

CSS

kaskádové styly

praktický manuál

Miroslav Cyroň

- ❑ Základní popis využití CSS
- ❑ Detailní popis vlastností s přihlédnutím k praxi
- ❑ Praktické návody, postupy, tipy a triky
- ❑ Řešení problémů pomocí JavaScriptu
- ❑ Kompatibilita CSS v nejčastěji používaných prohlížečích
- ❑ Doprovodné CD s elektronickou verzí knihy a příklady



Upozornění pro čtenáře a uživatele této knihy

Všechna práva vyhrazena. Žádná část této tištěné či elektronické knihy nesmí být reprodukována a šířena v papírové, elektronické či jiné podobě bez předchozího písemného souhlasu nakladatele. Neoprávněné užití této knihy bude **trestně stíháno**.

Používání elektronické verze knihy je umožněno jen osobě, která ji legálně nabyla a jen pro její osobní a vnitřní potřeby v rozsahu stanoveném autorským zákonem. Elektronická kniha je datový soubor, který lze užívat pouze v takové formě, v jaké jej lze stáhnout s portálu. Jakékoliv neoprávněné užití elektronické knihy nebo její části, spočívající např. v kopírování, úpravách, prodeji, pronajímání, půjčování, sdělování veřejnosti nebo jakémkoliv druhu obchodování nebo neobchodního šíření je zakázáno! Zejména je zakázána jakákoliv konverze datového souboru nebo extrakce části nebo celého textu, umístování textu na servery, ze kterých je možno tento soubor dále stahovat, přitom není rozhodující, kdo takovéto sdílení umožnil. Je zakázáno sdělování údajů o uživatelském účtu jiným osobám, zasahování do technických prostředků, které chrání elektronickou knihu, případně omezují rozsah jejího užití. Uživatel také není oprávněn jakkoliv testovat, zkoušet či obcházet technické zabezpečení elektronické knihy.





Copyright © Grada Publishing, a.s.

Obsah

Úvod	15
1. Základy CSS	25
1.1 Základem je dobrý dokument	26
1.2 Vytvoření stylopisu	26
Základní syntaxe.....	26
Komentáře	27
1.3 Připojení stylopisu k dokumentu XHTML	27
Zápis přímo do dokumentu	27
Načtení externího stylopisu	27
1.4 Hodnoty	28
Klíčová slova.....	28
<číslo> , <celé_číslo>	29
<velikost> , <délka>	29
Absolutní jednotky	29
Relativní jednotky	29
<procenta>.....	29
<barva>.....	30
<URI>	30
<řetězec>.....	31
<čas>.....	31
1.5 Selektory	31
Selektory typu	31
Univerzální selektor	31
Selektory třídy.....	31
Násobné třídy	31
Selektory ID	32
Selektory s atributy	32
Existence atributu	32
Atribut hodnoty.....	32
Atribut obsahující hodnotu	32
Atribut obsahující podhodnotu	32
Vícenásobné selektory s atributy.....	33
Selektory následníků	33

Selektory potomků	33
Selektory sourozenců	33
1.6 Pseudotřídy a pseudoprvky	34
Pseudotřídy :link a :visited	34
Dynamické pseudotřídy :hover, :active a :focus	34
Pseudotřída :first-child	35
Jazyková pseudotřída :lang	35
Pseudoprvek :first-line	35
Pseudoprvek :first-letter	36
Pseudoprvky :before a :after	37
1.7 Kombinování a seskupování selektorů, pseudotříd a pseudoprvků ...	38
Kombinované selektory	38
Seskupování selektorů	38
1.8 Média.....	38
Typy médií	38
Určení médií v dokumentu	39
Určení médií přímo ve stylopisu	39
1.9 Dědičnost.....	39
Klíčové slovo inherit	39
1.10 Vypočítané hodnoty a kaskáda	40
Vypočítané hodnoty	40
Kaskáda	40
Konkrétnost selektorů	40
Pravidlo !important	41
1.11 Podpora vlastností v nejběžnějších prohlížečích	41
2. CSS a JavaScript	45
2.1 Připojení souboru JavaScriptu k dokumentu XHTML.....	45
2.2 Identifikace prohlížečů JavaScriptem (příklady)	46
2.3 Podstrčení vhodného stylu na základě identifikace prohlížeče.....	47
3. Rozměry prvků a přetékaní obsahu	49
3.1 Výpočet rozměrů prvků	50
Výpočet rozměrů prvků podle Specifikace W3C (model „pouze obsah“)	50
Výpočet rozměrů prvků podle Microsoftu (model „včetně rámečku a výplně“)	50

Hodnota auto	51
Blokové (nenahrazované) prvky	51
Řádkové (nenahrazované) prvky	51
Nahrazované (blokové i řádkové) prvky	52
Absolutně pozicované (nenahrazované) prvky	52
Absolutně pozicované (nahrazované) prvky	52
3.2 Šířka obsahu prvku – vlastnost width	52
Problém chybného počítání rozměrů v Internet Exploreru 5–6	53
Řešení	56
Starší způsoby řešení	56
Řešení pomocí CSS	60
3.3 Minimální šířka prvku – vlastnost min-width	61
Příklad	62
3.4 Maximální šířka prvku – vlastnost max-width	66
3.5 Šířka prvku „od hrany rámečku po hranu rámečku“ – vlastnost box-width	67
3.6 Výška obsahu prvku – vlastnost height	68
Problém chybného počítání rozměrů v Internet Exploreru 5–6	68
Možnosti řešení	70
Starší způsoby řešení	70
Řešení pomocí CSS	73
Procentní výška prvku umístěného v <body>	74
3.7 Minimální výška prvku – vlastnost min-height	74
3.8 Maximální výška prvku – vlastnost max-height	78
3.9 Výška prvku „od hrany rámečku po hranu rámečku“ – vlastnost box-height	78
3.10 Výpočetní model rozměrů prvků – vlastnost box-sizing	79
3.11 Přetékání obsahu – vlastnosti overflow, overflow-x a overflow-y	80
Přetékání v praxi	81
3.12 Viditelná oblast – vlastnost clip	84
3.13 Viditelná oblast v CSS 3 – vlastnost overflow-clip	84
3.14 Viditelná oblast nahrazovaných prvků – vlastnost crop	85
3.15 Podpora vlastností v nejběžnějších prohlížečích	86

4. Barvy a pozadí	89
4.1 Barva textu – vlastnost color	89
4.2 Styl pozadí – vlastnost background	90
Barva pozadí – vlastnost background-color	90
Obrázek na pozadí – vlastnost background-image	90
Opakování obrázku na pozadí – vlastnost background-repeat	91
Umístění obrázku na pozadí – vlastnost background-position	91
Ukotvení obrázku na pozadí – vlastnost background-attachment	92
Sdružená vlastnost – background	92
4.3 Podpora vlastností v nejběžnějších prohlížečích	93
5. Okraje, výplně a rámečky	95
5.1 Základní koncepce rámců	95
Základní podpora v různých prohlížečích	96
5.2 Okraje – vlastnost margin	101
Slučování okrajů	102
Hodnota auto – snadné centrování blokových prvků	104
Stránka na střed v Mozille	106
5.3 Výplně – vlastnost padding	106
5.4 Rámečky – vlastnost border	107
Šířka rámečku – vlastnost border-width	107
Barva rámečku – vlastnost border-color	108
Styl rámečku – vlastnost border-style	108
Sdružená vlastnost – border	109
5.5 Zaoblení rámečku – vlastnost border-radius	111
Složitější příklad	111
5.6 Obrazy v rámečku – vlastnosti border-image, border-corner-image, border-fit, border-image-transform, border-corner-image-transform	115
Obrázek v rámečku – vlastnost border-image	115
Obrázek v rohu rámečku – vlastnost border-corner-image	116
Opakování obrázků v rámečku – vlastnost border-fit	116
Natočení obrázku v rámečku – vlastnosti border-image-transform, border-corner-image-transform	118
Sdružená vlastnost border dle CSS 3	119

5.7	Obrys prvku – vlastnost outline.....	119
5.8	Stín prvku – vlastnost box-shadow	120
5.9	Podpora vlastností v nejběžnějších prohlížečích	121
6.	Vlastnosti písma	125
6.1	Typ písma – font-family.....	125
	Písma pro starší verze Linuxu	127
	Písma pro Windows 3.x	127
	Písma pro Mac OS	127
	Písma pro moderní Linux.....	127
	Bezpatková písma	127
	Patková písma	128
	Psaná písma	129
	Písma pro Windows 95/98/ME/NT/2000/XP	129
	Bezpatková písma	129
	Ostatní písma	130
	Příklady definice písem.....	130
	Bezpatkové písmo – malé	130
	Bezpatkové písmo – střední	131
	Bezpatkové písmo – velké.....	131
	Patkové písmo – malé	131
	Patkové písmo – větší	131
	Proložené písmo – menší	131
	Proložené písmo – větší	131
	Dekorativní písmo.....	131
	„Psané“ písmo	131
	Dynamické vkládání fontů.....	131
6.2	Velikost písma – font-size	132
6.3	Styl písma – font-style.....	133
6.4	Síla písma – font-weight	133
6.5	Kapitálky – font-variant.....	134
6.6	Výška řádek – line-height.....	134
6.7	Sdružená vlastnost – font.....	135
6.8	Hustota písma – font-stretch.....	135
6.9	Výška nízkých liter – font-size-adjust	136
6.10	Podpora vlastností v nejběžnějších prohlížečích	137

7. Vlastnosti textu.....	139
7.1 Odsazení první řádky – vlastnost text-indent	139
7.2 Rozestup znaků – vlastnost letter-spacing.....	140
7.3 Rozestup slov – vlastnost word-spacing.....	140
7.4 Vodorovné zarovnání – vlastnost text-align	140
7.5 Svislé zarovnání – vlastnost vertical-align.....	141
7.6 Velikost písmen – vlastnost text-transform.....	142
7.7 Dekorace textu – vlastnost text-decoration.....	142
7.8 Stín kolem textu – vlastnost text-shadow.....	143
7.9 Rolování textu – vlastnosti marquee.....	144
Styl rolování textu – vlastnost marquee-style	144
Směr rolování textu – vlastnost marquee-direction	145
Rychlost rolování textu – vlastnost marquee-speed	145
Opakování rolování textu – vlastnost marquee-repetition	145
Sdružená vlastnost – marquee	146
7.10 Efekty písma – vlastnost font-effect	146
7.11 Podpora vlastností v nejběžnějších prohlížečích	147
8. Typ prvku a viditelnost.....	149
8.1 Typ prvku – vlastnost display	149
block, inline, none	150
inline-block.....	150
run-in.....	152
compact	158
table, table-cell a inline-table	158
Svislé centrování obsahu pomocí vlastnosti display.....	166
8.2 Viditelnost prvku – vlastnost visibility	167
8.3 Podpora vlastností v nejběžnějších prohlížečích	168
9. Pozice prvku.....	169
9.1 Pozice prvku – vlastnosti position, top, right, bottom, right, z-index	169
Pozice prvku – position	169
Souřadnice prvku – top, right, bottom, left	170

Pozice prvku ve třetím rozměru – z-index	170
Praktické příklady pozicování	171
Kompletní rozvržení stránky pomocí position: absolute – pevný tříslopcový layout	171
Kompletní rozvržení stránky pomocí position: absolute – pružný tříslopcový layout	177
Kompletní rozvržení stránky pomocí position: absolute – částečně pružný tříslopcový layout	178
Kompletní rozvržení stránky pomocí position: absolute – částečně pružný tříslopcový layout – rozměry v em.	185
Kompletní rozvržení stránky pomocí position: fixed – pružný tříslopcový layout s fixně pozicovaným ovládacím menu – rozměry v em.	192
Relativní pozicování řádkových prvků	201
Plovoucí „vykřičník“	205
Pružné poznámky ve zdrojovém kódu	207
Svislé centrování obsahu – verze pro Internet Explorer	210
Svislé centrování obsahu – kompletní verze (pro všechny prohlížeče)	210
9.2 Plovoucí prvky – vlastnost float	213
Praktické příklady použití vlastnosti float	213
Plovoucí nadpisy	213
Dva sloupce s obrázkem uprostřed	217
Tři sloupce s obrázkem uprostřed	219
9.3 Zrušení obtékání – vlastnost clear	222
9.4 Podpora vlastností v nejběžnějších prohlížečích	225
10. Tabulky	227
10.1 Tabulky a CSS	227
10.2 Tabulkový formátovací model CSS	228
10.3 Tabulkové prvky	229
Přehled předdefinovaných hodnot display v XHTML	229
10.4 Šířka tabulky	229
10.5 Formátovací modely – vlastnost table-layout	229
Automatický model	229
Fixní model	230
10.6 Výška tabulky	230
10.7 Vodorovné zarovnání obsahu buněk	231
Teoretický výsledek:	231

10.8	Svislé zarovnání obsahu buněk – vlastnost vertical-align.....	231
10.9	Formátování sloupců.....	237
10.10	Dynamické efekty – skrytí či zobrazení řádků a sloupců – vlastnost visibility.....	240
10.11	Slučování rámců buněk – vlastnost border-collapse.....	245
10.12	Rozestup mezi rámy buněk – vlastnost border-spacing.....	245
10.13	Zobrazení prázdných buněk – vlastnost empty-cells.....	246
10.14	Pozice titulku tabulky – vlastnost caption-side	246
10.15	Praktické příklady formátování tabulek v CSS.....	247
	Příklad 1 – jednoduchá tabulka	247
	Příklad 2 – trochu složitější tabulka	249
	Celý kód opravné funkce	253
	Celé řešení	254
10.16	Automatické nastavení barev v buňkách podle obsahu	256
	Základní řešení	256
10.17	Podpora CSS v nejběžnějších prohlížečích	265
11.	Generovaný obsah	267
11.1	Generování obsahu – vlastnost content	267
	<řetězec>	268
	attr(X)	270
	<uri>	271
	Čítače a uvozovky	272
11.2	Čítače – vlastnosti counter-increment a counter-reset.....	272
11.3	Uvozovky – vlastnost quotes	275
11.4	Podpora vlastností v nejběžnějších prohlížečích	277
12.	Vlastnosti seznamů.....	279
12.1	Typ odrážky seznamu – vlastnost list-style-type	279
12.2	Obrázek jako odrážka seznamu – vlastnost list-style-image.....	283
12.3	Umístění odrážky seznamu – vlastnost list-style-position.....	283
12.4	Sdružená vlastnost list-style.....	284
12.5	Podpora vlastností v nejběžnějších prohlížečích	285

13. Uživatelské efekty.....	287
13.1 Tvar kurzoru – vlastnost cursor	287
13.2 Průhlednost prvku – vlastnost opacity	288
13.3 Podpora vlastností v nejběžnějších prohlížečích	289
14. Vlastnosti tisku	291
14.1 Zalamování textu – vlastnosti page-break-before, page-break-after, page-break-inside	291
14.2 Minimální počet řádků na stránce – vlastnosti orphans a widows	292
14.3 Tisknutelná oblast – pravidlo @page.....	293
14.4 Velikost a orientace tisknutelné oblasti – vlastnost size	293
14.5 Pojmenované stránky – vlastnost page	294
14.6 Adresování první, levé a pravé strany – pseudotřídy :first, :left a :right.....	294
14.7 Další tipy k formátování pro tisk.....	295
Zrušte tisk zbytečnosti	295
Barvy a pozadí.....	295
Tisk odkazů.....	295
14.8 Podpora vlastností v nejběžnějších prohlížečích	296
15. Odstrizení starých prohlížečů od stylů a skriptů, individualizace stylopisů.....	297
15.1 Individualizace stylů.....	297
Připojení externích stylů	298
Link a import	298
Podmíněné komentáře	298
Příklad odřiznutí starších prohlížečů od stylu	299
Triky na úrovni pravidel stylů	299
Triky se selektory	299
Další CSS triky.....	299
15.2 Individualizace skriptů.....	300
Identifikace prohlížeče JavaScriptem – příklady „userAgent“	300
Opera 7	301
Opera 7.0-7.5.....	301

Internet Explorer 5.5 a 6	301
Konqueror 3.0-3.5	301
Staré prohlížeče, Mozilla 0, Opera 6, IE 5.0	302
16. Další tipy a řešení	303
16.1 Režimy prohlížečů	303
16.2 Zapsání stylu JavaScriptem	306
16.3 Registrace ovladačů událostí	306
16.4 Jak na onload	311
16.5 Rozbalovací nabídky	314
Svislá rozbalovací nabídka (změna výšky)	314
Svislá rozbalovací nabídka (pomocí display)	319
Víceúrovňová rozbalovací nabídka	325
Několik poznámek	335
16.6 Nezákladnější styly píšete přímo do dokumentu	336
Rejstřík	337

Úvod

CSS (Tabulky kaskádových stylů, Cascading Style Sheets) jsou skutečným přelomem ve způsobu formátování webových prezentací. Protože CSS neovlivňuje obsah dokumentů samotných, umožňuje vytvářet čisté, přísně strukturované dokumenty, vyhovující logice i standardům, umožňující bezproblémovou práci s obsahem vašich stránek i jiným aplikacím, než jakými jsou běžné prohlížeče (včetně čtecích zařízení pro nevidomé, mobilních zařízení, XML aplikací, fulltextových vyhledávačů na webu, ale třeba i starých textových prohlížečů).

Jak je tato kniha napsána

Tato kniha je psána trochu jinak, než je u většiny podobných knih o CSS zvykem. Najdete tu jen velmi málo teorie, kniha je koncipována spíše jako praktický manuál pro tvůrce webových stránek. Obsahuje základní popis využití CSS, dále detailní popis vlastností s přihlédnutím k praxi, řešení problémů pomocí JavaScriptu, využití spojení CSS a JavaScriptu, kompatibilitu CSS v nejčastěji používaných prohlížečích a další témata.

Další rozdíl je v tom, že zde nezatajují žádné problémy, které vás při vytváření dokumentů za pomoci CSS mohou potkat. Většina publikací o CSS se spokojí s tím, že popíší teorii, popíší stručně jednotlivé vlastnosti a připojí test podpory v jednotlivých prohlížečích v základních konstrukcích, avšak chování prohlížečů ve složitějších situacích příliš nerozebírají, aby čtenáře neodradili od CSS zcela. V této knize je tomu jinak – pokud jsem zjistil v určitých situacích nějaké nekorrektní chování prohlížečů, neváhám na problém upozornit a nabídnout případně řešení. Rovněž se snažím nepreferovat žádný prohlížeč, nejsem ani stoupenec tzv. „IE ONLY“, ani bezvýhradný obdivovatel Mozilly, jsem zastáncem hesla „padni komu padni“, jen když to bude užitečné pro čtenáře.

Co byste měli znát

Studium této knihy předpokládá, že máte alespoň základní znalosti jazyka HTML či XHTML. Není nutná žádná znalost CSS (...to vás má tato kniha přece naučit...). Vítána je znalost jazyka JavaScript – není však vůbec nutná –, při jeho neznalosti se pouze připravíte o spoustu zajímavých možností.

Struktura knihy

Jak již bylo uvedeno výše, tato kniha má posloužit hlavně praktickým potřebám webových tvůrců, proto zde praxe vysoce převažuje nad teorií. Zájemce o studium teorie CSS odkazují na jinou literaturu či ještě lépe přímo na stránky W3C (www.w3.org), kde naleznete přesné znění všech specifikací (CSS 1, CSS 2, CSS 2,1, jakož i stále ještě rozpracované CSS 3).

Největší část obsahu knihy (kapitoly 3 až 14) proto tvoří popis jednotlivých vlastností, návody na jejich praktické použití (a využití), řešení případných problémů s nekompatibilitou v prohlížečích a zajímavé tipy. Na konci každé kapitoly je pak ještě tabulka s detailním popisem podpory vlastností popsaných v této kapitole, a to ve všech běžně používaných prohlížečích (novější verze prohlížečů Mozilla, Opera, Internet Explorer a Konqueror). Popisované vlastnosti jsou do jednotlivých kapitol zařazovány podle toho, v jakých situacích jsou nejčastěji využívány (např. vlastnost font-size spolehlivě najdete v kapitole Vlastnosti písma).

Historie formátování (X)HTML

Ve svých počátcích v roce 1990 byla jediným cílem jazyka HTML prezentace obsahu (především textu) v čisté, strukturované formě, pouze s několika značkami. Mělo být pouze zřejmé, co je nadpis, co odstavec apod. Již v těchto dobách se předpokládalo, že způsob zobrazení se bude řídit stylovými předpisy. První webový prohlížeč na světě – NeXT – byl navržen tak, že výsledné formátování dokumentu bylo řízeno jednoduchým stylovým předpisem. Další prohlížeč, Mozaic z roku 1993, však tuto koncepci zcela opustil a možnosti práce se styly dokumentu prakticky znemožnil.

Později si začalo mnoho autorů stěžovat, že nemohou nikterak ovlivnit vzhled svých HTML stránek. Proto byl již v roce 1994 vypracován první návrh „Cascading Style Sheets“. Bohužel se však v téže době objevil webový prohlížeč Netscape 1, což s odstupem doby byla pro web katastrofa. Firma Netscape totiž místo podpory stylů zabudovala vlastnosti pro řízení vzhledu přímo do značek HTML. Toto řešení započalo éru nekompatibility webových stránek i prohlížečů, a hlavně zanesení původně čistého a strukturovaného kódu HTML dokumentu různým formátovacím balastem.

Postupně přišli na trh webových prohlížečů další výrobci, kteří v tomto trendu pokračovali, a HTML kód se tak zanášel dalšími značkami a jejich atributy, které nesloužili ničemu jinému,

nežli k řízení vzhledu dokumentů. V jazyce HTML byly k dispozici i tabulky, a to výhradně k prezentaci tabulkových dat. Jenže autoři brzy přišli na to, že tabulky se dají využít k rozvržení stránek a nalili do buněk sloupce textu a dalšího obsahu. Autoři byli později „obohaceni“ o tzv. rámce, pomocí kterých bylo možno vzhled prezentace na obrazovce (ale pouze na ní) zobrazit do té doby nevídaným způsobem. Tak bylo například možno zajistit, aby v jednom místě obrazovky byly ovládací prvky, které se nerolovaly s ostatní prezentací. Rámce však přinášely spoustu problémů (ukládání, záložky, tisk, čtečky pro nevidomé atd.).

Později začalo být i výrobcům prohlížečů jasné, že tudy cesta nevede. Prvním novodobým prohlížečem, který začal styly částečně podporovat, byl Internet Explorer 3. Jelikož v té době CSS však ještě nebyly nikým standardizovány, implementace CSS se v tomto prohlížeči s pozdějšími standardy příliš neshoduje. První verze specifikace standardizačního konsorcia W3C (CSS 1) byla uvedena v roce 1996, těsně po příchodu Internet Exploreru 3. Za otce dnešních CSS je považován Håkon Wium Lie, vývojář (dnes šéfvývojář) Opery. Další prohlížeče firmy Microsoft – verze 4 a 5.0 už měly zabudovanou poměrně slušnou podporu CSS (i když z dnešního pohledu naprosto nedostatečnou). Konkurenční Netscape byl zpočátku ke kaskádovým stylům velmi skeptický. První prohlížeč firmy Netscape, který začal CSS podporovat, byl Netscape 4. Podpora CSS v tomto prohlížeči je však minimální a velmi chybová. Podpora stylů v Netscape byla spíše důsledkem konkurenčního boje než touhy po standardech. V podstatě šlo jen o „z nouze ctnost“. Netscape 4 rozhodně nelze považovat za prohlížeč podporující CSS.

Firma Netscape totiž v té době slavila velké úspěchy se svou velkou zbraní – skriptovacím jazykem JavaScript. Velká část programového kódu pro formátování HTML dokumentů již byla v JavaScriptu k dispozici. Vývojářům ve firmě Netscape se tedy zdálo jen přirozené využít JavaScript k práci se styly. Tak vznikl JSS (JavaScript Style Sheets). Tato alternativní technologie se však neujala a nikdy nebyla standardizována. Její podpora byla zabudována pouze do Netscape 4.

Netscape později uvolnil svůj kód, který začali pod názvem Mozilla dále rozvíjet nezávislí tvůrci. Jádro Mozilly 0.6 poté zpětně firma Netscape implementovala do svého prohlížeče Netscape 6, a to přesto, že samotní tvůrci Mozilly upozorňovali na její přetrvávající chyby. Dá se říci, že ačkoliv je teoretická podpora standardů CSS, JavaScriptu i XHTML v Mozille 0 / Netscape 6 velmi rozsáhlá, v případě složitých konstrukcí zcela selhává. Další verze – Mozilla 1 / Netscape 7 vrátila firmu Netscape opět na piedestal tvůrce nejlepšího prohlížeče a svého konkurenta – Internet Explorer 6 – s přehledem poráží. Přesněji však nejde ani tak o Netscape, jako spíše o Mozillu a ostatní open source prohlížeče s jádrem Gecko (Phoenix, Firefox, Camino).

O místo na slunci se však začíná hlásit další hráč – Opera. Zatímco do verze 6 nebyla považována za nijak oslnivý prohlížeč (pro nás zvláště kvůli problémům se zobrazováním češtiny, slovenštiny a středoevropských jazyků vůbec), verze 6 se stala celkem oblíbenou. Vývojáři Opery však nelenili, a jejich další verze – Opera 7 – byla bomba. Atraktivní design, bohaté možnosti nastavení, výborná podpora webových technologií. Další verze – Opera 7.5 – již v podstatě mírně předehnala Mozillu v podpoře CSS, slabší je však v JavaScriptu, ale jedná se jen o pokročilejší funkce.

Současné verze prohlížečů (Mozilla 1.5+, Opera 7.5+, Konqueror 3.2+, Safari 1.2+, s (většími) výhradami i Internet Explorer 6) jsou již z hlediska podpory CSS i JavaScriptu na velmi vysoké úrovni. Není problém vytvořit pro tyto prohlížeče jakkoliv složitý XHTML dokument zcela bez použití HTML formátovacích značek, a pochopitelně i bez použití tabulek a rámců.

Co dokáže CSS

Před nástupem CSS se HTML dokumenty formátovaly pomocí značek a jejich atributů (jmenujme například značky , či <center>, nebo atributy align či bgcolor). K rozvržení stránky se používaly výhradně tabulky (co se dříve nedělalo pomocí tabulek).

S narůstající složitostí HTML dokumentů však narůstaly i problémy. Datový objem různých formátovacích značek a atributů byl často vyšší, než objem vlastního obsahu. Tím docházelo ke zpomalení načítání stránek, zbytečně narůstaly i nároky na servery (jakož i náklady). Formátování stránek pomocí skrytých tabulek pak přinášelo spoustu problémů pro jiná zařízení, nežli běžné prohlížeče. V podstatě byli vyřazeni zrakově handicapovaní, jejichž čtecí zařízení si s takovou stránkou obvykle nedokáže poradit.

Formátování dokumentů pomocí CSS tyto problémy odstraňuje. Umožňuje vytvořit čistý XHTML dokument, plně vyhovující současným standardům (např. velmi přísné normě XHTML 1.1), a hlavně 100% přístupný všem aplikacím – od velmi starých textových prohlížečů, přes zvuková zařízení, až po nejmodernější mobilní telefony.



Pochopitelně zde záleží na obsahu a účelu prezentace – např. e-obchod s rozsáhlou ukázkou zboží, formuláře řízené JavaScriptem apod., na mobilu či na papírovém výstupu z tiskárny asi nezpřevodíte...

Používání CSS vám však přinese mnoho dalších, praktických výhod:

- ✓ širší možnosti formátování,
- ✓ snadnou tvorbu a údržbu stylu,
- ✓ dynamickou práci se styly,
- ✓ dopřednou kompatibilitu,
- ✓ možnosti řízení tisku,
- ✓ formátování XML dokumentů.

Širší možnosti formátování

CSS nabízí mnohem širší možnosti formátování oproti HTML. Jeden příklad – seznamy. V HTML se rozlišuje mezi uspořádanými a neuspořádanými seznamy. Neuspořádanému seznamu lze přiřadit jeden ze tří možných stylů odrážek, uspořádanému jeden z pěti. Pokud chcete změnit styl odrážek ze symbolů třeba na čísla (nebo naopak), musíte změnit i základní značky, a to včetně koncových. CSS mezi značkami a nerozlišuje, je možno nastavit více druhů odrážek, dále libovolný obrázek či none. Změna je dílem okamžiku.

Některé potřebné vlastnosti klasický způsob formátování pomocí HTML ani realizovat neumožňuje (nebo jen ve značně omezené míře) – např. vlastnosti okrajů, výplně, orámování, pozadí jiných než tabulkových prvků, nebo pozicování (přesné nastavení polohy). V CSS lze přesně umístit nejen jakýkoliv prvek, ale třeba i obrázek na pozadí.

Snadná tvorba a údržba stylu

Obtížnost tvorby a především změn patří mezi největší slabiny klasického HTML formátování. Všechny související stránky by měly být formátovány shodně (tzv. zásada konzistentního stylu). Tedy např. barvy, písmo, ale i styl nadpisů, odstavců, odkazů, apod. Toho se při klasickém HTML formátování docílí jen velmi obtížně. Ještě obtížnější je provést změnu vzhledu celého webu. Znamená to projít všechny HTML dokumenty, nalézt a změnit stovky značek a jejich atributů, či dokonce zcela „překopat“ tabulky užité k pozicování prvků.

Naproti tomu změna webu používajícího CSS je snadná a rychlá. Pokud se například rozhodnete změnit barvu nadpisů, stačí jednoduchá úprava v tabulce stylů. Pokud byste chtěli totéž provést u webu formátovaného „postaru“, bylo by třeba projít všechny HTML dokumenty, vyhledat v nich všechny nadpisy a ručně změnit jejich atributy ve značkách . Pokud jsou tyto stránky vytvořeny dokonce pomocí WYSIWYG editoru, bývá balast několikrát se opakujících a vzájemně se „přebíjejících“ formátovacích značek často tak „neproniknutelný“, že jediným východiskem je provést změny zase jen v tomto editoru.

Dynamická práce se styly

Pomocí kombinace CSS a JavaScriptu lze vaše stránky obohatit o spoustu efektů, lze však i zcela změnit chování prezentace. Například rozbalovací nabídky či automatické nastavení barvy pozadí tabulkových buněk v závislosti na jejich obsahu.

Dopředná kompatibilita

Dopředná kompatibilita znamená, že vaše dokumenty, vytvořené například podle standardu XHTML 1.1, budou zaručeně funkční i v budoucích generacích webových prohlížečů, neboť současné verze standardů (XHTML, CSS, JavaScript) budou dozajista platit velmi dlouho, a prohlížeče je tedy budou podporovat, což se o formátování HTML tvrdit nedá. Normy HTML 4.01 a XHTML 1.0 již mnoho starších značek a atributů prohlásily za odmítnuté, a jejich užívání je povoleno pouze v tzv. přechodových verzích těchto norem (Transitional), verze XHTML 1.1 pak většinu formátovacích značek a atributů zakázala zcela. A nejde jen o „jakési standardy, které stejně nikoho nezajímají“, nejmodernější prohlížeče už nyní přestávají zvláště zastaralé značky podporovat a tento trend bude dozajista pokračovat.

Navíc, dokument vytvořený zcela bez formátovacích značek HTML dokáží zpracovat i další aplikace. Můžete sice namítnout, že vytváříte stránku určenou výhradně k prezentaci na počítačovém monitoru, nikoliv kód pro řízení mikrovlnné trouby, avšak:

- ✓ Nikdy nevíte, pro jaké zařízení budete v budoucnu chtít svůj dokument zpřístupnit (třeba mobilní telefon).
- ✓ Tím, že se naučíte vytvářet bezchybné dokumenty XHTML, bude pro vás budoucí přechod k jazyku XML mnohem snazší.

Řízení tisku

Pomocí CSS můžete též dokonale ovládat chování dokumentu při tisku. Můžete například určit, že na začátku každé vytištěné stránky bude logo vaší firmy, nebo můžete zakázat tisk ovládacích tlačítek a určit, že po celé šířce tištěné plochy bude pouze vlastní obsah.

Formátování XML dokumentů

Formátovat pomocí CSS lze nejen tradiční (X)HTML dokumenty, ale též dokumenty XML. Tato možnost je velmi důležitá, protože XML dokumenty sice současné prohlížeče již zobrazit dokáží, avšak bez jakéhokoliv formátování. Nepočítáme-li stylování pomocí XSL, je použití CSS jedinou použitelnou metodou, jak dokument vhodně zobrazit.

Vlastnosti prohlížečů a jejich použitelnost

Protože technologie CSS je poměrně nová, starší prohlížeče nedokáží dokumenty stylované pomocí CSS správně zobrazit. V zásadě je možné prohlížeče rozdělit do čtyř skupin:

- ✓ **Archaické** – tyto prohlížeče používá minimum uživatelů. Žádná podpora CSS, žádná nebo jen symbolická podpora JavaScriptu. Těmto prohlížečům můžeme nabídnout pouze čisté XHTML, jinak se jimi nemusíme zabývat. Příklady těchto prohlížečů:
 - ✓ textové prohlížeče,
 - ✓ Netscape 1, 2, 3,
 - ✓ Internet Explorer 1, 2, 3,
 - ✓ Opera 1, 2, 3.
- ✓ **Silně zastaralé** – tyto prohlížeče se občas ještě provozují, jejich podíl však (naštěstí) rychle klesá. Je pro ně charakteristická nízká podpora CSS i JavaScriptu (především standardní DOM). Také touto skupinou je zbytečné se zabývat, chceme-li, můžeme jim nabídnout pouze pár jednoduchých stylů a JavaScriptových efektů. Složitější styly a skripty by v nich napáchaly více škody než užitku. Příklady těchto prohlížečů:
 - ✓ Netscape 4,
 - ✓ Opera 4 a 5,
 - ✓ Internet Explorer 4 a 5.0.
- ✓ **Zastaralé** – tyto prohlížeče se vyznačují na jedné straně velmi vysokou podporou CSS i JavaScriptu, na straně druhé však také značnou chybovostí a mezerami. Vzhledem k jejich velkému zastoupení na webu však musíme s těmito prohlížeči vážně počítat, a prezentaci navrhnout tak, aby i těmto prohlížečům byla přístupná v „plné parádě“. Bohužel, chování Mozilly 0 a Opery 6 je však ve složitějších konstrukcích velmi problematické, někdy v nich takovouto složitější prezentaci ani nezprovozníme. O problémech Internet Exploreru se dají psát romány. Příklady těchto prohlížečů:
 - ✓ Mozilla 0 / Netscape 6,
 - ✓ Opera 6,
 - ✓ Internet Explorer 5.5 a 6.
- ✓ **Moderní** – současné verze kvalitních prohlížečů mají vysokou podporu CSS i JavaScriptu, není problém v nich zprovoznit jakkoliv složitou a efektní prezentaci zcela bez použití HTML formátování, tabulek a rámců. Příklady těchto prohlížečů:
 - ✓ Mozilla 1 / Netscape 7,
 - ✓ Opera 7,

- ✓ Konqueror 3,
- ✓ Safari 1.
- ✓ **Supermoderní** – nejnovější verze prohlížečů jsou zatím ve stadiu betaverzí a prvních subverzí. Mají velmi vysokou podporu CSS i JavaScriptu, není problém v nich zprovoznit jakkoliv složitou a efektní prezentaci zcela bez použití HTML formátování, tabulek a rámců. Příklady těchto prohlížečů:
 - ✓ Mozilla 1.75+ / Netscape 8+,
 - ✓ Opera 8+,
 - ✓ Konqueror 3, 4+.

Nejčastější problémy používaných prohlížečů

I novější prohlížeče mají v některých náročnějších konstrukcích problémy. Jde většinou o velmi složité prezentace (e-obchod na straně klienta, skrývání a zobrazování celých bloků, fixní polohování, čtení a zápis formulářových prvků apod.). Nejvážnější problémy těchto prohlížečů uvádím níže. (Kategorie archaických a silně zastaralých prohlížečů zde již nerozebírám.)

- ✓ **Mozilla 0 / Netscape 6** – v těchto prohlížečích je teoretická podpora CSS i JavaScriptu velmi vysoká. Bohužel platí, že čím složitější konstrukce a čím vyšší důraz klademe na nejnovější standardy, tím je chování Mozilly 0 / Netscape 6 problematičtější. Do jisté míry lze tyto chyby kompenzovat JavaScriptem, nebo zvolit méně náročnou normu (např. HTML 4.0 Transitional). Pokud se však nehodláme vzdát možnosti naší prezentace, ani striktního dodržování standardů (samozřejmě za předpokladu, že naše prezentace bez problémů funguje v Mozille 1 – tam „šlapat“ musí, jinak je chyba na naší straně), nezbyvá nic jiného, než i tomuto prohlížeči nabídnout pouze základní (X)HTML dokument. Problémy s týkají především JavaScriptu (například standard XHTML 1.1 nepovoluje atribut target u formulářů. Pokud tedy chceme zaslanou odpověď ze serveru zobrazit v novém okně, musíme jej otevřít JavaScriptem. Mozilla 0 toto okno sice otevře, ale odpovědi se nedočkáme. Ve složitých konstrukcích jsem pozoroval i chyby v použití stylů, například rozměry grafiky, rodina písma nebo zarovnání textu. Samozřejmě šlo o poněkud extrémně složité konstrukce, jež se však v Mozille 1 zobrazily bez problémů). Pokud ovšem tvoříte pouze „běžné“ WWW stránky, problémy se nevyskytují, a můžete se spolehnout na vysokou podporu CSS i JavaScriptu. Platí, že v Mozille 0 / Netscape 6 zprovozníme i velice složitou prezentaci, pokud slevíme z dodržování standardů.

Tyto staré verze Mozilly se dnes již téměř nevyskytují.



- ✓ **Opera 6** – problémy se týkají CSS a ještě více JavaScriptu. Opera 6 při pokusu číst a zapisovat hodnoty formulářových prvků skrytých vlastností display:none pomocí JavaScriptu končí pádem. Rovněž Opera 7 toto nedokáže, avšak pádem nekončí. Pokud naše prezentace ke své činnosti formulářové prvky nezbytně potřebuje, je nutné dosti složité obcházení tohoto problému opravným kódem JavaScriptu.