



OKAMŽITĚ

OVLÁDNĚTE PHP ZA VÍKEND



Callum Hopkins

PHP Okamžitě

Computer Press Brno 2014

PHP Okamžitě

Callum Hopkins

Překlad: Ondřej Baše Odpovědný redaktor: Martin Herodek Technický redaktor: Jiří Matoušek

Authorized Czech translation of the English edition of Jump Start PHP, 1st Edition (ISBN 9780987467409) © 2013 Sitepoint Pty. Ltd. This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to sell the same.

Translation © Ondřej Baše, 2014

Objednávky knih: http://knihy.cpress.cz www.albatrosmedia.cz eshop@albatrosmedia.cz bezplatná linka 800 555 513 ISBN 978-80-251-4196-0

Vydalo nakladatelství Computer Press v Brně roku 2014 ve společnosti Albatros Media a. s. se sídlem Na Pankráci 30, Praha 4. Číslo publikace 18477.

© Albatros Media a. s., 2014. Všechna práva vyhrazena. Žádná část této publikace nesmí být kopírována a rozmnožována za účelem rozšiřování v jakékoli formě či jakýmkoli způsobem bez písemného souhlasu vydavatele.

1. vydání



Obsah

Poděkování	7
O autorovi	7
Úvod	9
Komu je tato kniha určena	9
Použité konvence	9
Ukázky zdrojového kódu	9
Tipy, poznámky a varování	10
Zpětná vazba od čtenářů	11
Zdrojové kódy ke knize	11
Errata	11
KAPITOLA 1	
Server	13
Co je PHP	13
Co budeme potřebovat	14
Začínáme	15
Windows	15
Mac OS X	16
Linux	19
Nastavení jazyka PHP	20
Ahoj světe	21
Proměnné v jazyce PHP	25
Pole	27
Komentáře	28
Připravujeme si náš projekt	28
Shrnutí	29

KAPITOLA 2	
Jazyk PHP a data	31
Operátory	31
Podmíněné příkazy	33
Příkaz if	33
Příkaz else	33
Příkaz elseif	33
Příkaz switch	34
Cykly	35
Cyklus for	35
Cyklus while	36
Cyklus foreach	37
Databáze, MySQL a PHP	37
Shrnutí	45
KAPITOLA 3	
Objekty a OOP	47
První kroky s OOP	47
Rozšiřujeme třídy	50
Šablony	53
Soubory projektu	55
Shrnutí	62
KAPITOLA 4	
Formuláře	63
Formulářové elementy	63
POST a GET	65
Cílová adresa URL formuláře	69
Superglobální proměnné a proměnná \$_REQUEST	70
Formuláře a databáze	71

Stavíme na našich základech	74
Třída pro komunikaci s databází	74
Příspěvky blogu	80
Vkládané šablony	83
Administrační část blogu	85
Shrnutí	95
KAPITOLA 5	
Relace a cookies	97
Cookies – popis	97
Relace – popis	98
Relace vs. cookies	98
Cookies	98
Relace	99
Relace a cookies v jazyce PHP	100
Cookies v jazyce PHP	100
Relace v jazyce PHP	102
Projekt	103
Shrnutí	121
KAPITOLA 6	
Jazyk PHP a bezpečnost	123
Soubor php.ini	123
Možnost nastavení allow_url_include	124
Možnost nastavení open_basedir	124
Nakládání s chybami	124
Vylepseni zabezpečeni relači	125
Validace odesílaných dat	126
Shrnutí	128
Závěr	128
Rejstřík	129

Poděkování

Děkuji své rodině a přátelům za neutuchající podporu. Miluju vás všechny.

O autorovi

Callum Hopkins se živí jako programátor webových aplikací, a navíc rád navrhuje. Vyzbrojen znalostmi návrhu i programování umí postavit celou webovou aplikaci bez cizí pomoci. Jeho touha po znalosti složitých funkcí a tvorby nádherného designu ho nutí hledat nové způsoby, jak navrhnout, naprogramovat a optimalizovat složitá webová řešení pro své klienty po celém světě.

Úvod

Jazyk PHP mnozí považují za jeden z nejoblíbenějších programovacích jazyků určených pro vývoj webových aplikací. Původně byl tento jazyk navržen, aby vylepšoval webové stránky – měl umožnit generovat jejich obsah dynamicky, ale postupně se z něj stal plnohodnotný programovací jazyk. S jazykem PHP lze jednoduše vytvářet komplexní aplikace, jako jsou diskuzní fóra, galerie obrázků atd.

V této knize se naučíte základům vývoje aplikací v jazyce PHP. Začnete jednoduchými webovými stránkami s dynamickým obsahem a dostanete se až k interaktivním webovým aplikacím. Naučíte se o bezpečnosti, interakci s databázemi a nastavení vývojového prostředí pro tvorbu aplikací v jazyce PHP.

V průběhu této knihy budeme průběžně pracovat na projektu, kterým bude malá (avšak robustní) blogovací aplikace. Ta nám poslouží jako praktický příklad pro teoretické informace, které si budeme uvádět v jednotlivých kapitolách. V tomto projektu implementujeme řadu užitečných funkcí (prezentační část s příspěvky, komentáře a administrační nástroje) a objasníme si na něm všechny popisované koncepce.

Komu je tato kniha určena

Všem vývojářům, kteří se chtějí rychle seznámit s jazykem PHP. Musíte znát jazyky HTML a CSS. Zkušenost s jinými programovacími jazyky jistě nebude na škodu.

Použité konvence

V této knize narazíte na řadu typografických konvencí a různých stylů rozvržení, které budou zvýrazňovat různé typy informací. Prohlédněte si proto následující konvence.

Ukázky zdrojového kódu

Zdrojový kód bude napsán neproporcionálním písmem, například takto:

```
<h1>Skvělý letní den</h1>
Byl krásný letní den, který přímo vybízel na procházku parkem. Ptáčci
zpívali a všechny děti byly zpátky ve škole.
```

Pokud se daný zdrojový kód nachází v archivu zdrojových kódů k této knize, příslušné jméno souboru se objeví nad tímto výpisem:

priklad.css

```
.zapati {
   background-color: #CCC;
   border-top: 1px solid #333;
}
```

Pokud bude výpis zobrazovat jen část obsahu souboru, bude označen slovem úryvek:

priklad.css (úryvek)

```
border-top: 1px solid #333;
```

Když budeme k současnému příkladu přidávat další zdrojový kód, takový kód se bude zobrazovat tučně:

```
function animuj() {
    var nova_promenna = 'Ahoj';
}
```

Místo již napsaného zdrojového kódu bude výpis obvykle obsahovat tři tečky:

```
function animuj() {
    ...
    return nova_promenna;
}
```

Tipy, poznámky a varování

Tip: Haló, vy tam! Tipy vám poskytují malé cenné rady.

Poznámka: Ehm, promiňte... Poznámky jsou doplňující informace, které se týkají tématu, ale nejsou kriticky důležité.

```
Upozornění: Nezapomeňte vždy...
... věnovat pozornost těmto sdělením.
```

Zpětná vazba od čtenářů

Nakladatelství a vydavatelství Computer Press, které pro vás tuto knihu přeložilo, stojí o zpětnou vazbu a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

Computer Press Albatros Media a. s., pobočka Brno IBC Příkop 4 602 00 Brno

nebo

sefredaktor.pc@albatrosmedia.cz

Computer Press neposkytuje rady ani jakýkoli servis pro aplikace třetích stran. Pokud budete mít dotaz k programu, obraťte se prosím na jeho tvůrce.

Zdrojové kódy ke knize

Z adresy *http://knihy.cpress.cz/K2152* si po klepnutí na odkaz Soubory ke stažení můžete přímo stáhnout archiv s ukázkovými kódy.

Errata

Přestože jsme udělali maximum pro to, abychom zajistili přesnost a správnost obsahu, chybám se úplně vyhnout nelze. Pokud v některé z našich knih najdete chybu, ať už chybu v textu nebo v kódu, budeme rádi, pokud nám ji oznámíte. Ostatní uživatele tak můžete ušetřit frustrace a pomoci nám zlepšit následující vydání této knihy.

Veškerá existující errata zobrazíte na adrese *http://knihy.cpress.cz/K2152* po klepnutí na odkaz Soubory ke stažení.

Server

V této kapitole:

- Co je PHP
- Co budeme potřebovat
- Začínáme
- Nastavení jazyka PHP
- Ahoj světe
- Proměnné v jazyce PHP
- Pole
- Komentáře
- Připravujeme si náš projekt

Co je PHP

PHP je nejoblíbenější skriptovací jazyk na straně serveru v oblasti webového vývoje. Pohání téměř 78,9 % všech webových stránek.

Vytvořil ho Rasmus Lerdorf v roce 1995. PHP byla původně zkratka pro "Personal Home Page (Tools)" (což bychom mohli přeložit jako "Nástroje pro osobní domovské stránky"), ale nyní je spíše rekurzivní zkratkou pro "PHP: Hypertext Preprocessor" (v překladu "Hypertextový preprocesor PHP"). Tento jazyk spravuje, sleduje a vyvíjí skupina vývojářů s názvem The PHP Group, která ho stále distribuuje jako volně dostupný skriptovací jazyk. Jazyk PHP je k dispozici ke stažení přímo na oficiálních webových stránkách.¹

Jazyk PHP bývá většinou interpretován, zpracováván a jeho výsledky zobrazovány webovým serverem s nainstalovaným modulem PHP. Kód jazyka PHP můžeme vkládat do značkovaného kódu jazyka HTML v souborech s příponou *.php*. Jazyk PHP je možné zprovoznit v téměř všech operačních systémech a na všech platformách, a to zdarma. Nejčastější volbou jsou osobní počítače s operačním systémem Linux.

V dnešní době vývojáři používají jazyk PHP pro řešení úloh na straně serveru, a ne jako obecný skriptovací jazyk. Tento jazyk je konkurentem pro jazyk ASP.NET společnosti Microsoft, modul mod_perl od společnosti Apache Software Foundation a platformu Node.js od společnosti Joynet. Hlavním cílem jazyka PHP je zpracovávat data tak, aby je bylo možné dynamicky zobrazovat do webových stránek – provádí matematické výpočty, převádí formáty a spolupracuje s databázemi. Umožňuje vývojářům vylepšit statické stránky o reakce na uživatelské požadavky, nebo dokonce ukládat načtená data do databáze a později je uživatelům vypisovat.

¹ http://php.net/

Jazyk PHP se specializuje na webový vývoj, a proto představuje obvyklou volbu pro vývojáře, kteří vytvářejí webové aplikace nebo stránky. Jeho strmá křivka učení dovoluje programátorům, aby v něm rychle začali vytvářet vlastní aplikace, přičemž poskytuje dostatek funkcí na to, aby se programátoři nemuseli v budoucnu obracet na jiné programovací jazyky. Převážnou část úspěšných webových projektů pohání jazyk PHP – jedná se například o projekty Digg, Etsy, Yahoo, Facebook a Wikipedia.

Jednoduchým příkladem použití jazyka PHP může být zobrazování počtu návštěvníků dané stránky. Do databáze bychom uložili počet lidí, kteří navštívili tuto stránku, a potom vygenerovali kód jazyka HTML, v němž bychom zobrazili aktuální počet. S pomocí jazyka PHP můžeme ale vytvářet také složité webové stránky s víceúrovňovou navigací, která obsahuje odkazy na spoustu dílčích stránek. Tento jazyk bývá také obvyklým nástrojem pro tvorbu firemních prezentací. Dokonce v něm můžeme přizpůsobit stránky konkrétnímu uživateli na základě informací, které o něm posbíráme.

Velká popularita jazyka PHP dala vzniknout rovněž rozsáhlé komunitě vývojářů, kteří nabízejí pomoc každému, kdo o ni stojí (a to převážně zdarma), a také rozrůstající se knihovně, jež poskytuje informace online i offline.

Co budeme potřebovat

Jazyk PHP je k dispozici na téměř jakémkoli webovém hostingu. Můžeme ho ale nainstalovat spolu s webovým serverem Apache HTTP Server na domácí počítač, čímž si vytvoříme vlastní lokální vývojový server. V závislosti na kvalitě připojení k Internetu můžeme dokonce zpřístupnit soukromý webový server přes Internet.

Lokální servery obvykle instalujeme pomocí oblíbených balíků LAMP (pro operační systém Linux; obsahuje Apache, MySQL a PHP), MAMP (pro operační systém Mac OS X; obsahuje Apache, MySQL a PHP) nebo WAMP (pro operační systém Windows; obsahuje Apache, MySQL a PHP), které jsou volně ke stažení.

Tyto balíky tvoří instalační program, s nímž na jedno klepnutí nainstalujeme webový server s výchozí konfigurací. Díky tomu si vytvoříme obdobné prostředí, jaké poskytují společnosti nabízející hosting webových projektů. Programátoři standardně vytvářejí aplikace na svých lokálních serverech, protože mají okamžitý přístup k souborům a nemusejí čekat, než se jejich úpravy nahrají na vzdálený server. Navíc se nemusejí obávat, že by nějaká nedokončená změna pronikla do ostrého provozu, a ušetří přenos dat na webovém hostingu (často se za nadměrný přenos dat připlácí).

Začínáme

Nainstalovat vlastní webový server na svůj počítač se může zdát složité, ale většinou to vyřeší instalační program jedním klepnutím. K dispozici máte několik možností podle toho, který operační systém používáte.

Windows

V operačním systému Windows si můžete vybrat jeden ze dvou oblíbených instalačních programů. Prvním z nich je WAMP,² jenž nainstaluje Apache, PHP, MySQL a phpMyAdmin na váš počítač (phpMyAdmin je webové rozhraní pro správu databází MySQL). Druhým je XAMPP³ od týmu Apache Friends, který obsahuje Apache, MySQL, PHP, Perl a phpMyAdmin.

V této knize najdete jen postup instalace balíku WAMP, ale instalace balíku XAMPP je velmi podobná, kdybyste se rozhodli pro něj.

Nejprve si stáhněte balík WAMP.⁴ Můžete si vybrat z několika verzí v závislosti na vašem procesoru a operačním systému.



Poznámka: Jaký máte procesor?

Klepněte pravým tlačítkem myši na ikonu **Počítač** a vyberte položku **Vlastnosti**. Alternativně můžete postupovat přes **Ovládací panely**, kde klepnete na položku **Systém a zabezpečení** a posléze na položku **Systém**. V okně, které se objeví, uvidíte v poli **Typ systému** buď hodnotu 32bitový, nebo 64bitový operační systém.

Upozornění: Visual C++

Na svůj počítač si musíte nainstalovat také balík Visual C++ 2010 SP1 Redistributable Package x86 nebo x64. Odkaz na jeho stažení najdete na stránkách balíku WAMP, ale musíte si vybrat správnou verzi pro svůj operační systém (32bitovou, nebo 64bitovou). Tento balík byste měli nainstalovat před balíkem WAMP.

Po stažení balíku WAMP spusťte instalační program.

Jakmile nainstalujete balík WAMP, měli byste vidět ikonu v oznamovací oblasti systému Windows. Tato ikona mění svou barvu podle aktuálního stavu služeb:

- Červená webový server Apache a databázový systém MySQL jsou vypnuté. Důvodem může být, že jste je nespustili, nebo jsou špatně nastavené.
- Oranžová jedna ze služeb se nespustila. Viníkem bývá obvykle drobná chyba v konfiguraci nelze načíst doplňující knihovnu nebo jiný program používá výchozí port. Řešení tohoto problému byste měli hledat na helpdesku nebo diskuzním fóru.

² http://www.wampserver.com/en/

³ http://www.apachefriends.org/index.html

⁴ http://www.wampserver.com/en/#download-wrapper

Zelená – všechny služby běží a nenastaly žádné chyby. Jinými slovy – vše v pořádku.

Pokud klepnete levým tlačítkem myši na ikonu balíku WAMP, můžete začít pracovat se službami. Konkrétně můžete dělat následující věci:

- Spravovat webový server Apache a databázový systém MySQL.
- Vypínat a zapínat služby.
- Instalovat a přepínat verze webového serveru Apache, databázového systému MySQL a jazyka PHP.
- Přistupovat k záznamovým souborům.
- Přistupovat ke konfiguračním souborům.
- Vytvářet aliasy.

Jestliže klepnete pravým tlačítkem myši na tuto ikonu, můžete ukončit program WAMP, změnit jeho jazyk nebo otevřít nápovědu na jeho oficiálních webových stránkách. V případě, že si s něčím ohledně balíku WAMP nevíte rady, zkuste hledat odpověď v této nápovědě.

Tip: Problémy s portem 80

Nejčastější problém s balíkem WAMP nastává, pokud nějaký jiný program blokuje port 80 (výchozí port určený pro komunikaci s webovým serverem Apache). Pokud máte kupříkladu spuštěný program Skype v okamžiku, kdy spouštíte balík WAMP, můžete narazit na tuto chybu, jelikož program Skype taktéž používá port 80. Tento problém snadno vyřešíte tak, že vypnete program Skype a znovu spustíte program WAMP. Na webových stránkách Stack Overflow⁵ ale najdete lepší řešení.

Správnou funkčnost lokálního serveru ověříte tak, že otevřete svůj webový prohlížeč a do jeho adresního panelu zadáte adresu *localhost* nebo *127.0.0.1*. Měla by vás uvítat domovská stránka balíku WAMP.

Mac OS X

Pro uživatele Maců je k dispozici jednoduchý instalační program (na jedno klepnutí) lokálního serveru s podporou jazyka PHP, který se jmenuje MAMP. Podobně jako u operačního systému Windows můžete alternativně použít balík XAMPP. Raději byste ale měli použít balík MAMP, protože ten je skvěle přednastavený pro vývoj webových aplikací pomocí jazyka PHP.

V první řadě si stáhněte tuto aplikaci z oficiálních webových stránek projektu MAMP,⁶ a jakmile stáhnete soubor *.pkg*, otevřete ho, čímž zahájíte instalační proces.

⁵ http://stackoverflow.com/questions/4705005/how-to-solve-wamp-and-skype-conflict-on-windows-7/ 4705033#4705033

⁶ http://www.mamp.info/en/index.html

Version 2.2 Version Française Version 2.4 Version 4.4 Version Votentity Version 2.4 Version 4.4 Version Votentity Version 2.4 Version 4.4 Version Version 4.4 Version Votentity Version 4.4 Version Ver	Firefox *	age +									×
► Constant of the second seco	(Iocalhost				☆ マ C 関 - G	ioogle 🖌	+	÷	D -	₩	* -
Server Configuration Apache Version : 2.4.2 PHP Version : 5.4.3 Loaded Extensions : Core * bcmath * calendar * com_dotnet * type * date * ereg * filter * type * date * session * standard * mysqlnd * tokenizer * zip * zilb * libomi * dom * PDO * Phar * SimpleXML * apache2hand(* curl * mysql * mysql * mash * xwlebug MySQL Version : 5.5.24 Tools * phpinfo() * phpinmyadmin Ceh * ceh * ceh * ceh * ceh	Wa	W pServer									^
Server Configuration Apache Version : 2.4.2 PHP Version : 5.4.3 Loaded Extensions : Core * bcmath * calendar * com_dotnet * ctype * date * ereg * filter * tp * hash * iconv * json * mcrypt * SPL * odbc * pcre * Reflection * session * standard * mysqlnd * tokenizer * zlp * zlib * libuml * dom * PDO * Phar * SimpleXML * dom * PDO * Phar * SimpleXML * mysql * mysqli * mysql * mysql * mysql * pdo_sqlte * mhash * xdebug MySQL Version : 5.5.24 Tools / phpinfo() / phpmyadmin Your Projects - ceh - ceh						Version 2.2 Ver	sion Fra	nçaise			
Apache Version : 2.4.2 PHP Version : 5.4.3 Loaded Extensions : Core bornath calender com_dotnet ctype dete ereg filter to p hash bronc pore Reflection session standard proysqlnd tokenizer 2 zip 2 zib blownl dom PDO Phar SimpleXML apacheZhandle curl mobstring gd mysql mysql pdo_mysql pdo_sqlite mhssh xudebug MySQL Version : 5.5.24 Tools Phpmyadmin Your Projects ceh ceh	Se	erver Configur	ration								
PHP Version : 5.4.3 Loaded Extensions : • Core • bcmath • calendar • com_dotnet • can_dotnet • can_dotn		Apache Version :	2.4.2								
Loaded Extensions : Core & borneth & calender & con_dotnet		PHP Version :	5.4.3								
Ceh	To	MySQL Version : MySQL Version : phpinfo() phpmyadmin Dur Projects	Core ctype ftp mcrypt Reflection tokenizer dom widdx apache2har mysql mhash 5.5.24	bcmath b	 calendar ereg iconv odbc standard zlib Phar xmireader mbstring pdo_mysql 	<pre>com_dotnet filter filter</pre>					
a ceh-bkup	En l	ceh									
		ceh-bkup									
🔄 ceh-dev	6	ceh-dev									
🖾 ceh2	6	ceh2									
iquid_css		liquid_css									
🖾 lovely		lovely									
C mvc		mvc									
Permisson-Bar	Ca (Permisson-Bar									
Pro-Candy-Golf	6	Pro-Candy-Golf									
🗇 project1	-	project1									
test		test									

Obrázek 1.1. Domovská stránka balíku WAMP



Poznámka: MAMP Pro

Instalační program MAMP pravděpodobně vytvoří dva adresáře – jeden s názvem *MAMP* a druhý s názvem *MAMP Pro*. Pokud ve svém počítači objevíte adresář *MAMP Pro*, odstraňte ho i příslušnou aplikaci. MAMP Pro je placený program, kdežto program MAMP je zdarma.

Po dokončení instalace spusťte program MAMP klepnutím na jeho ikonu. Měli byste vidět úvodní obrazovku.

	MAMP Age your website locally Test MAMP PRO
Status Apache Server MySQL Server	Stop Servers Open start page Preferences Quit

Obrázek 1.2. Uvítací obrazovka programu MAMP

Na úvodní obrazovce (viz obrázek 1.2) byste měli vidět dvě červené ikony před položkami **Apache Server** a **MySQL Server**. Tyto červené ikony označují, že jste tyto služby prozatím nespustili. Spustíte je klepnutím na tlačítko **Start Server**. Červené body by měly zezelenat. Krátce po klepnutí na tlačítko **Start Server** by se měl otevřít váš výchozí webový prohlížeč a v něm uvítací stránka balíku MAMP.



Obrázek 1.3. Uvítací stránka balíku MAMP

Úspěšně jste nainstalovali balík MAMP. Jestliže však chcete přistupovat k webovému serveru Apache, musíte použít port 8888. To není tak pohodlné, jako když můžete napsat do adresního řádku jen *localhost*, jako to umožňuje balík WAMP. Nápravu ale sjednáte jednoduše – přepněte se na úvodní obrazovku programu MAMP a klepněte na tlačítko **Preferences**. V horní části panelu **Ports** byste měli najít příslušnou položku nabídky.

Otevřete tím okno, v němž byste měli vidět vstupní pole s popiskem **Apache Port**. Pokud změníte jeho hodnotu z 8888 na 80, můžete přistoupit k uvítací stránce balíku MAMP jednoduše přes adresu *localhost/MAMP*.

Linux

Jestliže používáte operační systém Linux (například Ubuntu nebo Debian), můžete z terminálu nainstalovat balík LAMP z repozitáře systému. Instalace lokálního serveru v systému Linux se mírně liší od postupu ze systémů Windows a Mac OS X. Tato instalační metoda je o něco složitější, ale poskytuje více volnosti. Ve skutečnosti téměř stejný postup používají společnosti poskytující webový hosting na Internetu (pokud používají systém Linux). Zde uvedený příklad demonstruje postup instalace balíku LAMP na systému Ubuntu.

Nejprve spusťte terminál a potom začněte instalací webového serveru Apache. Do terminálu napíšete tento příkaz:

sudo apt-get install apache2

Stiskněte klávesu Enter a váš terminál se zblázní, ale nebojte se, je to normální.

Tím by měla být instalace serveru Apache u konce, což si ověříte tak, že otevřete webový prohlížeč a vepíšete adresu *localhost* do panelu adresy. Úspěšně nainstalovaný webový server by vás měl přivítat stránkou s textem "It works!" jako na obrázku 1.4.



Obrázek 1.4. Stránka na adrese localhost na Linuxu

Nyní musíte nainstalovat ještě jazyk PHP, proto se vraťte do terminálu a napište:

```
sudo apt-get install php5 libapache2-mod-php5*
```

Zahájíte tak instalaci jazyka PHP5 spolu s doplňujícími knihovnami, které umožní spolupráci tohoto jazyka s webovým serverem Apache, jejž jste nainstalovali před chvílí. Po dokončení instalace musíte restartovat server Apache, aby se dozvěděl, že přibyl jazyk PHP. Do terminálu napište níže uvedený příkaz:

sudo service apache2 restart

Právě jste restartovali svůj server Apache, aby si mohl načíst jazyk PHP a všechny nezbytné knihovny, které bude potřebovat pro zpracování kódu jazyka PHP.

Váš lokální server už běží, a dokonce umí pracovat i s jazykem PHP. Při instalaci systému MySQL a nástroje phpMyAdmin se můžete řídit návodem⁷ od Mhabuba Mamuna.

Nakonec můžete povolit modul mod_rewrite serveru Apache, který umožňuje vývojářům přesměrovávat uživatele na různé sekce webových stránek přepisováním požadovaných adres URL. Postup, jak používat modul mod_rewrite, najdete kupříkladu na stránkách:

- http://www.jakpsatweb.cz/server/mod-rewrite.html
- http://net.tutsplus.com/tutorials/other/a-deeper-look-at-mod_rewrite-for-apache/

Nastavení jazyka PHP

Než začneme psát kód jazyka PHP, můžeme si nastavit tento jazyk tak, aby lépe vyhovoval našemu projektu. Jazyk PHP nastavujeme v souboru *php.ini*, jenž ovlivňuje, jak se bude tento jazyk chovat na našem serveru. Rychlý přehled aktuální konfigurace si můžeme vypsat funkcí phpinfo() ve skriptu jazyka PHP. Za tímto účelem si vytvoříme nový soubor s názvem *info.php* v adresáři webového serveru (následující část této kapitoly popisuje, kde tento adresář hledat) a přidáme do něho následující zdrojový kód:

```
<?php
phpinfo();
```

Když otevřete svůj skript *info.php* v prohlížeči (přes adresu *http://localhost/info.php*), měli byste vidět stránku s informacemi o aktuální konfiguraci jazyka PHP, jak ukazuje obrázek 1.5.

Najděte řádek označený **Configuration File (php.ini) Path**, jenž obsahuje cestu k vašemu souboru *php.ini*. Na obrázku 1.5 můžete vidět, že soubor *php.ini* se nachází v adresáři *C*:*devprogphp5*\ *php.ini*. Vaše cesta bude nejspíše úplně jiná, proto ji musíte najít ve výstupu funkce phpinfo().

Soubor *php.ini* obsahuje velké množství možností nastavení. Většinu z nich nikdy nebudete potřebovat. S některými z nich si ale můžete přizpůsobit jazyk PHP svému stylu a prostředí, v němž vyvíjíte aplikace. Pokud chcete prozkoumat soubor *php.ini* blíže, přečtěte si můj článek "A Tour of php.ini".⁸

⁷ http://www.developmentwall.com/install-apache-php-mysql-phpmyadmin-ubuntu/4/

⁸ http://www.sitepoint.com/a-tour-of-php-ini/

PHP Version 5.4.25

System	Windows NT ONDRA-THINK 6.1 build 7601 (Windows 7 Home Premium Edition Service Pack 1) i586
Build Date	Feb 5 2014 21:16:25
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	cscript /nologo configure.js "enable-snapshot-build" "disable-isapi" "enable- debug-pack" "without-mssql" "without-pdo-mssql" "without-pi3web" "with- pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared" "with-oci8=C:\php-sdk\oracle \instantclient10\sdk,shared" "with-oci8-11g=C:\php-sdk\oracle\instantclient11 \sdk,shared" "enable-object-out-dir=/obj/" "enable-com-dotnet=shared" "with- mcrypt=static" "disable-static-analyze" "with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\dev\prog\php5\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525,TS,VC9

Obrázek 1.5. Výstup funkce phpinfo()

Ahoj světe

Protože už jste si nainstalovali a spustili svůj lokální webový server, můžete začít programovat. Otevřete proto adresář webového serveru Apache (zkráceně označovaný jako **webový adresář**), jehož umístění závisí na operačním systému, který používáte:

Pokud používáte operační systém Windows a nainstalovali jste si balík WAMP, otevřete si adresář C:\Program Files\wamp\www\.

P

- Jestliže máte systém Mac OS X s balíkem MAMP, webový adresář najdete na cestě /Aplikace/MAMP/htdocs.
- V operačním systému Linux se webový adresář skrývá pod cestou /var/www. Vývojáři, kteří pracují v systému Linux, si rádi upravují cestu k webovému adresáři v konfiguračním souboru webového serveru. Pokud se k nim chcete přidat, zkuste vyhledat řešení na webových stránkách Stack Overflow.⁹

Až najdete webový adresář, vytvořte v něm adresář, jenž bude kořenovým adresářem pro všechny soubory (a adresáře) vašeho projektu. Zatím tedy vytvořte adresář s názvem *muj_projekt*. V názvu adresáře byste neměli používat mezery, speciální znaky ani velká písmena, protože na jeho soubory se budete odkazovat adresami URL.

V adresáři *muj_projekt* vytvořte nový soubor PHP a pojmenujte ho *index.php*. Nově vytvořený soubor si otevřete v textovém editoru, který umí upravovat soubory s kódem jazyka PHP.

Poznámka: Výběr textového editoru

Zkuste si najít textový editor, který zvýrazňuje syntaxi jazyka PHP. Volba editoru je velice osobní záležitost, jelikož různé editory nabízejí různé funkce. Některé textové editory můžou podpořit váš styl vývoje, zatímco jiné s ním můžou kolidovat. Závěrečné rozhodnutí závisí na tom, v kterém editoru se vám programuje nejjednodušeji a jestli vás v něm programování baví. Proto byste si měli vyzkoušet mnoho rozdílných textových editorů, než si vyberete ten svůj.

Následuje stručný seznam nejoblíbenějších editorů. Nachází se v něm kombinace volně dostupných i placených programů:

- Adobe Dreamweaver¹⁰ (Windows a Mac OS X) jeden z nejznámějších editorů, který nabízí mnoho jednoduchých funkcí a vestavěného klienta FTP.
- Sublime Text¹¹ (Windows, Mac OS X a Linux) editor, který si lze vyzkoušet zdarma a poskytuje užitečné zkratky a makra.
- Komodo Edit¹² (Windows, Mac OS X a Linux) mocný volně dostupný editor, jenž nabízí úhledné rozhraní a vestavěného klienta FTP.
- Coda¹³ (Mac OS X) uživatelsky přívětivý editor se širokou škálou funkcí.
- Notepad++¹⁴ (Windows) velmi oblíbený volně dostupný editor.

Jakmile si vyberete editor, otevřete v něm svůj soubor *index.php* a můžete začít psát kód jazyka PHP. Vložte do něj následující kód a uložte ho:

<?php echo "Ahoj světe"; ?>

14 http://www.notepad-plus-plus.org/

⁹ http://stackoverflow.com/questions/5891802/how-do-i-change-the-root-directory-of-an-apache-server/ 5891858#5891858

¹⁰ http://www.adobe.com/products/dreamweaver.html

¹¹ http://www.sublimetext.com/

¹² http://www.activestate.com/komodo-edit

¹³ http://panic.com/coda/

Zkuste spustit tento kód. Otevřete webový prohlížeč a do pole s adresou zadejte adresu *http://localhost/muj_projekt/index.php*. Měli byste vidět něco podobného jako na obrázku 1.6.

	Firefox http://localhost/muj_projekt/index.php +		1	-
		٩	÷	⋒
	Ahoj svÄ>te			
Ĺ				

Obrázek 1.6. Ahoj světe

Gratuluji. Právě jste napsali svůj první kód v jazyce PHP. Do webového prohlížeče jste vypsali sdělení "Ahoj světe". Nevšímejte si toho, že prohlížeč zobrazuje písmeno ě špatně. Nenastavili jste totiž znakovou sadu, která by obsahovala česká písmena s diakritickými znaménky, což se naučíte za malou chvíli. Prozatím si prohlédněte, co se děje na předchozím řádku kódu:

- <?php: tento výraz označujeme jako otevírací značku. Tou sdělujeme webovému serveru, že následující kód by měl interpretovat jako kód jazyka PHP. Webový server tedy zpracuje tento kód pomocí interpreteru jazyka PHP. Tuto značku musíme uvést v souboru vždy, když chceme psát kód v jazyce PHP.
- echo "Ahoj světe";: echo je základní konstrukcí jazyka PHP, s jejíž pomocí mu sdělujeme, že to, co následuje za tímto příkazem, chceme zobrazit do webového prohlížeče. Do uvozovek vkládáme text, který by měl uživatel vidět na obrazovce. Kromě toho jsme nezapomněli ani na středník, jímž ukončujeme příkaz echo.
- ?>: jedná se o uzavírací značku, která říká serveru, že by měl přestat zpracovávat následující kód jako kód jazyka PHP. Jestliže ponecháme kód jazyka PHP "neuzavřený," můžeme zabránit serveru, aby poslal do prohlížeče neúmyslně nadbytečný obsah. Pokud uzavírací značku nepotřebujeme, je lepší ji vynechat, abychom se vyhnuli problémům, které můžou nastat ve složitém kódu jazyka PHP. Dále v této knize se budeme často setkávat s kódem jazyka PHP bez uzavírací značky. Alespoň budeme vědět, že tato značka chybí záměrně.

Upozornění: Nezapomínejte na středníky

Je velmi podstatné, abyste nezapomněli psát středníky na konec všech příkazů, jinak by jazyk PHP nevěděl, kde končí jeden příkaz a začíná druhý.

Do souborů s kódem jazyka PHP můžeme vkládat i kód jazyka HTML. Díky kombinování obou typů zdrojových kódů můžeme vytvářet skutečně nádherné webové stránky s dynamickým obsahem, který jazyk HTML sám nepodporuje.

Poznámka: Znaková sada

Výše jsme narazili na problém, že webový prohlížeč nezobrazil české písmeno **ě** správně. V první řadě bychom měli všem souborům, které budou obsahovat texty vypisované do prohlížeče (nebo raději preventivně všem souborům) nastavit ve vybraném editoru znakovou sadu UTF-8 (pokud bude na výběr, tak UTF-8 bez BOM, protože s touto značkou si webový server mnohdy neumí poradit). Také nesmíme zapomínat, že webový prohlížeč nezobrazuje uživatelům text, ale zpracovaný kód jazyka HTML. Nesmíme proto zapomenout uvést alespoň záhlaví dokumentu HTML a nastavit v něm pomocí elementu **meta** znakovou sadu UTF-8, jako kdybychom vytvářeli běžnou webovou stránku. Když to neuděláme, prohlížeč zobrazí české znaky chybně, jak jsme si ukázali ve výše uvedeném příkladu.

Pro zachování jednoduchosti příkladů v této knize nebudou jednotlivé příklady obvykle obsahovat element **meta**, který by nastavil znakovou sadu UTF-8, a proto se budou české znaky zobrazovat v prohlížeči špatně. Už ale víme, jak bychom to případně opravili.

Zde je náš příklad "Ahoj světe" v podobě dokumentu HTML s dynamickým obsahem (s textem "Ahoj světe"):

```
<!DOCTYPE html>
<html lang="cs" dir="ltr">
<head>
<meta charset="utf-8" />
<title>Ahoj světe</title>
</head>
<body>
<?php echo "Ahoj světe"; ?>
</body>
</html>
```

Teď už prohlížeč zobrazuje písmeno **ě** správně. Zde je další příklad spolupráce jazyků PHP a HTML:

```
<!DOCTYPE html>
<html lang="cs" dir="ltr">
<head>
<meta charset="utf-8" />
<title>Spolupráce jazyků PHP a HTML</title>
</head>
<body>
<h1><?php echo 'Toto je element h1 s daty z PHP'; ?></h1>
<?php echo 'Toto je element p také s daty z PHP.'; ?>
</body>
</html>
```

Pokud změníte obsah souboru *index.php* na výše uvedený zdrojový kód a zadáte do svého webového prohlížeče adresu *http://localhost/muj_projekt/*, uvidíte, že jazyk HTML zobrazuje všechna data zpracovaná v jazyce PHP správně.

Proměnné v jazyce PHP

Posuňme naše schopnosti psát kód o něco dále – prozkoumejme společně svět proměnných. Proměnná reprezentuje místo v počítačové paměti, kam můžeme dočasně ukládat informace složené z bitů (jedniček a nul). Proměnné lze přidělit kus dat a pak opakovaně používat tuto proměnnou napříč skriptem. Kdykoli **interpreter jazyka PHP** (dále budeme zjednodušeně označovat jen jako **jazyk PHP**) narazí na proměnnou v našem kódu, ví, že má přistoupit k datům, které tato proměnná představuje. Do proměnné můžeme ukládat jakákoli data – od jednoduchých textových řetězců ("Ahoj světe") přes čísla (1234) až po složité datové struktury, o nichž si povíme za chvíli.

Před název proměnné vždy zapisujeme symbol \$ (znak dolaru). Název proměnné nesmí obsahovat mezery a může obsahovat písmena (malá i velká), čísla a podtržítka. První znak za symbolem \$ navíc může být jen písmeno nebo podtržítko. Platným jménem proměnné může být kupříkladu \$mojeUzasnaPromenna, nebo dokonce \$moje_Uzasna_Promenna_1.

Praxí osvědčený postup spočívá v pojmenovávání proměnných smysluplně tak, aby kterýkoli programátor, jenž si je přečte, věděl, jaká data reprezentují. Kdybychom například chtěli vytvořit proměnnou, do které uložíme něčí jméno, pojmenovali bychom ji \$jmeno0soby.



Poznámka: Velbloudí zápis

Pokud se název proměnné skládá z více slov, všechna slova kromě prvního zahájíme velkým písmenem. Tím vylepšíme čitelnost kódu.

Ukažme si, jak proměnné používat. Upravíme soubor index.php následovně:

```
<?php
$maPromenna = 'Ahoj světe, používám proměnnou.';
echo $maPromenna;
```

Co jsme právě teď naprogramovali? Nejprve jsme uložili hodnotu (textový řetězec 'Ahoj světe, používám proměnnou.') do proměnné \$maPromenna. Protože jsme poprvé použili proměnnou \$maPromenna, jazyk PHP ji automaticky vytvořil. Když teď napíšeme kdekoli ve skriptu \$maPromenna, PHP ví, že má načíst data, která jsme přidělili této proměnné.

Pokud obnovíte stránku *index.php* ve svém prohlížeči, měli byste vidět text "Ahoj světe, používám proměnnou". Nezapomeňte, že chybné české znaky můžete ignorovat.

Pokračujme dále v úpravách zdrojového kódu ze souboru *index.php*, jenž by měl teď vypadat následovně:

```
<?php
$maPromenna;
$maPromenna = 'Ahoj světe, používám proměnnou.';
echo $maPromenna;
$maPromenna = 'Na shledanou, světe.';
echo $maPromenna;
```

Po obnovení stránky byste měli vidět věty "Ahoj světe, používám proměnnou." a "Na shledanou, světe". Data v proměnné můžete totiž kdykoli přepisovat.

Výše uvedený příklad rovněž demonstruje, jak PHP zobrazuje výstup z našeho kódu řádek po řádku směrem shora dolů. Prvním příkazem echo vypisujeme první větu, která je tou dobou uložená v proměnné \$maPromenna. Po prvním příkazu echo přepisujeme ale její hodnotu. Neměli bychom zapomínat na uspořádání "shora dolů", když určujeme, co se má dít s daty na konkrétním místě v našem skriptu. Jestliže přepíšeme obsah proměnné dříve, než ho stačíme vypsat, ztratíme data a výstup skriptu nebude správný.

Do proměnných je možné ukládat různé typy dat, nejen textové řetězce. Představme si datové typy, které můžeme používat:

php</th <th></th> <th></th>		
<pre>\$maPromenna = 0;</pre>	//	Integer
<pre>\$maPromenna = 3.14;</pre>	//	Float
\$maPromenna = "Od začátku roku";	//	String
\$maPromenna = true;	//	Boolean
<pre>\$maPromenna = array(250, 300, 325, 475);</pre>	//	Array

Ve výše uvedeném zdrojovém kódu používáme nejrůznější typy proměnných, proto si je teď v rychlosti projděme:

- Integer celá čísla. Můžou být kladná, záporná nebo nula.
- Float desetinná čísla. Ta můžou být také kladná, záporná nebo nula.
- String textové řetězce, což jsou posloupnosti písmen, čísel a speciálních znaků. Textové řetězce zapisujeme do uvozovek (ať už jednoduchých, nebo dvojitých).
- Boolean pravdivostní hodnota; jedna ze dvou hodnot true nebo false. Kolem této hodnoty nesmějí být uvozovky, jinak by se stala textovým řetězcem.
- Array pole; tj. víceúrovňová datová struktura, která se podobá tabulce.

Toto je pouze náhled elektronické knihy. Zakoupení její plné verze je možné v elektronickém obchodě společnosti eReading.