

KAPITOLA 5

Písma

Autoři specifikace CSS bez diskuse uznávají, že výběr nejvhodnějšího písma je oblíbenou (a zároveň klíčovou) schopností. Ostatně – kolik stránek je zaneřáděno desítkami, či dokonce stovkami značek `?` A skutečně – sekce "Font Properties" specifikace CSS začíná větou: "Nastavování vlastností písma je jedním z nejběžnějších využití stylových předpisů".

Navzdory této důležitosti není v současné době k dispozici žádný způsob, který by zaručoval konzistentní práci s fonty na webu, protože neexistuje jednotný způsob pro popis fontů a jejich variant. Například písma Times, Times New Roman a TimesNR si mohou být velmi podobná nebo dokonce stejná, ale jak to má chudák uživatelský agent vědět? Autor sice může v dokumentu specifikovat "TimesNR", ovšem co se stane, když si bude dokument prohlížet uživatel, na jehož stroji není tento konkrétní font nainstalovaný? A pokud je náhodou na počítači nainstalovaný Times New Roman, uživatelský agent nemá žádnou šanci se dozvědět, že tyhle dva fonty jsou vzájemně zaměnitelné. A věříte-li, že se dá u čtenáře nějaký konkrétní font vynutit, tak na to rychle zapomeňte.

Přestože CCS2 definovalo výbavu pro fonty, které bude možné stahovat, webové prohlížeče je neimplementovaly uspokojivě, a kromě toho čtenář může vždy stahování fontů zamítnout, když chce získat co nejvyšší výkon. CSS tak neposkytuje žádnou definitivní kontrolu nad fonty – nemá ji o nic větší než nějaký textový editor: když někdo načte dokument Microsoft Office, který jste vytvořili, jeho zobrazení závisí na tom, jaké fonty má dotyčná osoba nainstalované. Nemá-li stejné fonty jako vy, bude dokument vypadat jinak. Totéž platí pro dokumenty navrhované pomocí CSS.

S názvy písem začínou vznikat zmatky tehdy, vstoupíte-li do světa variant písma, jako jsou tučný text nebo kurzíva. Většina lidí sice ví, jak vypadá text psaný kurzívou (*italic*), ale jen málo z nich dokáže vysvětlit, jak a čím se liší od nakloněného textu, i když jsou mezi nimi rozdíly. "Nakloněný" (*slanted*) není jediný další termín pro text psaný kurzívou – existuje řada dalších, mj. *oblique*, *incline* (nebo *inclined*), *cursive* či *kursiv* (v češtině to vyjadřují slova *šikmý*, *kosý*, *skloněný* apod.). Takže – jedno písmo má variantu, která se dejme tomu jmenuje *TimesItalic*, ale jiné písmo zase používá název *GeorgiaOblique*. Ačkoliv jsou obě varianty možná více či méně ekvivalentní s kurzívou (*italic*), v názvech jsou označeny zcela odlišně. A obdobně – tučné písmo se může nazývat **bold**, **black** nebo **heavy**, přičemž tyto termíny mohou, ale nemusejí znamenat totéž.

CSS se pokouší poskytnout jistý mechanismus pro vyřešení všech těchto otázek spojených s písmem, kompletní řešení však poskytnout nemůže. Nejkomplikovanějšími částmi ohledně zpracování písma v CSS je soulad rodin písma a soulad tučnosti písma, přičemž z hlediska obtížnosti úlohy jsou na třetím místě výpočty velikosti písma. Další aspekty písma, k nimž se obrací CSS, jsou řezy písma, jako je kurzíva (italics), a různé varianty písma, jako jsou kapitálky – tohle jsou ovšem věci, které jsou docela bezproblémové. Různé aspekty řezů písma je možné hromadně nastavovat pomocí jediné vlastnosti `font`, kterou probereme v této kapitole později. Nejprve ale prodiskutujeme pojem rodina písma, protože je to ten nejzákladnější krok k výběru správného písma pro váš dokument.

Rodiny písma

Jak jsme prodiskutovali výše, existuje řada způsobů, jak označit různými názvy to, co ve skutečnosti znamená jedno a totéž písmo. CSS se udatně pokouší pomoci uživatelským agentům, aby se v tom zmatku lépe vyznali. Ostatně – to, co my chápeme jako "písmo", se může skládat z mnoha různých variant, jimiž se popisuje tučné písmo, text kurzívou atd. Například asi dobře znáte písmo Times. Ovšem Times je ve skutečnosti kombinace mnoha variant, mezi něž patří TimesRegular, TimesBold, TimesItalic, TimesOblique, TimesBoldItalic, TimesBoldOblique a další. Každá z těchto variant písma Times je skutečným existujícím řezem písma, nicméně my pod pojmem Times chápeme všechny tyto varianty dohromady. Jinak řečeno – Times je ve skutečnosti rodina písma, nikoliv pouze jediné písmo. Toto je velmi důležité pro zapamatování.

Kromě jednotlivých rodin písem, jako jsou Times, Verdana, Helvetica nebo Arial, definuje CSS pět všeobecných rodin písem:

■ Písma patková (serif)

Tato písma jsou proporcionální a mají patky. Písmo je proporcionální, mají-li jednotlivé znaky písma různou šířku, protože jsou různě veliká. Například – malé písmeno "i" a malé písmeno "m" jsou různě široké. (I tento odstavec knihy je napsaný proporcionálním písmem.) Patky jsou ozdoby na konci tahů, z nichž se skládá daný znak, například ty malinké čárky nahoře a dole u písmene "l", nebo konce obou nožiček velkého písmene "A". Mezi patková písma patří například Times, Georgia a New Century Schoolbook.

■ Písma bezpatková (sans-serif)

Tato písma jsou proporcionální a nemají patky. Patří mezi ně Helvetica, Geneva, Verdana, Arial a Univers.

■ Neproportionální písma (monospace)

Jak jejich název říká, nejsou proporcionální. Obvykle se pomocí nich simuluje podoba textu vytvářeného na psacím stroji, výstup ze zastaralých jehličkových tiskáren, nebo dokonce zobrazení na staříčkých terminálech. V těchto písmech jsou všechny znaky přesně stejně široké, takže malé písmeno "i" a malé písmeno "m" mají stejnou šířku. Některá z písem mají

patky, jiná ne. Má-li písmo jednotnou šířku znaků, klasifikuje se jako neproporcionální (monospace), bez ohledu na to, má-li patky nebo ne. Mezi neproporcionální písma patří Courier, Courier New a Andale Mono. V této knihu je pro výpisy zdrojových kódů použito právě neproporcionální písmo.

■ Kurzívní písma (cursive)

Tato písma se pokoušejí emulovat text psaný rukou. Znaky se obvykle většinou skládají z křivek a jsou bohatěji zdobená než patková písma. Například velké písmeno "A" může mít dole na nožičkách malé kudrlinky, nebo je rovnou celé tvořeno vlnovkami a kudrlinkami. Mezi kurzívní písma patří Zapf Chancery, Author a Comic Sans.

■ Ozdobná písma (fantasy)

Jedná se o písma, které nelze rozumě definovat pomocí jediné charakteristiky kromě té, že se nedají zařadit ani do jedné z rodin uvedených výše. Není jich mnoho a patří mezi ně Western, Woodblock a Klingon.

Teoreticky by měla každá rodina písem, kterou si uživatel může do počítače nainstalovat, spadat do jedné z těchto pěti všeobecných rodin. V praxi to ovšem může být jinak, i když výjimek je málo a jsou od sebe daleko.

Práce s všeobecnými rodinami písma

Kteroukoliv z těchto rodin písma můžete začlenit do svého dokumentu vlastností `font-family`.

font-family

Hodnoty:	<code>[[<název-rodiny> <všeobecná-rodina>],]* [<název-rodiny> <všeobecná-rodina>] inherit</code>
Výchozí hodnota:	<i>Specifická pro jednotlivé uživatelské agenty</i>
Aplikuje se na:	<i>Všechny prvky</i>
Dědí se:	<i>Ano</i>
Vypočtená hodnota:	<i>Jak je specifikována</i>

Chcete-li ve vašem dokumentu používat nějaké písmo bez patek, přičemž je vám jedno, které konkrétní písmo bude použito, pak je vhodná tato deklarace:

```
body {font-family: sans-serif;}
```

Toto způsobí, že uživatelský agent vybere nějakou rodinu proporcionálního písma bez patek (jako je Helvetica) a bude ji aplikovat na prvek `body`. Díky dědění se stejná volba písma bude aplikovat i na všechny prvky, které jsou následníky prvku `body` – pokud toto nastavení samozřejmě nepotlačí nějaký specifitější selektor.

I pouze s těmito všeobecnými rodinami písma může autor vytvořit poměrně sofistikovaný stylový předpis. Například následující sadu pravidel ilustruje obrázek 5-1.

```
body {font-family: serif;}
h1, h2, h3, h4 {font-family: sans-serif;}
code, pre, tt, span.input {font-family: monospace;}
p.signature {font-family: cursive;}
```

Obyčejný dokument

Tohle je směsice prvků, které se běžně vyskytují v normálních dokumentech. Jsou to záhlaví, odstavce, fragmenty kódu a mnoho dalších inline prvků. Jaká písma se použijí pro jednotlivé typy prvků, závisí na tom, co deklaroval autor, jaké jsou výchozí styly prohlížeče a jak se obojí prolíná.

Nadpis oddílu

Tohle je kousek předem naformátovaného textu, abychom tu nějaký takový měli

Chcete-li udělat nějaké změny ve svém spouštěcím skriptu v prostředí operačního systému DOS, napište `edit autoexec.bat`. Ale jestliže pracujete pod DOSem, pravděpodobně to už víte

—Neznámý autor

Obrázek 5-1. Různé rodiny písem.

Tento stylový předpis zajistí, že většina dokumentu bude zobrazena v nějakém písmu s patkami (jako je Times), kromě těch odstavců, které mají atribut `class` nastavený na `signature`, protože tyto odstavce se budou realizovat nějakým kurzívním písmem, mezi něž patří Author. Nadpisy prvních čtyř úrovní budou nějakým písmem bez patek, jako je Helvetica, zatímco prvky `code`, `pre`, `tt` a `span.input` budou zobrazeny nějakým neproporcionálním písmem, jako je Courier (mimořádně – neproporcionálním písmem jsou v této knize vysázeny všechny zdrojové kódy).

Specifikace rodiny písma

Autor ovšem může chtít mnohem konkrétněji vyjádřit své preference, co se týče písem, v nichž se mají zobrazovat texty prvků z jeho dokumentu. V obdobném duchu může chtít uživatel vytvořit svůj vlastní stylový předpis, v němž bude definovat zcela konkrétní fonty, v nichž se mají zobrazovat dokumenty, které si prohlíží. V obou případech stále mluvíme o vlastnosti `font-family`.

Předpokládejme na chvíli, že chce mít všechny nadpisy `h1` zobrazeny písmem Georgia. Nejjednodušší pravidlo, jak toho docílit, vypadá takhle:

```
h1 {font-family: Georgia;}
```

Toto způsobí, že uživatelský agent zobrazí v dokumentech všechny nadpisy h1 v písmu Georgia, jak to ukazuje obrázek 5-2:

Prvek záhlaví h1

Obrázek 5-2. Prvek nadpisu h1 v písmu Georgia.

Toto pravidlo samozřejmě předpokládá, že uživatelský agent bude mít rodinu písma Georgia k dispozici. Pokud ne, uživatelský agent nebude schopen použít toho písma. Pravidlo jako takové se ovšem nebude ignorovat. Nepodaří-li se agentovi najít font s názvem "Georgia", nezbude mu nic jiného, než zobrazit prvky h1 ve výchozím fontu uživatelského agenta.

Ale všechno ještě není ztraceno. Když zkombinujete názvy konkrétních fontů s názvy všeobecných rodin, dají se vytvořit vyhovující dokumenty, které mohou do značné míry splňovat vaše záměry. Pokračujme v předchozím příkladu. Následující značkování sdělí uživatelskému agentovi, aby použil rodinu Georgia, je-li dostupná, a pokud není, aby použil nějaký jiný patkový font:

```
h1 {font-family: Georgia, serif;}
```

Nemá-li čtenář nainstalovanou rodinu písma Georgia, ale má Times, bude moci uživatelský agent použít pro nadpisy h1 rodinu Times. Přestože Times není přesně totéž co Georgia, obě rodiny písem se sobě dost podobají.

Z tohoto důvodu vás důrazně vybízím, abyste vždy uvedli – jako součást každé deklarace `font-family` – nějakou všeobecnou rodinu písma. Tím poskytnete uživatelskému agentovi mechanismus zpětné vazby, který mu umožní vybrat alternativu, když nemá k dispozici přesně shodný font. Takové záložní opatření je prospěšné zejména v prostředí sahajícím přes několik platforem, protože v něm nemáte žádnou šanci zjistit, kdo má nainstalované jaké fonty. Je jasné, že na každém stroji s Windows budou po celém světě nainstalované písma Arial a Times New Roman, ale na některých (zejména starších) strojích Macintosh určitě nebudou. Totéž pravděpodobně platí pro stroje s Unixem. A naopak – zatímco na všech novějších strojích Macintosh jsou běžné fonty MarkerFelt a Charcoal, není pravděpodobné, že by některý z těchto fontů měli k dispozici uživatelé Windows a Unix, a ještě mnohem méně pravděpodobné je, že by měli rovnou oba. Několik ukázek:

```
h1 {font-family: Arial, sans-serif;}
h2 {font-family: Charcoal, sans-serif;}
p {font-family: TimesNR, serif;}
address {font-family: Chicago, sans-serif;}
```

Vyznáte-li se alespoň trochu v písmech, určitě vás napadla řada obdobných rodin, jimiž se dá zobrazit daný typ prvku. Řekněme, že chcete mít všechny odstavce v dokumentu zobrazené pomocí Times, nicméně v případě nutnosti byste akceptovali i TimesNR, Georgia, New Century School-Book a New York (všechny zmíněné písma mají patky). Takže nejprve rozhodněte o jejich pořadí

(tedy to, které písmo se má hledat jako první, které jako druhé, až když to první písmo není k dispozici atd.), pak je sestavte do řetězce a oddělte pomocí čárek:

```
p {font-family: Times, TimesNR, 'New Century Schoolbook',  
    Georgia, 'New York', serif;}
```

Na základě tohoto seznamu bude uživatelský agent hledat fonty v tom pořadí, v jakém jste je uvedli. Nenajde-li žádný z nich, jednoduše použije takový patkový font, který je k dispozici.

Apostrofy v názvech rodin písma

Možná jste si v předchozím příkladu všimli, že názvy některých rodin jsou v apostrofech. Apostrofy jsou v deklaraci `font-family` nutné tehdy, když se název rodiny skládá z několika slov, jako je písmo "New York", nebo když se v názvu vyskytují symboly, jako # nebo \$. V obou případech by měl být celý název písma v apostrofech, aby uživatelský agent nezmatkoval, protože jinak mu nemusí být jasné, co vlastně tvoří název fontu. (Možná si myslíte, že by stačily čárky, ale nestačí.) Takže – pokud bychom měli písmo s názvem `Karrank%`, musíte dát jeho název do apostrofů:

```
p {font-family: Wedgie, 'Karrank%', Klingon, fantasy;}
```

Jestliže apostrofy neuvedete, existuje nebezpečí, že uživatelský agent bude tento konkrétní název zcela ignorovat, přestože bude schopen zpracovat zbytek pravidla. Připomínám, že v současné době není ve specifikaci CSS 2.1 stanoveno jako povinné dávat do apostrofů názvy, které obsahují symboly – pouze se to doporučuje, což je v CSS velmi blízko tomu, čemu se tam říká "nejlepší praktiky". Dále se doporučuje vkládat do apostrofů názvy fontů, které obsahují mezery. Jediný povinný požadavek na apostrofy je u fontů, jejichž názvy se shodují s klíčovými slovy. Takže – voláte-li nějaký font, jehož skutečný název je *cursive*, musíte dát tento název do apostrofů.

Je evidentní, že názvy fontů, které jsou tvořeny jediným slovem – takové, které nekolidují s nějakým klíčovým slovem pro `font-family` – se nemusejí dávat do apostrofů. Všeobecné názvy rodin (jako jsou `serif`, `monospace` atd.) by se nikdy neměly dávat do apostrofů, odkazují-li se na skutečné názvy všeobecných rodin. Dáte-li do apostrofů všeobecný název rodiny, bude uživatelský agent předpokládat, že požadujete konkrétní font s tímto názvem (například font s názvem "serif"), a nikoliv všeobecnou rodinu patkových písem.

Místo apostrofů se dají také použít uvozovky. Je potřeba ale dávat pozor, pokud umísťujete pravidlo `font-family` do atributu `style`, protože tam musíte použít jiné oddělovače, než jaké jste použili pro samotný atribut (obvykle uvozovky). A proto – pokud jste uzavřeli pravidlo `font-family` do uvozovek, musíte uvnitř pravidla použít apostrofy, jako je tomu v následujícím značkování:

```
p {font-family: sans-serif;} /* nastaví odstavce na bezpatkový font */  
  
<!-- následující ukázka je správná (používají se apostrofy) -->  
<p style="font-family: 'Century Schoolbook', Times, serif;"> ... </p>  
<!--následující ukázka je NESPRÁVNÁ (používají se uvozovky) -->  
<p style="font-family: "Century Schoolbook", Times, serif;"> ... </p>
```

Použijete-li uvozovky za těchto okolností, narušíte syntaxi atributu, jak to předvádí obrázek 5-3:

Dobrý den! Tento odstavec má použít pro své zobrazení buď 'Century Schoolbook', nebo Times, nebo alternativní patkový font.

Dobrý den! Tento odstavec má také použít pro své zobrazení buď 'Century Schoolbook', nebo Times, nebo alternativní patkový font.

Obrázek 5-3. Nebezpečí skrývající se za nesprávnými oddělovači.

Tučnost písma

I když si to možná neuvědomujete, tučnost písma důvěrně znáte už nyní – nejběžnějším příkladem zdůrazněného textu je tučný řez písma (tučný text). CSS ovšem nabízí mnohem větší kontrolu (přinejmenším teoreticky) nad tučností textu pomocí vlastnosti `font-weight`.

font-weight

Hodnoty:	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit
Výchozí hodnota:	normal
Aplikuje se na:	Všechny prvky
Dědí se:	Ano
Vypočtená hodnota:	Jedna z číselných hodnot (100 atd.) nebo jedna z číselných hodnot plus jedna z relativních hodnot (bolder nebo lighter)

Všeobecně řečeno – čím je písmo tučnější, tím je tmavší a vypadá "macatější". Existuje mnoho způsobů, jimiž se označuje tučnější písmo. Například rodina písma Zurich má řadu variant, jako jsou Zurich Bold, Zurich Black, Zurich UltraBlack, Zurich Light a Zurich Regular. Každá z těchto variant je založena na tomtéž základním písmu, nicméně každá varianta má jinou tučnost.

Takže řekněme, že chcete v nějakém dokumentu použít všechny úrovně tučnosti písma Zurich. Dá se na ně odkazovat přímo (prostřednictvím vlastnosti `font-family`), nicméně v praxi tento způsob pravděpodobně vůbec nebudete používat. Ostatně by nebylo nijak příjemné, kdybyste museli psát svůj stylový předpis takto:

```
h1 {font-family: 'Zurich UltraBlack', sans-serif;}
h2 {font-family: 'Zurich Black', sans-serif;}
h3 {font-family: 'Zurich Bold', sans-serif;}
h4, p {font-family: 'Zurich', sans-serif;}
small {font-family: 'Zurich Light', sans-serif;}
```

Pokud pomineme, že je poměrně pracné psát stylový předpis v tomto duchu, je důležité vědět, že tento stylový předpis bude fungovat pouze u těch uživatelů, kteří mají nainstalované patřičné fonty. Klidně se s vámi vsadím, že většina uživatelů je mít nebude. Mnohem rozumnější by bylo, kdybychom mohli pro celý dokument specifikovat pouze jedinou rodinu písma, a pak přiřadit jednotlivým prvkům různé tučnosti. Toto je možné teoreticky udělat – pomocí hodnot vlastnosti `font-weight`. Podívejte se na následující triviální deklaraci `font-weight`:

```
b {font-weight: bold;}
```

Tato deklarace říká, že prvek `b` se má zobrazovat pomocí tučného řezu písma, nebo – pokud to mám říci jinak – nějakým písmem, které je tučnější než normální písmo dokumentu. Na to jsme zvyklí, protože `b` opravdu způsobí, že text se zobrazí tučným řezem písma.

Ve skutečnosti se ale stane to, že pro zobrazení prvku `b` se použije nějaká tučnější varianta písma. Tedy – máte-li nějaký odstavec, který je zobrazen písmem Times, a část textu z odstavce je uzavřena ve značkách `b`, v odstavci vlastně používáte dvě varianty téhož písma: Times a TimesBold. Normální text se zobrazí písmem Times, text ve značkách `b` pak písmem TimesBold.

Jak tučnosti fungují

Abyste porozuměli tomu, jak uživatelský agent určuje tučnost dané varianty písma (nemluvě o tom, jak se tučnost dědí), bude nejjednodušší, když začneme hovořit o klíčových slovech 100 až 900. Tato klíčová slova (resp. čísla) byla definovaná proto, aby zmapovala relativně běžnou schopnost návrhu písma, kdy je písmu přiděleno devět úrovní tučnosti. Například – OpenType používá číselnou škálu devíti hodnot. Má-li nějaký font v sobě zabudované tyto úrovně tučnosti, pak se výše uvedená čísla mapují přímo na předdefinované úrovně, kdy 100 je nejméně tučná (nejlehčí) varianta, přičemž 900 je nejtučnější (nejtěžší).

A skutečně – za těmito čísly není žádná konkrétní tučnost. Specifikace CSS říká jen to, že každému číslu odpovídá nejméně taková tučnost, jaká je asociována s předchozím číslem. Takže tučnosti 100, 200, 300 a 400 se mohou mapovat na stejnou (relativně málo tučnou) variantu. Hodnoty 500 a 600 mohou odpovídat nějaké tučnější variantě písma. A konečně – zbývající hodnoty 700, 800 a 900 mohou produkovat stejnou, ovšem velmi tučnou variantu písma. Nebude-li žádné klíčové slovo odpovídat variantě, které je méně tučná než je ta, kterou jsme uváděli výše, bude vše v nejlepším pořádku.

Často se stává, že tato čísla se specifikují tak, že jsou ekvivalentní s určitými běžnými pojmenovanými variantami (nemluvě o dalších hodnotách pro `font-weight`). 400 může být ekvivalentem pro `normal`, 700 odpovídá `bold`. Zbývající čísla nemusejí odpovídat žádné z ostatních hodnot pro `font-weight`, mohou však odpovídat běžným názvům variant. Existuje-li nějaká varianta písma nazvaná třeba `Normal`, `Regular`, `Roman` nebo `Book`, sdruží se s ní číslo 400, přičemž jakákoliv varianta s názvem `"Medium"` se sdruží s číslem 500. Je-li však varianta `"Medium"` jedinou dostupnou variantou, nepřihadí se k 500, ale k 400.