

PHP

- 1 Kratak kurs PHP-a
- 2 Snimanje i učitavanje podataka
- 3 Upotreba nizova
- 4 Rad sa znakovnim nizovima i regularnim izrazima
- 5 Višekratna upotreba koda i pisanje funkcija
- 6 Objektno orijentisani PHP
- 7 Obrada grešaka i izuzetaka

Kratak kurs PHP-a

Ovo poglavlje sadrži kratak pregled sintakse i jezičkih konstrukcija PHP-a. Ako već programirate na PHP-u, popunićete praznine u svom znanju, a ako ste ranije koristili C, Perl, Python ili neki drugi programski jezik, ovo poglavlje će vam pomoći da brzo pređete na PHP.

Radićete na primerima iz stvarnog sveta, zasnovanim na iskustvu koje smo stekli pri izradi pravih veb lokacija, i tako naučiti kako da koristite PHP. Programerski udžbenici vas često uče osnovnoj sintaksi, ali s veoma jednostavnim primerima. Mi smo odlučili da ne radimo tako. Svesni smo da čitaoci žele da što pre nešto naprave i pokrenu, da shvate kako se jezik koristi, umesto da se muče s još jednom sintaksom i objašnjenjima funkcija koja nisu ništa bolja od uputstava na vebu.

Isprobajte naredne primere – otkucajte ih ili ih preuzmite na veb strani prikaza ove knjige (www.mikroknjiga.rs/store/prikaz.php?ref=978-86-7555-421-9&pismo=latinica), menjajte ih, rastavljajte i učite kako da ih popravite.

Poglavlje počinje primerom on-lajn narudžbenice iz koga ćete naučiti kako se u PHP-u koriste promenljive, operatori i izrazi. Govorićemo o tipovima promenljivih i prioritetima operatora. Naučićete da pristupate promenljivama obrasca izračunavajući ukupnu vrednost i porez na porudžbini.

Zatim ćemo napraviti obrazac za unošenje podataka u kojem ćemo pomoću PHP skripta proveriti ulazne podatke. Proučićemo koncept logičkih vrednosti i primere upotrebe naredbi `if`, `else`, operatora `?:` i naredbe `switch`. I najzad, napisaćemo petlje koje generišu HTML tabele.

U ovom poglavlju obrađene su sledeće teme:

- Ugradnja PHP-a u HTML
- Generisanje dinamičkog sadržaja
- Pristup promenljivama obrasca
- Identifikatori
- Promenljive
- Tipovi promenljivih
- Dodela vrednosti promenljivama
- Konstante
- Oblast važenja promenljivih
- Operatori i prioriteti

- Izrazi
- Funkcije
- Donošenje odluka pomoću naredbi `if`, `else` i `switch`
- Iteracija: petlje `while`, `do` i `for`

Pre nego što počnete: pristupanje PHP-u

Da biste obradili primere iz ovog poglavlja i ostatka knjige, morate imati pristup veb serveru na koji je instaliran PHP. Da biste iz tih primera naučili što više, trebalo bi da ih pokrenete i pokušate da ih izmenite. Za to će vam biti potreban server na kome ćete eksperimentisati.

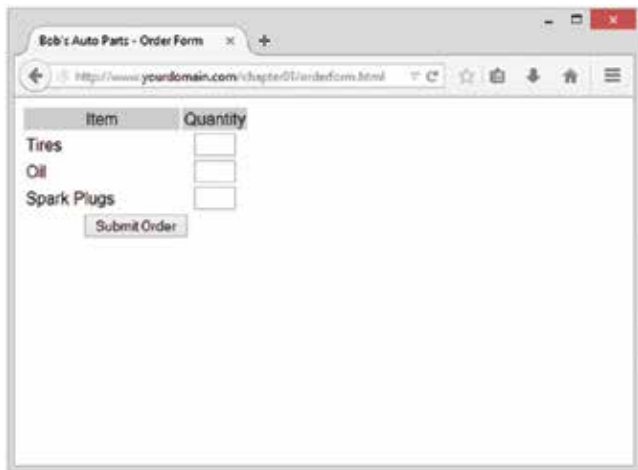
Ako PHP nije instaliran na vašem računaru, moraćete prvo da ga instalirate ili da se za to obratite administratoru sistema. Uputstvo za instaliranje pronaći ćete u dodatku A.

Probna aplikacija: Bobovi auto-delovi

Jedna od najčešćih primena serverskih skript jezika jeste obrada HTML obrazaca. Učenje PHP-a počecete tako što ćete napraviti narudžbenicu za fiktivnu veb prodavnicu „Bobovi auto-delovi“. Izvorni kôd primera iz ovog poglavlja nalazi se na veb strani prikaza ove knjige (www.mikroknjiga.rs/store/prikaz.php?ref=978-86-7555-421-9&pismo=latinica), u direktorijumu Chapter01.

Izrada obrasca za narudžbenicu

Veb dizajner je napravio narudžbenicu za auto-delove koje Bob prodaje. To je relativno jednostavna narudžbenica (slika 1.1), slična mnogima koje ste do sada verovatno sretali na vebu. Bob bi želeo da zna šta su njegovi kupci naručili i da izračuna ukupnu vrednost naručene robe i iznos poreza na tu robu.



Slika 1.1 Početna verzija narudžbenice beleži samo proizvode i količine.

Deo HTML koda narudžbenice prikazan je u listingu 1.1.

Listing 1.1 **orderform.html** – HTML kôd osnovne verzije narudžbenice

```

<form action="processor.php" method="post">
<table style="border: 0px;">
<tr style="background: #cccccc;">
  <td style="width: 150px; text-align: center;">Item</td>
  <td style="width: 15px; text-align: center;">Quantity</td>
</tr>
<tr>
  <td>Tires</td>
  <td><input type="text" name="tireqty" size="3"
maxlength="3" /></td>
</tr>
<tr>
  <td>Oil</td>
  <td><input type="text" name="oilqty" size="3"
maxlength="3" /></td>
</tr>
<tr>
  <td>Spark Plugs</td>
  <td><input type="text" name="sparkqty" size="3"
maxlength="3" /></td>
</tr>
<tr>
  <td colspan="2" style="text-align: center;"><input type="submit" value="Submit Order" /></td>
</tr>
</table>
</form>

```

Obratite pažnju na to da je u atributu `action` obrasca naveden PHP skript koji će obraditi porudžbinu (kasnije ćemo napisati taj skript). Vrednost atributa `action` je URL koji će biti učitan kada korisnik pritisne dugme `Submit Order`. Podaci koje je korisnik uneo u obrazac biće poslani stranici na toj adresi HTML metodom zadatom u atributu `method`. To može biti metoda `get` (podaci se dodaju na kraj URL-a) ili metoda `post` (podaci se šalju zasebno).

Sledeće što treba uočiti jesu imena polja u obrascu – `tireqty`, `oilqty` i `sparkqty`. Ta imena ćemo upotrebiti i u PHP skriptu. Zbog toga je važno da poljima u obrascu date smislena imena kojih ćete se lako setiti dok budete pisali PHP skript (na primer, `gumekol`, `uljekol` i `modelkol`). Neki editori za HTML daju poljima podrazumevana imena – recimo, `field23` – koja se teško pamte. Ako imena polja asociiraju na podatke koji se u njih unose, lakše ćete programirati.

Trebalo bi da razmislite i usvajanje određenog standarda za imena polja, koji ćete primenjivati na svim stranicama. Tako će vam biti lakše da zapamtite da li ste, na primer, u imenu polja skratili reč ili ste umesto razmaka stavili donju crtu.

Obrada podataka sa obrasca

Da biste obradili podatke sa obrasca, treba da napravite skript pomenut u atributu `action` oznake `form`. Taj skript će se zvati `processor.php`. Pokrenite svoj editor teksta i napravite tu datoteku. Otkucajte zatim sledeći kôd:

```

<!DOCTYPE html>
<html>
  <head>
    <title>Bob's Auto Parts – Order Results</title>
  </head>
  <body>
    <h1>Bob's Auto Parts</h1>
    <h2>Order Results</h2>
  </body>
</html>

```

Obratite pažnju na to da je sve što ste do sada otkucali samo običan HTML. Vreme je da dodate nešto jednostavnog PHP koda u skript.

Ugrađivanje PHP koda u HTML kôd

Ispod naslova <h2> u datoteci, dodajte sledeće redove:

```

<?php
  echo '<p>Order processed.</p>';
?>

```

Snimite datoteku i pogledajte je u čitaču veba (engl. *web browser*) tako što ćete popuniti obrazac i pritisnuti dugme Submit Order. Trebalo bi da se pojavi rezultat nalik na onaj sa slike 1.2.



Slika 1.2 Tekst prosleđen PHP naredbi echo poslat je čitaču veba.

Obratite pažnju na to da je PHP kôd koji ste napisali ugrađen u običnu HTML datoteku. Pogledajte izvorni kôd stranice u čitaču veba. Trebalo bi da vidite nešto slično sledećem:

```

<html>
  <head>
    <title>Bob's Auto Parts – Order Results</title>
  </head>
  <body>

```

```

<h1>Bob's Auto Parts</h1>
<h2>Order Results</h2>
<p>Order processed.</p>
</body>
</html>

```

PHP naredbe se ne vide zato što ih je interpreter PHP koda zamenio rezultatom. To znači da se iz PHP koda dobija čisti HTML kôd, koji se može gledati pomoću svakog čitača; drugim rečima, čitač ne mora da razume PHP.

Prethodni primer ilustruje način rada serverskih skriptova. PHP kôd se interpretira i izvršava na veb serveru, za razliku od JavaScripta i drugih klijentskih jezika koji se prevode i izvršavaju unutar čitača veba na korisnikovom računaru.

Kôd koji se sada nalazi u PHP datoteci sadrži četiri vrste tekstualnih elemenata:

- HTML oznake (engl. *HTML tags*)
- PHP oznake (engl. *PHP tags*)
- PHP naredbe (engl. *PHP statements*)
- Beline (engl. *whitespace*)

Možete dodati i komentare.

Većinu redova u navedenom primeru čini običan HTML kôd.

PHP oznake

PHP kôd iz prethodnog primera počeo je sa `<?php` i završio se sa `?>`. To je slično HTML oznakama jer sve one počinju znakom „manje od“ (`<`) i završavaju se znakom „veće od“ (`>`). Te kombinacije znakova nazivaju se *PHP oznake* (engl. *PHP tags*) – one veb serveru pokazuju gde PHP kôd počinje i gde se završava. Ceo tekst između te dve oznake smatra se PHP kodom. Tekst izvan tih oznaka samo se prosleđuje čitaču jer predstavlja običan HTML kôd. PHP oznake omogućavaju da *izađemo* (engl. *escape*) iz bloka HTML koda.

Postoje dva stila PHP oznaka. Sledeći primeri koda međusobno su ekvivalentni.

- **XML stil**

```
<?php echo ' <p>Order processed.</p>'; ?>
```

Ovaj stil oznaka koristimo u knjizi, a to je i stil čija se upotreba preporučuje. Pošto ga administrator servera ne može isključiti, on će sigurno biti dostupan na svim serverima, što je naročito korisno kada se aplikacija koristi na više različitih servera. Može se koristiti s XML dokumentima. U opštem slučaju, preporučujemo da koristite ovaj stil oznaka.

- **Skraćeni stil**

```
<? echo ' <p>Order processed.</p>'; ?>
```

Ovaj stil je jednostavniji i oznake liče na instrukcije za obradu u SGML-u. Ako hoćete da ga koristite, potrebno je da u konfiguracijskoj datoteci PHP okruženja aktivirate opciju `short_tags` ili da prevedete PHP sa uključenim kratkim oznakama. Više informacija o upotrebi skraćenog stila oznaka naći ćete u dodatku A. Međutim, upotreba ovog stila se ne preporučuje u kodu koji će se distribuirati jer neće raditi u mnogim okruženjima pošto više nije podrazumevano uključen.

PHP naredbe

PHP naredbe (engl. *PHP statements*) nalaze se između početnih i završnih oznaka i određuju šta interpreter PHP koda treba da uradi. U navedenom primeru upotrebili smo samo jedan tip naredbe:

```
echo '<p>Order processed.</p>';
```

Kao što ste verovatno pretpostavili, naredba `echo` je veoma jednostavna: ona u čitaču ispisuje tekst koji joj prosledite. Na slici 1.2 vidite da se tekst `Order processed` pojavljuje u prozoru čitača.

Obratite pažnju na to da se na kraju naredbe `echo` nalazi znak tačka i zarez. On razdvaja izraze u PHP-u – kao što se u govornom jeziku za razdvajanje rečenica koristi tačka. Ako ste programirali na C-u ili na Javi, takva upotreba tačke i zareza biće vam poznata.

Izostavljanje tačke i zareza česta je sintaksna greška koja se lako pravi. Srećom, takva greška se isto tako lako otkriva i ispravlja.

Beline

Znakovi za razdvajanje – poput znaka za novi red (za povratak na početak reda), razmaka i tabulatora – nazivaju se *beline* (engl. *white spaces*). Kao što verovatno već znate, čitači veća zanemaruju beline u HTML kodu. Isto tako postupa i izvršno okruženje za PHP. Pogledajte ova dva bloka HTML koda:

```
<h1>Welcome to Bob's Auto Parts!</h1><p>What would you like to order today?</p>
i
<h1>Welcome           to Bob's
Auto Parts!</h1>
<p>What would you like
to order today?</p>
```

Oba bloka koda daju identičan rezultat zato što čitaču izgledaju isto. Preporučljivo je da u HTML kodu razumno koristite beline kao pomoćno sredstvo kojim ćete kod stranica učiniti čitljivijim. Isto važi i za PHP. Nema potrebe da između PHP naredbi stoje beline, ali se kod s njima lakše čita. Na primer, rezultati naredbi:

```
echo 'hello ';
echo 'world';
i
echo 'hello ';echo 'world';
```

identični su, ali se prva verzija lakše čita.

Komentari

Komentari su upravo to što im samo ime kaže i služe kao obaveštenja ljudima koji čitaju kod. Komentari se mogu upotrebiti da objasne skript, da opišu ko ga je napisao, zašto je baš tako napisan, kada je poslednji put izmenjen itd. Komentare ćete nalaziti gotovo u svim PHP skriptovima, osim u onim najjednostavnijim.

U suštini, pošto PHP interpreter zanemaruje tekst u komentaru, može se smatrati da je ekvivalentan belini.

PHP podržava komentare u stilu jezika C, C++ i skriptova komandnog okruženja.

Sledi višeredni komentar u stilu jezika C, koji bi mogao da se pojavi na početku PHP skripta:

```
/* Pera Petrović
   Izmenjen: 10. aprila
   Skript obrađuje narudžbine.
*/
```

Višeredni komentari treba da počinju sa `/*` i završavaju se sa `*/`. Isto kao u jeziku C, višeredni komentari ne mogu biti ugnežđeni jedni u druge.

Za komentare u jednom redu možete koristiti stil jezika C++:

```
echo '<p>Order processed.</p>'; // Štampaj narudžbenicu
```

ili stil skriptova komandnog okruženja:

```
echo '<p>Order processed.</p>'; # Štampaj narudžbenicu
```

U oba stila, sve što se nalazi iza simbola jednorednog komentara (`#` ili `//`) zanemaruje se do kraja reda ili do završne PHP oznake.

U narednom redu koda, tekst ispred završne PHP oznake, ovo je komentar, deo je komentara. Tekst iza završne PHP oznake, a ovo nije, obrađivaće se kao HTML kôd jer se nalazi iza završne PHP oznake:

```
// ovo je komentar ?> a ovo nije
```

Dodavanje dinamičkog sadržaja stranici

Sve što smo do sada uradili u PHP-u, mogli smo da uradimo i pomoću običnog HTML koda.

Glavni razlog upotrebe serverskog skript jezika jeste mogućnost prikazivanja dinamičkog sadržaja. To je vrlo važna primena, zato što će sadržaj koji se menja periodično ili prema potrebama korisnika, podstaći posetioce da se vraćaju na veb lokaciju. PHP omogućava laku izradu dinamičkog sadržaja.

Počnimo jednostavnim primerom. Zamenimo PHP u datoteci `processorder.php` sledećim kodom:

```
<?php
echo "<p>Order processed at ";
echo date('H:i, jS F Y');
echo "</p>";
?>
```

Isto možete napisati i u jednom redu, koristeći operator nadovezivanja (`.`), kao

```
<?php
echo "<p>Order processed at ".date('H:i, jS F Y')."</p>";
?>
```

U ovom kodu koristimo ugrađenu PHP funkciju `date()`, pomoću koje kupcu saopštavamo datum i vreme kada je njegova porudžbina obrađena. Prilikom svakog izvršavanja skripta biće ispisane različite vrednosti. Rezultat posle jednog izvršenja skripta prikazan je na slici 1.3.