Zahvalnost

Ovu knjigu posvećujem svom fakultetu (Northeast Missouri State University). Posebno bih želeo da zahvalim Dr. Moniki Barron, Dr. Denisu Leavens-u, Dr. Edu Tyler-u, i Dr. Kolu Woodcox-u koji je takođe i moj prijatelj. Da nije bilo njihovih veličanstvenih i dirljivih predavanja ne bih ni ja postao ono što jesam, kao pisac, student i nastavnik.

Posebnu zahvalnost dugujem:

Kao i uvek svima koji rade za Peachpit Press. Veliku zahvalnost dugujem divnom uredniku na ovom projektu, Marku Taberu koji je rukovodio njime i koji je imao mnogo strpljenja da izađe na kraj sa gomilom email-ova i poglavlja koja su kasnila!

Takođe zahvaljujem Davidu Van Ness-u i Elizabet Welch koji su naporno radili i davali mi korisne sugestije, sa izuzetnim osećajem za detalje. Zahvalnost dugujem i Skautu Festu koji se potrudio da sve bude na svom mestu što se tiče ove knjige. Takođe, veliko hvala i Valeriji Perry koja je radila indeks i Danijeli Foster koja se bavila dizajnom kao i Timotiju Boronczyk-u koji se bavio tehničkim aspektima ove knjige.

Naravno, ogromnu zahvalnost i poštovanje dugujem svim dobrim ljudima koji su pored ostalih sjajnih projekata radili i na PHP-u, MySQL-u, Apache-u, phpMyAdmin-u, MAMP-u i XAMPP-u. Ovom prilikom želim da iskreno zahvalim svim učesnicima na forumima za podršku kao i pripadnicima različitih grupa koji su svima kojima je to bilo potrebno nudili svoju pomoć i savet.

Kao i uvek, želeo bih da zahvalim čitaocima čija podrška daje pun smisao mom poslu. Posebno bih zahvalio svima koji su pomogli oko prevoda u poglavlju 17 „Primer – Tabla za poruke”, kao i onima koji su davali sugestije šta bi želeli da vide u ovom izdanju.

Na kraju, da nije bilo ljubavi i podrške koju mi je pružala moja supruga Džesika, ne bih bio u stanju da napišem nijednu knjigu. Posebno bih zahvalio Zoe i Samu koji su mi davali razloge za i protiv pisanja knjiga!

Sadržaj

Uvod ix

Poglavlje 1 Uvod u PHP 1

Osnovna sintaksa 2
Slanje podataka u veb pretraživač 6
Pisanje komentara 10
Šta su to promenljive? 14
Uvođenje stringova 18
Nadovezivanje stringova 21
Uvođenje brojeva 23
Uvođenje konstanti 26
Jednostruki i dvostruki navodnici 29
Osnovni koraci za otkrivanje i uklanjanje grešaka 32
Pregled naučenog i unapređivanje stečenih veština 34

Poglavlje 2 Programiranje sa PHP-om 35

Kreiranje HTML obrasca 36
Rukovanje HTML obrascem 41
Uslovi i operatori 45
Ispitivanje validnosti podataka u obrascu 49
Uvođenje nizova 55
Petlje For i While 70
Pregled naučenog i unapređivanje stečenih veština 73

Poglavlje 3 Pravljenje dinamičkih veb sajtova 75

Uključivanje većeg broja fajlova 76
Ponovno rukovanje HTML obrascima 85
Pravljenje trajnih obrazaca 91
Kreiranje vaših sopstvenih funkcija 95
Pregled naučenog i unapređivanje stečenih veština 112

Poglavlje 4 Uvod u MySQL 113

Imenovanje elemenata baze podataka 114
Biranje tipova za kolone. 116
Biranje ostalih svojstava kolona 120
Pristupanje MySQL-u 123
Pregled naučenog i unapređivanje stečenih veština. . . . 130

Poglavlje 5 Uvod u SQL 131

Pravljenje baza podataka i tabela 132
Dodavanje zapisa 135
Selektovanje podataka 140
Upotreba uslova 142
LIKE i NOT LIKE 145
Sortiranje rezultata upita 147
Ograničavanje rezultata upita 149
Ažuriranje podataka 151
Brisanje podataka 153
Upotreba funkcija 155
Pregled naučenog i unapređivanje stečenih veština. . . . 166

Poglavlje 6 Projektovanje baze podataka. 167

Normalizacija 168
Pravljenje indeksa 181
Upotreba različitih tipova podataka 184
Jezici i MySQL 186
Vremenske zone i MySQL. 191
Ograničenja stranih ključeva 197
Pregled naučenog i unapređivanje stečenih veština: . . . 204

Poglavlje 7 Napredni SQL i MySQL 205

Izvođenje operacija JOIN 206
Grupisanje selektovanih rezultata 216
Napredne selekcije 220
FULLTEXT pretraživanja 224
Optimizacija upita 232
Transakcije 236
Šifrovanje baza podataka 239
Pregled naučenog i unapređivanje stečenih veština: . . . 242

Poglavlje 8 Rukovanje greškama i njihovo otklanjanje 243

Vrste grešaka i osnove njihovog otklanjanja 244
Prikazivanje PHP grešaka 250
Podešavanje izveštavanja o greškama u PHP-u 252
Pravljenje prilagođenih rukovalaca za obradu grešaka . . 255
Otklanjanje grešaka u PHP-u 260
Otklanjanje grešaka u SQL-u i MySQL-u. 264
Pregled naučenog i unapređivanje stečenih veština 266

Poglavlje 9 PHP i MySQL 267

Menjanje šablona 268
Povezivanje sa MySQL-om 270
Izvođenje jednostavnih upita 275
Izdvajanje rezultata upita 284
Garantovanje bezbednog SQL-a 288
Brojanje vraćenih zapisa 293
Ažuriranje zapisa pomoću PHP-a. 296
Pregled naučenog i unapređivanje stečenih veština 304

Poglavlje 10 Uobičajene tehnike programiranja 305

Slanje vrednosti u skript. 306
Korišćenje skrivenih ulaznih polja obrasca 310
Uređivanje postojećih zapisa 316
Paginacija rezultata upita 323
Prikaz koji može da se sortira. 331
Pregled naučenog i unapređivanje stečenih veština 336

Poglavlje 11 Razvijanje veb aplikacija. 337

Slanje elektronske pošte 338
Otpremanje fajlova 344
PHP i JavaScript 356
Razumevanje HTTP zaglavlja 364
Funkcije za datum i vreme 370
Obavljanje transakcija 374
Pregled naučenog i unapređivanje stečenih veština 380

Poglavlje 12 Kolačići i sesije 381

Izrada stranice za prijavljivanje 382
Funkcije za prijavljivanje 385
Korišćenje kolačića 390
Sesije 404
Unapređena bezbednost sesije 412
Pregled naučenog i unapređivanje stečenih veština 416

Poglavlje 13 Bezbednosne metode 417

Sprečavanje neželjene pošte 418
Proveravanje validnosti po tipu podataka 425
Proveravanje ispravnosti po tipu fajla 431
Sprečavanje XSS napada 435
Upotreba Filter ekstenzije 438
Sprečavanje napada SQL Injection 442
Obezbeđivanje lozinki pomoću PHP-a 449
Pregled naučenog i unapređivanje stečenih veština 458

Poglavlje 14 Perl-kompatibilni regularni izrazi 459

Probni skript 460
Definisanje jednostavnih uzoraka 464
Korišćenje kvantifikatora 467
Korišćenje klase znakova 469
Pronalaženje svih uparivanja 472
Korišćenje modifikatora 476
Pronalaženje i zamena uzoraka 478
Pregled naučenog i unapređivanje stečenih veština 482

Poglavlje 15 Uvod u jQuery 483

Šta je to jQuery? 484
Uključivanje radnog okruženja jQuery 486
Korišćenje radnog okruženja jQuery 489
Biranje elemenata stranice 492
Rukovanje događajima 495
Manipulisanje objektnim modelom dokumenta (DOM) . . . 499
Korišćenje Ajax-a 505
Pregled naučenog i unapređivanje stečenih veština 518

Poglavlje 16 Osnove objektno orijentisanog programiranja . 519

Osnove i sintaksa 520
Rad sa MySQL-om 523
Klasa `DateTime` 538
Pregled naučenog i unapređivanje stečenih veština 546

Poglavlje 17 Primer – Tabla za poruke 547

Pravljenje baze podataka 548
Pisanje šablona 557
Izrada stranice Index 565
Izrada stranice Forum 566
Izrada stranice Thread 571
Rukovanje obrascem 580
Pregled naučenog i unapređivanje stečenih veština 586

Poglavlje 18 Primer – registracija korisnika 587

Kreiranje šablona 588
Pisanje konfiguracionih skriptova 594
Izrada početne stranice 602
Registracija 604
Aktiviranje naloga 614
Prijavljivanje i odjavljivanje 617
Upravljanje lozinkama 624
Pregled naučenog i unapređivanje stečenih veština 634

Dodatak A Instalacija 635

Proces instaliranja na Windows-u 636
Proces instaliranja na Mac-u 639
Upravljanje MySQL korisnicima 641
Testiranje instalacije 646
Konfigurisanje PHP-a 649
Konfigurisanje Apache-a 652

Indeks 662

Uvod

Korisnici veba u naše vreme očekuju da stranice budu uzbudljive, da se često ažuriraju i da pružaju doživljaj na koji su navikli. Za njih su veb sajtovi kao društva kojima se više puta vraćaju. U isto vreme, administratori veb sajtova žele sajtove koji se lako ažuriraju i održavaju, smatrajući da je to jedini razuman način da ispune očekivanja svojih posetilaca. Iz tih i iz još nekih razloga su PHP i MySQL postali de facto standardi za izradu dinamičkih veb sajtova zasnovanih na bazama podataka.

Ova knjiga predstavlja vrhunac mog višegodišnjeg iskustva na razvoju veb sajtova, uz napomenu da sam napisao nekoliko prethodnih knjiga o tehnologijama koje ovde opisujem. Cilj ove knjige je da se najvažnija znanja iznesu na najefikasniji način. Ona će vas naučiti kako da počnete da pravite dinamičke veb sajtove i kroz primere vam ponuditi velike količine koda od kojeg možete da počnete. Na vama je jedino da pokažete želju za učenjem.

Naravno, tu je i računar.

Šta su to dinamički veb sajtovi?

Dinamički veb sajtovi su fleksibilne i moćne tvorevine, koje bi tačnije bilo nazvati *aplikacijama* nego samim sajtovima. Dinamički veb sajtovi imaju sledeće karakteristike:

- Reaguju na različite parametre (na primer, na doba dana ili na verziju veb pretraživača koju koristi posetilac sajta)
- Imaju „memoriju", čime se omogućavaju registrovanje i prijavljivanje korisnika, e-trgovina i slični procesi
- Skoro uvek sadrže HTML obrasce, pomoću kojih korisnik vrši pretraživanja, daje povratne informacije i tako dalje
- Često imaju interfejse putem kojih administratori upravljaju sadržajem sajta
- Lakše se održavaju, ažuriraju i nadograđuju u poređenju sa statičkim sajtovima

Postoje mnoge tehnologije za pravljenje dinamičkih veb sajtova. Najuobičajenije su ASP.NET (Active Server Pages, iz Microsofta), JSP (Java Server Pages), ColdFusion, Ruby on Rails (radni okvir za razvoj veba programskim jezikom Ruby) i PHP. Dinamički veb sajtovi se ne oslanjaju uvek na neku bazu podataka, ali to je sve češći slučaj, pogotovo što su dostupne izvrsne aplikacije za baze podataka kao što je MySQL i MongoDB koje su jeftine ili čak besplatne.

Šta se desilo sa PHP-om 6?

Kada sam napisao prethodno izdanje ove knjige, *PHP 6 and MySQL 5 for Dynamic Web Sites:Visual QuickPro Guide*, naredna verzija PHP-a – PHP 6 – bila je približno 50% dovršena. Misleći da će, dakle, PHP 6 izaći nešto nakon moje knjige, u nekim delovima tog izdanja sam se oslanjao na beta verziju PHP-a 6. A onda... PHP 6 je obustavljen.

Jedna od ključnih karakteristika planiranih za PHP 6 bila je podrška za Unicode, to jest, PHP 6 je trebalo da podrazumevano može da koristi bilo koji jezik. To bi bilo veliko unapređenje ovog već popularnog programskog alata. Nažalost, kad se uvođenje podrške za Unicode pokazalo ne samo složenim već i veoma teškim, autori jezika su odložili PHP 6. Međutim, nije sve izgubljeno: neke od drugih mogućnosti planiranih za PHP 6, kao što je podrška za imenske prostore (pojam iz objektno orijentisanog programiranja), dodate su u verziju PHP-a 5.3.

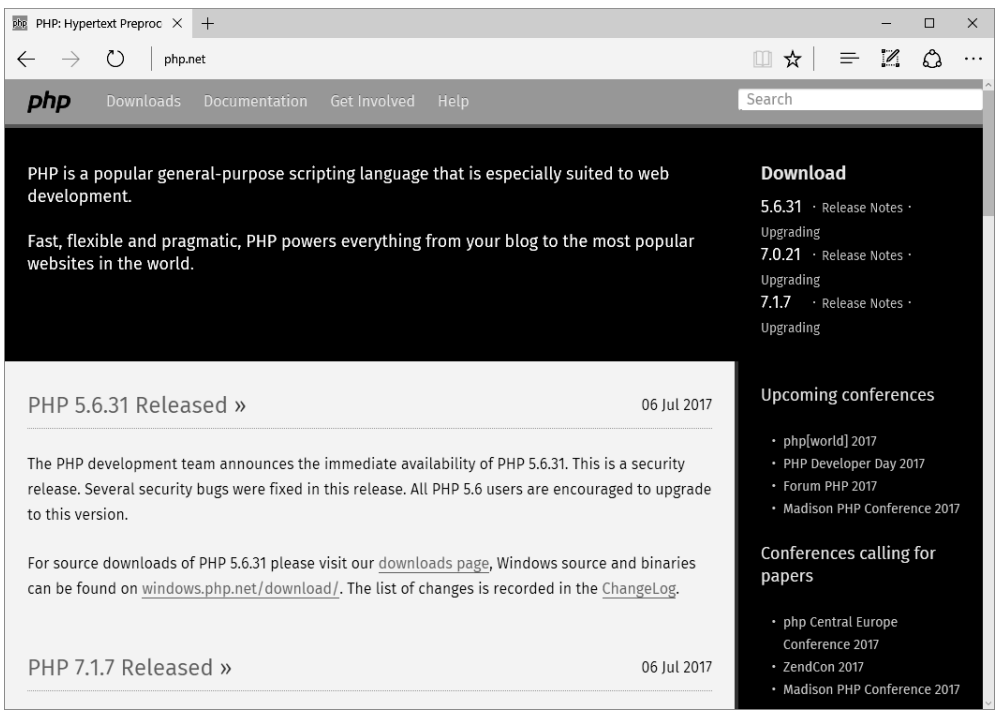
Kada jer došao trenutak da se izbaci na tržište nova verzija PHP-a, odlučeno je da se ona nazove PHP 7 kako bi se izbegla konfuzija sa verzijom PHP 6 koja je započeta ali nikada nije dovršena.

Šta je to PHP?

PHP je prvobitno značio „Personal Home Page” kad ga je Rasmus Lendorf 1994 napravio da bi pratio broj posetilaca svog onlajn rezimea. Kako su mu rasle mogućnosti i upotrebljivost (počeo je da se koristi u stručnijim situacijama), skraćenica se promenila u „PHP: HyperText Preprocessor”.

Na zvaničnom sajtu, koji se nalazi na adresi **`www.php.net`** Ⓐ, PHP se definiše kao „jezik za skriptovanje opšte namene u širokoj upotrebi koji je naročito podesan za veb programiranje”. To je dugačka i opisna definicija, čije ću značenje objasniti.

nastavak na sledećoj strani



Ⓐ Početna stranica za PHP.

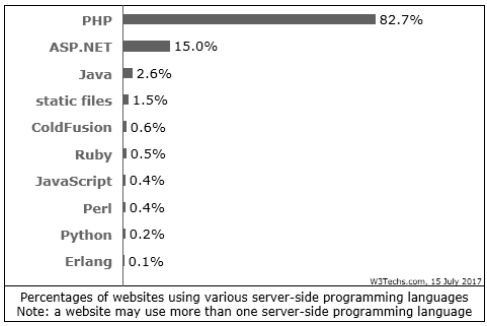
Počevši od kraja te definicije, koja kaže da je PHP *naročito podesan za veb programiranje*, zapravo znači da iako PHP možete koristiti za programiranje van veba, on je ipak najprikladniji za korišćenje prilikom veb programiranja. Iz ovoga sledi, da iako se mnoštvo drugih tehnologija može koristiti za veb programiranje one ipak nisu najprikladnije za to. Jednostavno rečeno, ako želite da se bavite veb programiranjem, PHP je zaista odličan izbor.

Osim toga, PHP je jezik za *skriptovanje* za razliku od jezika koji se *kompajliraju*: PHP je projektovan da bi se pisali veb skriptovi, nije namenjen za samostalne aplikacije (mada se uz dodatni trud sada mogu praviti i aplikacije u PHP-u). PHP skript se izvršava tek nakon nekog događaja – na primer, kada korisnik pošalje obrazac ili ode na URL (Uniform Resource Locator), što je tehnički izraz za veb adresu.

Ovoj definiciji bi trebalo da dodam napomenu da je PHP serverska višeplatformska tehnologija. Oba opisa su važna. *Serverska* se odnosi na činjenicu da se sve što PHP radi odvija na serveru: neophodna je aplikacija veb servera, kao što su Apache ili Majkrosoftov IIS (Internet Information Services), a svim PHP skriptovima mora da se pristupa kroz URL (**`http://`*`nešto`***). Višeplatformska znači da se PHP izvršava na većini operativnih sistema, uključujući Windows, Unix (i njegove mnogobrojne varijante) i Macintosh. Ono što je još važnije jeste činjenica da će PHP skript napisan na jednom serveru normalno funkcionisati na drugom uz minimalne izmene ili bez njih.

U vreme kad je ova knjiga pisana, aktuelna verzija PHP-a je bila 7.1.7. Iako je PHP 7 glavna verzija, najbitnije izmene se nalaze u njoj, tako da je PHP 7 daleko efikasniji u odnosu na verziju PHP 5.

U većini slučajeva, primeri će funkcionisati kako treba pod uslovom da koristite najmanje verziju 5.4. Neke funkcije i



B Na sajtu Web Technology Surveys naći ćete ovaj grafikon koji se odnosi na tehnologije za serversku stranu (**`www.w3techs.com/technologies/overview/programming_language/all`**).

karakteristike koje se obrađuju u ovoj knjizi zahtevaće novije verzije, kao što je PHP 5.6 ili novije. U takvim slučajevima, objasniću vam sve kada dođe do dodavanja funkcionalnosti u PHP-u i ponudiću vam alternativno rešenje ako nekim slučajem koristite stariju verziju jezika.

Više informacija o PHP-u uvek možete naći na adresi PHP.net.

Zašto biste koristili PHP?

Jednostavno rečeno, kada treba razviti dinamički veb sajt, PHP je bolji i brži nego njegove alternative i lakše ga je naučiti. Sa njim dobijate odlične performanse, čvrstu integrisanost sa takoreći svakom dostupnom bazom podataka, stabilnost, prenosivost i skoro neograničen skup funkcija zahvaljujući njegovoj proširivosti. Sve to ne zahteva nikakve troškove (PHP je softver sa otvorenim kodom) i omogućava prilagodljivu krivu učenja. Povezujući lakoću sa kojom programeri početnici mogu da počnu da ga koriste i sposobnost da iskusni programeri sa njim urade sve što im je potrebno, PHP je jedna od najboljih kombinacija koje sam ikad video.

Konačno, dokaz je u upotrebi: otkako se PHP pojavio, njegovo korišćenje pokazuje eksponencijalni rast i predstavlja izabranu tehnologiju na preko 82% veb sajtova **B**. Što se tiče programskih jezika, PHP je šesti po popularnosti **C**.

nastavak na sledećoj strani

Jul 2017	Jul 2016	Change	Programming Language	Ratings	Change
1	1		Java	13.774%	-6.03%
2	2		C	7.321%	-4.92%
3	3		C++	5.576%	-0.73%
4	4		Python	3.543%	-0.62%
5	5		C#	3.518%	-0.40%
6	6		PHP	3.093%	-0.18%

C Tiobe indeks (**`https://www.tiobe.com/tiobe-index/`**) koristi kombinaciju faktora kako bi rangirao programske jezike po popularnosti.

Naravno, mogli biste da pomislite da sam ja, kao autor knjige o PHP-u (u stvari, nekoliko knjiga), pristrasan. Mada u manjoj meri nego PHP, ja sam za razvijanje sajtova koristio i JavaServerPages (JSP), Ruby on Rails (RoR), Sinatra (još jedna vrsta radnog okruženja za Ruby), i ASP.NET. Svaki od njih ima dobre i loše strane, ali je PHP tehnologija kojoj se uvek vraćam. Možda ćete čuti da su mu performanse ili mogućnosti skaliranja lošije u odnosu na druge tehnologije, ali Yahoo!, Wikipedia i Facebook koriste PHP, a teško ćete naći posećenije i zahtevnije sajtove od njih.

Možda ste čuli da je PHP manje bezbedan. Međutim, *bezbednost nije u jeziku* već u načinu na koji ga koristite, Budite sigurni da se u ovoj knjizi nalazi potpun i najsavremeniji opis relevantnih bezbednosnih pitanja.

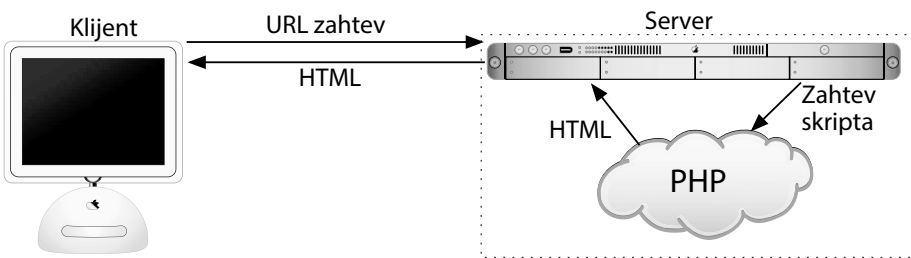
Kako PHP funkcioniše

Kako je već rečeno, PHP je jezik za serversku stranu. To znači da se kôd koji pišete u PHP-u nalazi na host računaru koji zovemo *server*. Server isporučuje veb stranice na zahtev posetilaca (vas, klijenta, sa vašim veb pretraživačem).

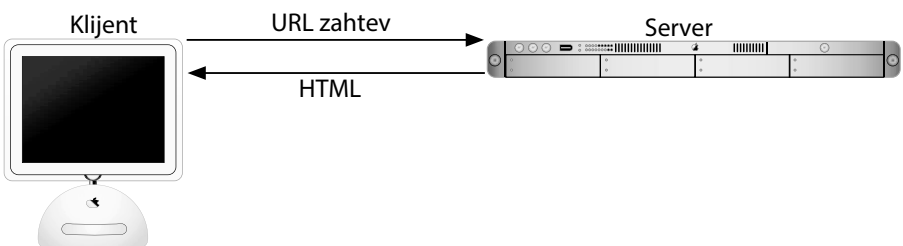
Kada posetilac stigne na veb sajt pisan u PHP-u, server pročita PHP kôd i zatim ga obradi prema skriptovanim uputstvima. U primeru na slici **D**, PHP kôd kaže serveru da pošalje odgovarajuće podatke – HTML kôd – veb pretraživaču, koji obrađuje primljeni kôd kao svaku standardnu HTML stranicu.

Tu je razlika u odnosu na statički HTML sajt gde na primljeni zahtev server samo pošalje HTML sadržaj veb pretraživaču i gde ne dolazi do interpretacije na serverskoj strani **E**. Pošto nema potrebe za aktivnošću servera, HTML stranice se izvršavaju u veb pretraživaču i server se uopšte ne koristi.

nastavak na sledećoj strani



D Kako se PHP uklapa u model klijent/server kada korisnik zatraži veb stranicu.



E Proces klijent/server kada se zahteva statična HTML stranica.

Za krajnjeg korisnika kao i za veb pretraživač, što se tiče izgleda nema uočljive razlike između **`home.html`** i **`home.php`**, ali način na koji se taj sadržaj pravi biće znatno drugačiji.

Šta je to MySQL?

MySQL (**`www.mysql.com`**) Ⓕ je najpopularniji sistem baza podataka sa otvorenim kodom na svetu. U stvari, danas MySQL ravnopravno konkuriše skupim gigantima kao što su Oracle i Majkrosoftov SQL Server (ironija je u tome što je MySQL zapravo vlasništvo Oracla). Poput PHP-a, MySQL nudi izvrsne performanse, prenosivost i pouzdanost, sa umerenom krivom učenja uz male troškove ili bez njih.

MySQL je sistem za upravljanje bazom podataka (DBMS – database management system) za relacione baze podataka (znači, MySQL je RDBMS). Baza podataka je, najjednostavnije rečeno, kolekcija podataka, koja može da sadrži tekstualne, numeričke ili binarne fajlove, koje čuva i organizuje DBMS.

Postoji mnogo vrsta baza podataka, od jednostavnog ravnog fajla do relacionih i objektno orijentisanih i do *NoSQL* baza podataka. Relaciona baza podataka u svojim najupadljivijim delovima koristi više tabela za čuvanje informacija. Mada je za relacione baze podataka potrebno više razmišljanja u fazama projektovanja i programiranja, time se dobija veća pouzdanost i integritet podataka što u velikoj meri nadoknađuje višak uloženog truda. Osim toga, relacione baze podataka se bolje pretražuju i dozvoljavaju pristup većem broju korisnika istovremeno.



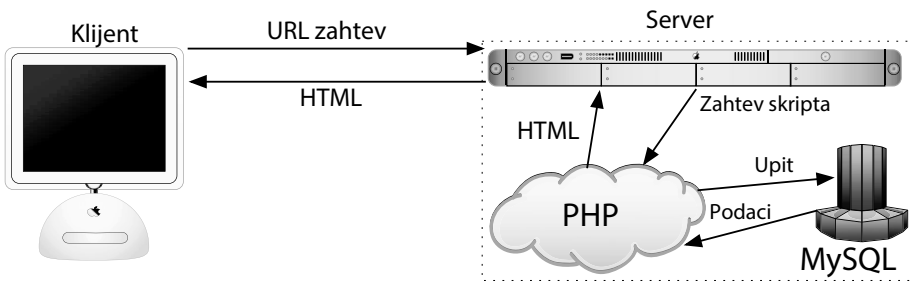
Ⓕ Početna stranica za aplikaciju baze podataka MySQL.

Kada se baza podataka ugradi u veb aplikaciju, neki podaci koje generiše PHP mogu se izdvojiti iz MySQL-a Ⓖ. Time se sadržaj sajta od statičke (ukodirane) osnove još više kreće ka fleksibilnom, gde je fleksibilnost ključna za dinamički veb sajt.

MySQL je aplikacija otvorenog koda, kao i PHP, što znači da je besplatan za korišćenje i čak za menjanje (može se preuzeti i sam izvorni kôd). Postoje slučajevi u kojima treba da se plati MySQL licenca, pogotovo ako zarađujete na prodaji ili na ugrađivanju MySQL proizvoda. Proverite politiku licenciranja MySQL-a gde se nalazi više informacija o tome.

MySQL softver se sastoji od više delova, uključujući MySQL server (*mysqld*, koji izvršava baze podataka i upravlja njima), MySQL klijent (*mysql*, koji vam daje interfejs prema serveru) i niz pomoćnih programa za održavanje kao i za druge svrhe. PHP je oduvek imao dobru podršku za MySQL, a to je još izraženije u novijim verzijama jezika.

nastavak na sledećoj strani



Ⓖ Ovako će funkcionisati većina dinamičkih aplikacija u ovoj knjizi, kada se koriste i PHP i MySQL.

MySQL je poznat po tome da može da upravlja bazama podataka sa 60.000 tabela i nekoliko milijardi redova. MySQL može da radi sa tabelama od čak nekoliko hiljada terabajta na pojedinim operativnim sistemima, a inače podržava dobrih 4 GB. MySQL, među ostalima, koriste NASA i uprava za popis SAD.

Dok ovo pišem, MySQL je na verziji 5.7.18. Koje funkcije ćete moći da koristite zavisi od toga koju verziju imate, zato je važno da znate sa čim radite. Za ovu knjigu smo koristili MySQL 5.7.14, mada bi trebalo da budete u stanju da uradite sve primere iz ove knjige ako imate verziju noviju od 5.0.

Uputstvo za izgovor

Mada zvuči trivijalno, naglasiću da se MySQL kaže „maj es kju el”, kao što se SQL kaže „es kju el”. To se pitanje često javlja kada ljudi počnu da koriste ove tehnologije. Mada nije bitno, uvek je najbolje da se akronimi pravilno izgovaraju.

Šta vam je sve potrebno

Da biste uradili primere iz ove knjige, potrebni su vam sledeći alati:

- Aplikacija veb servera (na primer, Apache, Nginx, ili IIS)
- PHP
- MySQL
- Veb pretraživač (Majkrosoftov Internet Explorer ili Edge, Mozilin Firefox, Apple-ov Safari, Google-ov Chrome i tako dalje)
- Tekst editor, WYSIWYG aplikacija podobna za PHP (Adobe-jev Dreamweaver završava posao), ili IDE (Integrated development environment – razvojno okruženje)
- Neka FTP aplikacija ako koristite udaljeni server

Jedna od odličnih stvari u razvijanju dinamičkih veb sajtova sa PHP-om i MySQL-om je to što se svi preduslovi mogu zadovoljiti sasvim besplatno, bez obzira na to koji operativni sistem koristite! Apache, PHP i MySQL su besplatni; veb pretraživači se dobijaju besplatno i mnogi dobri tekst editori mogu da se nabave besplatno.

U dodatku je opisan postupak instalacije na operativnim sistemima Windows i Mac. Ako imate računar, nedostaje vam samo nekoliko preuzimanja i moći ćete da pravite dinamičke veb sajtove (u tom slučaju bi vaš računar bio i klijent i server, kao što je prikazano na slikama D i E). Ili obratno, mogli biste da iznajmite prostor na Vebu za nekoliko dolara mesečno i tu bi vam PHP i MySQL već bili omogućeni onlajn.

O ovoj knjizi

Ova knjiga vas uči kako da razvijate dinamičke veb sajtove pomoću PHP-a i MySQL-a, obuhvatajući pritom materiju potrebnu većini programera. Prema formatu uobičajenom u seriji Visual QuickPro, informacije se opisuju korak po korak uz odgovarajuće slike. Sve je usredsređeno na praktične primere iz realnog života, i izbegavaju se scenariji koji su „mogući ali se nikad ne dešavaju". Pošto imam puno iskustva kao veb programer, pisao sam o informacijama koje koristim, a izbegavao teme koje su nebitne za zadatak. Kao praktičar i pisac, vodio sam računa da uključim teme i postupke za koje znam da ih traže čitaoci.

Struktura knjige je linearna i predviđena je da se čita redom. Počinje sa tri poglavlja koja obuhvataju osnove PHP-a (u drugom poglavlju ćete već dovršiti svoju prvu dinamičku veb stranicu). Pored toga, slede četiri poglavlja o SQL-u (Structured Query Language, koji se koristi za rad sa svim bazama podataka) i MySQL-u. U tim poglavljima, naučićete SQL, projektovanje baza podataka i posebno aplikaciju MySQL. Zatim sledi jedno poglavlje o traženju i ispravljanju grešaka kao i o upravljanju greškama, a to su informacije koje su svima potrebne. Naredno poglavlje predstavlja uvod u to kako se zajedno koriste PHP i MySQL, što je neverovatno jednostavno.

U sledećih pet poglavlja izlažu se dodatne tehnike primene kojima ćete zaokružiti svoje znanje. Bezbednost se na ovim stranicama neprestano razmatra. Dva nova poglavlja o kojima će odmah biti reči, proširiće vaše novostečeno znanje. Na kraju knjige sam uvrstio i dva poglavlja primera, u kojima se razvija osnova različitih aplikacija, sa uputstvima.

Da li je ovo knjiga za vas?

Ova knjiga je pisana za širok spektar ljudi u rasponu od početnika pa do onih čija su znanja srednjeg nivoa. U knjizi se koristi HTML5, pa je obavezno imati solidno iskustvo u radu sa HTML-om. Mada ova knjiga obuhvata mnoge stvari, ona nije namenjena za učenje HTML-a kao ni za učenje projektovanja veb stranica. Na nekim mestima se koristi CSS, ali se takođe o njemu ne uči.

Drugo, knjiga je pisana polazeći od pretpostavke da imate nešto od sledećeg:

- Želju i sposobnost za učenje bez mnogo vođenja za ruku, ili...
- Poznavanje nekog drugog programskog jezika (odgovaralo bi i solidno znanje JavaScript-a), ili...
- Površno poznavanje PHP-a

Nemojte imati dileme: u ovoj knjizi su PHP i MySQL obrađeni od početka pa do kraja, objašnjeno je sve što treba da znate da biste razvijali veb sajtove iz realnog života, ali se u prvim poglavljima PHP obrađuje velikom brzinom. Iz tog razloga preporučujem bilo programersko iskustvo, bilo radoznalost i samostalnost u učenju novih stvari. Ako smatrate da je tempo brz, verovatno bi trebalo da počnete sa mojom knjigom *PHP for the World Wide Web: Visual QuickStart Guide* u kojoj se izlaže mnogo umerenijom brzinom.

Iskustvo sa bazama podataka nije uopšte potrebno, pošto se SQL i MySQL objašnjavaju od sasvim početnog nivoa.

Šta je novo u ovom izdanju

Prva četiri izdanja ove knjige su bila veoma popularna i primio sam mnogo pozitivnih povratnih informacija o njima (hvala!). Radeći na ovom novom izdanju, fokusirao sam se na to da materijal bude tačan, ažuriran i u skladu sa današnjim standardima i najboljom praksom. Izmene u ovom izdanju obuhvataju sledeće:

- Ažuriranje celokupnog koda tako da može da koristi HTML5
- Upotreba modernijih tehnika za dizajniranje HTML-a, uključujući višestruke primere za radno okruženje Twitter Bootstrap
- Ažuriranje svega što se odnosi na najnoviju verziju PHP-a i MySQL-a
- Dodatni primeri sa PHP-om i MySQL-om. Kao što je izvođenje transakcija iz PHP skripta
- Još više informacija i primera koji unapređuju bezbednost vaših skriptova i sajtova
- Uklanjanje zastarelog sadržaja (na primer, stvari koje su korišćene u starijim verzijama PHP-a ili više ne mogu da se primene)
- Dodatak posvećen instalaciji ponovo se nalazi u štampanoj verziji knjige (u četvrtom izdanju ovaj dodatak je bio dostupan onlajn)

Za one od vas koji već imaju prethodno izdanje (hvala, hvala, hvala!), i nadam se da će vam ovo ažurirano izdanje biti odličan izvor za učenje kao i prethodno.

Poređenje ove knjige sa mojim ranijim knjigama

Ovo je moja četvrta knjiga sa tematikom koja se odnosi na PHP, odnosno MySQL. Prethodne su bile (redom):

- *PHP for the World Wide Web: Visual QuickStart Guide*
- *PHP Advanced and Object-Oriented Programming: Visual QuickPro Guide*
- *MySQL: Visual QuickStart Guide*

Nadam se da ovaj spisak ukazuje na to da imam određen stepen kvalifikovanosti za pisanje ove knjige. Ali kako da vi, kao čitalac koji stoji u prodavnici, odlučite koji naslov je za vas? Naravno, ako želite da se razbacujete parama, izvolite i kupite ceo komplet, biću vam večno zahvalan, ali...

Knjiga *PHP for the World Wide Web: Visual QuickStart Guide* je pre svega namenjena onima koji tek počinju da koriste PHP. U sadašnjoj knjizi se, uglavnom u prva tri poglavlja, materija preklapa sa tom knjigom ali su ovde dati novi primeri kako bi se izbegla redundantnost. Za početnika, ova knjiga može da se uzme kao nastavak prethodne. *PHP Advanced* predstavlja nastavak ove knjige, jer se u njoj pretpostavlja poprilično dobro znanje o materiji koja se ovde tek objašnjava. Knjiga MySQL je skoro isključivo usredsređena na MySQL (samo u dva poglavlja se koristi PHP).

nastavak na sledećoj strani

Imajući sve to na umu, pročitajte odeljak „Da li je ovo knjiga za vas?" da biste videli da li ispunjavate uslove. Ako nemate nikakvo programersko iskustvo i želite natenane da učite PHP, bolje će biti da uzmete moju prvu knjigu. Ako ste već navikli na PHP, pa vas zanimaju naprednije mogućnosti, uzmite knjigu *PHP Advanced and Object-Oriented Programming: Visual QuickPro Guide*. Ako vas pre svega zanima MySQL, pa vas PHP mnogo ne zanima, pogledajte knjigu *MySQL: Visual QuickStart Guide*.

Konačno, ako želite da naučite sve što je potrebno da biste danas počeli da pravite dinamičke veb sajtove sa PHP-om i MySQL-om, onda je ova knjiga za vas! Ona se odnosi na najnovije verzije obe tehnologije, koristi tehnike koje prethodno nisu opisane u drugim knjigama i sadrži vlastite jedinstvene primere.

Koju god knjigu da izaberete, vodite računa o tome da uzmete najnovije izdanje, ili svakako ono izdanje koje je najprimerenije verzijama tehnologija koje ćete vi da koristite.

Veb sajt koji je posvećen ovoj knjizi

Napravio sam jedan veb sajt posebno za ovu knjigu, i možete ga pronaći na adresi **`www.LarryUllman.com`**. Na njemu možete pronaći sve skriptove iz ove knjige, jedan tekstualni fajl u kojem se nalaze dugačke SQL komande, i jedan spisak grešaka (errata) do kojih je došlo tokom izdavanja knjige. (Ako imate problema sa nekom komandom ili skriptom, a tačno ste ga kopirali iz knjige, dok niste sasvim poludeli proverite u spisku grešaka da li je reč o štamparskoj grešci). Na ovom veb sajtu ćete naći i korisne linkove, jedan popularni forum na kojem čitaoci mogu da postavljaju pitanja i odgovaraju na tuđa (na mnoga pitanja odgovaram i ja) i još mnogo toga!

Pitanja, komentari ili predlozi?

Ako imate neko pitanje u vezi sa PHP-om ili MySQL-om, možete da pogledate na nekom od više veb sajtova, listi za slanje elektronske pošte, diskusionih grupa i već postojećih često postavljenih pitanja (FAQ). Brzom onlajn pretragom naći ćete praktično beskonačan izvor odgovora. Što se toga tiče, ako vam je potreban trenutni odgovor, takvi izvori ili onlajn pretraga će skoro sigurno zadovoljiti vaše potrebe (veoma je verovatno da je još neko imao i rešio problem identičan vašem).

Takođe, možete meni da uputite svoja pitanja, komentare i predloge. Najbrži odgovor ćete dobiti preko foruma koji je posvećen ovoj knjizi (uvek prvo odgovaram na ta pitanja). Ako više volite e-poštu, moje kontakt informacije su na mom veb sajtu. Pokušavam da odgovorim na svaku e-poruku koju primim, mada ne mogu da garantujem brz odgovor.

1

Uvod u PHP

Iako se ova knjiga bazira na zajedničkoj upotrebi PHP-a i MySQL-a, za veći deo rada koristićete samo PHP. U ovom kao i u narednom poglavlju, učićete osnove, od sintakse do promenljivih, operatora i jezičkih konstrukcija (uslova, petlji i još mnogo toga). Istovremeno ćete praviti upotrebljiv kôd koji kasnije možete da koristite i pri kreiranju većih aplikacija.

Ovo uvodno poglavlje vas vodi kroz veći deo osnova jezika PHP. Pored toga, učićete sintaksu PHP kodiranja, kako da šaljete podatke veb pretraživaču i kako da koristite dve vrste promenljivih (stringove i brojeve) kao i konstante. Možda vam se neki od datih primera učine irelevantnim, ali će vam oni na najbolji način prikazati pojmove koje treba da savladate kako biste kasnije pisali naprednije skriptove. Na kraju ovog poglavlja, u slučaju da vam zatreba, možete pronaći savete koji se odnose na brzo otkrivanje i uklanjanje grešaka.

U ovom poglavlju

Osnovna sintaksa 2

Slanje podataka u veb pretraživač 6

Pisanje komentara 10

Šta su to promenljive? 14

Uvođenje stringova 18

Nadovezivanje stringova 21

Uvođenje brojeva 23

Uvođenje konstanti 26

Jednostruki i dvostruki navodnici 29

Osnovni koraci za otkrivanje i uklanjanje grešaka 32

Pregled naučenog i unapređivanje stečenih veština 34

Osnovna sintaksa

Kao što je rečeno u uvodnom delu knjige, PHP predstavlja skript jezik koji se *ugrađuje u HTML*, što znači da se PHP kôd kao i HTML kôd mogu zajedno koristiti u istom fajlu. Dakle, da biste počeli da programirate pomoću PHP-a, počnite od jednostavne veb stranice. Skript 1.1 je primer HTML5 dokumenta bez suvišnih ukrasa i bez sadržaja, koji će se koristiti kao polazna tačka prilikom kreiranja većine veb stranica u knjizi (U ovoj knjizi se HTML5 ne objašnjava; više informacija potražite u dodatnim izvorima posvećenim toj temi). Takođe, obratite pažnju da se u ovom šablonu koristi kodiranje UTF-8, koje je detaljnije opisano u komentaru koji sledi.

Da biste veb stranici dodali PHP kôd, stavite ga među PHP oznake:

```
<?php
?>
```

Skript 1.1 osnovna veb stranica HTML5

```
<   !doctype html>
2   <html lang="en">
3   <head>
4      <meta charset="utf-8">
5      <title>Page Title</title>
6   </head>
7   <body>
8      <!-- Script 1.1 - template.html -->
9   </body>
10  </html>
```

Uloga kodiranja

Kodiranje je opširna tema, ali je najbitnije da obratite pažnju na sledeće: *kodiranje koje koristite u datoteci zapravo vam propisuje koji znakovi* mogu da se predstave (pa, prema tome, i koje jezike možete da koristite). Prilikom odabira kodiranja, najpre treba da se uverite da vaš tekst editor ili IDE (Integrated Development Environment – integrisano okruženje za razvoj)– bilo koja aplikacija koju koristite za pisanje HTML i PHP skriptova – može da sačuva dokumente sa tim kodiranjem. U nekim aplikacijama možete da postavite tip kodiranja u podešavanjima ili delu sa opcijama, dok se kod drugih kodiranje podešava prilikom čuvanja datoteke.

Da biste veb pretraživaču ukazali na vrstu kodiranja, koristite sledeću **meta** oznaku:

```
<meta charset="utf-8">
```

Element *charset=utf-8* ukazuje nam na korišćenje kodiranja UTF- 8, što ujedno i predstavlja skraćenicu za *8-bit Unicode Transformation Format*. Unicode nam omogućava da pouzdano predstavimo svaki simbol u svakom alfabetu. Unicode verzija 9.0.0 – trenutno važeća verzija dok ovo pišem – podržava preko 128.000 znakova!

Ako nameravate da kreirate višejezičnu veb stranicu, treba da koristite UTF-8, a ja ću ga koristiti u primerima u ovoj knjizi. Naravno, to nije neophodno. Međutim, koje god kodiranje da odaberete, obratite pažnju da kodiranje označeno na HTML stranici odgovara kodiranju podešenom u tekst editoru ili okruženju za razvoj (IDE). U suprotnom, verovatno ćete ugledati neke čudne znakove kada budete prikazali stranicu u svom veb pretraživaču.

Skript 1.2 ovaj prvi PHP skript ništa ne radi, ali se u njemu vidi kako se piše PHP skript. Koristiće se i za testiranje PHP skripta pre prelaska na razrađen PHP kôd.

```
1   <!doctype html>
2   <html lang="en">
3   <head>
4      <meta charset="utf-8">
5      <title>Basic PHP Page</title>
6   </head>
7   <body>
8      <!-- Script 1.2 - first.php -->
9      <p>This is standard HTML.</p>
10  <?php
11  ?>
12  </body>
13  </html>
```

Veb server će sve što se upiše između tih oznaka prepoznavati kao PHP. Što znači da će PHP interpretator obraditi taj kôd. Celokupan tekst izvan PHP oznaka se odmah šalje veb pretraživaču kao običan HTML. S obzirom na činjenicu da se PHP najčešće koristi za kreiranje sadržaja koji će biti prikazan u veb pretraživaču, PHP oznake se obično postavljaju na odgovarajućim mestima unutar tela stranice.

Osim postavljanja PHP koda unutar PHP oznake, vaše PHP datoteke moraju da imaju i odgovarajuće ekstenzije. Ekstenzija zapravo samom serveru skreće pažnju da je potrebno taj skript obraditi na poseban način, što znači kao PHP stranicu. Većina veb servera koristi ekstenziju **`.html`** za standardne HTML stranice, a **`.php`** za PHP stranice.

Pre nego što nastavite rad po uputstvu koje sledi, morate već da imate *PHP instalaciju u funkciji!* Ona se može nalaziti na nekom zakupljenom sajtu ili na vašem računaru, kao što je navedeno u Dodatku A, „Instaliranje."

Kako biste kreirali osnovni PHP skript, potrebno je da uradite sledeće:

1. Kreirajte nov dokument u svom tekst editoru ili razvojnom okruženju. Dodelite mu naziv **`first.php`** (**Skript 1.2**).

   Uglavnom nije važno koju aplikaciju koristite. To može biti Adobe Dreamweaver (veoma elegantan IDE), Sublime Text (odličan i veoma popularan editor teksta), ili vi (Unix editor čistog teksta, bez grafičkog interfejsa). Sa druge strane, neki tekst editori i IDE okruženja olakšavaju unošenje i čišćenje HTML i PHP sadržaja (za razliku od Notepada u Windowsu koji radi neke stvari koje otežavaju kodiranje: *nemojte koristiti Notepad!*). Ako još uvek

nastavak na sledećoj strani

nemate instaliranu aplikaciju koju rado koristite, možete je pronaći na vebu ili se raspitajte na forumu koji je posvećen ovoj knjizi (www.LarryUllman.com/forums/)

2. Kreirajte osnovni HTML dokument:

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Basic PHP Page</title>
</head>
<body>
  <!-- Script 1.2 - first.php -->
  <p>This is standard HTML.</p>
</body>
</html>
```

   Ovo je zapravo osnovna HTML5 stranica. Jedan od kvaliteta HTML5 leži u njegovoj jednostavnosti koja se odnosi na doctype kao i na sintaksu.

3. Pre završne oznake **body** ubacite PHP oznake:

```
<?php
?>
```

   To su zvanične PHP oznake, koje su poznate i pod imenom oznake *tipa XML*. Iako PHP podržava i druge vrste oznaka, moja preporuka je da koristite ipak zvanične oznake, a ja ću ih koristiti u celoj knjizi.

4. Sačuvajte datoteku pod imenom **first.php**.

   Obratite pažnju na činjenicu da se skript neće pravilno izvršavati ako ne sačuvate datoteku sa odgovarajućom PHP ekstenzijom. (Jedan od razloga zašto ne treba da se koristi Notepad leži u činjenici da Notepad PHP datoteci neprimetno dodaje ekstenziju **.txt**, što može dovesti do velikih problema).

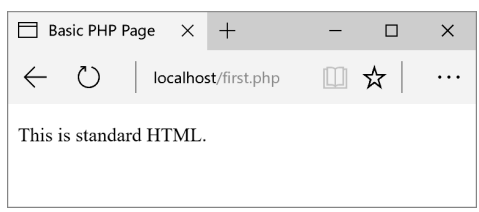
5. Smestite datoteku u odgovarajući direktorijum svog veb servera.

   Ako izvršavate PHP na vašem računaru (recimo, nakon instaliranja po uputstvima iz Dodatka A), dovoljno je da premestite, kopirate ili kreirate svoju datoteku u određenom folderu na vašem računaru. Pogledajte Dodatak A ili dokumentaciju svog konkretnog veb servera kako biste utvrdili koji je pravi direktorijum, ako još ne znate koji je.

   Sa druge strane, ako izvršavate PHP na zakupljenom serveru (tj. na udaljenom računaru), biće vam potrebna FTP aplikacija (File Transfer Protocol) kako biste postavili datoteku u odgovarajući direktorijum. Kompanija kod koje iznajmljujete serverske kapacitete dostaviće vam potrebne parametre za pristup.

6. Izvršite **first.php** u svom veb pretraživaču Ⓐ

   Pošto server mora da raščlani (engl. *parse*) svaki PHP skript, *neophodno je* da mu pristupate putem URL-a (tj. adresa u pretraživaču mora da počinje sa **http://** ili **https://**). Ne možete prosto da ga otvorite u pretraživaču, kao što biste otvorili datoteku drugih aplikacija (u tom slučaju bi adresa počinjala sa **file://** ili **C:** ili slično tome).



Ⓐ Iako izgleda kao svaka druga (jednostavna) HTML stranica, ovo je zapravo PHP skript koji predstavlja polaznu osnovu za ostale primere u knjizi.

Ako PHP izvršavate na vašem računaru, moraćete da koristite neke od sledećih URL-ova :http://localhost/first.php, **`http://127.0.0.1/first.php`** ili **`http://localhost/-<user>/first.php`** (U slučaju da koristite macOS, potrebno je da umesto <user> upotrebite svoje stvarno korisničko ime). Međutim, ako koristite zakupljeni veb server, potrebno je da koristite: **`http://vaše-ime-domena/first.php`** (npr. **`http://www.example.com/first.php`**).

7. U slučaju da ne vidite rezultat kakav je na slici A, počnite da tražite grešku!

   U učenje bilo kojeg programskog jezika spada i savladavanje otkrivanja grešaka. Taj proces je ponekad težak ali je apsolutno neophodan. Ako u ovom prvom primeru ne dobijete jednostavnu ali potpuno ispravnu veb stranicu, potrebno je da uradite sledeće:

   A. Proverite da li imate aktivnu PHP instalaciju (U dodatku A imate uputstva kako da to proverite).

   B. Proverite da li ste skript pokrenuli preko URL-a. Adresa u veb pretraživaču mora da počinje sa **`http`**. Međutim, ako počinje sa **`file://`**, to je uzrok samog problema B.

   C. Ako dobijete poruku koja vas obaveštava da datoteka nije pronađena („file not found”) ili nešto slično, verovatno ste postavili datoteku u pogrešan direktorijum ili ste pogrešno otkucali ime datoteke (ili tokom pohranjivanja ili u samom veb pretraživaču).

   Ako ste sve ovo proverili, a još uvek imate probleme, posetite stranicu foruma koja je posvećena ovoj knjizi (**`www.LarryUllman.com/forums/`**).

SAVET Više informacija o HTML-u možete da pronađete u izuzetnoj knjizi Elizabete Castro pod imenom: *HTML and CSS: Visual QuickStart Guide* (Peachpit, 2013) ili vam preostaje da tražite po Vebu.

SAVET Moguće je u isti HTML dokument ubaciti više segmenata PHP koda (tj. Možete da prelazite iz jednog jezika u drugi). Čitajući ovu knjigu susretaćete se sa takvim primerima.

SAVET Kodiranje eksterne CSS datoteke možete deklarisati tako što ćete u prvom redu datoteke upisati `@charset "utf-8"`. U slučaju da ne koristite UTF-8, potrebno je da promenite taj red na odgovarajući način.



B PHP kôd može da se izvrši samo kroz http://

Slanje podataka u veb pretraživač

Da biste kreirali dinamične veb sajtove pomoću PHP-a, morate znati kako da podatke pošaljete veb pretraživaču. PHP ima niz ugrađenih funkcija koje se koriste za tu namenu. Uobičajene su **`echo`**`print`. Ja lično više volim **`echo`**.

```
echo 'Hello, world!';
echo "What's new?";
```

Ako više volite, mogli biste da upotrebite **`print`** (to ime bolje ukazuje na to šta sama funkcija radi):

```
print 'Hello, world!';
print "What's new?";
```

Na osnovu ovih primera, možemo izvesti zaključak da je moguće koristiti jednostruke ili dvostruke navodnike (mada se te dve vrste navodnika razlikuju, što će biti detaljnije objašnjeno kasnije u ovom poglavlju). Prvi navodnik nakon imena funkcije označava početak poruke koja treba da se štampa. Sledeći navodnik koji se slaže (tj. sledeći navodnik iste vrste kao početni navodnik) označava kraj poruke koja treba da se štampa.

Uporedo sa učenjem kako se podaci šalju u veb pretraživač, potrebno je obratiti pažnju na to da u PHP-u sve naredbe – laički rečeno, svaki red izvršnog koda – treba završiti znakom tačka zarez. Osim toga, PHP ne pravi razliku između velikih i malih slova kada su u pitanju imena funkcija. Shodno tome, potpuno je svejedno da li ćete napisati **`ECHO`**, **`echo`** ili **`eCHo`**. Naravno, najlakše je kad pišete sve malim slovima.

Potreba da se izbegne specijalno značenje

Ubrzo ćete videti da jedan od problema prilikom slanja podataka pretraživaču predstavlja pitanje kako da mu se pošalje jednostruki ili dvostruki navodnik. Upotreba ove dve varijante dovešće do greške:

```
echo "She said, "How are you?"";
echo 'I'm just ducky.';
```

Za ovaj problem postoje dva rešenja. Prvo je da se koriste jednostruki navodnici kada se šalje dupli navodnik i obratno:

```
echo 'She said, "How are you?"';
echo "I'm just ducky.";
```

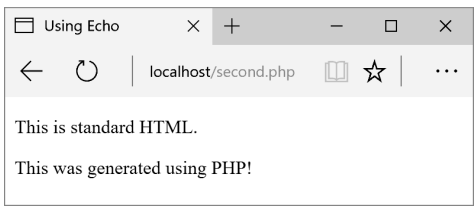
Drugo rešenje podrazumeva izbegavanje specijalnog značenja navodnika (koristeći obrnutu kosu crtu \):

```
echo "She said, \"How are you?\"";
echo 'I\'m just ducky.';
```

Poništeni navodnik će biti štampan kao i svaki drugi znak. Važno je shvatiti značaj upotrebe obrnute kose crte kojom se izbegava specijalno značenje. Do kraja ovog poglavlja biće još reči o tome.

Skript 1.3 komandama **echo** ili **print** PHP šalje podatke u veb pretraživač.

```
1   <!doctype html>
2   <html lang="en">
3   <head>
4      <meta charset="utf-8">
5      <title>Using Echo</title>
6   </head>
7   <body>
8      <!-- Script 1.3 - second.php -->
9      <p>This is standard HTML.</p>
10  <?php
11  echo 'This was generated using PHP!';
12  ?>
13  </body>
14  </html>
```



Ⓐ Rezultati još uvek nisu sjajni, ali ova stranica je jednim svojim delom ipak kreirana od strane PHP-a.

Da biste poslali podatke u veb pretraživač

1. Otvorite first.php (to je Skript 1.2) u tekst editoru ili IDE okruženju.
2. Između PHP oznaka (redovi 10 i 11) dodajte jednostavnu poruku (**Skript 1.3**):

   echo 'This was generated using → PHP!';

   U stvari, sasvim je nevažno kakvu poruku unosite, koju funkciju ili koje navodnike koristite – samo vodite računa o tome da li poruka sadrži jednostruke ili dvostruke navodnike (pogledajte komentar „Potreba da se izbegne specijalno značenje").
3. Ako želite, promenite naslov stranice kako biste na taj način bolje definisali skript (red broj 5).

   <title>Using Echo</title>

   Ova izmena utiče samo na naslovnu liniju u prozoru pretraživača.
4. Sačuvajte datoteku pod imenom **second.php** u svom veb direktorijumu i testirajte je u svom pretraživaču Ⓐ.

   Nemojte izgubiti iz vida činjenicu da svi PHP skriptovi moraju da se izvršavaju kao URL (**http://nešto**) !

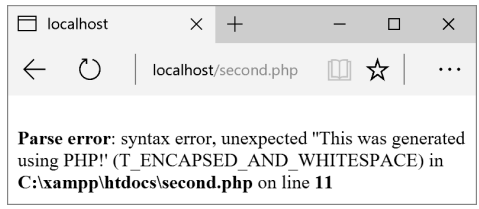
Nastavak na sledećoj strani

5. Ako je to neophodno, pokušajte da potražite grešku u skriptu. Ako vam se javi greška raščlanjivanja (engl. *parse error*) B, proverite da li ste otvorili i zatvorili navodnike i stavili obrnutu kosu crtu ispred svakog problematičnog znaka (pročitajte komentar na tu temu). Takođe, proverite da li ste svaku naredbu završili znakom tačka zarez. Ako dobijete praznu belu stranicu, to bi moglo da bude iz jednog od sledeća dva razloga:
   - Problem je u HTML-u. Pregledajte HTML naredbe u izvornom kodu vaše stranice C.
   - Došlo je do greške, ali je prilikom konfigurisanja PHP-a isključena opcija za prikazivanje grešaka (display_errors) i zato se ništa ne prikazuje. U tom slučaju, pročitajte odeljak o konfigurisanju PHP-a koji se nalazi u Dodatku A, kako biste ponovo uključili opciju display_errors.

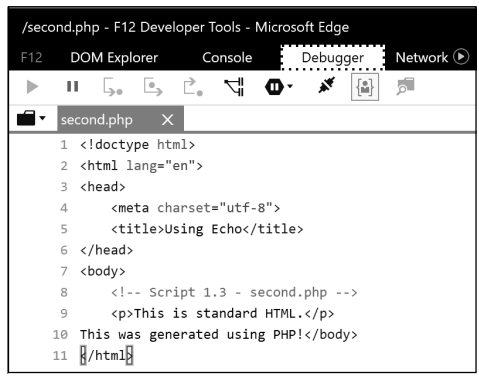
SAVET Sa tehničkog aspekta, `echo` i `print` su jezičke konstrukcije a ne funkcije u pravom smislu. Stoga, nemojte da vas buni što ih i dalje iz praktičnih razloga nazivam „funkcijama." Osim toga, videćete kasnije u knjizi da dodajem zagrade kada opisujem funkcije – na primer `number_format()` a ne prosto `number_format` kako bi se funkcije lakše razlikovale od promenljivih kao i od drugih elemenata PHP-a. To je prosto moja lična odluka.

SAVET Za slanje HTML koda u pretraživač moguće je, što ćete svakako i sami često raditi, koristiti `echo` i `print`, i to na sledeći način D:

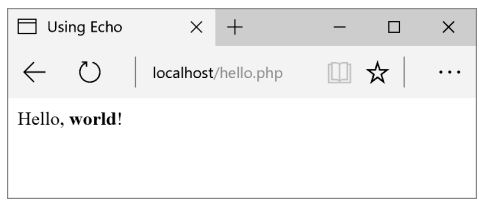
```
echo '<p>Hello,
→ <strong>world</strong>!</p>';
```



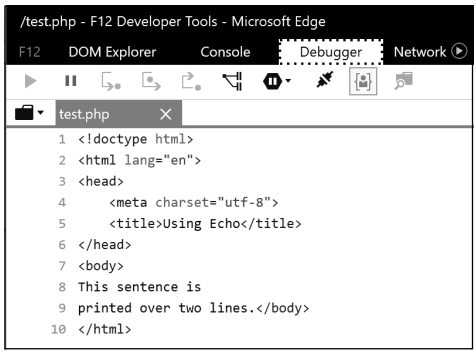
B Ovo bi mogla da bude prva od mnogih poruka o greškama raščlanjivanja sa kojima ćete se susretati kao PHP programer (razlog njenog pojavljivanja jeste nedostatak završnog znaka navoda).



C Jedan od razloga pojavljivanja prazne PHP stranice može da bude jednostavna HTML greška, kao što je ovde završna oznaka **`title`** (nedostaje kosa crta).



D PHP može u pretraživač da pošalje I HTML kôd (u našem slučaju to je kôd za formatiranje) a ne samo običan tekst A.



E Kada pišete tekst I HTML u više PHP redova, dobićete izvorni HTML kôd takođe u više redova. Imajte na umu da višak razmaka u izvornom HTML-u ne utiče na krajnji izgled same stranice **F** ali sa duge strane pomaže da izvorni kôd bude pregledniji.



F Prelaz u novi red u HTML izvoru **E** nema uticaja na krajnji izgled. Jedini način da se postigne prelaz u novi red jeste upotreba HTML oznake (na primer, **`<br>`** ili **`<p></p>`**).

SAVET **Echo i `print` mogu da se pišu u više redova:**

```
echo 'Ova rečenica se
ispisuje u dva reda.';
```

U ovom slučaju će prelaz u novi red (dobijen pritiskom na taster Enter ili Return) postati sastavni deo štampane poruke, koja se završava tek na završnom znaku navoda. Rezultat će biti „štampanje" prelaza u novi red u izvornom HTML kodu E. To neće uticati na generisanu stranicu F. Opširnije objašnjenje ove teme nalazi se u komentaru „Uloga razmaka".

Pisanje komentara

Kreiranje izvršnog PHP koda predstavlja samo deo programiranja (nesumnjivo, to je i najvažniji deo). Manje važan, ali ipak bitan aspekt svakog programerskog poduhvata jeste i dokumentovanje vašeg koda. U HTML-u se komentari dodaju pomoću specijalnih oznaka:

```
<!—Ovde dolazi komentar. -->
```

HTML komentari su vidljivi u izvornom kodu ali se ne vide u gotovoj stranici (pogledajte slike E i F u prethodnom odeljku).

PHP komentari se razlikuju po tome što se uopšte ne šalju u veb pretraživač, što znači da ih neće videti krajnji korisnik, čak i kad bude gledao izvorni HTML.

PHP podržava tri vrste sintaksi za komentare. Prva je simbol za brojeve, poznat i kao hešili taraba (**#**):

```
# Ovo je komentar
```

Drugi tip sintakse predstavljaju dve kose crte:

```
// Ovo je takođe komentar
```

U oba slučaja PHP ignoriše sve nakon toga pa do kraja reda (kada pritisnete taster Enter ili Return). Dakle, ova dva znaka se koriste samo za pojedinačni red. Oni se često koriste da bi se komentar stavio u isti red sa PHP kodom:

```
print 'Hello!'; // Say hello.
```

Treći način omogućava pisanje komentara u više redova:

```
/* Ovo je dugačak komentar koji
obuhvata više redova. */
```

Uloga razmaka

Vi pomoću PHP-a šaljete podatke (kao što su HTML oznake i tekst) u veb pretraživač, a on zatim pretvara to u veb stranicu koju krajnji korisnik vidi. Na taj način vi pomoću PHP-a često pravite izvorni HTML kôd jedne veb stranice. Uzevši to u obzir, postoje tri područja gde je prazan prostor (dodatni razmaci, tabulatori i prazni redovi) očigledan: u vašem PHP skriptu, u vašem izvornom HTML-u i u pripremljenoj veb stranici.

PHP uglavnom ne reaguje na prazan prostor, što znači da možete unutar koda da pravite razmake kako to vama odgovara, i na taj način sam kôd bi bio čitljiviji. Pored toga, HTML uglavnom ne reaguje na prazan prostor. Konkretno, jedini prazan prostor u HTML-u koji utiče na konačan izgled veb stranice jeste zapravo jednostruki razmak (ako ih stavite nekoliko, opet dobijate jedan). Ako u izvornom HTML-u imate tekst koji obuhvata više redova, to ne znači da će na gotovoj stranici biti više redova (E i F).

Da biste dobili prazan prostor na gotovoj veb stranici, upotrebite HTML oznake **
** (novi red) i ** ** (pasus). Da biste dobili prazan prostor u izvornom HTML-u koji će PHP napraviti, možete da:

- Upotrebite **echo** ili **print** u više redova.

  ili

- Pošaljete znak za novi red (**\n**) unutar dvostrukih navodnika, što je ekvivalentno korišćenju tastera Enter ili Return.

Skript 1.4 Ovi osnovni komentari dati su u tri moguće sintakse za komentar koje mogu da se koriste za PHP.

```
1   <!doctype html>
2   <html lang="en">
3   <head>
4      <meta charset="utf-8">
5      <title>Comments</title>
6   </head>
7   <body>
8   <?php
9
10  # Script 1.4 - comments.php
11  # Created March 16, 2011
12  # Created by Larry E. Ullman
13  # This script does nothing much.
14
15  echo '<p>This is a line of text.<br>This
    is another line of text.</p>';
16
17  /*
18  echo 'Ovaj red se neće
    izvršiti.';
19  */
20
21  echo "<p>Now I'm done.</p>";
    // Kraj PHP koda.
22
23  ?>
24  </body>
25  </html>
```

Da biste dodali komentare u svoj skript, potrebno je da uradite sledeće:

1. Otvorite u tekst editoru ili u IDE okruženju nov PHP dokument pod imenom **comments.php**, ali će on u početku sadržati samo HTML kôd (**Skript 1.4**):

   ```
   <!doctype html>
   <html lang="en">
   <head>
     <meta charset="utf-8">
     <title>Comments</title>
   </head>
   <body>
   ```

2. Dodajte početnu PHP oznaku i upišite svoje prve komentare:

   ```
   <?php
   # Script 1.4 - comments.php
   # Created April 23, 2017
   # Created by Larry E. Ullman
   # This script does nothing much.
   ```

   Jedan od prvih komentara u svakom skriptu trebalo bi da sadrži uvodni blok sa datumom izrade, datumom izmene, autora, informacije kako stupiti u kontakt sa autorom, namenu skripta i tako dalje. Neki smatraju da se sintaksa sa znakom **#** (stil koji se koristi u ljusci – shell) najbolje uočava i da je upravo zbog toga najprimerenija za ovu vrstu komentara.

3. Pošaljite sledeći HTML u pretraživač:

   ```
   echo '<p>This is a line of text.
   → <br>This is another line of
   → text.</p>';
   ```

   Nije važno šta ćete ovde da uradite, već je bitno da to bude nešto što će pretraživač da prikaže. Radi raznovrsnosti, naredba **echo** šalje i neke HTML oznake, uključujući znak za novi red (**
**) kako bi se na HTML stranici dobio određeni razmak.

Nastavak na sledećoj strani

4. Kreirajte komentar sastavljen od više redova kako biste pretvorili drugu naredbu **echo** u komentar:

```
/*
echo Ovaj red se neće
→izvršiti.';
*/
```

Kada neki blok PHP koda uokvirite znacima **/*** i ***/**, vi ga na taj način izuzimate od izvršenja a da pritom ne morate da ga brišete iz skripta. Ako kasnije uklonite ove znakove, taj kôd će ponovo biti aktivan.

5. Na kraju, dodajte komentar iza treće **echo** naredbe:

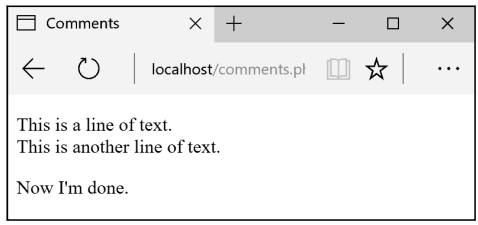
```
echo "<p>Now I'm done.</p>";
→// Kraj PHP koda.
```

Ovaj poslednji (suvišan) komentar upotrebljen je u svrhu prikazivanja kako se komentar stavlja na kraj reda, što je često i praksa. Obratite pažnju na to da su za poruku upotrebljeni dvostruki navodnici, jer bi jednostruki bili u konfliktu sa apostrofom koji poruka sadrži (pročitajte komentar „Potreba da se izbegne specijalno značenje", koji je dat u ranijem delu ovog poglavlja).

6. Zatvorite PHP odeljak i dovršite HTML stranicu:

```
?>
</body>
</html>
```

7. Sačuvajte fajl pod imenom **comments.php** u svom veb direktorijumu i proverite kako funkcioniše u veb pretraživaču A.



A PHP komentari u skriptu 1.4 ne pojavljuju se na veb stranici niti u izvornom HTML-u B

8. Ako ste po prirodi radoznali, pregledajte izvorni kôd u veb pretraživaču kako biste se uverili da se PHP komentari ne nalaze u njemu Ⓑ.

SAVET **Ne biste smeli da ugnežđujete (da postavljate jedan u drugi) komentare od više redova (/* */). To bi moglo da dovede do problema.**

SAVET **Moguće je postaviti bilo koju vrstu PHP komentara na kraj reda samog koda (na primer, iza poziva funkcije):**

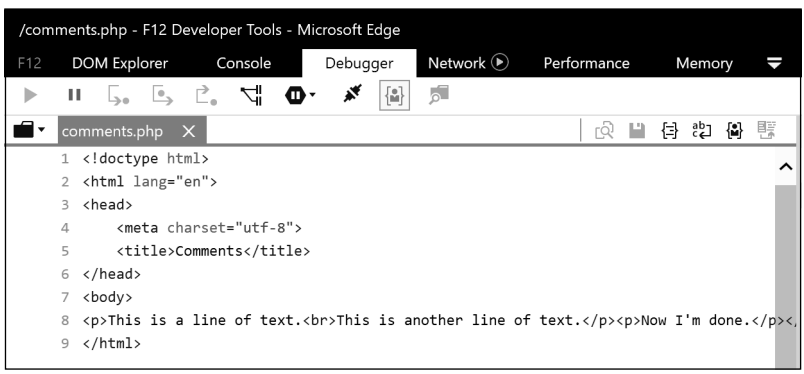
```
echo 'Howdy'; /* Pozdrav 'Howdy' */
```

Mada je dozvoljeno, ovo nije tako česta praksa.

SAVET **Radi uštede prostora, skriptovi u ovoj knjizi nisu tako detaljni kao što bih ja to želeo.**

SAVET **Takođe je bitno da prilikom promene skripta promenite i komentare, kako bi bili ažurni i tačni. Toliko je zbunjujuće kada se naiđe na komentar koji kaže jedno, a kôd radi nešto sasvim drugo.**

SAVET **Neki programeri smatraju da nije neophodno komentarisati pojedinačne delove koda zato što je sam kôd po sebi jasan. Ipak, na osnovu mog iskustva, smatram da dodavanje komentara pomaže.**



Ⓑ PHP komentari iz skripta 1.4 se nigde ne vide u klijentovom pretraživaču.

Šta su to promenljive?

Promenljive su kontejneri za privremeno čuvanje vrednosti. Te vrednosti mogu da budu brojevi, tekst ili mnogo složeniji podaci. PHP podržava osam tipova promenljivih. Postoje četiri skalarna tipa (pojedinačne vrednosti) – *Boolean* (sa vrednostima tačno **`TRUE`** i netačno **`FALSE`**), *integer*, *floating point* (sa decimalama) i *string* (jedan ili više znakova); dva neskalarna tipa (više vrednosti) – *array* i *object*; kao i *resource* (koje ćete videti prilikom rada sa bazama podataka) i *NULL* (što predstavlja specijalan tip koji nema vrednost).

Bez obzira na to koji tip promenljive želite da kreirate, u PHP-u postoje određena sintaksna pravila kojih se morate pridržavati a koja se odnose na imena svih promenljivih:

- Ime promenljive mora da počinje dolarskim znakom **`($)`**, na primer **`$ime`**.
- Dozvoljeno je da ime promenljive sadrži kombinaciju slova, brojeva i donje crte, na primer **`$my_report1`**.
- Prvi znak nakon dolarskog znaka mora da bude slovo ili donja crta (ne sme da bude broj).
- *Kod imena promenljivih u PHP-u postoji razlika između malih i velikih slova!* Ovo je veoma važno pravilo. To znači da su **`$ime`** i **`$Ime`** dve potpuno različite promenljive.

Kako biste počeli da radite sa promenljivama, sledeći skript će ispisati vrednosti tri *unapred definisane predefinisane promenljive* (engl. *predefined variables*). Dok se standardnim promenljivima vrednost dodeljuje tokom izvršavanja skripta, predefinisana promenljiva će već imati vrednost kada skript počne da se izvršava. Većina tih predefinisanih promenljivih odražava svojstva servera kao celine, kao što je, na primer, operativni sistem koji se koristi.

Pre nego što pređemo na ovaj skript, trebalo bi obratiti pažnju na još dve stvari. Kao prvo, promenljivama se vrednost može dodeliti pomoću znaka jednakosti (**`=`**), koji se takođe zove *operator dodeljivanja* (engl. *assignment operator*). Kao drugo, da biste prikazali vrednost promenljive, možete da štampate promenljivu bez znakova navoda:

```
print $neka_var;
```

ili možete da štampate promenljivu unutar dvostrukih navodnika:

```
print "Hello, $ime";
```

Vrednost promenljive ne može da se prikaže unutar jednostrukih navodnika:

```
print 'Hello, $name';
→// neće uspeti!
```

Da biste koristili promenljive, potrebno je da uradite sledeće:

1. Započnite u svom tekst editoru ili IDE okruženju nov PHP dokument, čije ime će biti **predefined.php** i počnite sa osnovnim HTML-om (**Skript 1.5**)

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Predefined Variables</
title>
</head>
<body>
```

2. Dodajte početnu PHP oznaku i prvi komentar

```
<?php # Script 1.5 - predefined.php
```

Od sada pa nadalje, u skriptovima se neće navoditi autor, datum nastanka i ostala svojstva, mada bi vi svoje skriptove trebalo temeljno da dokumentujete. U našim skriptovima će se ipak nalaziti komentari sa brojem skripta i imenom datoteke, a sve u cilju lakšeg unakrsnog referenciranja (u knjizi i kada ih preuzimate sa sajta koji se odnosi na ovu knjigu, **www.LarryUllman.com**).

nastavak na sledećoj strani

Skript 1.5 Ovaj skript štampa tri od mnogih unapred definisanih promenljivih iz PHP-a.

```
1   <!doctype html>
2   <html lang="en">
3   <head>
4      <meta charset="utf-8">
5      <title>Predefined Variables</title>
6   </head>
7   <body>
8   <?php # Script 1.5 - predefined.php
9
10  // Napravi promenljive sa kraćim imenima:
11  $file = $_SERVER['SCRIPT_FILENAME'];
12  $user = $_SERVER['HTTP_USER_AGENT'];
13  $server = $_SERVER['SERVER_SOFTWARE'];
14
15  // Štampaj ime ovog skripta:
16  echo "<p>You are running the file:<br><strong>$file</strong>.</p>\n";
17
18  // Štampaj informacije o korisniku:
19  echo "<p>You are viewing this page using:<br><strong>$user</strong></p>\n";
20
21  // Štampaj informacije o serveru:
22  echo "<p>This server is running:<br><strong>$server</strong>.</p>\n";
23
24  ?>
25  </body>
26  </html>
```

3. Kreirajte skraćenu verziju prve promenljive koja će se koristiti u ovom skriptu:

```
$file = $_SERVER['SCRIPT_FILENAME'];
```

U ovom skriptu će se koristiti tri promenljive, od kojih svaka potiče iz veće predefinisane promenljive **$_SERVER**. Promenljiva **$_SERVER** odnosi se na niz informacija vezanih za server. Prva promenljiva koju skript koristi je **$_SERVER['SCRIPT_FILENAME']**.

U ovoj promenljivoj se čuva puna putanja i ime skripta koji se trenutno izvršava (na primer, **C:\Program Files\Apache\htdocs\predefined.php**). Vrednost koja se čuva u promenljivoj **$_SERVER ['SCRIPT_FILENAME']** dodeliće se novoj promenljivoj $file. Kreirali smo novu promenljivu kraćeg imena i dodelili joj vrednost iz promenljive **$_SERVER**, kako bi nam bilo lakše da je koristimo prilikom štampanja. (Na ovaj način rešavamo još jedan problem, ali o tome će biti reči kasnije.)

4. Kreirajte skraćene verzije za preostale dve promenljive:

```
$user = $_SERVER
→['HTTP_USER_AGENT'];
$server = $_SERVER
→['SERVER_SOFTWARE'];
```

$_SERVER ['HTTP_USER_AGENT'] predstavlja pretraživač i operativni sistem korisnika koji pristupa skriptu. Ta vrednost se dodeljuje promenljivoj **$user**.

$_SERVER ['SERVER_SOFTWARE'] predstavlja veb aplikaciju na serveru gde se izvršava PHP (na primer, Apache, Abyss, Xitami, IIS). To je program koji mora da bude instaliran (pogledajte Dodatak A) kako bi PHP skriptovi mogli da se izvršavaju na tom računaru.

5. Štampajte ime skripta koji se izvršava:

```
echo "<p>You are running the
→file:<br /><strong>$file
→</strong>.</p>\n";
```

Prva promenljiva koju štampamo je **$file**. Obratite pažnju na to da promenljiva mora da se upotrebi unutar dvostrukih navodnika, a da naredba koristi i znak za novi red (**\n**), kojim se dobija novi red u generisanom izvornom HTML kodu. Neke osnovne HTML oznake – za pasus i za tamna slova – upotrebljene su kako bi generisana stranica bila privlačnija.

6. Štampajte informaciju o korisniku koji pristupa skriptu:

```
echo "<p>You are viewing this page
→using:<br><strong>$user</strong>
→</p>\n";
```

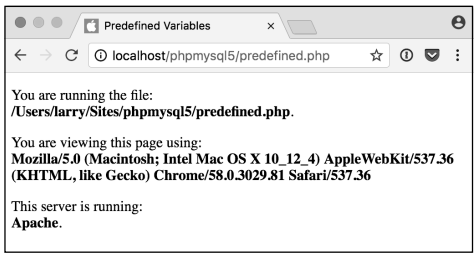
Upotrebom ovog reda štampa se druga promenljiva **$user**. Obratimo pažnju na ono što je rečeno u četvrtom koraku: **$user** je promenljiva vezana sa **$_SERVER ['HTTP_USER_AGENT']** i odnosi se na operativni sistem, vrstu i verziju pretraživača koji se koristi za pristupanje veb stranici.

7. Štampajte informacije o serveru:

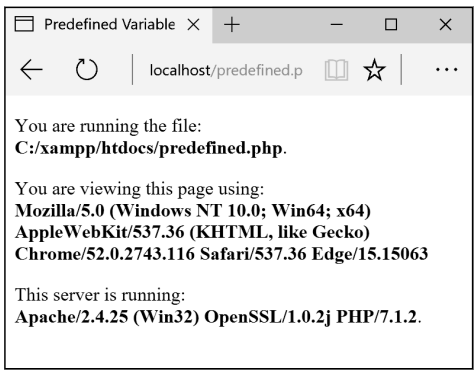
```
echo "<p>This server is running:
→<br><strong>$server</strong>.
→</p>\n";
```

8. Zatvorite PHP blok i HTML stranicu:

```
?>
</body>
</html>
```



A Skript predefined.php pruža korisniku informacije o skriptu, veb pretraživaču koji se koristi za to prikazivanje kao i o samom serveru.



B Ovo je prvi stvarno dinamičan skript u knjizi, što možemo zaključiti na osnovu same veb stranice koja se menja zavisno od servera na kojem se izvršava i od veb pretraživača kojim se posmatra (uporedite sa **A**).

9. Sačuvajte fajl pod imenom **`predefined.php`** u svom veb direktorijumu a potom proverite kako funkcioniše u veb pretraživaču **A**.

SAVET Ako imate problem sa ovim ili bilo kojim drugim skriptom, potražite pomoć na forumu posvećenom ovoj knjizi (`www.LarryUllman.com/forums/`).

SAVET Ako je moguće, izvršite ovaj skript pomoću drugog veb pretraživača i/ili na drugom serveru B.

SAVET Ime promenljive ne sme da sadrži razmak. Umesto razmaka obično se koristi donja crta.

SAVET Najvažnije pitanje prilikom kreiranja promenljivih jeste dosledno korišćenje šeme imenovanja. U ovoj knjizi ćete videti da za imena promenljivih koristim samo mala slova, a reči odvajam donjom crtom (`$first_name`). Sa druge strane, neki programeri više vole da umesto donje crte koriste veliko slovo: `$FirstName` (kamilja notacija – camel-case style).

SAVET PHP nema tako stroga pravila što se tiče promenljivih, što znači da nije obavezno da im zadate početnu vrednost (odredite tačnu vrednost) niti da ih deklarišete (odredite im tip), a promenljivu možete da konvertujete bez problema u raznovrsne tipove.

Uvođenje stringova

Sada kad ste upoznati sa opštim konceptom promenljivih, pogledajmo promenljive detaljnije. Prvi tip promenljivih koji obrađujemo je *string*. To su jednostavno neki znakovi u navodnicima: slova, brojevi, razmaci, znakovi interpunkcije i tako dalje. Ovo su primeri stringova:

- 'Tobias'
- "In watermelon sugar"
- '100'
- 'August 2, 2017'

Da bi se kreirala promenljiva tipa string, potrebno je validnom imenu promenljive dodeliti vrednost tipa stringa:

```
$first_name = 'Tobias';
$today = 'August 2, 2011';
```

Prilikom kreiranja samog stringa, mogu se oko znakova koristiti jednostruki ili dvostruki navodnici, isto kao kada se štampa tekst. Ono što je bitno, string mora da se otvori i zatvori istim tipom navodnika. Ako se isti tip navodnika javlja unutar stringa, mora se izbeći njegovo specijalno značenje:

```
$var = "Define \"platitude\", please.";
```

Takođe, postoji mogućnost da se upotrebi druga vrsta navodnika:

```
$var = 'Define "platitude", please.';
```

Za štampanje sadržaja promenljive, koristi se **`echo`** ili **`print`**:

```
echo $first_name;
```

Da bi se sadržaj promenljive štampao u okviru dodatnog teksta, moraju se koristiti dvostruki navodnici:

```
echo "Hello, $first_name";
```

Već ste jednom radili sa stringovima – kad ste u prethodnom odeljku upotrebili unapred definisane promenljive (vrednosti tih promenljivih su slučajno bili stringovi). U narednom primeru potrebno je da kreirate i upotrebite vlastite stringove.

Skript 1.6 U ovom skriptu se prave promenljive tipa string i njihove vrednosti se potom šalju u veb pretraživač.

```
1   <!doctype html>
2   <html lang="en">
3   <head>
4      <meta charset="utf-8">
5      <title>Strings</title>
6   </head>
7   <body>
8   <?php # Script 1.6 - strings.php
9
10  // Kreiranje promenljivih:
11  $first_name = 'Haruki';
12  $last_name = 'Murakami';
13  $book = 'Kafka on the Shore';
14
15  // Štampanje vrednosti:
16  echo "<p>The book <em>$book</em> was
    written by $first_name
    $last_name.</p>";
17
18  ?>
19  </body>
20  </html>
```

Da biste koristili string, potrebno je da uradite sledeće:

1. Započnite u svom tekst editoru ili IDE okruženju nov PHP dokument, čije ime će biti **strings.php**. Počnite sa osnovnim HTML-om i dodajte početnu PHP oznaku (**Skript 1.6**):

   ```
   <!doctype html>
   <html lang="en">
   <head>
     <meta charset="utf-8">
     <title>Strings</title>
   </head>
   <body>
   <?php # Script 1.6 - strings.php
   ```

2. U okviru PHP oznaka, kreirajte tri promenljive:

   ```
   $first_name = 'Haruki';
   $last_name = 'Murakami';
   $book = 'Kafka on the Shore';
   ```

   Ovim najjednostavnijim primerima kreiraju se promenljive **$first_name**, **$last_name**, i **$book** koje će se nakon toga štampati u poruci.

Nastavak na sledećoj stranici.

3. Dodajte naredbu **echo**:

```
echo "<p>The book <em>$book
→</em> was written by
→$first_name $last_name.</p>";
```

Ovaj skript samo štampa izjavu o autorstvu pomoću tri definisane promenljive. Nešto malo HTML formatiranja (isticanje naslova knjige) korišćeno je kako bi tekst izgledao lepše. Ne zaboravite da upotrebite dvostruke navodnike kako bi se pravilno prikazao sadržaj promenljivih (o značaju dvostrukih navodnika biće reči i na kraju poglavlja).

4. Zatvorite PHP blok i HTML stranicu:

```
?>
</body>
</html>
```

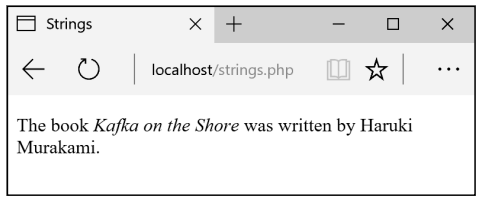
5. Sačuvajte fajl pod imenom **strings.php** u svom veb direktorijumu i potom proverite kako funkcioniše u veb pretraživaču Ⓐ.

6. Ako želite, promenite vrednosti tri promenljive, sačuvajte datoteku i ponovo izvršite skript Ⓑ.

SAVET **Ako dodelite drugu vrednost nekoj postojećoj promenljivoj (na primer, $book), nova vrednost će zameniti (poništiti) prethodnu, Na primer:**

```
$book = 'High Fidelity';
$book = 'The Corrections';
/* $book sada ima vrednost
'The Corrections'. */
```

SAVET **PHP nema zadati maksimum za dužinu stringa. Teoretski je moguće da naiđete na ograničenje resursa na serveru, ali mala je verovatnoća da ćete u praksi imati takav problem.**



Ⓐ Veb stranica je dobijena štampanjem vrednosti tri promenljive.



Ⓑ Rezultat skripta se promenio tako što su izmenjene promenljive u njemu.

Skript 1.7 Nadovezivanje (konkatenacija) omogućava pridruživanje dodatnih znakova samom stringu

```
1   <!doctype html>
2   <html lang="en">
3   <head>
4      <meta charset="utf-8">
5      <title>Concatenation</title>
6   </head>
7   <body>
8   <?php # Script 1.7 - concat.php
9
10  // Kreiranje promenljivih:
11  $first_name = 'Melissa';
12  $last_name = 'Bank';
13  $author = $first_name . ' ' .
    $last_name;
14
15  $book = 'The Girls\' Guide to Hunting
    and Fishing';
16
17  //Štampanje vrednosti:
18  echo "<p>The book <em>$book</em> was
    written by $author.</p>";
19
20  ?>
21  </body>
22  </html>
```

Nadovezivanje stringova

Nadovezivanje (konkatenacija) je zapravo sabiranje za stringove, što znači da se znakovi dodaju na kraj stringa. Ono se izvodi pomoću *operatora sastavljanja*, a to je tačka (**.**):

```
$city= 'Seattle';
$state = 'Washington';
$address = $city . $state;
```

Promenljiva **$address** sad ima vrednost *SeattleWashington* što je takoreći ono što se želelo postići (*Seattle, Washington*). Da bismo dobili precizniji rezultat, mogli bismo da napišemo:

```
$address = $city . ', ' . $state;
```

Što znači da se osim promenljivih nadovezuje još i zarez sa razmakom. Pošto PHP nema strogih ograničenja što se tiče promenljivih, moguće je nadovezivati stringove i brojeve. Korišćenjem obe ove naredbe dobićete isti rezultat (Seattle, Washington 98101):

```
$address = $city . ', ' . $state .
' 98101';
$address = $city . ', ' . $state .
' ' . 98101;
```

Sada ćemo promeniti skript **scripts.php** upotrebom ovog novog operatora.

Upotreba nadovezivanja:

1. Otvorite **scripts.php** (vidite ga kao Skript 1.6) u svom tekst editoru ili IDE okruženju.
2. Pošto ste odredili promenljive **$first_name** i **$last_name** (u redovima 11 i 12), dodajte ovaj red (**Skript 1.7**)

   ```
   $author = $first_name . ' ' .
   → $last_name;
   ```

Nastavak na sledećoj strani

U cilju demonstracije procesa nadovezivanja kreiraćemo novu promenljivu – **$author** – kao konkatenaciju dva postojeća stringa i razmaka između njih.

3. Izmenite naredbu **echo** tako da se upotrebi nova promenljiva:

```
echo "<p>The book <em>$book</em>
→was written by $author.</p>";
```

Pošto su dve promenljive spojene u jednu, naredba **echo** je morala da se prilagodi.

4. Ako želite, izmenite naslov HTML stranice i vrednosti promenljivih **$first_name**, **$last_name**, **$book**.

5. Sačuvajte fajl pod imenom **concat.php** u svom veb direktorijumu i proverite kako funkcioniše u veb pretraživaču Ⓐ.

SAVET **PHP sadrži veliki broj funkcija koje se odnose na stringove, što ćete i videti u ovoj knjizi. Na primer, da biste izračunali dužinu stringa (koliko znakova sadrži), upotrebićete funkciju `strlen()`:**

```
$num = strlen('some string'); // 11
```

SAVET **Možete da zadate PHP-u da izvrši promenu veličine slova. Pomoću funkcije `strtolower()` dobijaju se sve mala slova; sa funkcijom `strtoupper()` dobijaju se sve velika slova; pomoću funkcije `ucfirst()` dobijamo veliko prvo slovo rečenice a sva ostala su mala, dok upotrebom funkcije `ucwords()` dobijamo veliko prvo slovo svake reči a ostala mala.**

SAVET **Ako samo spajate dve vrednosti, možete da upotrebite operator nadovezivanja (`.=`) (engl. *concatenation assignment operator*). Sledeće naredbe su ekvivalentne:**

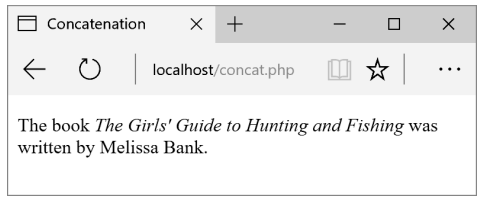
```
$title = $title . $subtitle;
$title .= $subtitle;
```

SAVET **Početni primer u ovom odeljku mogao bi da se napiše kao:**

```
$address = "$city, $state";
```

ili

```
$address = $city;
$address .= ',';
$address .= $state;
```



Ⓐ U ovom izmenjenom skriptu krajnji rezultat nadovezivanja korisnik i ne primećuje.

Korišćenje PHP priručnika

PHP priručnik – može se naći na adresi **www.php.net/manual** – sadrži spisak svih funkcija i mogućnosti koje poseduje ovaj jezik. U priručniku su prvo objašnjeni opšti pojmovi (instaliranje, sintaksa, promenljive), a zatim su funkcije grupisane po temama (MySQL, funkcije za stringove i tako dalje).

Funkciju ćete u priručniku najbrže naći ako u veb pretraživaču ukucate sledeće: **www.php.net/nazivfunkcije** (na primer **www.php.net/print**). Za svaku funkciju priručnik navodi sledeće:

- Verzije PHP-a u kojima je ta funkcija dostupna.
- Koliko i koje tipove argumenata funkcija prihvata (neobavezni argumenti su navedeni u uglastim zagradama).
- Kojeg tipa je vraćena vrednost funkcije.

Pored toga, priručnik takođe sadrži opis svake funkcije. Trebalo bi da steknete naviku da koristite priručnik kad god ste u dilemi u vezi sa nekom funkcijom i njenim pravilnim korišćenjem, ili kada vam je potrebno da više naučite o nekoj mogućnosti jezika. Takođe, veoma je bitno da znate koju verziju PHP-a imate, jer se funkcije i drugi detalji u PHP-u vremenom menjaju.

Uvođenje brojeva

Prilikom uvođenja promenljivih napomenuo sam da PHP ima numeričke tipove integer (cele brojeve) i floating-point (decimale). Međutim, na osnovu mog iskustva, oba ova tipa mogu da se klasifikuju pod opšti pojam brojevi (engl. *numbers*) bez ikakve razlike. Validne numeričke vrednosti u PHP-u mogu da budu:

- 8
- 3.14
- 10980843985
- –4.2398508
- 4.4e2

Primetićete da te vrednosti nisu nikad u navodnicima – brojevi koji se nalaze u navodnicima su zapravo stringovi sa numeričkim vrednostima – i ne koriste zarez za odvajanje hiljada. Osim toga, za broj se smatra da je pozitivan ako ispred njega nema znaka minus (**-**). Pored standardnih artitmetičkih operatora koje možete da primenite na brojeve (**Tabela 1.1**), u PHP-u je ugrađeno na desetine funkcija. Dve funkcije koje se vrlo često koriste jesu **`round ()`** i **`number_format ()`**. Prva zaokružuje broj na najbliži ceo broj:

```
$n = 3.14;
$n = round($n); // 3
```

Ista funkcija takođe može da zaokruži broj na zadati broj decimalnih mesta:

```
$n = 3.141592;
$n = round($n, 3); // 3.142
```

Funkcija **`number_format ()`** pretvara broj u češće korišćen format, sa zarezima za odvajanje hiljada:

```
$n = 20943;
$n = number_format($n); // 20,943
```

I ova funkcija može da kreira zadati broj decimalnih mesta:

```
$n = 20943;
$n = number_format($n, 2); //
20,943.00
```

Da bismo uvežbali brojeve, napisaćemo model skripta koji izvodi izračunavanja koja bi mogla da se koriste za potrošačku korpu u e-trgovini.

TABELA 1.1 **Aritmetički operatori**

Operator	Značenje
+	Sabiranje
-	Oduzimanje
*	Množenje
/	Deljenje
%	Modulo
++	Uvećanje
--	Umanjenje

Da biste koristili brojeve, potrebno je da uradite sledeće:

1. Započnite u svom tekst editoru ili IDE okruženju nov PHP dokument, čije ime će biti **numbers.php** (**Skript 1.8**):

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Numbers</title>
</head>
<body>
<?php # Script 1.8 - numbers.php
```

2. Podesite parametre za promenljive koje su vam potrebne (količina, cena i poreska stopa):

```
$quantity = 30;
$price = 119.95;
$taxrate = .05;
```

   U ovom skriptu će se koristiti tri ukodirane promenljive nad kojima će se vršiti izračunavanja. Kasnije ćete videti kako te vrednosti mogu da se odrede dinamički (tj. interakcijom sa korisnikom putem HTML obrasca).

3. Obavite sledeća izračunavanja:

```
$total = $quantity * $price;
$total = $total + ($total *
→$taxrate);
```

   U prvom redu se izračunava iznos porudžbine kao broj komada pomnožen sa jediničnom cenom. Nakon toga, u drugom redu se ukupnom iznosu dodaje iznos poreza (koji se izračunava množenjem zbira sa poreskom stopom).

4. Formatirajte ukupan iznos:

```
$total = number_format($total, 2);
```

   Funkcija **number_format()** ima zadatak da grupiše ukupan iznos u hiljade koji će potom i zaokružiti i to na dve decimale. Primenom ove funkcije dobija se pravilan format izračunate vrednosti.

Skript 1.8 Ovaj skript vrši osnovna matematička izračunavanja, kakva se inače koriste za aplikacije u e-trgovini.

```
1   <!doctype html>
2   <html lang="en">
3   <head>
4      <meta charset="utf-8">
5      <title>Numbers</title>
6   </head>
7   <body>
8   <?php # Script 1.8 - numbers.php
9
10  // Set the variables:
11  $quantity = 30; // Kupovina 30 predmeta.
12  $price = 119.95;
13  $taxrate = .05; // 5% porez na promet.
14
15  // Izračunavanje zbira:
16  $total = $quantity * $price;
17  $total = $total + ($total * $taxrate);
    // Izračunavanje i dodavanje poreza.
18
19  // Formatiranje zbira:
20  $total = number_format ($total, 2);
21
22  // Štampanje rezultata:
23  echo '<p>You are purchasing <strong>' .
    $quantity . '</strong> widget(s) at a
    cost of <strong>$' . $price . '</strong>
    each. With tax, the total comes to
    <strong>$' . $total . '</strong>.</p>';
24
25  ?>
26  </body>
27  </html>
```

5. Štampajte rezultate:

```
echo '<p>You are purchasing
→<strong>' . $quantity .
→'</strong> widget(s) at a cost
→of <strong>$' . $price .
→'</strong> each. With tax, the
→total comes to <strong>$' .
→$total . '</strong>.</p>';
```

Poslednji korak u skriptu predstavlja štampanje rezultata. Naredba **echo** koristi i tekst u jednostrukim navodnicima kao i konkatenirane promenljive prilikom štampanja cele kombinacije HTML-a, dolarskih znakova i vrednosti promenljivih. U poslednjem primeru ovog poglavlja videćete drugi način prikazivanja.

6. Zatvorite PHP kôd i HTML stranicu:

```
?>
</body>
</html>
```

7. Sačuvajte fajl pod imenom **numbers.php** u svom veb direktorijumu i proverite kako funkcioniše u veb pretraživaču Ⓐ.

8. Ako želite, izmenite tri početne promenljive i ponovo izvršite skript Ⓑ.

SAVET PHP može na većini platformi da podrži maksimalan ceo broj od oko dve milijarde. Kod većih vrednosti, PHP će automatski preći na tip floating point.

SAVET U aritmetici se javlja pitanje prvenstva (redosleda kojim se vrše složene operacije). Mada se u PHP priručniku i drugim izvorima navodi hijerarhija prvenstva, sa mog stanovišta samo programiranje je sigurnije i čitljivije kada se redosled izračunavanja dobija korišćenjem zagrada (pogledajte red 17 u skriptu 1.8).

SAVET Poznato je koliko su računari loši kada je u pitanju rad sa decimalama. Na primer, broj 2.0 može da se čuva kao 1.99999. U većini slučajeva to neće predstavljati problem, ali u situacijama u kojima je najvažnija matematička preciznost, bolje bi bilo da se oslonite na cele brojeve, a ne na decimalne. PHP priručnik sadrži informacije o ovoj temi, ali i alternativne funkcije radi dobijanja veće preciznosti izračunavanja.

SAVET Mnogi aritmetički operatori imaju i odgovarajući operator dodeljivanja, što predstavlja skraćeno dodeljivanje vrednosti. Ovaj red:

```
$total = $total + ($total * $taxrate);
```

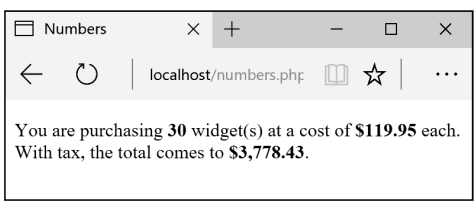
bi mogao da se zameni sledećim redom:

```
$total += ($total * $taxrate);
```

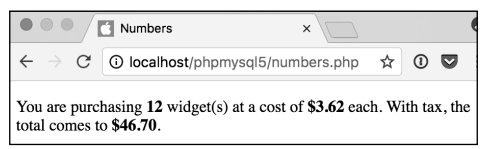
SAVET Ako prilikom određivanja vrednosti `$price` ne upotrebite dve decimale (npr. 119,9 ili 34), potrebno je pre štampanja da na `$price` primenite funkciju `number_format()`.

SAVET U verziji PHP7 postoji nova funkcija `intdiv()` koja vraća celobrojnu vrednost prilikom deljenja.

```
echo intdiv(7, 3); // 2
```



Ⓐ PHP stranica za brojeve (Skript 1.8) vrši izračunavanja sa fiksiranim vrednostima.



Ⓑ Da biste promenili generisanu veb stranicu, izmenite bilo koju od tri promenljive (uporedite sa Ⓐ).

Uvođenje konstanti

Konstante se, poput promenljivih, koriste za privremeno čuvanje vrednosti. Međutim, konstante i promenljive se razlikuju na više načina. Kao prvo, za kreiranje konstante koristimo funkciju **define()** a ne operator dodeljivanja (=).

```
define('IME', vrednost);
```

Primetićete da je uobičajeno da se za ime konstante koriste samo velika slova, mada to nije neophodno. Najvažnije je što se za razliku od promenljivih kod imena konstanti ne koristi početni dolarski znak (zato što konstante ne podležu izmenama).

Konstanti može da se dodeli samo *skalarna* vrednost, kao što je string ili broj:

```
define('USERNAME', 'troutocity');
define('PI', 3.14);
```

Takođe, za razliku od promenljivih, vrednost konstante ne može da se menja. Za pristupanje vrednosti konstante, na primer radi štampanja, ne možete da stavite konstantu unutar navodnika:

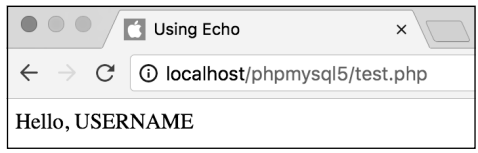
```
echo "Hello, USERNAME"; // Neće valjati!
```

Ovaj kôd će PHP štampati bukvalno kao *Hello USERNAME* Ⓐ a neće pritom staviti vrednost konstante **USERNAME** (zato što ništa ne ukazuje na to da **USERNAME** nije samo tekst). Umesto toga, možete konstantu da štampate samu:

```
echo 'Hello, ';
echo USERNAME;
```

ili da upotrebite operator za nadovezivanje:

```
echo 'Hello, ' . USERNAME;
```



Ⓐ Konstante ne mogu da se koriste unutar navodnika.

PHP sadrži nekoliko unapred definisanih konstanti, slično unapred definisanim promenljivama koje smo već koristili u ovom poglavlju. One uključuju **PHP_VERSION** (verzija pokrenutog PHP-a) i **PHP_OS** (operativni sistem na kojem je pokrenut server). Sledeći skript će odštampati ove vrednosti kao i vrednosti korisnički definisanih konstanti.

Da biste koristili konstante, potrebno je da uradite sledeće:

1. Započnite u svom tekst editoru ili IDE okruženju nov PHP dokument, čije ime će biti **constants.php** (**Skript 1.9**):

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Constants</title>
</head>
<body>
<?php # Script 1.9 - constants.php
```

2. Kreirajte novu konstantu za datum:

```
define('TODAY', 'April 23, 2017');
```

Priznajem da je ovo trivijalna upotreba konstante, ali je primer dovoljan za ilustraciju. U poglavlju 9 „PHP i MySQL", videćete kako se u konstantama čuvaju informacije za pristupanje bazi podataka.

3. Štampajte datum, informacije o PHP verziji i operativnom sistemu:

```
echo '<p>Today is ' . TODAY .
→ '.<br>This server is running
→ version<strong>' . PHP_VERSION .
→ '</strong> of PHP on the
→ <strong>' . PHP_OS . '</strong>
→ operating system.</p>';
```

Pošto konstante ne mogu da se štampaju unutar znakova navoda, u naredbi **echo** koriste operator konkatenacije.

Nastavak na sledećoj strani

Skript 1.9 Pored promenljivih, konstante predstavljaju još jedan način za privremeno smeštanje podataka koje možete koristiti u PHP-u.

```
1   <!doctype html>
2   <html lang="en">
3   <head>
4      <meta charset="utf-8">
5      <title>Constants</title>
6   </head>
7   <body>
8   <?php # Script 1.9 - constants.php
9
10  // Postavi današnji datum kao konstantu:
11  define('TODAY', 'April 23, 2017');
12
13  // Štampanje poruke, uz upotrebu unapred definisanih konstanti i konstante TODAY:
14  echo '<p>Today is ' .TODAY .'.<br>This server is running version <strong>' .
    PHP_VERSION . '</strong> of PHP on the <strong>' . PHP_OS . '</strong> operating
    system.</p>';
15
16  ?>
17  </body>
18  </html>
```

4. Zatvorite PHP kôd i HTML stranicu:

```
?>
</body>
</html>
```

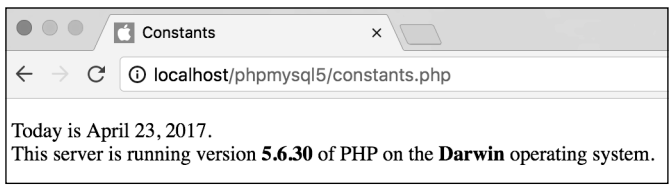
5. Sačuvajte fajl pod imenom **`constants.php`** u svom veb direktorijumu i proverite kako funkcioniše u veb pretraživaču B.

SAVET Ako imate mogućnosti, izvršite ovaj skript na nekom drugom serveru na kojem može da se koristi PHP C.

SAVET Operativni sistem po imenu `Darwin` B je tehnički naziv za macOS.

SAVET U poglavlju 12 „Kolačići i sesije”, učićete o još jednoj konstanti, SID (identifikator sesije).

SAVET Počevši od verzije PHP 7, imate mogućnost da kreirate i konstantu niza. Više o nizovima učićete u poglavlju 2, „Programiranje sa PHP-om”.



B Kad budete upotrebili PHP konstante, videćete detalje vaših podešavanja u PHP-u.



C Kada isti skript (Skript 1.9) izvršite na drugom serveru, dobijate drugačije rezultate.

Jednostruki i dvostruki navodnici

Kada je u pitanju korišćenje PHP-a, veoma je važno razumeti razliku između jednostrukih i dvostrukih navodnika. Kao što ste videli u dosadašnjim primerima, za **`echo`** i **`print`** kao i prilikom dodeljivanja vrednosti stringovima možete da koristite bilo koje. Međutim, postoji jedna ključna razlika između te dve vrste navodnika i njihove upotrebe. Već smo videli tu razliku ali je važno da se taj pojam bolje objasni.

U PHP-u se *vrednosti u jednostrukim navodnicima uzimaju doslovno, dok se one u dvostrukim navodnicima interpretiraju.* Drugim rečima, kada se promenljive i specijalni znaci (Tabela 1.2) postave u dvostruke navodnike, prikazaće se vrednosti koje oni predstavljaju a ne njihove prave vrednosti. Na primer, recimo da imate

```
$var = 'test';
```

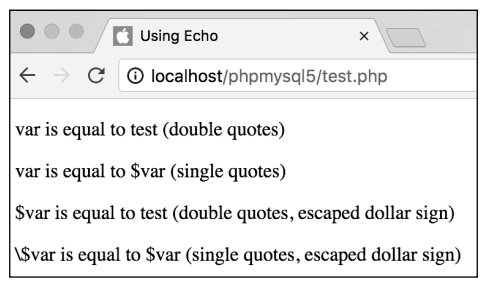
TABELA 1.2 Kontrolne (izlazne) sekvence

Kôd	Značenje
\”	Dvostruki navodnik
\’	Jednostruki navodnik
\\	Obrnuta kosa crta
\n	Novi red
\r	Početak reda
\t	Tabulator
\$	Dolarski znak

Kodom **`echo "var is equal to $var";`** bi se dobilo *var is equal to test*, dok bi se kodom **`echo 'var is equal to $var';`** dobilo *var is equal to $var*. Kad se upotrebi obrnuta kosa crta da bi se izbeglo specijalno značenje, kodom **`echo "\$var is equal to $var";`** bi se dobilo *$var is equal to test*, a kodom **`echo '\$var is equal to $var';`** bi se dobilo *\$var is equal to $var* Ⓐ.

Na osnovu ovih primera trebalo bi da se vidi da će dvostruki navodnici zameniti ime promenljive **`($var)`** njenom vrednošću (*test*) i specijalan znak **`(\$)`** onom vrednošću koju on predstavlja (*$*). Kada se koriste jednostruki navodnici, dobija se tačno ono što je napisano, osim što se kombinacija **`\'`** (jednostruki navodnik sa izbegnutim specijalnim značenjem) štampa samo kao jednostruki navodnik, a **`\\`** (obrnuta kosa crta sa izbegnutim specijalnim značenjem) se štampa kao samo jedna obrnuta kosa crta.

Još jedan primer različitog ponašanja jednostrukih i dvostrukih navodnika videćemo u eksperimentu sa izmenjenim skriptom **`numbers.php`**.



Ⓐ Kako jednostruki i dvostruki navodnici u PHP-u utiču na rezultat štampanja.

Da bi se koristili jednostruki i dvostruki navodnici, potrebno je da uradite sledeće:

1. Otvorite **numbers.php** u tekst editoru ili u IDE okruženju (Skript 1.8).
2. Izbacite (uklonite) postojeću naredbu **echo** (**Skript 1.10**).
3. Štampajte natpis a zatim ponovite prvobitnu naredbu **echo**, ali ovoga puta sa dvostrukim znacima navoda:

```
echo "<h3>Using double quotation
→ marks:</h3>";
echo "<p>You are purchasing
→ <strong>$quantity</strong>
→ widget(s) at a cost of
→ <strong>\$$price</strong> each.
→ With tax, the total comes to
→ <strong>\$$total</strong>.</p>\n";
```

U prvobitnom skriptu su rezultati štampani koristeći jednostruke navodnike i operator nadovezivanja. Isti rezultat može da se postigne pomoću dvostrukih navodnika. Kada se koriste dvostruki navodnici, promenljive mogu da se uvrste u string.

Međutim, tu postoji jedan problem: ako želite da štampate iznos u dolarima kao $12.34 (gde 12.34 potiče iz promenljive), verovatno biste pomislili da stavite **$$var**. Ali, to neće biti moguće (razlozi su kompleksni). Umesto toga izbegnite specijalno značenje prvog dolarskog znaka i dobićete **\$$var**, što se javlja dva puta u ovom kodu. Prvi dolarski znak se štampa, a drugi označava početak imena promenljive.

Skript 1.10 Ovaj poslednji skript u prvom poglavlju prikazuje razlike do kojih dolazi prilikom korišćenja jednostrukih i dvostrukih navodnika.

```
1   <!doctype html>
2   <html lang="en">
3   <head>
4      <meta charset="utf-8">
5      <title>Quotation Marks</title>
6   </head>
7   <body>
8   <?php # Script 1.10 - quotes.php
9
10  // Postavljanje promenljivih:
11  $quantity = 30; // Buying 30 widgets.
12  $price = 119.95;
13  $taxrate = .05; // 5% sales tax.
14
15  // Računanje zbira.
16  $total = $quantity * $price;
17  $total = $total + ($total * $taxrate);
// Calculate and add the tax.
18
19  // Formatiranje zbira:
20  $total = number_format ($total, 2);
21
22  // Štampanje rezultata sa dvostrukim
    navodnicima:
23  echo "<h3>Using double quotation
    marks:</h3>";
24  echo "<p>You are purchasing
    <strong>$quantity</strong> widget(s)
    at a cost of <strong>\$$price
    </strong> each. With tax, the total
    comes to <strong>\$$total</strong>.
    </p>\n";
25
26  // Štampanje rezultata sa jednostrukim
    navodnicima:
27  echo '<h3>Using single quotation
    marks:</h3>';
28  echo '<p>You are purchasing
    <strong>$quantity</strong> widget(s)
    at a cost of <strong>\$$price
    </strong> each. With tax, the total
    comes to <strong>\$$total</strong>.
    </p>\n';
29
30  ?>
31  </body>
32  </html>
```

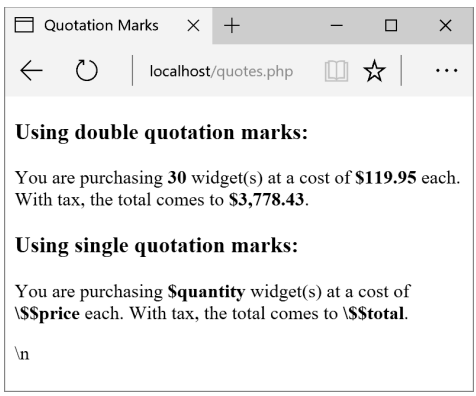
4. Ponovite naredbe **echo**, ali ovoga puta sa jednostrukim navodnicima:

```
echo '<h3>Using single quotation
→marks:</h3>';
echo '<p>You are purchasing
→<strong>$quantity</strong>
→widget(s) at a cost of
→<strong>\$$price</strong> each.
→With tax, the total comes to
→<strong>\$$total</strong>.</p>\n';
```

   Naredba **echo** se ovde koristi kako bi se naglasila razlika prilikom korišćenja jednostrukih ili dvostrukih navodnika. Nećete dobiti željeni rezultat, a na dobijenoj stranici se tačno vidi šta se dešava.

5. Ako želite, promenite naslov stranice.

6. Sačuvajte fajl pod imenom **quotes.php** u svom veb direktorijumu i proverite kako funkcioniše u veb pretraživaču **B**.



B Ovde se vide rezultati korišćenja dvostrukih (double) i jednostrukih (single) navodnika.

7. Pregledajte sam izvor veb stranice gde ćete videti kakva je razlika kada se znak za novi red (**\n**) koristi u svakoj od dve vrste navodnika.

   Trebalo bi da se vidi da kad znak za novi red (**\n**) stavite u dvostruke navodnike, dobijate novi red u izvornom HTML-u. Međutim, kada ga stavite u jednostruke navodnike, dobijate samo znakove **** i **n**.

SAVET S obzirom na to da PHP traži imena promenljivih u dvostrukim navodnicima, jednostruki navodnici teoretski brže rade. Međutim, ako treba da štampate vrednost promenljive, morate da upotrebite dvostruke navodnike.

Kako se u ispravnom HTML-u često javljaju atributi u dvostrukim navodnicima, često je praktičnije da se upotrebe jednostruki navodnici kad se iz PHP-a prikazuje HTML:

```
echo '<table class="data">';
```

Da biste taj HTML štampali pomoću dvostrukih navodnika, morali biste da izbegnete specijalno značenje svakog dvostrukog navodnika u stringu:

```
echo "<table class=\"data\">";
```

SAVET U novijim verzijama PHP-a sada može da se piše $$price i $$total bez obrnute kose crte ispred (zahvaljujući nekoj internoj čaroliji). U starijim verzijama to nije bilo moguće. Da biste garantovali dobijanje pouzdanih rezultata, koji pritom neće zavisiti od verzije PHP-a, moj vam je savet da se pridržavate sintakse \$$var kada treba da štampate dolarski znak a odmah zatim i vrednost promenljive.

SAVET Ako vam je još uvek nejasna razlika između jednostrukih i dvostrukih navodnika, upotrebite dvostruke pa ćete verovatno imati manje problema.

Osnovni koraci za otkrivanje i uklanjanje grešaka

Otkrivanje i uklanjanje grešaka (engl. *debugging*) nikako nije jednostavno za razumevanje, a nažalost, moguće ga je savladati jedino u praksi. Mogli bismo narednih 50 stranica da posvetimo toj temi i opet biste uhvatili tek delić veštine otkrivanja i otklanjanja grešaka koju ćete vremenom da steknete jer će vam biti potrebna.

Razlog zbog kojeg vam predstavljam ovu temu na tako preteći način je što je važno da se ne zavaravate kad se prihvatite programiranja. Ponekad vaš kôd neće raditi ono što od njega očekujete, neizostavno ćete praviti greške iz nepažnje, a biće i momenata kad ćete čupati kosu od besa, čak i prilikom korišćenja jezika koji je relativno blizak korisniku kakav je PHP. U kratkim crtama, budite spremni da ponekad budete zbunjeni i frustrirani. Ja koristim PHP još od davne 1999 godine pa se još uvek povremeno zaglibim u programerskom blatu. Ali, imajte na umu da je otkrivanje i otklanjanje grešaka veoma važna veština koju ćete vremenom da savladate iz nužde i iskustva. Kako počinjete da se upuštate u avanturu zvanu PHP programiranje, otkriću vam neke osnovne ali konkretne savete koji se odnose na uspešno otkrivanje i uklanjanje grešaka. Imajte u vidu da su to samo opšte tehnike otkrivanja i uklanjanja grešaka, posebno prilagođene PHP programeru početniku. U poglavlju 8 „Rukovanje greškama i njihovo ispravljanje”, detaljnije se obrađuju i druge tehnike.

Da biste pronašli i uklonili greške u PHP skriptu:

- Vodite računa da uvek izvršavate PHP skript preko URL-a!

  Ovo je verovatno najčešća početnička greška. PHP kôd mora da se izvršava kroz aplikaciju veb servera, što znači da mora da se pozove kao http://nešto. Kada ugledate sam PHP kôd umesto rezultata koji je taj kôd trebalo da generiše, najverovatnije taj PHP skript niste izvršili preko URL-a.

- Morate da znate koju verziju PHP-a koristite.

  Neki problemi nastaju zbog verzije PHP-a koja se koristi. Pre nego što počnete da koristite bilo koji server na kojem je omogućen PHP, izvršite preko njega skript **`phpinfo.php`** (pogledajte Dodatak A) ili ispitajte konstantu **`PHP_VERSION`** gde je upisana informacija o verziji koja se koristi.

- Opcija **`display_errors`** mora da bude uključena.

  Ovo je osnovno podešavanje PHP konfiguracije (takođe se opisuje u Dodatku A). To možete da proverite ako izvršite funkciju **`phpinfo()`** (pregledajte dobijenu stranicu u svom pretraživaču i proverite stavku *display_errors*). Može se desiti da je PHP iz bezbednosnih razloga podešen tako da ne prikazuje nastale greške. Ako je tako, u slučaju greške ćete ugledati samo praznu stranicu.Za rešavanje većine problema morate da vidite koja je greška prijavljena, zato potvrdite ovu opciju dok učite. U Dodatku A nalaze se uputstva kako se to radi.

- Proverite izvorni HTML kôd.

  Ponekad je problem sakriven u izvornom HTML-u za stranicu. U stvari, ponekad se tamo nalazi poruka o PHP grešci!

- Verujte poruci o grešci.

  Još jedna česta početnička greška je da se ne pročita cela poruka o grešci koju PHP prijavljuje, ili da joj se ne pokloni puno poverenje. Mada poruka o grešci može često da izgleda zagonetno i beznačajno, ne sme da se zanemari. U najmanju ruku, PHP obično ne greši bar što se tiče reda u kojem se greška može pronaći. U slučaju da je neophodno da tu poruku prenesete nekom drugom (na primer, kada se obratite meni za pomoć), svakako prepišite celu poruku o grešci!

- Napravite pauzu!

  Veliki broj programskih problema sa kojima sam se susretao tokom godina, rešen je, a to važi i za većinu onih najtežih, tako što se programer na neko vreme udaljio od računara. U takvim situacijama se čovek lako frustrira i zbuni, pa što god uradi može samo da pogorša situaciju.

Pregled naučenog i unapređivanje stečenih veština

Na kraju svakog poglavlja ove knjige nalazi se odeljak „Pregled naučenog i unapređivanje stečenih veština". U tom odeljku ćete naći pitanja koja se odnose na materiju koju smo upravo obradili, a zatim i predloge o načinima da sami proširite svoje znanje i iskustvo. Ako naiđete na neki problem, bilo u odgovaranju na pitanja ili u sprovođenju sopstvenih poduhvata, posetite forum ove knjige koji se nalazi na adresi: **`www.LarryUllman.com/forums/`**.

Pregled naučenog

- Koje oznake se koriste oko PHP koda?
- Koju ekstenziju treba da ima PHP datoteka?
- Šta znači *kodiranje stranice*? Kako kodiranje utiče na samu stranicu?
- Koje PHP funkcije ili jezičke konstrukcije možete da koristite da biste poslali podatke veb pretraživaču?
- Kako se jednostruki navodnici razlikuju od dvostrukih prilikom kreiranja stringova ili prilikom štampanja stringova?
- Šta podrazumevamo pod „izbeći specijalno značenje" nekog znaka u stringu?
- Koje tri sintakse postoje za komentare u PHP-u? Koja od njih može da se koristi za više redova?
- Kojim znakom počinju imena svih promenljivih? Koji znakovi mogu da se upotrebe nakon toga? Koji još znakovi mogu da se koriste u imenu promenljive?
- Da li se u imenima promenljivih velika slova razlikuju od malih?
- Koji operator koristimo za dodeljivanje?
- Kako se kreira promenljiva tipa string?
- Koji operator koristimo za nadovezivanje? Koji operator koristimo za dodeljivanje nadovezivanja?
- Kako se definišu i koriste konstante?

Unapređivanje stečenih veština

- Ako još ne znate – sasvim pouzdano – koju verziju PHP-a koristite, proverite to sada.
- Potražite neku od pomenutih funkcija za stringove u PHP priručniku. Nakon toga pogledajte i neku od ostalih dostupnih funkcija za stringove u priručniku.
- Potražite neku od pomenutih funkcija za brojeve u PHP priručniku. Nakon toga pogledajte i neku od ostalih dostupnih numeričkih funkcija u priručniku.
- Potražite promenljivu **`$_SERVER`** u PHP priručniku i proverite koje još informacije ona sadrži.
- Napravite sami novi skript koji definiše i prikazuje vrednosti nekih promenljivih tipa string. Upotrebite dvostruke znake navoda u naredbi **`echo`** ili **`print`** koja štampa vrednosti. Da biste sve malo zakomplikovali, dodajte nešto HTML-a u tekst. Zatim ponovo napišite isti skript, ali umesto dvostrukih upotrebite jednostruke navodnike i operator nadovezivanja (konkatenacije).
- Napravite potpuno novi skript koji definiše neke promenljive numeričkog tipa, vrši neke manipulacije sa njima i prikazuje vrednosti nekih numeričkih promenljivih.