

KAPITOLA 1

Tři vrstvy webu

"Byl jednou jeden ... král", doplní promptně mí malí posluchači. "Ne, děti, špatně. Byl jednou jeden kus dřeva ..."

– Pinocchiova dobrodružství

I bez JavaScriptu se dá hodně udělat. S jazykem HTML¹ (Hypertext Markup Language) dokážete vyprodukovat komplikované dokumenty, které spletně popisují obsah stránky – a význam tohoto obsahu – až do těch nejmenších detailů. Pomocí kaskádových stylových předpisů (Cascading Style Sheets, CSS) můžete obsah prezentovat nesčíslnými způsoby a s obměnami, které mohou být nenápadné (jako je změna jen jediné barvy), nebo naopak velmi nápadné (náhrada textu obrázkem).

Bez ohledu na to, do čeho vaši stránku oblečete, s HTML a CSS se dá dosáhnout jen statické krásy figuríny ve výloze obchodního domu – nebo přinejlepším toho, že jde-li někdo okolo, figurína se začne vratce a toporně hýbat, s monstrózně robotickými gesty. S JavaScriptem vdechnete nemotorné loutce život, čímž zároveň sami sebe, jakožto stvořitele nové živé bytosti, pozdvihnete z nevýrazného prodejce na mistra webového designu!

Ovšem to, zdali se bude vaše nová kreace pohybovat ladně jako modelka na přehlídkovém molu, nebo se bude kulhavě belhat a šourat jako zrůda doktora Frankensteina, závisí nejenom na kvalitě výchozího kódu HTML a CSS, ale také na kódu JavaScriptu, kterým jste vaši kreaci vdechli život.

A proto, než se společně začneme učit dělat zázraky, věnujme trochu času na rekapitulaci, jak se vytvářejí weby, které vypadají dobře uvnitř i navenek, a jak do celé této skládačky zapadá JavaScript.

1 V knize se zkratkou HTML odkazujeme na HTML i XHTML. Co zvolíte vy, záleží jen na vás, JavaScriptu se to v podstatě netýká. Pro případ, že by to pro vás bylo podstatné, vězte, že kód HTML uváděný v této knize je platný XHTML.

Udržujte věci oddělené

Není to tak dávno, co profesionální weboví designéři škodolibě nacpali HTML, CSS a JavaScript do jediného souboru, který pak pojmenovali `index.html`² a říkali mu webová stránka. I dnes to můžete dělat tímto způsobem, ale připravte se na to, že v takovém případě se o vás vaši spolupracovníci nebudou vyjadřovat příliš lichotivě.



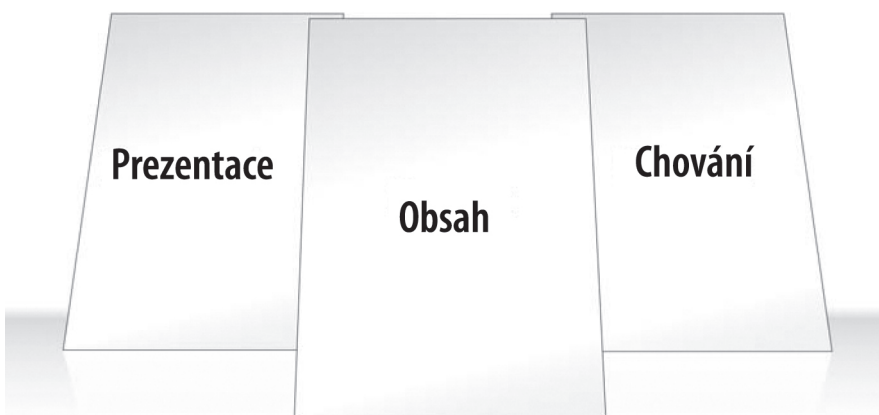
V jisté fázi vývoje WWW stránek designéři zjistili, že kód, který píše při vytváření webové stránky, dělá následující tři zásadní věci:

- Popisuje obsah stránky.
- Specifikuje prezentaci obsahu.
- Řídí chování obsahu.

Dále přišli na to, že budou-li tyto tři typy věci od sebe oddělovat – jak je to znázorněno na obrázku 1.1 – hodně jim to usnadní práci, a také to napomůže k tomu, aby webové stránky lépe fungovaly i za nepříznivých okolností, např. když mají uživatelé ve svých prohlížečích vypnutý JavaScript.

Počítačová experta ovšem tohle vědí už léta a dokonce tomuto principu přidělili odbornou zkratku SoC (Separation of Concerns, doslova "oddělení věcí").

2 Nebo `default.htm`, pokud mají mozek vymytý společností Microsoft.



Obrázek 1.1. Oddělení věcí.

Musíme ale říci, že jedna věc je vědět, že to tak má být, a druhá věc je skutečně to tak dělat – zejména tehdy, když nejste počítačový expert. Já jsem počítačový expert, a přesto jsem neustále v pokušení dělat věci nesprávně.

V pohodičce si edituji kód HTML, který popisuje obsah nějaké webové stránky, když mě najednou napadne, jak by bylo pěkné, kdyby text měl poněkud jiný odstín šedé, kdyby byl posunutý kousíček doleva, a kdyby na pozadí byl můj legrační obličej, jak jsem vyfotil sám sebe na posledním firemním večírku vydavatelství. Protože o sobě moc dobře vím, jak jsem často roztěkaný, chtěl bych ty změny udělat raději ihned, než na ně zapomenu.

Co myslíte, že je snazší: otevřít samostatný soubor CSS, abych modifikoval stylový předpis stránky, nebo jednoduše napsat příslušné vlastnosti stylu rovnou do HTML kódu, který zrovna edituji?

Podobně jako u jiné pracovní činnosti, i zde je potřebná určitá sebekázeň, pokud chcete udržet různé typy kódu oddělené od sebe. Jakmile však pochopíte, jaké to přináší výhody, budete i vy schopni najít dost pevné vůle, abyste odolali pokušení udělat to tou přímou a zdánlivě snazší cestou.

Tři vrstvy

Udržovat různé druhy kódu od sebe oddělené vždy, když to jenom trochu jde, je dobrý návyk v každém druhu programování. Části takového kódu se snadněji opětovně využívají v budoucích projektech, nehledě na to, že až dokončíte kód, který píšete, bude objem duplicitního kódu menší (takže po uplynutí několika měsíců – nebo let – se budou snadněji hledat a opravovat případné vzniklé chyby a problémy).

Pokud jde o weby, je tu ještě jeden důvod navíc, proč udržovat kód oddělený. Umožní vám totiž brát v úvahu mnoho různých způsobů, jimiž lidé přistupují k webovým stránkám.

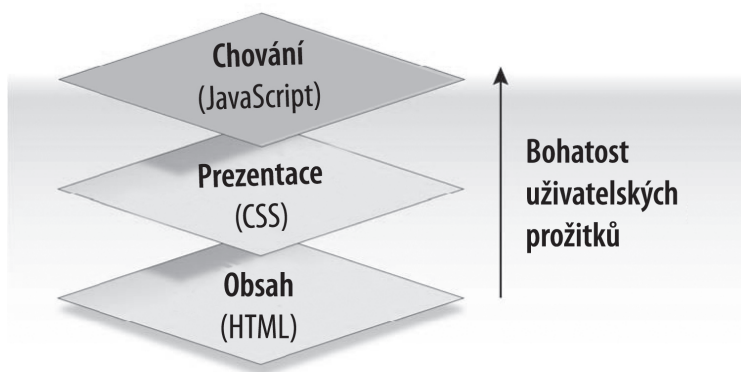
Podle toho, jaké máte publikum, většina návštěvníků pravděpodobně používá dobře vybavené prohlížeče s podporou nejnovějších verzí CSS a JavaScriptu, ovšem spousta z nich může trpět pod re-

striktními zásadami IT divize jejich společnosti, jež je nutí, aby používali starší verze prohlížečů (popř. používají prohlížeče, ve kterých mají určité funkce – jako je JavaScript – vypnuté).

Uživatelé, kteří špatně vidí, často stránky prohlížejí pomocí nějaké čtečky obrazovky (screen reader) nebo pomocí speciálního programu zvětšujícího obsah obrazovky. Pro takové uživatele může být váš brilantní vizuální design spíše překážkou než pomocí.

Někteří uživatelé váš web ani nenavštíví, protože dají přednost tomu, aby si jeho obsah přečetli pomocí RSS kanálu nebo pomocí obdobného formátu (pokud jim něco takového nabízíte). Až nastane vhodná chvíle vytvořit takové kanály, budete moci těmto uživatelům odeslat obsah HTML bez veškerého harampádí v podobě JavaScriptu nebo CSS.

Klíčem k tomu, aby vás web dokázal zaujmout co nejširší okruh návštěvníků, je přemýšlet o webu v termínech tří vrstev, které prakticky odpovídají třem druhům kódu, o nichž jsem se zmínil výše. Tyto vrstvy jsou graficky znázorněné na obrázku 1.2.



Obrázek 1.2. Tři vrstvy webu.

Když budujeme web, postupujeme těmito vrstvami zdola nahoru:

1. Začneme tím, že vyprodukujeme obsah ve formátu HTML. To je základní vrstva, kterou by měli být schopni vidět všichni návštěvníci vašeho webu, bez ohledu na to, jaký druh či verzi prohlížeče používají.
2. Když máme hotovou základní vrstvu, můžeme se soustředit na to, aby web vypadal lépe, což uděláme tím, že pomocí CSS přidáme vrstvu prezentačních informací. Web nyní bude vypadat pěkně u těch uživatelů, u nichž je možné zobrazovat styly CSS.
3. A nakonec můžeme s JavaScriptem zavést dodatečnou vrstvu interaktivity a dynamického chování (behavior), takže váš web se bude snadněji používat v prohlížečích s podporou JavaScriptu.

Udržíte-li kódy HTML, CSS a JavaScriptu vzájemně od sebe oddělené, nepochybně pak zjistíte, že je mnohem snadnější zajistit, aby vrstva obsahu zůstala čitelná i v takových prohlížečích prostředích, v nichž vrstva prezentace (a/nebo vrstva chování) nejsou funkční. Tento přístup "start na dně" k webovému designu je v branži známý jako progresivní vylepšování (progressive enhancement).

Podívejme se nyní na jednotlivé vrstvy samostatně, abyste si mohli lépe uvědomit, jakým způsobem je nejlépe od sebe navzájem oddělit.

Obsah s HTML

Všechno, co je potřeba k tomu, aby se dal přečíst a pochopit obsah webové stránky, patří do kódu HTML dané stránky – nic víc, nic méně. Tak prostě to je. Weboví designéři sami sobě přivodí trouble, když zapomenou na princip K.I.S.S.³ a cpou do svého HTML kódu informace, které nelze označit za obsah, nebo když přesouvají část obsahu stránky do kódu CSS nebo JavaScriptu.

Běžným příkladem informace, která není obsahem, a přitom se často cpe do stránek, je prezentační HTML kód. Jedná se o kód, který popisuje, jak má vypadat obsah při svém zobrazování v prohlížeči. Patří sem staromódní značky HTML, jako jsou ``, `<i>`, `<u>`, `<tt>` a ``:

```
<p>At' děláte cokoliv, <a href="666.html"><font color="red">neklíkejte  
na tomto odkazu</font></a>!/</p>
```

Může mít také formu inline CSS aplikovaného pomocí atributu `style`:

```
<p>At' děláte cokoliv, <a href="666.html" style="color: red;">neklíkejte  
na tomto odkazu</a>!/</p>
```

Prezentační kód dále může obsahovat utajenou hanbu mnoha webových designérů s dobrými úmysly – styly CSS, které jsou aplikovány v HTML s prezentačními názvy tříd:

```
<p>At' děláte cokoliv, <a href="666.html" class="red">neklíkejte  
na tomto odkazu</a>!/</p>
```

Poznámka – prezentační názvy tříd?

Jestliže vám připadá poslední příklad v pořádku, nejste sami; rozhodně to ale není dobrý postup. Usoudíte-li později, že odkaz by byl lepší žlutý, budete muset buď aktualizovat název třídy a styly CSS, které se k ní vztahují, nebo se smířit s tím (což je dost trapné), že budete mít třídu pojmenovanou "red" (červená), jež ve skutečnosti styluje žlutě. Z toho byste měli začít žloutnout – ach, promiňte, červenat!

Místo toho, abyste prezentační informace vkládali do kódu HTML, měli byste se soustředit na důvod takové akce. Například – proč chcete zobrazit nějaký odkaz jinou barvou. Je daný odkaz opravdu důležitý? Co kdybyste ho obklopili značkou popisující zdůraznění, které mu chcete udělit:

```
<p>At' děláte cokoliv, <em><a href="evil.html">neklíkejte  
na tomto odkazu</a></em>!/</p>
```

3 Keep It Simple, Stupid, neboli "dělej to jednoduše, idiote!"

Ale co když má daný odkaz sloužit jako nějaké upozornění? HTML nemá značku, která by popisovala upozornění, nicméně můžete použít takový název třídy CSS, který bude označovat tento druh informace:

```
<p>At' děláte cokoli v, <a href="evil.html" class="warning">neklíkejte  
na tomto odkazu</a>!!</p>
```

Takový přístup se dá samozřejmě přehnat. Např. někteří designéři chybně považují značky jako je `<h1>` za prezentační a pokoušejí se takový prezentační kód odstranit ze svého kódu HTML:

```
<p class="heading">Záhlaví, když mám krizi identity</p>
```

Prezentační informace, které máte opravdu držet mimo dokument, jsou zejména písmo, velikost písma a barva, v nichž se má záhlaví zobrazit. Skutečnost, že kus nějakého textu je záhlaví, takže tvoří část obsahu, by se rozhodně měla odrazit v kódu HTML. Takže tento kód je v pořádku:

```
<h1>Záhlaví, když mám mír v duši</h1>
```

Stručně řečeno – HTML by se mělo používat pro všechno, co vyjadřuje význam, neboli sémantiku obsahu ve stránce. Důsledně byste se měli vyvarovat popisů, jak má obsah vypadat. Experti na webové standardy nazývají takový HTML kód, který popisuje význam, jako sémantické značkování.

Pokud používáte sémantické značkování, jsou soubory HTML samy o sobě smysluplnými dokumenty. Lidé, kteří z nějakého důvodu nemohou vaše dokumenty číst tak, že si je prohlížíjí v typickém webovém prohlížeči, budou schopni se s nimi seznámit jinak. Například lidé, kteří velmi špatně vidí, si budou moci nechat stránku přečíst nahlas nějakým pomocným softwarem, například nějakou čtečkou obrazovky. Čím jasněji bude kód HTML popisovat význam obsahu, tím srozumitelnější bude pro ty, kteří pro jeho přečtení použijí takové smyslové nástroje.

Nejlepší ze všeho je ale to, že sémantické značkování vám umožňuje aplikovat nové styly (pro prezentaci) a interaktivní funkce (chování), aniž byste museli v kódu HTML provádět velké množství změn (pokud vůbec nějaké!).

Prezentace s CSS

Je zřejmé, že pokud má být obsah stránky kompletně obsažen uvnitř svého kódu HTML, jeho styl – neboli prezentace – by měl být plně definován v kódu CSS, který se aplikuje na stránku.

Když už jste si dali takovou práci, abyste měli HTML kód bez prezentačního kódu a sémanticky bohatý, byla by hanba pokazit takový soubor tím, že ho zaplníte kousky CSS. Jak patrně víte, styly CSS se dají aplikovat na webové stránky trojím způsobem:

■ Inline styly.

```
<a href="evil.html" style="color: red;">
```

Inline styly jsou lákavé z důvodů, které jsem už vysvětlil výše: styly můžete aplikovat na obsah hned, když ho vytváříte, aniž byste museli přehodit rychlost a editovat oddělený stylový

předpis. Ale jak už jste se dozvěděli v předchozí sekci, měli byste se snažit těmto přímým stylům vyhýbat jako čert kříži, chcete-li udržet kód HTML maximálně smysluplný.

■ Vložené styly.

```
<style type="text/css">
.warning {
color: red;
}
</style>
...
<a href="evil.html" class="warning">
```

Vložené styly sice zachovávají přehlednost značkování, nicméně takové styly jsou svázané s jediným dokumentem. Ve většině případů chcete své styly sdílet na mnoha stránkách vašeho webu, takže je nejlepší opustit i tento přístup.

■ Externí styly.

```
<link rel="stylesheet" href="styles.css" />
...
<a href="evil.html" class="warning">
```

styles.css

```
.warning {
color: red;
}
```

Externí styly jsou tou správnou cestou, kterou je potřeba preferovat, protože umožňují sdílet styl mezi více dokumenty, snižují objem kódu, který musí prohlížeč stáhnout, a v neposlední řadě umožňují změnit vzhled webu, aniž byste si museli špinit ruce editací HTML kódu.

Ale tohle všechno přece už dávno víte, že? Tohle má být ostatně kniha o JavaScriptu, takže si pojďme pohovořit o tom, jak dostanete na vaše stránky JavaScript.

Chování s JavaScriptem

Podobně jako u CSS, i JavaScript se dá do webových stránek přidávat mnoha způsoby:

■ Kód JavaScriptu se dá vložit přímo do obsahu HTML stránky:

```
<a href="evil.html" onclick="sem přijde kód JavaScriptu">
```

■ Kód JavaScriptu se také dá vložit na začátek HTML dokumentu – do značky <script>:

```
<script type="text/javascript"><!--//--><![CDATA[//><!--
sem přijde kód JavaScriptu
//--><!]]></script>
```

```
...
<a href="evil.html" class="warning">
```

Poznámka – CDATA?

Pokud se divíte, co znamená ta hatmatilka před značkou <script> a před značkou </script>, tak vězte, že s její pomocí legitimně vložíte JavaScript do XHTML dokumentu, aniž byste tím popletli ty webové prohlížeče, které nerozumí jazyku XHTML (jako je např. Internet Explorer).

Pokud vaše stránky obsahují HTML (ne XHTML), vystačíte s mnohem jednodušší syntaxí:

```
<script type="text/javascript">
sem přijde kód JavaScriptu
</script>
```

- Kód JavaScriptu vložíte do samostatného souboru a pak v dokumentech HTML uvedete odkaz na tento soubor vždy, když to potřebujete:

```
<script type="text/javascript" src="script.js"></script>
...
<a href="evil.html" class="warning">
```

script.js (výňatek)

```
sem přijde kód JavaScriptu
```

Hádejte, kterou z metod máte používat? Pokud píšete JavaScript, který vylepšuje použitelnost webových stránek, který neznečišťuje HTML dokumenty, na něž se aplikuje, který nezavírá dveře uživatelům, kteří mají ve svých prohlížečích JavaScript vypnutý, a který nenarušuje jiný kód JavaScriptu, jenž by se mohl aplikovat na tutéž webovou stránku, pak vězte, že se jedná o tzv. nevtíravé skriptování (unobtrusive scripting).

Bohužel – i když mnozí profesionální weboví vývojáři mají povědomí o výhodách, které získají, budou-li svůj kód CSS udržovat v samostatných souborech, pořád se najde dost kódu JavaScriptu vloženého přímo do HTML stránky. Když vám v této knize ukážeme správnou cestu, jak používat JavaScript, věříme, že napomůžeme tomu, aby se tato nepříjemná skutečnost změnila.

Správná cesta

Takže – jak moc je všechna tahle látka důležitá? Proč se ptáme? Inu, lidé budují weby se smíchaným kódem HTML, CSS a JavaScriptem už léta, přičemž většině lidí brouzdajících po webu takové webové stránky fungují zcela bez problémů.

Ovšem proto, že v této knize se vás chystáme naučit JavaScript, je velmi důležité, abyste se jej rovnou naučili používat tím správným způsobem. JavaScript je jednoznačně nejmocnější ze všech tří

jazyků, které používáte při tvorbě designu webových stránek, a jako takový vám poskytuje naprostou bezprecedentní volnost, jak kompletně všechno zpackat.

Jeden příklad z mnoha. Máte-li JavaScript rádi, opravdu rádi, můžete zajít až tak daleko, že umístíte úplně všechno – tedy obsah, prezentaci a chování – do kódu JavaScriptu. Už jsem to viděl na vlastní oči, a není to nic pěkného (zejména v případě, kdy je použit prohlížeč, který má JavaScript vypnutý, nebo jej vůbec nepodporuje).

Ještě více vypovídající je fakt, že JavaScript je jediný z těchto tří jazyků, který je tak schopný, že může způsobit, že prohlížeč uživateli jednoduše "zamrzne", takže nebude vůbec reagovat⁴.

Ve zbývajících částech knihy budeme tudíž maximálně usilovat o to, abychom vám neustále ukazovali správnou cestu, jak používat JavaScript. Nejenom proto, že pak budete mít kód úhledný, ale také kvůli tomu, že to napomáhá, aby web pracoval tak, jak se od něho očekává (tzn. aby zpřístupnil svůj obsah co největšímu počtu lidí, bez ohledu na to, jaký webový prohlížeč se rozhodli používat).

Knihovny JavaScriptu

Jak už jsem zmínil, oddělení různých druhů kódu přináší mnoho výhod. Jednou z nich je, že kód, který jste napsali pro jeden web, můžete snadno opětovně využít pro jiné weby. Někteří velcí příznivci JavaScriptu (od této chvíle se na ně budeme odkazovat jako na "lidi") si udělali čas na to, aby sestavili obrovské knihovny (libraries) užitečného, nevtíravého kódu JavaScriptu, které si můžete stáhnout a zdarma používat ve vašich webech.

V knize budujeme všechny příklady od začátku. To znamená, že veškerý kód JavaScriptu, který budete k těmto příkladům potřebovat, najdete zde, na stránkách knihy. Protože ve skutečném světě není vždy možné postupovat tímto způsobem, a protože knihovny se velmi rychle staly významnou součástí světa JavaScriptu, podíváme se také, jak se pracuje s některými populárními knihovnami JavaScriptu. V této knize budeme používat následující knihovny:

- Prototype – <http://www.prototypejs.org/>
- script.aculo.us – <http://script.aculo.us/>
- Yahoo! User Interface Library, YUI – <http://developer.yahoo.com/yui/>
- Dojo – <http://dojotoolkit.org/>
- jQuery – <http://jquery.com/>
- MooTools – <http://mootools.net/>

⁴ Názornou ukázkou uvidíte v kapitole 7.

Varování – všechny knihovny nejsou stejně dobré

Dávejte si pozor na weby, na nichž se nabízejí fragmenty kódu JavaScriptu, které stačí zkopírovat pomocí schránky a vložit do vašich webových stránek pro docílení nějakého konkrétního efektu. Existuje velké množství takového zdarma poskytovaného kódu. Jeho kvalita je ovšem rozdílná.

Dobrá knihovna je obvykle poskytována ve formě javascriptového souboru .js, který můžete nevtíravě vložit do vašich stránek, takže nemusíte vkládat kód JavaScriptu přímo do HTML kódu.

Necítíte-li se dost povolaní, abyste rozhodli, zdali je nějaká konkrétní knihovna JavaScriptu dobrá nebo ne, požádejte o radu na fórech vývojářského serveru [interval.cz](http://interforum.interval.cz)⁵, nebo se prostě spokojte pouze s knihovnami zmiňovanými v této knize, protože ty všechny jsou velmi dobré.

Začněte!

Dost bylo kázání – tuto knihu jste si pořídili proto, abyste se naučili JavaScript, je to tak? (Jestliže tomu tak není, obávám se, že asi budete trochu zklamaní.) Čistý a přehledný kód HTML i CSS je sice velmi pěkný, nicméně nyní je čas skočit rovnýma nohama přímo do třetí vrstvy webů: chování. Otočte několik stránek a hned narazíte na skvělý (a nevtíravý) JavaScript.

⁵ <http://interforum.interval.cz>