

# BGWE2

Mgr. Vlastislav Kučera, Ph.D.

CSS - opakování, kaskádování, pseudotřídy

# Selektory

- ▶ univerzální
- ▶ typový
- ▶ následníka
- ▶ potomka
- ▶ sousedních sourozenců
- ▶ třídy
- ▶ ID

# Selektory - 1. část

- ▶ univerzální
  - ▶ \*
  - ▶ váže se k prvku libovolného typu, tj. ke všem prvkům ve stromu dokumentu
- ▶ typový
  - ▶ váže se ke jménu prvku v jazyce dokumentu
  - ▶ `H2{}` - pravidlo, které se váže na všechny tagy h2
- ▶ následníka
  - ▶ váže se k prvkům, které jsou následníky jiného prvku
  - ▶ skládá se ze dvou či více selektorů oddělených mezerou
  - ▶ `p em{}` - váže na všechny prvky em, které jsou obsaženy v p; pokud bude em mimo p, toto pravidlo se nepoužije

# Selektory - 2. část

- ▶ potomka
  - ▶ váže se k prvku, který je potomkem jiného prvku
  - ▶ skládá se ze dvou či více selektorů, které jsou odděleny >
  - ▶ p > b
  - ▶ !! termín potomek označuje pouze bezprostředního následníka !!
  - ▶ <p> ..... <b>...<em>...</em>...</b>....</p>
    - ▶ b je potomkem p, em je pouze následníkem p

# Selektory - 3. část

- ▶ sousedních sourozenců
  - ▶ zápis: E1 + E2, kde E1, E2 jsou selektory
  - ▶ pravidlo se váže k E2 jen tehdy, kdy E1 a E2 mají stejného rodiče (bezprostředního předchůdce) a E1 bezprostředně předchází E2
  - ▶ př:
    - ▶ `<p>...<b>...</b>...<em>...</em>...<code>...</code>...</p>`
      - ▶ b a em jsou sourozenci, kdežto b a code sourozenci nejsou
- ▶ př.: [kaskadovani.html](http://kaskadovani.html)

# Selektory - 4. část - selektor třídy

- ▶ prvky jazyka HTML můžeme rozlišit (přiřadit k prvkům jazyka HTML) pomocí:
  - ▶ tříd - použijeme parametr class - hodnota je libovolný text
- ▶ je-li atribut prvku class, používá se ve styl. předpisech "."
- ▶ název třídy (hodnota atributu class musí následovat bezprostředně za tečkou)
- ▶ př:
  - ▶ .trida
    - ▶ vztahuje se na všechny prvky, které mají hodnotu atributu class trida
  - ▶ div.trida
    - ▶ vztahuje se pouze na prvek/tag div, které má hodnotu atributu class nastavenou na trida

# Selektory - 5. část - selektor id

- ▶ prvky jazyka HTML můžeme rozlišit (přiřadit k prvkům jazyka HTML) pomocí:
  - ▶ identifikátorů - použijeme parametr ID - hodnota je libovolný text
- ▶ jeho zvláštností je, že musí nabývat vždy unikátních hodnot (v jednom dokumentu nemůže být víc prvků, jejichž atributy id měly stejnou hodnotu)
- ▶ ve stylových předpisech: "#"
- ▶ př:
  - ▶ #blok
    - ▶ váže se k jakémukoliv prvku, který má atribut id s hodnotou blok
  - ▶ p#blok
    - ▶ váže se k pouze k prvku p, který má atribut id s hodnotou blok
    - ▶ pokud bychom jej použili, např. na div, pravidlo nebude uplatněno

# Pseudotřídy

- ▶ Rozlišují prvky podle jiných charakteristik než je jejich jméno, atributy, obsah
- ▶ Nejsou explicitně uvedeny ve zdrojovém kódu dokumentu



# Pseudotřídy - odkazy

- ▶ `:link`
  - ▶ vztahuje se k dosud nenavštíveným odkazům
- ▶ `:visited`
  - ▶ vztahuje se k již navštíveným odkazům

# Pseudotřídy 1

- ▶ :hover
  - ▶ působí na prvek v době, kdy na něj uživatel ukazuje např. kurzorem myši
- ▶ :active
  - ▶ působí na prvek v době, kdy je aktivován (v době mezi stiskem a uvolněním tlačítka)

# Pseudotřída 2

- ▶ :first-child
  - ▶ váže k prvku, který je 1. potomkem jiného prvku
- ▶ :last-child
  - ▶ váže se k prvku, který ke posledním potomkem jiného prvku
- ▶ :nth-child()
  - ▶ lze vybrat prvek na základě opakujícího se pořadí, např. liché položky v seznamu, sudé řádky v tabulce, ..
  - ▶ do závorky se zapisuje vzorec ve tvaru  $A_n+B$ , n začíná 0 nebo přímo číslo, např. 4.
  - ▶ liché řádky: buď odd nebo  $2n+1$
  - ▶ sudé řádky: buď even nebo  $2n$
- ▶ :nth-last-child()
  - ▶ podobné jako nth-child, ale počítá se od konce

# Pseudotřída 3

- ▶ :first-line
  - ▶ aplikuje styl na 1. formátovaný řádek odstavce
- ▶ :first-letter
  - ▶ váže se k 1. písmenu obsahu prvku
  - ▶ pouze tyto vlastnosti
    - ▶ vlastnosti písma, barvy, pozadí
    - ▶ line-height, text-decoration, margin, padding, border,

# Pseudotřídy 4

- ▶ `:has()`
  - ▶ je možné vybrat prvky, které obsahují prvky odpovídající prvku v závorce
  - ▶ př.: `a:has(img)` - vybere `<a>`, které obsahují `<img>`
  - ▶ př.: `h2:has(+ p)` - vybere `<h2>`, za kterým je `<p>`
- ▶ `:is()`
  - ▶ můžeme psát složené selektory jednodušeji
  - ▶ př.: `ul li, ol li` můžeme napsat jako `:is(ul, ol) li`
  - ▶ v závorce může být i třída (class) - `.navez_class`
- ▶ `:where()`
  - ▶ prakticky podobný s `:is()`, ale `:is()` je konkrétnější, viz dále

# Pseudotřídy - př.

- ▶ obarvení odkazů:
  - ▶ musí se zachovat pořadí definice:
    - ▶ a:link
    - ▶ a:visited
    - ▶ a:hover
    - ▶ a:active

# ::before, ::after

- ▶ ::before
  - ▶ vloží „něco“ před obsah daného prvku
  - ▶ můžeme vložit i text pomocí content: - obsah se definuje do uvozovek
- ▶ ::after
  - ▶ vloží „něco“ za obsah daného prvku

# Přiřazování hodnot

- ▶ Po analýze dokumentu musí klient všem prvkům přiřadit hodnoty všech vlastností
- ▶ Výsledná hodnota je výsledkem tohoto výpočtu:
  - ▶ Vezme se určená, tj. ve styl. předpisu explicitně nastavená hodnota - tzv. určená
  - ▶ Je-li určená hodnota relativní, převede se na absolutní - vypočítaná
  - ▶ Vypočítaná hodnota je nakonec transformována na skutečnou hodnotu



# Určená hodnota

- ▶ Klient přiřazuje vlastnostem určené hodnoty násl. algoritmem:
  - 1) Pokud existuje hodnota, která je výsledkem kaskádování, vezme se tato hodnota
  - 2) Neexistuje-li hodnota, která je výsledkem kaskádování, ale daná vlastnost je dědičná, použije se hodnota rodiče
  - 3) Nelze-li uplatnit pravidla 1) a 2), použije se výchozí hodnota vlastnosti
- ▶ Kořenový prvek nemá žádné rodiče => nemůže hodnoty vlastností dědit; použije se výchozí hodnota vlastnosti

# Vypočítaná hodnota

- ▶ určené hodnoty - absolutní (black,, 2cm, ...), relativní (auto, 2em, 150%, ...)
- ▶ absolutní hodnoty není třeba přepočítávat
- ▶ relativní hodnoty musí být převedeny na hodnoty vypočítané:
  - ▶ procenta se vypočítají z referenční hodnoty
  - ▶ hodnoty v relativních jednotkách se musí vynásobit velikostí příslušného písma, nebo rozměrem pixelu
  - ▶ hodnota auto se musí vypočítat způsobem definovaným u každé vlastnosti
- ▶ ve většině případů se dědí vypočítaná hodnota

# Skutečná hodnota

- ▶ Vypočítaná hodnota již většinou může být použita
- ▶ Jsou ale případy, kdy to z různých důvodů nelze
- ▶ Např.: klient vykreslí orámování, jehož síla je celé číslo. Pokud vyjde vypočítaná hodnota desetinné číslo, musí být převedena (zaokrouhlení, oříznutí)
- ▶ Skutečná hodnota je vypočítaná hodnota, která byla upravena v důsledku omezení

# Dědičnost

- ▶ některé hodnoty vlastnosti dědí potomci prvku ve stromu dokumentu
- ▶ pro každou vlastnost je definováno, zda se dědí, či nikoli
- ▶ tento princip umožňuje, aby autor určil výchozí hodnoty pro celý dokument
  - ▶ nastaví se hodnoty pro kořenový prvek stromu dokumentu (`<html>`, `<body>`)

# Kaskádování

- ▶ stylové předpisy mohou pocházet ze tří zdrojů:
  - ▶ autor webu
  - ▶ uživatel
  - ▶ klient/prohlížeč
- ▶ styl. předpisy z těchto zdrojů se mohou překrývat
- ▶ o tom, jaké styly budou použity pro formátování dokumentu rozhodují pravidla kaskádování
- ▶ ta přiřazují každému stylovému pravidlu určitou váhu; v případech, kdy se styl. pravidla dostanou do kolize, zvítězí to s největší váhou
- ▶ implicitně - pravidla autora mají vyšší váhu než pravidla uživatele
- ▶ autorská a uživatelská pravidla mají vyšší váhu než předpis klienta

# Pravidlo !important

- ▶ klíčové slovo - píše za vlastní deklaraci, od které je odděleno mezerou
- ▶ v případě konfliktu mají deklarace s !important přednost před normálními, tj. neoznačenými deklaracemi

# Kaskádovací pořadí

► pravidla pro určení vlastnosti určitého prvku:

- 1) klient nalezne všechny deklaráce, které se vztahují k danému prvku; jedná se o deklaráce obsahující danou vlastnost a jejichž selektor se váže k danému prvku
- 2) 1. třídění - dle váhy a původu zdroje; pro normální deklaráce platí, že stylový předpis autora přepíše předpis uživatele a ten přepíše výchozí předpis klienta
- 3) 2. třídění - provede se podle konkrétnosti selektorů - konkrétnější přepíše obecnější
- 4) 3. třídění - podle pořadí, v jakém jsou pravidla uvedena; jsou-li váha, původ i konkrétnost dvou pravidel shodné, zvítězí to, které je definováno později

# Výpočet konkrétnosti selektoru

- ▶ při výpočtu - určí se 3 základní hodnoty a, b, c takto:
  - ▶ a = počet ID atributů v selektoru
  - ▶ b = počet ostatních atributů a pseudotříd v selektoru
  - ▶ c = počet názvů prvků v selektoru
- ▶ následně se získaná čísla zřetězí (zapiší se bezprostředně za sebou) v pořadí a-b-c a výsledkem je konkrétnost selektoru



# Příklad výpočtu konkrétnosti

selektor	a	b	c	konkrétnost
*	0	0	0	0
li	0	0	1	1
ul li	0	0	2	2
ul ol+li	0	0	3	3
ul ol li.red	0	1	3	13
li.red.level	0	2	1	21
#blok	1	0	0	100