

Tworba webu

Kaskádové styly (CSS)



Martin Urza



Motivace



- Proč používat kaskádové styly k formátování HTML?
 - Dovolují více možností formátování než klasické HTML atributy a stále přibývají další (možnosti).
 - Umožňují jednou definovat formátování a opakovaně jej používat (na jiném webu a/nebo jiných stránkách).
 - Pro dynamicky generované stránky lze za pomoci CSS snadno vytvořit různé vzhledy pro různé uživatele.
 - Vzhled stránek lze oddělit od obsahu, který je možné umístit do jiného souboru.
 - Pomocí kaskádových stylů je možné vyřešit kompletně celý layout stránek.
 - Frikulíni (free, cool & in) používají zásadně CSSko ;o))



Syntax samotného CSS

- CSS nabízí mnoho vlastností, kterým lze přiřazovat různé hodnoty (každá vlastnost má doménu hodnot).
- Syntakticky zápis přiřazení je *vlastnost: hodnota*.
- Jednotlivé CSS výrazy jsou odděleny středníky:
 - vlastnost: hodnota; vlastnost: hodnota
- Příklady vlastností CSS: font-variant, font-size.
- Příklady hodnot: font-variant má hodnoty normal a small-caps (kapitálky); font-size má (kromě jiného) hodnoty small, medium, large (a mnoho dalších).
- Příklad: "font-variant: small-caps; font-size: large"
 - Text bude kapitálkami a navíc velký na výšku (šířku).

Přímý zápis CSS



- Kaskádové styly lze zapsat do atributu style k HTML tagu.
- **Je důležité rozlišovat atributy HTML tagů od CSS vlastností!!!!**
 - Typicky lze požadovaného výsledku dosáhnout oběma cestami.
 - Atributy HTML tagů se zapisují: **atribut="hodnota"**.
 - CSS výrazy se mají jiný zápis: **atribut: hodnota**.
 - **Celý zápis CSS výrazu patří do hodnoty atributu style nějakého HTML elementu!!!!**
- `<p style="font-size: large; font-variant: small-caps">`
 - Text v odstavci bude kapitálkami a navíc velký.



Nepřímý zápis CSS

- Jsou dvě možnosti nepřímého zápisu CSS výrazů:
 - Do hlavičky HTML dokumentu do tagu style.
 - `<style> CSS výrazy </style>`
 - Do souboru, na který se odkazuje tag v hlavičce:
 - `<link rel="stylesheet" type="text/css" href="styl.css" />`
 - V atributu href je URL souboru, tentokrát jen "styl.css".
- Text ve všech odstavcích bude velkými kapitálkami:
 - `<style>`
 `p {font-variant: small-caps; font-size: large}`
`</style>`
 - `<link rel="stylesheet" type="text/css" href="a.css" />`
Obsah a.css: `"p {font-variant: small-caps; font-size: large}"`

Příklady nepřímého zápisu CSS



- Řekněme, že chceme nastavit vzhled nadpisů celého HTML dokumentu tak, aby nadpisy stejných úrovní vypadaly stejně.

- `<style>`

- `h1 {font-variant: small-caps; font-size: xx-large;}`

- `h2 {font-variant: small-caps; font-size: x-large;}`

- `h3 {font-variant: normal; font-size: x-large;}`

- `h4 {font-variant: normal; font-size: large;}`

- `h5 {font-variant: normal; font-size: medium;}`

- `h6 {font-variant: normal; font-size: small;}`

- `</style>`

- Totéž lze napsat do souboru (místo do elementu style).
- Všimněte si, jak snadno lze vzhled nadpisů změnit bez jediného zásahu do HTML dokumentu!

Další možnosti deklarace v nepřímém zápisu



• Hromadná deklarace.

- Chceme-li deklarovat nějaké vlastnosti pro více tagů, napíšeme je na jednu řádku a oddělíme čárkou:

- `<style> h1, h2, h3, h4 {font-size: large} </style>`

- Nadpisy h1, h2, h3 a h4 budou všechny velkým písmem, což lze zapsat i na čtyři řádky pro každý nadpis zvlášť.

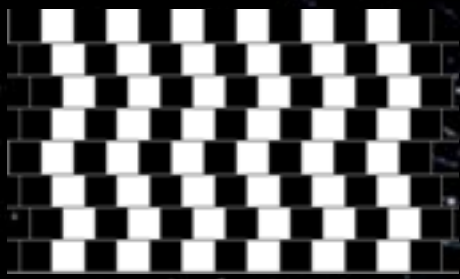
• Kontextová deklarace.

- Pro deklaraci vlastností tagu uvnitř jiného tagu jejich názvy uvedeme za sebou a oddělíme mezerou:

- `<style> h2 a {font-variant: small-caps} </style>`

- Odkaz v nadpisu druhé úrovně bude kapitálkami, avšak ostatních odkazů se to nijak nedotkne.

- Zanoření může být libovolně hluboké (více tagů v sobě).



Pseudoelementy

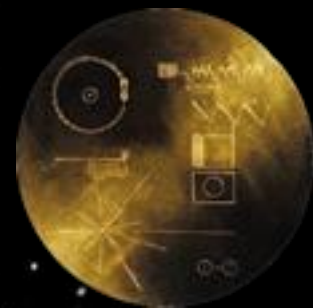
- Pseudoelement označuje nějakou speciální „oblast“ elementu, pro kterou lze nastavit nějaké vlastnosti kaskádových stylů; ta se může týkat například jen prvního řádku či prvního písmena.
 - `<style>`
 - `p:first-line {font-variant: small-caps}`
 - `p:first-letter {font-size: large}`
 - `</style>`
 - První řádka všech odstavců bude kapitálkami a každé první písmeno v každém odstavci bude velké.
- Některé elementy mají speciální pseudoelementy.
 - Tag `a` má tyto pseudoelementy: `link` (normální odkaz), `visited` (navštívený odkaz) a `hover` (odkaz pod myší).



Třídy a identifikátory

- Doposud jsme nepřímými zápisy nastavovali pouze vlastnosti všem elementům nějakého jména.
 - Někdy můžeme chtít nastavit vlastnosti pouze nějaké skupině elementů (či jen jednomu konkrétnímu tagu).
- Každý HTML element může mít atributy class a id, jehož hodnotu si volíme (vymýšlíme) sami.
- Stejná hodnota id nemůže být přiřazena více různým elementům, hodnota class ano (nesmí být více tagů se stejným identifikátorem, ale stejné třídy ano).
- V CSS výrazu značíme třídu tečkou (.) následovanou jménem dotyčné třídy, identifikátor pak křížkem (#), který je také následován jménem třídy.

Příklad použití tříd a identifikátorů

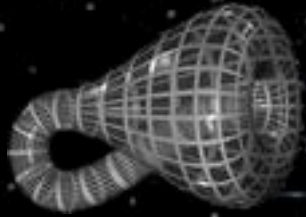


- `<style>`
 - `.dulezite_sdeleni {font-size: large}`
 - `#nejdulezitejsi_sdeleni {font-variant: small-caps}``</style>`
- Pak každý element (může jich být více) s atributem `class` s hodnotou `dulezite_sdeleni` bude svůj textový obsah zobrazovat velkým písmem.
 - Příklad: `<p class="dulezite_sdeleni">tučný odstavec</p>`
- Jeden jediný element (nutně unikátní) s atributem `id` s hodnotou `nejdulezitejsi_sdeleni` bude svůj textový obsah zobrazovat kapitálkami.
 - Příklad: `<p id="nejdulezitejsi_sdeleni">kapitálky</p>`

Cvičení



- Napište stránku (nebo použijte nějakou existující) a její vzhled upravte podle následujících parametrů:
 - Alespoň jednou užití přímý zápis CSS (atribut style).
 - Pomocí nepřímých zápisů v tagu style v hlavičce HTML stránky definujte vzhled všech úrovní nadpisů.
 - Ve zvláštním souboru (pomocí nepřímých zápisů CSS) definujte vzhled alespoň dvou tříd a identifikátorů.
 - Alespoň jednu třídu použijte pro více elementů.
 - Použijte kontextovou a hromadnou deklaraci a ověřte, zda fungují (použijte element i mimo kontext).
 - Alespoň jednou použijte pseudoelement.
 - Nemusíte formátovat více než font-size a font-variant.



Elementy div a span

- Tagy div a span se nazývají neutrální, protože samy o sobě nemají žádný význam, nic nedělají, respektive div na svém konci odřádkuje, span nedělá opravdu vůbec nic.
- Primární použití tagu div je, že do něj lze obalit více jiných elementů (například více odstavců nemůžete zabalit do odstavce, ale do divu ano).
- Tag span se používá typicky pro obalení nějakého kusu textu (neodřádkuje za sebou).
- Elementy div a span lze dobře využívat v kombinaci s kaskádovými styly (nastavením atributů class, id, případně style).

Nadužívání tagů div a span



- Tagy div a span lze v kombinaci s kaskádovými styly pohodlně používat místo většiny ostatních elementů jazyka HTML, což nemusí být vždy vhodné.
- Elementy jazyka HTML lze v zásadě rozdělit na dvě skupiny:
 - Tagy, které udávají vzhled svého obsahu.
 - Příklady: font, b, small, strike,
 - Tagy, které určují obsah (smysl, sémantiku) obsahu.
 - Příklady: p, cite, code, q,
- První skupinu je velmi žádoucí formátovat styly.
- Při nahrazení tagů druhé skupiny pak z kódu HTML stránky není patrný obsah, což je nepraktické.



Vlastnosti kaskádových stylů

- Doposud jsme používali jen dvě CSS vlastnosti, což stačilo pro objasnění způsobu práce s kaskádovými styly.
- Vlastností kaskádových stylů je pochopitelně hodně, každá má i mnoho různých možných hodnot, které mohou být vlastnosti přiřazeny.
 - Pro každou vlastnost existuje jedna defaultní hodnota, která je vlastnosti implicitně přiřazena v případě, že vlastnost ve stylu vůbec nebyla uvedena.
- Na následujících slajdech budou uvedeny další CSS vlastnosti a jejich možné hodnoty, mezi nimiž bude vždy podtržena hodnota defaultní.

Vlastnosti fontů



- font-family (*seznam fontů oddělených čárkami*): font textu; nemá-li klient první, je zvolen další a tak dále
- font-style (normal/italic/oblique): normalní, kurzíva, skloněné
- font-variant (normal/small-caps): normální, kapitálky
- font-size (*výška v pixelech/ procento/xx-small/x-small/small/medium/large/x-large/xx-large*): velikost písma
- font-weight (lighter/normal/bold/bolder/100/200/300/400/500/600/700/800/900): tloušťka písma (lze vyjádřit i číslem)
- line-height (normal/*výška v pixelech/násobek/procento*): výška řádky; násobky a procenta vzhledem k velikosti písma
- font (*font-style font-variant font-weight font-size/line-height font-family*): shrnuje vlastnosti fontu do jedné, je však nutné uvést je v daném pořadí, př.: italic normal bold large/1.5 Arial
- color (*barva jako v HTML*): barva písma, udává se jako v HTML



Vlastnosti textu

- text-decoration (none/underline/overline/line-through/blink): nic, podtržení, nadtržení, přeškrtnutí, blikání (pozor na různé „blogýsky“, používat s rozmyslem)
- text-transform (none/capitalize/uppercase/lowercase): nic, první písmeno každého slova velké, celý text velkými písmeny, celý text malými písmeny
- word-spacing (normal/ *délka v pixelech*): mezera mezi slovy; v případě zadání délky nebude standardní mezera nahrazena, nýbrž zvětšena o délku (lze zadat záporné číslo)
- letter-spacing (normal/ *délka v pixelech*): mezera mezi znaky; funguje stejně jako word-spacing
- white-space (normal/pre/nowrap): nic, text bude na stránce zobrazen jako ve zdroji (například mezery, odřádkování, ...), text nebude zalamován (automatické konce řádek)

Vlastnosti pozadí

C



- background-color (transparent / *barva jako v HTML*): průhlednost, barva pozadí (jakéhokoliv objektu – nejen body)
- background-image (none / *url('URL obrázku')*): obrázek na pozadí tagu (opět jakéhokoliv objektu), př.: *url('./img/x.png')*
- background-repeat (repeat / *no-repeat* / *repeat-x* / *repeat-y*): pozadí se opakuje, neopakuje, opakuje doleva, opakuje dolů
- background-attachment (scroll / *fixed*): má smysl jen u pozadí stránky, *fixed* pozadí zůstává na místě a vypadá divně
- background-position (left / *center* / *right* / *délka v px* / *procenta top* / *center* / *bottom* / *délka v px* / *procenta*): dvě hodnoty pro vertikální a horizontální pozici (oddělené mezerou)
- background (*background-image background-repeat background-attachment background-color background-position*): všechny vlastnosti pozadí oddělené mezerami (jako u fontu)

Vlastnosti rámečků a okrajů



- border-width, border-top-width, border-bottom-width, border-left-width, border-right-width (thin / medium / thick / *délka*): slabá, normální, tlustá, nastavená délka rámečku (případně jen jeho horní, spodní, levé, pravé části)
- border-color (*barva jako v HTML*): barva rámečku
- border-style (none / dotted / dashed / solid / double / groove / ridge / inset / outset): různé podoby vzhledů rámečků
- border, border-top, border-bottom, border-left, border-right (*border-width / border-style / border-color*): všechny vlastnosti rámečků v jedné (podobně jako u fontu)
- margin, margin-top, margin-bottom, margin-left, margin-right (auto / *délka* / *procento*): šířka **vnějších** okrajů
- padding, padding-top, padding-bottom, padding-left, padding-right (auto / *délka* / *procento*): šířka **vnitřních** okrajů

Vlastnosti seznamů



- `list-style-type` (disc/decimal/lower-alpha/upper-alpha/none/circle/square/lower-roman): vzhled odrážek v seznamu (puntík, číslo, písmeno, velké písmeno, nic, kroužek, čtverec, malá římská číslice)
- `list-style-image` (none/url('URL')): obrázek místo odrážky v seznamu (přebíjí `list-style-type`)
- `list-style-position` (inside/outside): odrážky jsou zarovnané s textem, nebo jsou posunuté doprava
- `list-style` (*list-style-type list-style-position list-style-image*): souhrn všech vlastností listů (jako u fontu)
- **Použití všech těchto vlastností se předpokládá u elementů seznamů, tedy například `li` a podobně.**
- Seznamy jsou typickým případem, ve kterém není moc vhodné použití přímého zápisu stylů.

Vlastnosti velikostí



- **width** (auto / *šířka v pixelech / procento*): šířka elementu, auto znamená přizpůsobení obsahu
- **height** (auto / *šířka v pixelech / procento*): výška elementu, auto znamená opět přizpůsobení obsahu; tato vlastnost lze nastavit pouze blokovým tagům (což je třeba div)
- **max-width, min-width, max-height, min-height** (*šířka v px / procento*): minimální a maximální šířka a výška; minimální funguje bez problémů (objekt je buď zvětšen na minimální hodnotu, nebo se přizpůsobí obsahu, je-li tento větší), použití maximální hodnoty (jedné) může způsobit protažení objektu do druhého rozměru, použití obou řeší overflow
- **overflow** (visible / hidden / scroll / auto): nevejde-li se element do stanovených rozměrů, lze přesahující část buď nechat přetéct, schovat, vždy nechat scrollovat, nebo jen když je třeba (auto)

Další vlastnosti CSS

- Výše zmíněné CSS vlastnosti tvoří jen malou část z celé množiny všech vlastností, které existují.
- **Protože jsou vlastností (malé) stovky a přibývají nové (případně nové hodnoty starých vlastností), stačí mít po ruce nějaké základní a na ostatní se ptát pana Googla.**
- Výše uvedené vlastnosti pokrývají skoro všechny ze základních potřebných oblastí krom vlastností, které jsou potřebné pro pozicování stránky pomocí CSS.
 - Pozicování pomocí kaskádových stylů (CSS layout) bude objasněno na příští přednášce poté, co si osvojíte základy zápisů CSS.

Cvičení



- Vezměte nějakou hodně rozsáhlou HTML stránku se smysluplným obsahem (třeba z Vašeho minulého webu těch osmi stránek, případně z webu, nebo ji vytvořte) a celou ji (s výjimkou pozicování) formátujte pomocí CSS vlastností tak, aby vypadala stejně jako předloha.
- Dbejte na to, aby přímé a nepřímé CSS zápisy byly použity rozumně.
 - Typické nerozumné použití spočívá v přímém zápisu místo nepřímého, opačné chyby typicky nenastávají, i ač samozřejmě vyloučeny nejsou.
- Snažte se o přehledný kód (díky CSS to jde snadno).

Rekapitulace



- **Vnímejte rozdíl mezi klasickým formátováním za pomoci atributů tagů a mezi CSS vlastnostmi, jež samy o sobě nejsou atributy HTML elementů!!!**
- Vlastnosti kaskádových stylů lze zapisovat přímým i nepřímým zápisem (nepřímý navíc lze provést jak do hlavičky dokumentu, tak do zvláštního souboru).
 - Typicky je vhodné většinu vlastností uvádět nepřímým zápisem a ideálně takovým, který je oddělen v souboru .css mimo samotný HTML dokument.
- Kaskádové styly jsou velmi mocný nástroj, za pomoci kterého lze naformátovat vše, čeho lze dosáhnout s atributy HTML tagů a ještě mnohem více.