1

Uvod u PHP

Mada je glavni predmet ove knjige kako da se PHP i MySQL koriste kao kombinacija, za veći deo rada koristićete samo PHP. U ovom i sledećem poglavlju, učićete osnove, od sintakse do promenljivih, operatora i jezičkih konstrukcija (uslova, petlji i šta sve ne). Istovremeno ćete praviti upotrebljiv kôd koji možete kasnije da uključite u veće aplikacije.

Ovo uvodno poglavlje vas vodi kroz veći deo osnova jezika PHP. Učićete sintaksu PHP kodiranja, kako da šaljete podatke veb pretraživaču i kako da koristite dve vrste promenljivih (stringove i brojeve) plus konstante. Neki od primera možda izgledaju irelevantni, ali će vam prikazati pojmove koje treba da savladate da biste kasnije pisali naprednije skriptove. Poglavlje se završava nekim savetima za brzo otkrivanje i uklanjanje grešaka... znate... za svaki slučaj.

U ovom poglavlju

Osnovna sintaksa 2
Slanje podataka u veb pretraživač 6
Pisanje komentara 10
Šta su to promenljive? 14
Uvođenje stringova 18
Sastavljanje stringova 21
Uvođenje brojeva 23
Uvođenje konstanti 26
Jednostruki i dvostruki navodnici 29
Osnovni koraci u otkrivanju grešaka 33
Pregled i unapređivanje 34

Osnovna sintaksa

Kao što je rečeno u uvodu knjige, PHP je jezik skriptovanja koji se *ugrađuje u HTML*, što znači da se PHP kôd i HTML kôd pomešaju u istom fajlu. Dakle, da biste počeli da programirate PHP, počnite od jednostavne veb stranice. **Skript 1.1** je primer XHTML Transitional dokumenta bez suvišnih ukrasa i bez sadržaja, koji će se koristiti kao temelj većine veb stranica u knjizi (u ovoj knjizi se [X]HTML formalno ne opisuje; više informacija potražite u izvorima posvećenim toj temi). Obratite takođe pažnju da se u ovom šablonu koristi kodiranje UTF-8, koje opisujemo u istaknutom komentaru.

Da biste na stranicu dodali PHP kôd, stavite ga među PHP oznake:

```
<?php

?>
```

Skript 1.1 osnovna veb stranica XHTML 1.0 Transitional

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD
   XHTML 1.0 Transitional//EN" "http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-
   transitional.dtd">
2  <html xmlns="http://www.w3.org/ 1999/
   xhtml" xml:lang="en" lang="en">
3  <head>
4    <meta http-equiv="Content-Type"
     content="text/html;
     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 se shvati sledeće: *kodiranje koje koristite u fajlu diktira koji znakovi mogu da se predstave* (pa, prema tome, koji jezici mogu da se koriste). Kada birate kodiranje, najpre treba da se uverite da vaš tekst editor ili IDE (Integrated Development Environment – integrisano okruženje za razvoj) – ono što koristite za pisanje HTML i PHP skriptova – može da sačuva dokumente sa tim kodiranjem. U nekim aplikacijama možete da postavite vrstu kodiranja u podešavanjima ili u delu sa opcijama, kod drugih se kodiranje određuje prilikom čuvanja fajla.

Da biste veb pretraživaču ukazali na vrstu kodiranja, imate odgovarajuću meta oznaku:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
```

Element **`charset=utf-8`** kaže da se koristi kodiranje UTF-8, što je skraćena oznaka za *8-bit Unicode Transformation Format*. Unicode je način da se pouzdano predstavi svaki simbol u svakom alfabetu. Unicode verzija 6 – trenutno važeća verzija dok ovo pišem – podržava preko 99.000 znakova!

Ako hoćete da pravite višejezičnu veb stranicu, treba da koristite UTF-8, a ja ću ga koristiti u primerima u ovoj knjizi. Naravno, to nije neophodno. Ali, koje god kodiranje da odaberete, pazite da kodiranje označeno na XHTML stranici odgovara kodiranju postavljenom u tekst editoru ili okruženju za razvoj (IDE). Inače ćete verovatno ugledati 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 PUBLIC "-//W3C//DTD
   XHTML 1.0 Transitional//EN" "http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-
   transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/
   xhtml" xml:lang="en" lang="en">
3  <head>
4    <meta http-equiv="Content-Type"
     content="text/html;
   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>
```

HTML 5

Dok je ovo pisano, aktivno se razvija i razmatra sledeća glavna verzija HTML-a – HTML5 – ali ona još nije gotova za proizvodnju, pa je zbog toga nisam koristio u ovoj knjizi. U stvari, ne bi me iznenadilo da HTML5 još uvek ne bude izdat kad budem počeo da pripremam peto izdanje ove knjige, a još će duže trajati da se taj jezik prihvati u većem broju pretraživača. Ipak, HTML5 je uzbudljiva perspektiva. U ovoj knjizi se mestimično pominju mogućnosti za koje se očekuje uvođenje i podrška.

Veb server će sve što se upiše između tih oznaka tretirati kao PHP, što znači da će PHP interpretator obraditi taj kôd. Sav tekst izvan PHP oznaka se trenutno šalje veb pretraživaču kao običan HTML. (Pošto se PHP najčešće koristi za pravljenje sadržaja koji će biti prikazan u veb pretraživaču, PHP oznake se obično postavljaju negde unutar tela (`<body> ... </body>`) stranice).

Osim postavljanja PHP koda među PHP oznake, vaši PHP fajlovi moraju da imaju odgovarajuću *ekstenziju*. Ekstenzija ukazuje serveru da taj skript treba da obradi na poseban način, naime, 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*! To može da bude na zakupljenom sajtu ili na vašem računaru, po uputstvima iz Dodatka A, »Instaliranje«, nakon besplatnog preuzimanja sa sajta peachpit.com.

Da biste napravili osnovni PHP skript:

1. Napravite nov dokument u svom tekst editoru ili okruženju za razvoj. Dokument će se zvati **`first.php`** (**Skript 1.2**).

   Uglavnom nije važno koju aplikaciju koristite, da li je to Adobe Dreamweaver (vrlo otmen IDE), TextMate (odličan i veoma popularan Macintosh editor teksta), ili vi (Unix editor čistog teksta, bez grafičkog interfejsa). Ipak, 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 da koristite Notepad*!). Ako još uvek nemate neku aplikaciju do koje vam je stalo,

potražite je na vebu ili se raspitajte na forumu pridruženom ovoj knjizi (**www.LarryUllman.com/forums/**).

2. Napravite osnovni HTML dokument:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD
→XHTML 1.0 Transitional//EN"
→"http://www.w3.org/TR/xhtml1/
DTD/
→xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/
→1999/xhtml" xml:lang="en"
→lang="en">
<head>
 <meta http-equiv="Content-Type"
→content="text/html;
→charset=utf-8"/>
 <title>Basic PHP Page</title>
</head>
<body>
 <!-- Script 1.2 - first.php -->
<p>This is standard HTML.</p>
</body>
</html>
```

Mada se ova sintaksa koristi u celoj ovoj knjizi, HTML možete da promenite prema bilo kojem standardu koji nameravate da koristite (Na primer, HTML 4.0 Strict). I opet, ako vam nešto iz tog HTML koda nije jasno, potražite neki resurs posvećen [X]HTML-u.

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

```
<?php
?>
```

To su *zvanične* PHP oznake, koje se zovu i oznake *tipa XML*. Mada PHP podržava i druge vrste oznaka, preporučujem da koristite zvanične oznake, a ja ću ih koristiti u celoj knjizi.

4. Sačuvajte fajl kao **first.php**.

Ne zaboravite da se skript neće pravilno izvršavati ako ne sačuvate fajl sa odgovarajućom PHP ekstenzijom. (Jedan od razloga da se ne koristi Notepad je da će on na PHP fajl tajno da doda ekstenziju **.txt** i tako da stvori mnoge glavobolje.)

5. Smestite fajl u odgovarajući direktorijum svog veb servera.

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

Ako izvršavate PHP na zakupljenom serveru (tj. na udaljenom računaru), biće vam potrebna FTP aplikacija (File Transfer Protocol) da prenesete (*engl.* upload) fajl u odgovarajući direktorijum. Kompanija kod koje iznajmljujete serverske kapacitete će vam dostaviti potrebne informacije za pristup.

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

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



(A) Iako izgleda kao svaka druga (jednostavna) HTML stranica, ovo je u stvari PHP skript i temelj za ostale primere u knjizi

Ako PHP izvršavate na vlastitom računaru, moraćete da koristite neki URL kao što je **`http://localhost/first.php`**, **`http://127.0.0.1/first.php`** ili **`http://localhost/~<user>/first.php`** (na Mac OS X-u ćete svoje stvarno korisničko ime da stavite umesto **`<user>`**). Ako koristite zakupljeni server na Vebu, moraćete da koristite **`http://`*`vaše-ime-domena`*`/first.php`** (npr, **`http://www.example.com/first.php`**).

7. Ako ne budete videli 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 validnu veb stranicu, postupite na sledeći način:

1. Uverite se da imate aktivnu PHP instalaciju (u Dodatku A imate uputstva kako da to proverite).
2. Proverite da li ste skript pokrenuli preko URL-a. Adresa u veb pretraživaču mora da počinje sa **`http://`**. Ako počinje sa **`file://`** problem je u tome **(B)**.
3. Ako vam se javlja poruka da fajl nije pronađen (»file not found«) ili nešto slično, verovatno ste postavili fajl u pogrešan direktorijum ili ste pogrešno otkucali ime fajla (bilo prilikom pohranjivanja, bilo u veb pretraživaču).

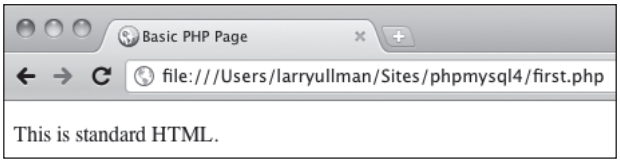
   Ako ste sve ovo proverili, a još uvek imate probleme, pregledajte forum posvećen ovoj knjizi (**www.LarryUllman.com/forums/**).

SAVET **Više informacija o HTML-u i XHTML-u možete da pronađete u izvrsnoj knjizi Elizabete Castro: *HTML, XHTML, and CSS, Sixth Edition: Visual QuickStart Guide* (Peachpit Press, 2006) ili tražite po Vebu.**

SAVET **Moguće je u isti HTML dokument ubaciti više odeljaka PHP koda (tj. možete da prelazite iz jednog jezika u drugi). Na takve primere ćete nailaziti u celoj ovoj knjizi.**

SAVET **Pre kodiranja UTF-8, jedno od najčešće korišćenih kodiranja bilo je ISO-8859-1. Ono predstavlja većinu zapadnoevropskih jezika. To je još uvek podrazumevano kodiranje u mnogim veb pretraživačima i u drugim aplikacijama.**

SAVET **Kodiranje eksternog CSS fajla možete da deklarišete tako što kao prvi red u fajlu upišete `@charset "utf-8"`. Ako ne koristite UTF-8, promenite taj red na odgovarajući način.**



(B) PHP kôd može da se izvrši samo kroz *http://* (ovaj konkretan skript ne zavisi od toga).

Slanje podataka u veb pretraživač

Da biste pravili dinamične veb sajtove pomoću PHP-a, morate znati kako da podatke pošaljete veb pretraživaču. PHP ima niz ugrađenih funkcija za tu svrhu, uobičajene su **`echo`** i **`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 funkcija radi):

```
print 'Hello, World!;
```

```
print "What's new?";
```

Kao što vidite u ovim primerima, mogu da se koriste jednostruki ili dvostruki navodnici (mada se te dve vrste navodnika razlikuju, što će se razjasniti još 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, trebalo bi da obratite pažnju na to da u PHP-u sve naredbe – laički rečeno, svaki red izvršnog koda – treba završiti znakom tačka zarez. Osim toga, PHP *ne razaznaje razliku između velikih i malih slova* u imenima funkcija. Prema tome, isto je da li napišete **ECHO** ili **echo** ili **eCHo** , sve važi. Naravno, najlakše je kad pišete sve malim slovima.

Potreba da se izbegne specijalno značenje

Ubrzo ćete videti da jednu od komplikacija prilikom slanja podataka pretraživaču predstavlja pitanje kako da mu se pošalje jednostruki ili dvostruki navodnik. Ove dve varijante izazvaće grešku:

```
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.";
```

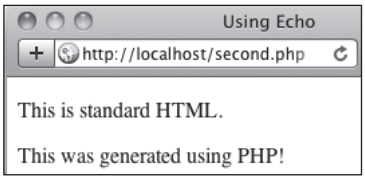
Drugo rešenje je da se specijalno značenje navodnika izbegne (obrnutom kosom crtom \):

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

Poništeni navodnik će prosto da se štampa kao i svaki drugi znak. Važno je da se razume upotreba obrnute kose crte kojom se izbegava specijalno značenje. Pri kraju poglavlja biće još reči o tome.

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

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD
   XHTML 1.0 Transitional//EN" "http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-
   transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/
   xhtml" xml:lang="en" lang="en">
3  <head>
4    <meta http-equiv="Content-Type"
     content="text/html; 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>
```



(A) Rezultati još uvek nisu bog zna kakvi, ali ovu stranicu je delom dinamički napravio PHP.

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 dodajte jednostavnu poruku (Skript 1.3):

   ```
   echo 'This was generated using PHP!';
   ```

   U stvari, sasvim je nevažno koju poruku stavljate, koju funkciju ili koje navodnike koristite – samo pazite ako poruka sadrži jednostruke ili dvostruke navodnike (istaknuti komentar »Potreba da se izbegne specijalno značenje«).
3. Ako želite, promenite naslov stranice u skladu sa sadržajem skripta (red broj 5).

   ```
   <title>Using Echo</title>
   ```

   Ova izmena utiče samo na naslovnu liniju u prozoru pretraživača.
4. Sačuvajte fajl kao **second.php** u svom veb direktorijumu i testirajte ga u svom pretraživaču **(A)**.

   Nemojte da zaboravite da svi PHP skriptovi moraju da se izvršavaju kao URL (**http://*nešto***)!

nastavalja se na sledećoj strani

5. Ako je potrebno, 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 istaknuti komentar). Takođe proverite da li ste svaku naredbu završili znakom tačka zarez.

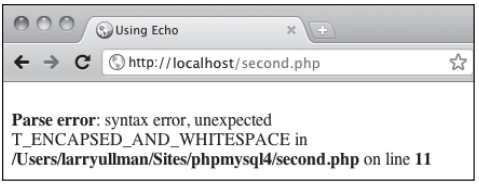
Ako dobijete čistu belu stranicu, to bi moglo da bude iz jednog od sledeća dva razloga:

- Problem je u HTML-u. Pregledajte HTML naredbe u izvornom kodu vaše stranice **(C)**.
- Došlo je do greške, ali je u konfigurisanju PHP-a isključeno prikazivanje grešaka (*display_errors*) i zato se ništa ne prikazuje. U tom slučaju, pročitajte odeljak o konfigurisanju PHP-a u Dodatku A, da biste ponovo uključili *display_errors*.

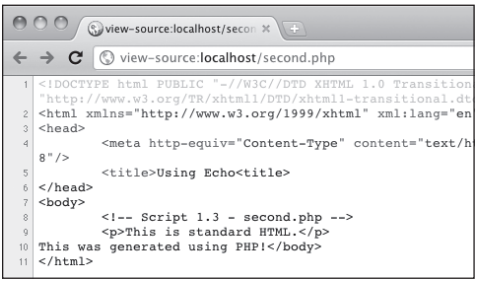
SAVET **Strogo govoreći `echo` i `print` su *jezičke konstrukcije* a ne funkcije u pravom smislu. Pošto sam to objasnio, nemojte da vas buni što ih i dalje iz praktičnih razloga nazivam »funkcijama«. Osim toga, videćete kasnije u knjizi da dodajem zagrade kada opisujem funkcije – na primer `number_format()` a ne prosto `number_format` – da bi se funkcije lakše razlikovale od promenljivih i od drugih elemenata PHP-a. To je prosto moja lična mala konvencija.**

SAVET **Za slanje HTML koda u pretraživač se može, i sami ćete često to raditi, koristiti `echo` i `print`, ovako (D):**

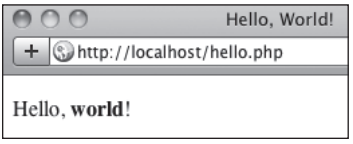
```
echo '<p>Hello, <b>world</b>!</p>';
```



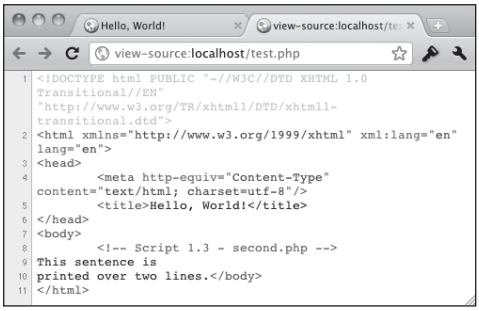
(B) Ovo bi mogla da bude prva od mnogih poruka o greškama raščlanjivanja na koje ćete nailaziti kao PHP programer (ova se javlja zato što nedostaje završni znak navoda).



(C) Jedan od razloga da se pojavi prazna PHP stranica 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 (ovde je to kôd za formatiranje) a ne samo običan tekst **(A)**.



(E) Kad 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 stranice **(F)** ali 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 je da se upotrebe HTML oznake (na primer, **
** ili ** **).

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

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

SAVET **U ovom slučaju će prelaz u novi red (dobijen pritiskom na Enter ili Return) postati 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 se neće odraziti na generisanoj stranici (F). Opširnije objašnjenje ove teme nalazi se u istaknutom komentaru »Uloga razmaka«).**

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. Kad imate to u vidu, postoje tri područja sa primetnim *praznim prostorom* (dodatni razmaci, tabulatori i prazni redovi): 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 kôd da razmičete kako god vam je zgodno da bi on bio čitljiviji. HTML takođe uglavnom ne reaguje na prazan prostor. Konkretno, jedini prazan prostor u HTML-u koji utiče na konačnu veb stranicu je jednostruki razmak (ako ih stavite nekoliko, opet dobijate jedan). Ako u izvornom HTML-u imate tekst na više redova, to ne znači da će na gotovoj stranici to biti više redova (**(E)** i **(F)**).

Da biste promenili prazan prostor na gotovoj veb stranici, upotrebite HTML oznake **
** (novi red, u starijim HTML standardima je bilo **
**) ili ** ** (pasus). Da biste postigli prazan prostor u *izvornom HTML*-u koji će PHP da napravi možete da

- upotrebite **`echo`** ili **`print`** na više redova sfjk skdjlfasdf lasdal sdjf alsdfj alsdjf halsdjf jklč dsdgjčgj lčsdfkjg čsdfk

  ili

- pošaljete znak za novi red (**`\n`**) unutar dvostrukih navodnika, što je ekvivalentno sa Enter ili Return.

Pisanje komentara

Pravljenje izvršnog PHP koda je samo deo programiranja (priznajem, to je najvažniji deo). Manje važan, ali ipak bitan aspekt svakog programerskog poduhvata je dokumentovanje izvršnog koda. U stvari, kada me pitaju po čemu se programer početnik razlikuje od iskusnog programera, moj nepokolebljiv odgovor jeste da je razlika u dobrim i temeljnim komentarima.

U HTML-u se komentari dodaju pomoću specijalnih oznaka:

```
<!-- Ovde dolazi komentar. -->
```

HTML komentari se vide 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 (#):

```
# Ovo je komentar
```

Druga su dve kose crte:

```
// Ovo je takođe komentar
```

U oba slučaja PHP ignoriše sve nakon toga do kraja reda (kada pritisnete Enter ili Return). Ti su znakovi, dakle, samo za pojedinačni red. Oni se često koriste da bi se komentar stavio u isti red sa PHP kodom:

```
print 'Hello!'; // štampanje
pozdrava
```

Treći način omogućava prostiranje komentara preko više redova:

```
/* Ovo je dugačak komentar koji
se prostire na više
redova. */
```

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

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD
   XHTML 1.0 Transitional//EN" "http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-
   transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/
   xhtml" xml:lang="en" lang="en">
3  <head>
4    <meta http-equiv="Content-Type"
     content="text/html; 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:

1. Otvorite u tekst editoru ili u IDE okruženju nov PHP dokument, nazvaćete ga **comments.php** a u početku će da sadrži samo HTML kôd (**Skript 1.4**):

```
<!DOCTYPE html PUBLIC "-//W3C//
→DTD XHTML 1.0 Transitional//
EN"
→"http://www.w3.org/TR/xhtml1/
DTD/
→xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/
→1999/xhtml" xml:lang="en"
→lang="en">
<head>
  <meta http-equiv="Content-Type"
  →content="text/html;
  →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 March 16, 2011
# Created by Larry E. Ullman
# This script does nothing much.
```

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

nastavalja se na sledećoj strani

3. Pošaljite malo HTML-a 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, važno je da to bude nešto što će pretraživač da prikaže. Radi raznovrsnosti, naredba **echo** šalje i neke HTML oznake, uključujući znak za novi red (**
**) da bi se na veb stranici dobilo nešto razmaka.

4. Poslužite se komentarom od više redova da pretvorite drugu naredbu **echo** u komentar

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

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

5. Dodajte na kraju komentar iza treće **echo** naredbe

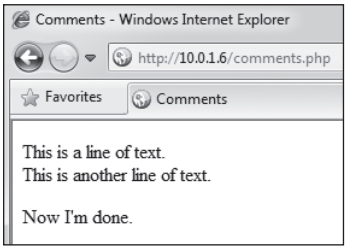
```
echo "<p>Now I'm done.</p>";
→// Kraj PHP koda.
```

Ovaj poslednji (suvišan) komentar sam stavio da bih prikazao kako se komentar stavlja na kraj reda, što se često koristi. Obratite pažnju na to da su za poruku upotrebljeni dvostruki navodnici, jer bi jednostruki bili u konfliktu sa apostrofom koji poruka sadrži (pročitajte istaknuti komentar »Potreba da se izbegne specijalno značenje« ranije u ovom poglavlju).

6. Zaključite PHP odeljak i dovršite HTML stranicu:

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

7. Sačuvajte fajl pod imenom **comments.php** u svom veb direktorijumu i isprobajte ga 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 **(B)**.

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

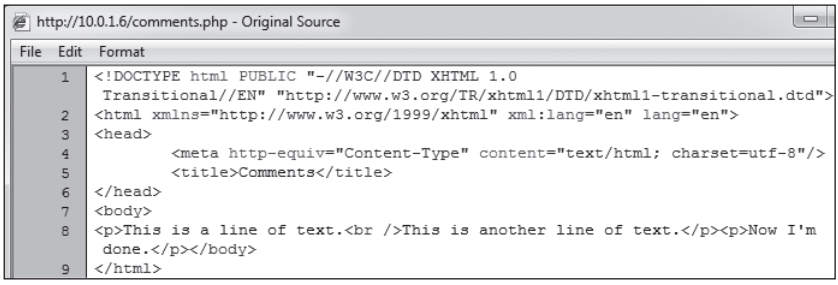
SAVET **Sve vrste PHP komentara mogu da se postave na kraj reda koda (na primer iza poziva funkcije):**

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

Mada je dozvoljeno, ovo nije tako često.

SAVET **Skoro je nemoguće da se pretera u komentarisanju skripta. Uvek je bolje da stavite previše komentara nego nedovoljno. Imajte to u vidu, mada zbog uštede prostora skriptovi u ovoj knjizi nisu dokumentovani koliko bih ja preporučio.**

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



(B) 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 vrsta promenljivih. Postoje četiri skalarna tipa (pojedinačne vrednosti) – *Boolean* (sa vrednostima tačno-**TRUE** i netačno-**FALSE**), *integer, floating point* (sa decimalima) i *string* (znakovi); dva neskalarna tipa (više vrednosti) – *array* i *object*; plus *resource* (koje ćete videti u radu sa bazama podataka) i *NULL* (što predstavlja specijalan tip koji nema vrednost).

Bez obzira na to koji tip pravite, u PHP-u postoje sintaktička pravila za sva imena promenljivih:

- Ime promenljive mora da počinje dolarskim znakom (**$**), na primer **$ime**.
- Ime promenljive sme da 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 sasvim različite promenljive.

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

Pre nego što pređemo na ovaj skript, treba da znate još dve stvari. Prvo, promenljivama se vrednost može dodeliti pomoću znaka jednakosti (**=**), koji se takođe zove *operator dodeljivanja* (*engl.* assignment operator). Drugo, da biste prikazali vrednost promenljive, možete da štampate promenljivu bez znakova navoda:

```
print $neka_prom;
```

Ili možete da štampate promenljivu unutar *dvostrukih* navodnika:

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

Vrednost promenljive ne može da se prikaže unutar *jednostrukih* navodnika:

```
print 'Hello, $ime'; // neće uspeti!
```

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

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD
   XHTML 1.0 Transitional//EN" "http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-
   transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/
   xhtml" xml:lang="en" lang="en">
3  <head>
4    <meta http-equiv="Content-Type"
   content="text/html; 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 = $_SERVER[,SERVER_
   SOFTWARE'];
14
15 // Štampaj ime ovog skripta:
16 echo "<p>You are running the
   file:<br /><b>$file</b>.</p>\n";
17
18 // Štampaj informacije o korišćenju:
19 echo "<p>You are viewing this page
   using:<br /><b>$user</b></p>\n";
20
21 // Štampaj informacije o serveru:
22 echo "<p>This server is running:
   <br /><b>$server</b>.</p>\n";
23
24 ?>
25 </body>
26 </html>
```

Da biste koristili promenljive:

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

```
<!DOCTYPE html PUBLIC «-//W3C//
→DTD XHTML 1.0 Transitional//EN"
→"http://www.w3.org/TR/xhtml1/DTD/
→xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/
→1999/xhtml» xml:lang="en"
→lang="en">
<head>
  <meta http-equiv="Content-Type"
→content="text/html;
→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 tako dalje, mada bi vi svoje skriptove trebalo temeljno da dokumentujete. U našim skriptovima će se ipak nalaziti komentari sa brojem skripta i imenom fajla, radi lakšeg unakrsnog referenciranja (u knjizi i kada ih preuzimate sa sajta pridruženog knjizi, **www.LarryUllman.com**).

nastavalja se na sledećoj strani

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

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

U ovom skriptu će se koristiti tri promenljive koje sve potiču iz veće unapred definisane 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**. Napravili smo novu promenljivu kraćeg imena i dodelili joj vrednost iz promenljive **$_SERVER**, pa će nam biti lakše da je koristimo prilikom štampanja. (Time se rešava još jedan problem, ali o tome će biti reči kasnije).

4. Napravite skraćene verzije još 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 (vidi Dodatak A) da 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 /><b>$file</b>.</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 postiže novi red u generisanom izvornom HTML kodu. Neke osnovne HTML oznake – za pasus i za tamna slova – stavljene su da bi generisana stranica bila privlačnija.

6. Štampajte informaciju o korisniku koji pristupa skriptu:

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

Ovim redom se štampa druga promenljiva **$user**. Da ponovimo ono što je rečeno u četvrtoj tački, **$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 /><b>$server</b>.</p>\n";
```

8. Zaključite PHP blok i HTML stranicu:

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

9. Sačuvajte fajl pod imenom **`predefined.php`** u svom veb direktorijumu i isprobajte ga u veb pretraživaču **(A)**.

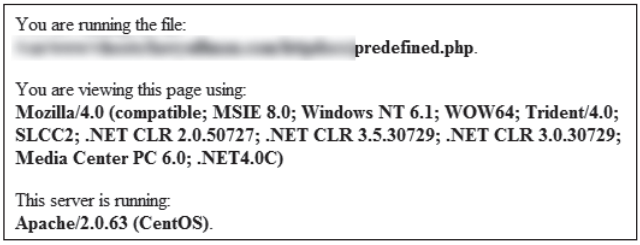
SAVET **Ako imate problem sa ovim ili bilo kojim drugim skriptom, potražite pomoć na forumu posvećenom ovoj knjizi (`www.LarryUllman.com/forums/`).**

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

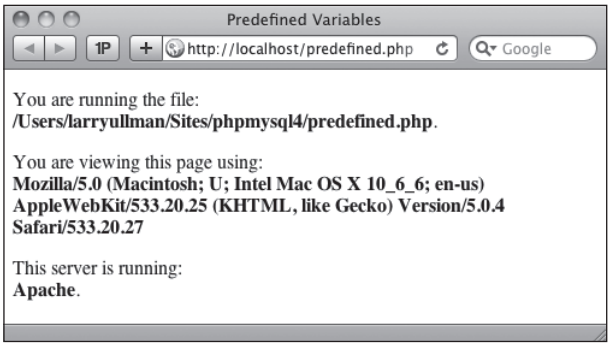
SAVET **Ime promenljive ne sme da sadrži razmak. Umesto razmaka obično se koristi donja crta.**

SAVET **Najvažnije pitanje prilikom pravljenja promenljivih je da se koristi konzistentna šema imenovanja. U ovoj knjizi ćete videti da ja za imena promenljivih koristim samo mala slova, a reči odvajam donjom crtom (`$first_name`). Neki programeri više vole da umesto donje crte koriste veliko slovo: `$FirstName` (kamilja notacija – camel-case style).**

SAVET **PHP nije strog 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.**



(A) Skript **`predefined.php`** prenosi gledaocu informacije o skriptu, veb pretraživaču koji se koristi za to prikazivanje i o samom serveru.



(B) Ovo je prvi stvarno dinamičan skript u knjizi, po tome što se veb stranica menja zavisno od servera na kojem se izvršava i od veb pretraživača kojim se posmatra (uporedite sa **(A)**).

Uvođenje stringova

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

- 'Tobias'
- "In watermelon sugar"
- '100'
- 'August 2, 2011'

Da bi se napravila promenljiva tipa string, validnom imenu promenljive se dodeli vrednost tipa string:

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

Kada se pravi string, mogu se oko znakova koristiti jednostruki ili dvostruki navodnici, isto kao kada se štampa tekst. Isto tako, string mora da se otvori i zatvori istom vrstom navodnika. Ako se ista vrsta navodnika javlja unutar stringa, mora se izbeći njegovo specijalno značenje:

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

Druga mogućnost je 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 da se stave 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 sledećem primeru ćete napraviti i upotrebiti vlastite stringove.

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

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD
   XHTML 1.0 Transitional//EN" "http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-
   transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/
   xhtml" xml:lang="en" lang="en">
3  <head>
4    <meta http-equiv="Content-Type"
     content="text/html; charset=utf-8"/>
5    <title>Strings</title>
6  </head>
7  <body>
8  <?php # Script 1.6 - strings.php
9
10 // Pravljenje 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:

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 PUBLIC "-//W3C//
→DTD XHTML 1.0 Transitional//EN"
→"http://www.w3.org/TR/xhtml1/DTD/
→xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/
→1999/xhtml" xml:lang="en"
→lang="en">
<head>
  <meta http-equiv="Content-Type"
  →content="text/html;
  →charset=utf-8"/>
  <title>Strings</title>
</head>
<body>
<?php # Script 1.6 - strings.php
```

2. U okviru PHP oznaka, napravite tri promenljive:

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

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

nastavalja se na sledećoj strani

3. Dodajte naredbu **echo**:

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

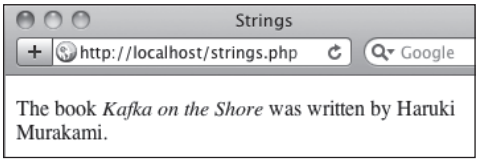
Ovaj skript samo štampa izjavu o autorstvu pomoću tri utvrđene promenljive. Malo HTML formatiranja (isticanje naslova knjige) ubačeno je da bi tekst izgledao lepše. Ne zaboravite da stavite dvostruke navodnike da bi se pravilno prikazao sadržaj promenljivih (o značaju dvostrukih navodnika biće reči i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

4. Zaključite PHP blok i HTML stranicu:

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

5. Sačuvajte fajl pod imenom **strings.php** u svom veb direktorijumu i isprobajte ga u veb pretraživaču **(A)**.

6. Po želji, promenite vrednosti tri promenljive, sačuvajte fajl i ponovo izvršite skript.



(A) Veb stranica je dobijena štampanjem vrednosti tri promenljive.



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

SAVET **Ako dodelite drugu vrednost nekoj postojećoj promenljivoj (na primer $book), nova vrednost će preklopiti 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 stvarno imati takav problem.**

Skript 1.7 Konkatenacija omogućava pripajanje dodatnih znakova na string.

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD
   XHTML 1.0 Transitional//EN" "http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-
   transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/
   xhtml" xml:lang="en" lang="en">
3  <head>
4    <meta http-equiv="Content-Type"
     content="text/html; charset=utf-8"/>
5    <title>Concatenation</title>
6  </head>
7  <body>
8  <?php # Script 1.7 - concat.php
9
10 // Pravljenje 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>
```

Sastavljanje stringova

Sastavljanje (*konkatenacija*) je kao sabiranje za stringove, tako što 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** sada ima vrednost *SeattleWashington* što je takoreći ono što se želelo (*Seattle, Washington*). Da bismo doterali rezultat, mogli bismo da napišemo:

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

tako se osim promenljivih sastavlja još i zarez sa razmakom. Pošto PHP nije strog što se tiče promenljivih, moguće je sastaviti stringove i brojeve. Obe ove naredbe proizvešće isti rezultat (*Seattle, Washington 98101*):

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

Hajde da promenimo skript **scripts.php** tako da primenimo ovaj novi operator.

Korišćenje sastavljanja:

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 ():

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

   Za demonstraciju sastavljanja, napraviće se nova promenljiva – **$author** – kao konkatenacija dva postojeća stringa i razmaka između njih.

nastavalja se na sledećoj strani

3. Izmenite naredbu **echo** tako da se upotrebi ova 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** i **$book**.

5. Sačuvajte fajl pod imenom **concat.php** u svom veb direktorijumu i isprobajte ga u veb pretraživaču **(A)**.

SAVET **PHP sadrži masu funkcija koje se odnose na stringove, videćete to u ovoj knjizi. Na primer, da biste izračunali dužinu stringa (koliko znakova sadrži) upotrebićete funkciju `strlen()`:**

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

SAVET **Možete da zadate da PHP pretvori veličinu slova. Sa `strtolower()` se dobiju sve mala slova, sa `strtoupper()` se dobiju sve velika, sa `ucfirst()` se dobija veliko prvo slovo rečenice a sva ostala mala, a sa `ucwords()` se dobija veliko prvo slovo svake reči a ostala mala.**

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

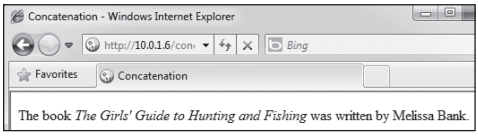
```
$title = $title . $subtitle;
$title .= $subtitle;
```

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

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

ili

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



(A) U ovom izmenjenom skriptu, krajnji rezultat sastavljanja koje korisnik ne primećuje.

PHP priručnik

PHP priručnik – može se naći na adresi **www.php.net/manual** - sadrži spisak svih funkcija i mogućnosti jezika. 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 potražite **www.php.net/*funkcija*** (na primer **www.php.net/print**). Za svaku funkciju priručnik navodi:

- 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.

Priručnik takođe sadrži opis funkcije.

Trebalo bi da steknete naviku da proverite u priručniku kad god ste u dilemi u vezi sa nekom funkcijom i njenim pravilnim korišćenjem, ili kada vam treba da više naučite o nekoj mogućnosti jezika. Takođe 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 rekao sam da PHP ima numeričke tipove integer (cele brojeve) i floating-point (decimale). Međutim, po mom iskustvu, oba ova tipa mogu da se klasifikuju pod opšti pojam *brojevi* (*engl.* numbers) bez ikakvog gubitka (uglavnom). 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 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 aritmetičkih operatora koje možete da primenite na brojeve (**Tabela 1.1**), u PHP je ugrađeno na desetine funkcija. Dve funkcije koje se vrlo često koriste su **`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 = 3.142857;
$n = round ($n, 3); // 3.143
```

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 proizvede zadati broj decimalnih mesta:

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

Da bismo vežbali sa brojevima, 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:

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

```
<!DOCTYPE html PUBLIC "-//W3C//
→DTD XHTML 1.0 Transitional//EN"
→"http://www.w3.org/TR/xhtml1/DTD/
→xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/
→1999/xhtml" xml:lang="en"
→lang="en">
<head>
  <meta http-equiv="Content-Type"
  →content="text/html;
  →charset=utf-8"/>
  <title>Numbers</title>
</head>
<body>
<?php # Script 1.8 - numbers.php
```

2. Uspostavite potrebne promenljive (količina, cena i procenat poreza):

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

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

3. Izvedite izračunavanja:

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

   U prvom redu se izračunava iznos porudžbine kao broj komada pomnožen sa jediničnom cenom. U drugom redu se zatim ukupnom iznosu dodaje iznos poreza (koji se izračunava množenjem zbira sa procentom poreza).

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

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD
   XHTML 1.0 Transitional//EN" "http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-
   transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/
   xhtml" xml:lang="en" lang="en">
3  <head>
4    <meta http-equiv="Content-Type"
     content="text/html; charset=utf-8"/>
5    <title>Numbers</title>
6  </head>
7  <body>
8  <?php # Script 1.8 - numbers.php
9
10 // Postavljanje promenljivih:
11 $quantity = 30; // Kupovanje 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 <b>' .
   $quantity . '</b> widget(s) at a cost
   of <b>$' . $price . '</b> each. With
   tax, the total comes to <b>$' .
   $total . '</b>.</p>';
24
25 ?>
26 </body>
27 </html>
```

4. Formatirajte ukupan iznos:

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

Funkcija **number_format()** će grupisati ukupan iznos u hiljade i zaokružiće ga na dve decimale. Primenom ove funkcije dobija se pravilan format izračunate vrednosti.

5. Štampajte rezultate:

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

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

6. Zaključite PHP kôd i HTML stranicu:

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

7. Sačuvajte fajl pod imenom **numbers.php** u svom veb direktorijumu i isprobajte ga u veb pretraživaču **(A)**.

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

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 operacije). Mada se u PHP priručniku i drugim izvorima navodi hijerarhija prvenstva, ja smatram da je programiranje sigurnije i čitljivije kada se redosled izračunavanja nametne zagradama (pogledajte red 17 u skriptu 1.8).**

SAVET **Poznato je koliko su računari loši u radu sa decimalama. Na primer, broj 2.0 može da se čuva kao 1.99999. U većini slučajeva to neće predstavljati problem, ali ako je najvažnija matematička preciznost, pouzdajte se u cele brojeve a ne u decimalne. PHP priručnik sadrži informacije o ovoj temi, a takođe i alternativne funkcije za veću preciznost izračunavanja.**

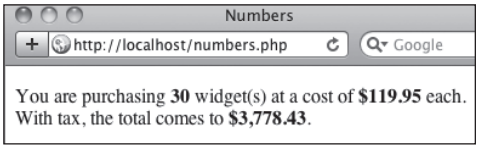
SAVET **Mnogi matematički operatori imaju i odgovarajući operator dodeljivanja, što predstavlja skraćeno dodeljivanje vrednosti. Ovaj red:**

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

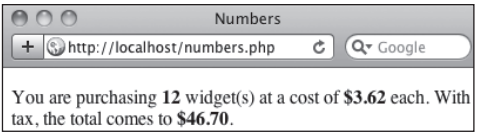
bi mogao da se zameni sa:

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

SAVET **Ako prilikom određivanja vrednosti `$price` ne stavite dve decimale (npr. 119.9 ili 34), biće korisno da na `$price` pre štampanja primenite funkciju `number_format()`.**



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



(B) Da biste promenili generisanu veb stranicu, izmenite bilo koju promenljivu ili sve tri (uporedite sa **(A)**).

Uvođenje konstanti

Konstante se, slično promenljivima, koriste za privremeno čuvanje vrednosti, ali se inače konstante i promenljive razlikuju na više načina. Kao prvo, za pravljenje konstante se koristi funkcija **define()** a ne operator dodeljivanja (=).

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

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

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
```

Ovo će PHP štampati bukvalno kao *Hello USERNAME* **(A)** a neće staviti sadržaj 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 sastavljanje:

```
echo "Hello, " . USERNAME;
```

PHP sadrži nekoliko unapred definisanih konstanti, slično unapred definisanim promenljivima 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.



(A) Konstante ne mogu da se koriste unutar navodnika.

Skript 1.9 Konstante su drugi način za privremeno smeštanje podataka koje možete koristiti u PHP-u za razliku od promenljivih.

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD
   XHTML 1.0 Transitional//EN" "http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-
   transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/
   xhtml" xml:lang="en" lang="en">
3  <head>
4    <meta http-equiv="Content-Type"
     content="text/html; 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', ,March 16, 2011');
12
13 // Štampanje poruke, uz upotrebu
   unapred definisanih konstanti i
   konstante TODAY:
14 echo '<p>Today is ' . TODAY . '.<br />
   This server is running version <b>' .
   PHP_VERSION . '</b> of PHP on the <b>'
   .
   PHP_OS . '</b> operating system.</p>';
15
16 ?>
17 </body>
18 </html>
```

Da biste koristili konstante:

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

```
<!DOCTYPE html PUBLIC "-//W3C//
→DTD XHTML 1.0 Transitional//EN"
→"http://www.w3.org/TR/xhtml1/DTD/
→xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/
→1999/xhtml" xml:lang="en"
→lang="en">
<head>
 <meta http-equiv="Content-Type"
 →content="text/html;
 →charset=utf-8"/>
 <title>Constants</title>
</head>
<body>
<?php # Script 1.9 - constants.
php
```

2. Napravite novu konstantu za datum

```
define (,TODAY', ,March 16, 2011');
```

   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 <b>' . PHP_VERSION .
→'</b> of PHP on the <b>' . PHP_OS .
→'</b> operating system.</p>';
```

   Pošto konstante ne mogu da se štampaju unutar znakova navoda, u nared-

nastavalja se na sledećoj strani

bi **echo** koristite operator sastavljanja.

4. Zaključite PHP kôd i HTML stranicu:

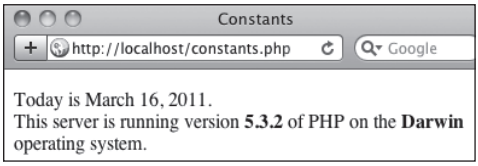
```
?>
</body>
</html>
```

5. Sačuvajte fajl pod imenom **constants.php** u svom veb direktorijumu i isprobajte ga u veb pretraživaču **(B)**.

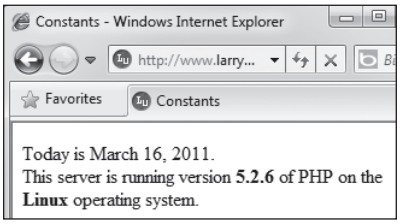
SAVET Ako imate mogućnosti, izvršite ovaj skript na nekom drugom serveru na kojem može da se koristi PHP (C).

SAVET Operativni sistem po imenu *Darwin* (B) je tehnički naziv za Mac OS X.

SAVET U poglavlju 12 »Kolačići i sesije«, učićete o još jednoj konstanti, SID (identifikacija sesije).



(B) Kad budete upotrebili PHP konstante, videćete detalje svoje PHP postavke.



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

Jednostruki i dvostruki navodnici

Za PHP je važno razumeti kako se jednostruki navodnici razlikuju od dvostrukih. 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 njihov bukvalni izgled. Na primer, uzmimo da imate

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

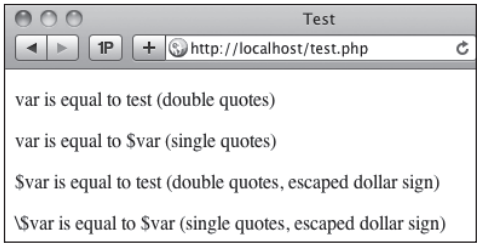
TABELA 1.2 Kontrolne (izlazne) sekvence

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* **(A)**.

Kao što bi iz ovih primera trebalo da se vidi, dvostruki navodnici će 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) štampa se kao samo jedna obrnuta kosa crta.

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



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

Da bi se koristili jednostruki i dvostruki navodnici:

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

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

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

Međutim, tu postoji začkoljica: ako želite da štampate iznos u dolarima kao $12.34 (gde 12.34 potiče iz promenljive), najpre biste pomislili da stavite **$$var**. To neće moći (razlozi su komplikovani). 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 prilikom korišćenja jednostrukih i dvostrukih navodnika.

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD
   XHTML 1.0 Transitional//EN" "http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-
   transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/
   xhtml" xml:lang="en" lang="en">
3  <head>
4    <meta http-equiv="Content-Type"
     content="text/html; 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
   <b>$quantity</b> widget(s) at a cost
   of <b>\$$price</b> each. With tax,
   the total comes to <b>\$$total</b>.
   </p>\n";
25
26 // Štampanje rezultata sa
   jednostrukim navodnicima:
27 echo '<h3>Using single quotation
   marks:</h3>';
28 echo '<p>You are purchasing
   <b>$quantity</b> widget(s) at a cost
   of <b>\$$price</b> each. With tax,
   the total comes to <b>\$$total</b>.
   </p>\n';
29
30 ?>
31 </body>
32 </html>
```

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

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

   Ova naredba **echo** se ovde koristi da bi se istakla razlika u korišćenju jednostrukih ili dvostrukih navodnika. Nećete dobiti željeni rezultat, a na dobijenoj stranici se vidi tačno šta se dešava.

5. Ako želite, promenite naslov stranice.

6. Sačuvajte fajl pod imenom **quotes.php** u svom veb direktorijumu i isprobajte ga u veb pretraživaču **(B)**

7. Pregledajte 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. Kada ga stavite u jednostruke navodnike, dobijate same znakove \ i n.

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

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

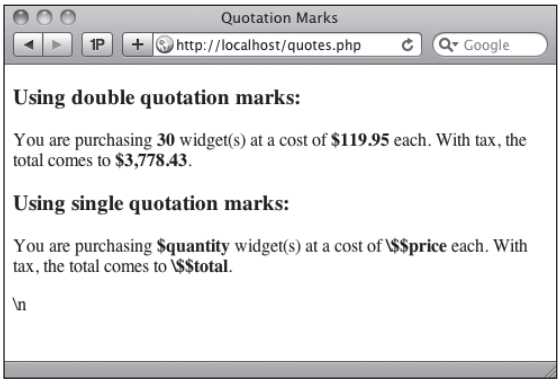
```
echo '<table width="80%" border="0"
→cellspacing="2" cellpadding="3"
→align="center">';
```

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

```
echo "<table width=\"80%\" border=
→\"0\" cellspacing=\"2\" cellpadding=
→\"3\" align=\"center\">";
```

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 moguće. Da biste garantovali pouzdane rezultate, koji neće zavisiti od verzije PHP-a, preporučujem da se držite sintakse `\$$var` kada treba da štampate dolarski znak a odmah zatim vrednost promenljive.**

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



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

Osnovni koraci za otkrivanje grešaka

Otkrivanje i uklanjanje grešaka (*engl.* debugging) nikako nije jednostavno za razumevanje, a nažalost, moguće ga je savladati jedino u praksi. Mogli bismo sledećih 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 što vam predstavljam ovu temu na tako preteći način je zato što je važno da se ne zavaravate kad se prihvatate programiranja. Ponekad vaš kôd neće raditi ono što od njega očekujete, neizostavno ćete praviti greške iz nepažnje, a biće dana kad ćete čupati kosu, čak i uz jezik relativno blizak korisniku kakav je PHP. Ukratko, budite spremni da ponekad budete zbunjeni i frustrirani. Ja koristim PHP još od 1999. godine pa se još uvek povremeno zaglibim u programerskom blatu. Ali otkrivanje i otklanjanje grešaka je veoma važna veština koju ćete vremenom da savladate iz nužde i iskustva. Kako počinjete da se upuštate u avanturu PHP programiranja, ponudiću vam sledeće osnovne ali konkretne savete za otkrivanje 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 ispravljanje grešaka« detaljnije se obrađuju druge tehnike.

Da biste pronašli i rešili greške u PHP skriptu:

- **Pazite 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 proizvede, najverovatnije taj PHP skript niste izvršili preko URL-a.

- **Morate da znate koju verziju PHP-a koristite.**

  Neki problemi nastaju zbog verzije PHP-a. Pre nego što počnete da koristite bilo koji server na kojem je omogućen PHP, izvršite preko njega skript **`phpinfo.php`** (pogledajte u dodatku A) ili ispitajte konstantu **PHP_VERSION** gde je upisana verzija 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*). Možda 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. A ako treba da prenesete tu poruku nekom drugom (na primer, kada se obratite meni za pomoć), svakako prepišite celu poruku o grešci!

- **Napravite pauzu!**

  Veliki broj programskih problema na koje sam naišao tokom godina, a većina onih najtežih, rešena je 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 stvari.

Pregled i unapređivanje

Novost u ovom izdanju knjige je odeljak »Pregled i unapređivanje« na kraju svakog poglavlja. U tom odeljku ćete naći pitanja koja se odnose na materiju koju smo upravo obradili, a zatim nagoveštaje o načinima da sami proširite svoje znanje i iskustvo. Ako naiđete na neki problem, bilo u odgovaranju na pitanja ili u sprovođenju sopstvenih poduhvata, obratite se na prateći forum za ovu knjigu (**`www.LarryUllman.com/forums/`**).

Pregled

- Koje oznake se koriste oko PHP koda?
- Koju ekstenziju treba da ima PHP fajl?
- Šta znači *kodiranje* stranice? Kako kodiranje utiče na stranicu?
- Koje PHP funkcije ili jezičke konstrukcije možete da koristite da biste poslali podatke veb pretraživaču?
- Kako se jednostruki navodnici razlikuju od dvostrukih prilikom pravljenja stringova ili prilikom štampanja stringova?
- Šta znači »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 dođu nakon toga? Koji još znakovi mogu da se koriste u imenu promenljive?
- Da li se u imenima promenljivih velika slova razlikuju od malih?
- Koji je operator za dodeljivanje?
- Kako se pravi promenljiva tipa string?
- Koji je operator za sastavljanje? Koji je operator za dodeljivanje sastavljanja?
- Kako se konstante definišu i koriste?

Unapređivanje

- Ako još ne znate – *sasvim pouzdano* – koju verziju PHP-a koristite, proverite 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 vidite koje još informacije ona sadrži.
- Napravite sasvim 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 sastavljanja (konkatenacije).
- Napravite sasvim novi skript koji definiše neke promenljive numeričkog tipa, vrši neke manipulacije sa njima i prikazuje vrednosti nekih numeričkih promenljivih.