře
- `git push -u origin master` - odeslání, -u = zapamatovat a příště mohu jen `git push`

Git - příkazy 2

- `git pull origin master` - stažení změn od ostatních
- `git diff HEAD` - změny od mého posledního commitu
- `git checkout` - vrácení k danému commitu
- `git branch test` - vytvoření větve test
 - `git checkout test` - přepnutí do branche test
 - `git rm '*.txt'` - odstranění všech .txt souborů
 - `git checkout master` - přepnutí do větve master
- `git merge test` - spojení změn do masteru
 - `git branch -d test` - smazání větve
- `git push` - odeslání změn

Github - zkuste si

- <https://try.github.io> - zkuste si práci s gitem v prohlížeči

Diskuze

- Dotazy?
 - Nepochopil jsem ...
 - Zajímá mě ...
- Opakování
 - K čemu je dobré CSS?
 - Proč zrovna kaskádové styly?
 - Rozdíl mezi id a class?
 - `<div id="blok1" />` a `<div class="blok" />`

CSS - zdroje pro samostudium

- <http://www.w3schools.com/css/>
 - doporučuji celé samostatně projít a vyzkoušet!!!

Následující přednáška:

- CSS 3
- Front-end framework Bootstrap - použití pro SP
- Preprocessory - Sass a Less