

Web programiranje

Vežbe 1

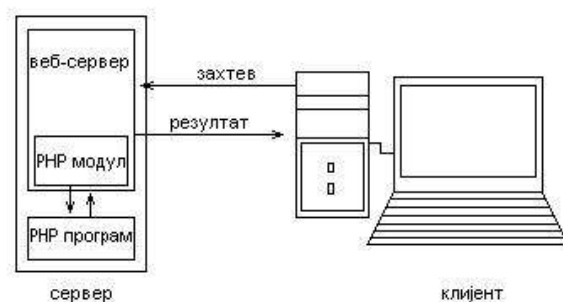
1. Sadržaj predmeta iz knjige predmeta

Студијски програм: ИНФОРМАТИКА/ФИЗИКА
Врста и ниво студија: Основне академске студије, први ниво
Назив предмета: WEB ПРОГРАМИРАЊЕ
Наставник (Презиме, средње слово, име): Цвјетковић Владимир
Статус предмета: Изборни на основним академским студијама Информатике и изборни на основним академским студијама Физике
Број ЕСПБ: 7
Услов: Уписан одговарајући семестар
Циљ предмета: Савладавање основа технологије, значаја и примене клијентског и серверског WEB програмирања
Исход предмета: Знања која ће студенти стећи после савладавања програма: Упознавање са технологијом, могућностима и применама клијентског и серверског WEB програмирања Вештине које ће стећи студенти после савладавања програма: Разумевање намене и могућности WEB програмирања и способност за самостално креирање клијентских и серверских програма. Ставови које ће стећи студенти после савладавања програма: Формирање схватања о могућностима, начинима примене, пројектовању и развоју клијентских и серверских програма за WEB.
Садржај предмета <i>Теоријска настава:</i> Web окружење. Дефинисање различитих приказа и претраживача, принципи пројектовања Увод у XML. Шта је XML, од чега потиче XML, намена XML, како се почиње са XML-ом. Маркуп и концепти језгра. Тагови, документи, елементи, ентитети, Моделирање информација. Једноставно меморисање података, наративни документи, сложени подаци, описивање документа документом. Графика. GIF, JPEG, PNG формат, креирање графика Web палетом. Мултимедија и интерактивност, анимирани GIF, аудио и видео на Web-у, увод у Java script. Серверско програмирање - CGI, PHP, Java, .NET Програмирање. Низови и догађаји, стабла и објекти, Pull Parsing, стандардни API, избор парсера, PEX, SAX, DOM, ostale opcije. Web servisi. Uvod u Web servise, opisivanje informacija:XML, OPIS Web servisa:WSDL, pristup Web servisima SOAP, налажење Web сервиса: UDDI регистри, алтернативни приступ: ebXML, архитектуре Web servisa, имплементација Web сервиса. Креирање портала. Креирање оквира за организовање информација. Коришћење трослојне архитектуре, коришћење оквира за интеграцију апликација, обезбеђивање прихватања портала, мерење повратка инвестиција. <i>Практична настава:</i> Креирање WEB апликација <i>Вежбе:</i> Самосталан рад студената на развоју WEB апликација.

2. Konkretna sadržaj predmeta za školsku 2019/2020

Za školsku 2019/2020 godinu, predmet *Web programiranje* bazira se na tehnologijama otvorenog koda PHP7, Apache i MySQL:

- **PHP** - programski jezik skript tipa posebno pogodan za razvoj web aplikacija na strani servera za dinamičko generisanje web stranica. Verzija 7 ovog programskog jezika takođe poseduje većinu objektno-orijentisanih karakteristika svojstvenih objektno-orijentisanim jezicima kao što su C++, Java, C# ili Python. Sintaksa PHP-a je slična sintaksi C-a i Perl-a. Podržava većinu popularnih baza podataka, uključujući tu MySQL, SQLite, MS SQL, Oracle itd, operativne sisteme Windows, Linux, FreeBSD, Solaris, OS X i različite web servere, kao što su Apache, MS IIS.
- **Apache** - Web server koji je odigrao ključnu ulogu u početnom rastu web-a sredinom devedesetih godina 20. veka. 2010. godine ovaj web server pogoni preko 100 miliona web lokacija širom sveta! U avgustu 2009, Apache opslužuje 54.32% svih svetskih sajtova i preko 66% najposećenijih sajtova na svetu. Apache je server modularne arhitekture, a različite programske platforme se s njim uvezuju preko specijalnih modula, na primer mod_php, mod_python, mod_mono, mod_aspdotnet itd.
- **MySQL** - open-source sistem za upravljanje relacionim bazama podataka. Ima preko 6 miliona instalacija. U sprezi sa prethodno navedenim tehnologijama, a zahvaljujući svojoj skalabilnosti, koristi se u projektima kao što su *WordPress*, *phpBB*, *Wikipedia*, *Facebook*, *Google*. Posедуje API (Applications Programming Interface) za sve popularne programske jezike.
- **LAMP** - kombinacija Linux-Apache-MySQL-PHP/Perl/Python softvera pogoni najveći broj web sajtova na svetu.



Literatura:

1. PHP 5 Beginning PHP and MySQL , from Novice to Professional , W. Jason Gilmore, APRESS, 2007.
2. PHP Cookbook, David Sklar, Adam Trachtenberg
3. PHP Zend Tutorial (<http://devzone.zend.com/article/627>)
4. PHP W3schools Tutorial (<http://w3schools.com/php/default.asp>)

3. PHP jezik - istorijat

PHP jezik (*PHP: Hypertext Preprocessor*) je, na mnoge načine, reprezentativni open-source projekat, koji je rastao od malog projekta koji je zadovoljavao potrebe male grupe programera koje ostali alati nisu mogli da zadovolje, da bi kasnijom rafinacijom uspevao da zadovolji potrebe sve veće baze programera. Trenutno aktuelne verzije jezika su 5 i 7 i o njima će posebno biti reči.

PHP je napisan kao skup CGI binarnih fajlova u C programskom jeziku od strane danskog/grenlandskog programera Rasmusa Lerdorfa 1994. godine, kako bi zamenio mali skup Perl skripti koje je koristio za održavanje lične stranice. Lerdorf je inicijalno napravio PHP kako bi prikazao svoj rezime i kako bi prikupio određene podatke, poput onih o broju poseta na svojoj stranici. *Personal Home Page Tools* se pojavio u javnosti 8. juna 1995. godine nakon što ga je Lerdorf kombinovao sa vlastitim *Form Interpreter*-om kako bi napravio PHP/FI (ovo izdanje se smatra drugom verzijom PHP-a).

Kako je ovo izdanje postalo ekstremno popularno, mnogi programeri su se priključili Lerdorfu. U junu 1998. godine, sa verzijom 3.0, više od 50000 korisnika je baziralo svoja web rešenja na PHP-u, ali je do početka 1999. premašio i najsmelija očekivanja autora: milion aktivnih korisnika. Tada je postalo očigledno da korisnici žele da pišu aplikacije koje daleko premašuju namenu za koju je jezik u startu napravljen.

PHP4

Dva programera PHP jezgra, **Zeev Suraski** i **Andi Gutmans** su preuzeli inicijativu da ponovo napišu samo jezgro i PHP parser, koje je rezultovalo novim jezgrom PHP-a - *Zend Engine*. Rezultat ovog rada bio je PHP4, izdat u maju 2000. godine. Samo par meseci pošto je zvanično izdat, PHP4 je bio instaliran na 3.6 miliona web lokacija. Mnogi izdanje PHP4 smatraju oficijelnim ulaskom PHP-a u *Enterprise* aplikacije. Evo nekoliko unapređenja PHP4 u odnosu na PHP3:

- **poboljšano rukovanje resursima** - znatno unapređena skalabilnost, jer u vreme PHP3 niko nije mogao da pretpostavi koliko će brzo porasti zahtevi da se u PHP-u razvijaju i web lokacije *enterprise* tipa
- **podrška objektno-orijentisanom pristupu** - u PHP4 dodati su osnovni koncepti objektno-orijentisane arhitekture. Mnogo napredniji i standardniji pristup OOP-u, koji će i biti predmet ovog kursa, prisutan je u PHP5.
- **Nativna podrška za sesije** - u PHP3 podržano je rukovanje sesijama uz pomoć spoljne biblioteke, ali je u PHP4 inkarnaciji dodato u samo jezgro jezika.
- **Kriptovanje** - u PHP4 inkorporirana je MCrypt biblioteka koja implementira različite algoritme za kriptovanje podataka, npr. MD5, SHA1, Blowfisk, itd.
- **Nativna COM/DCOM podrška** - otvorena mogućnost interakcije sa različitim MS Windows aplikacijama i servisima
- **Nativna Java podrška** - podrška za povezivanje sa Java objektima
- **Perl Compatible Regular Expressions (PCRE)** - danas je Perl tip regularnog izraza standardni način za rukovanje string pretragom u PHP-u.

PHP5

PHP5 je danas (2018. godina) najzastupljenija verzija, pa će iz tog razloga kurs biti baziran na njoj. Donosi ogroman broj poboljšanja kako bibliotečkog tipa, tako i samih koncepata programiranja u PHP-u. Evo nekih poboljšanja:

- **Veoma unapređen OOP** - eksplicitni konstruktori i destruktori, kloniranje objekata, apstrakcija klasa, public-private-protected deklaracije, interfejsi.
- **Try/catch rukovanje izuzecima** - radi na sličan način kao u Javi i C#.
- **Unapređena podrška za XML i web servise** - XML podrška se sada zasniva na libxml2 biblioteci, a takođe je tu i SOAP (*Simple Object Access Protocol*) podrška za komunikaciju sa web servisima.
- **Nativna podrška za SQLite bazu podataka** - ultralight baza koja podržava SQL standarde.

PHP6

PHP6 je još uvek u razvoju, tako da nije potpuno sigurno kako će tačno izgledati. Neke od njegovih naprednih osobina implementirane su (*backport*) na aktuelnu verziju 5.3. Evo nekih od sigurnih kandidata za funkcionalnost PHP6:

- **Unicode podrška** - dodata nativna *Unicode* podrška, čineći dosta lakšim posao održavanja multijezičkih web aplikacija
- **Veliki broj unapređenja sigurnosti** - neke ranije konstrukcije jezika koje su se pokazale kao nesigurne biće izbačene
- **Nove mogućnosti jezika i konstrukcije** - 64-bitni celobrojni tip, *foreach* konstrukcija za multidimenzione nizove, *closures*, prostori imena (*namespaces*), itd.

PHP7

PHP 7.0.0 Release Announcement

http://php.net/releases/7_0_0.php

Migrating from PHP 5.6.x to PHP 7.0.x

<http://php.net/manual/en/migration70.php>

4. Opšte osobine jezika

Praktičnost

Dato je upoređenje Java koda koji štampa današnji datum sa PHP kodom koji čini to isto:

Java: <pre>import java.util.*; public class JavaGetTodaysDate { public static void main(String[] args) { Date today = Calendar.getInstance().getTime(); System.out.println(today); } }</pre>	PHP: <pre><?php echo date("F j, Y"); ?></pre>
--	--

Dato poređenje ukazuje na to da je PHP skript jezik, slično JavaScript-u, Python-u, Perl-u, Ruby-ju itd. PHP je slabo tipiziran jezik, gde se konverzija tipova, ukoliko je moguća, dešava implicitno, tako da sledeći kod prolazi bez problema:

```
<?php
$broj = "5";           // $broj je a string
$zbir = 15 + $broj;     // zbraja ceo broj i string, konvertuje u ceo broj
$zbir = "dvadaset";    // prepisuje promenljivu $zbir stringom
?>
```

Oslobađanje sistemskih resursa po završetku skripta takođe je automatsko, tako da PHP programeru omogućava da se skoncentriše na samo rešavanje problema na kome radi, umesto da vodi računa o administrativnim zadacima.

Snaga

Programeri PHP jezgra, kao i zajednica, razvili su preko 180 biblioteka različitih namena, tako da je velika mogućnost da ono što novom programeru zatreba već negde postoji rešeno u okviru neke od biblioteka. Evo nekih značajnih mogućnosti:

- Kreiranje i manipulacija PDF i Flash fajlovima
- Parsiranje najkompleksnijih stringova PCRE regularnim izrazima
- Autentifikacija korisnika čiji su nalozi smešteni u fajlu, bazi podataka ili čak *MS Active Directory*
- Komunikacija korišćenjem širokog dijapazona protokola, recimo LDAP, IMAP, POP3, NNTP, DNS i mnogi drugi
- Integrisana rešenja za procesiranje kreditnih kartica

Funkcionalnost PHP-a u svakom trenutku se može proširiti ekstenzijama iz PEAR baze (*PHP Extension and Application Repository*), koja funkcioniše slično kao paket-menadžer Linux distribucija.

Mogućnosti

PHP danas (2019. godina) može da komunicira sa čak 25 baza podataka, i to: Adabas D, dBase, Empress, FilePro, FrontBase, Hyperwave, IBM DB2, Informix, Ingres, InterBase, mSQL, Microsoft SQL Server, MySQL, Oracle, Ovrimos, PostgreSQL, Solid, Sybase, Unix dbm i Velocis.

Ceo posao komunikacije sa bazom može se obaviti i na apstraktniji način, pomoću PDO tehnologije (*PHP Data Objects*), kada se pravi RSBMS skriva iza sloja apstrakcije, pa PHP aplikacija koja koristi PDO radi neizmenjena na bilo kom konkretnom RDBMS-u.

Još jedan nivo apstrakcije, popularan u nekim drugim tehnologijama (*Java Hibernate* i *ADO.NET*), a svrha mu je mapiranje tipa objekat-relacija (*ORM-Object Relational Mapper*), prisutan je i u PHP-u, recimo u vidu *Propel* biblioteke.

U PHP-u postoji mogućnost da se programira i proceduralno i objektno, a cela platforma je orijentisana da programeru, uz minimalno dodatno učenje omogući da uradi većinu poslova.

Cena

PHP tehnologija, kao i ceo LAMP stek je potpuno besplatan, i to bez ikakvih restrikcija, kako za neprofitnu, tako i za profitabilnu upotrebu.

Linkovi

Zvanični sajt: <http://www.php.net/>

Puna dokumentacija: <http://www.php.net/manual/en/>

Downloads: <http://www.php.net/downloads.php>

5. Osnove PHP jezika



Podešavanje radnog okruženja

Dva su osnovna načina pristupa AMP (Apache-Mysql-PHP) bateriji aplikacija u zavisnosti gde je AMP instaliran:

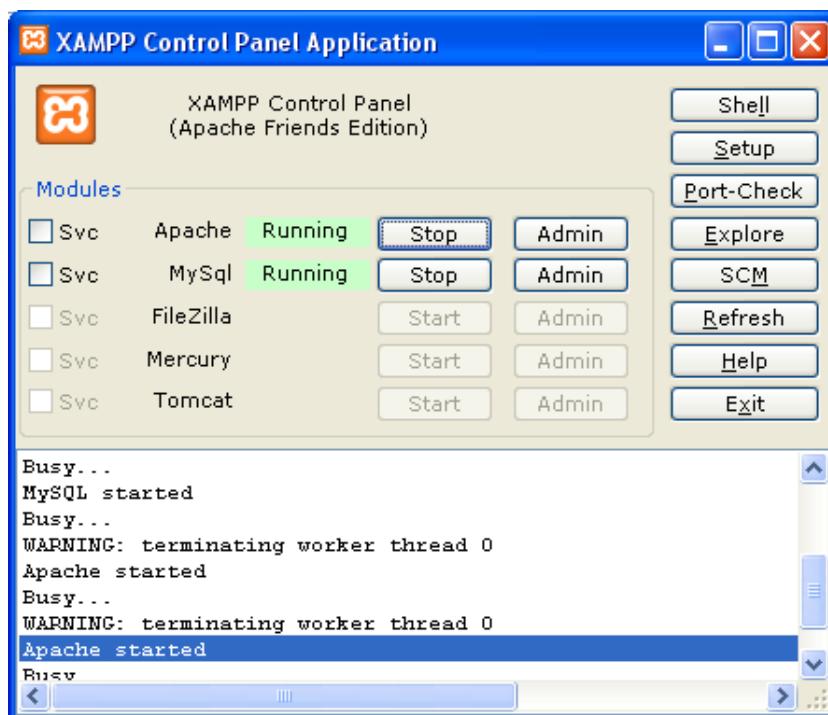
- **lokalni** - AMP aplikacije su instalirane na lokalnom računaru, tako da nema potrebe za kopiranjem fajlova unutar projekta na udaljeni LAMP ili WAMP server. Ovakav način rada se preporučuje u fazi razvoja web aplikacije.
- **udaljeni** - Da bi se web aplikacija pokrenula, potrebno je prvo je iskopirati na udaljeni LAMP ili WAMP server, obično korišćenjem FTP ili SFTP protokola (koje alat *NetBeans* automatski podržava).

U daljem tekstu biće data dva postupka za podešavanje radnog okruženja, i to: (1) preko ličnog studentskog naloga na serveru spider.pmf.kg.ac.rs, (2) lokalna AMP instalacija u *Windows*-u korišćenjem alata XAMPP i WAMP.

Lokalna instalacija na Windows-u

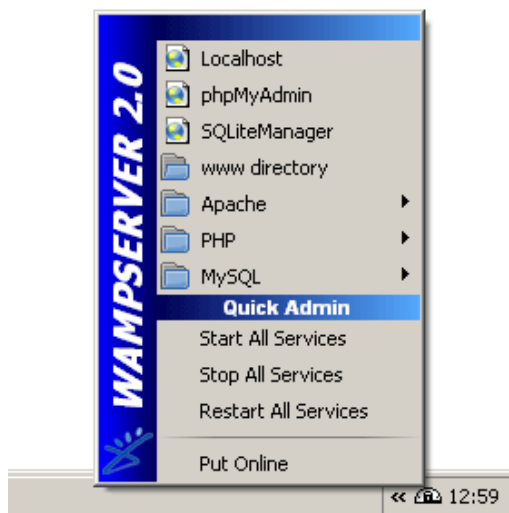
XAMPP

Na Windows sistemu je daleko najjednostavnija varijanta instalacija AMP baterije korišćenjem neke “spakovane” instalacije sa grafičkim kontrolnim panelom, kao što je npr. XAMPP (<http://www.apachefriends.org/en/xampp-windows.html>). Instalacija je krajnje pojednostavljena, a evo nekoliko slika ekrana kako podesiti XAMPP server:



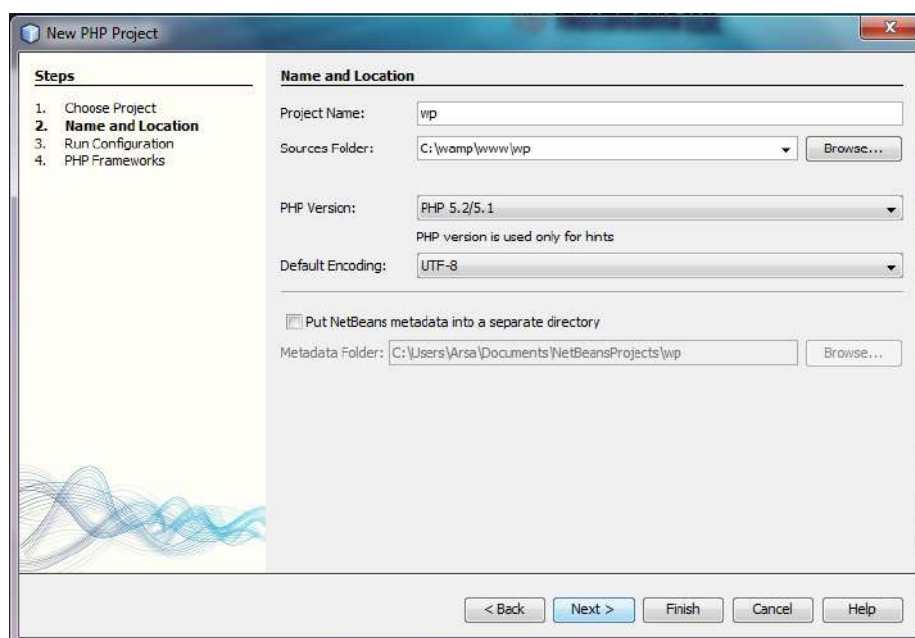
WAMP

Poslednju verziju WAMP servera možete preuzeti sa njegovog oficijelnog sajta:
<http://www.wampserver.com/en/download.php>



Slika: <http://localhost>

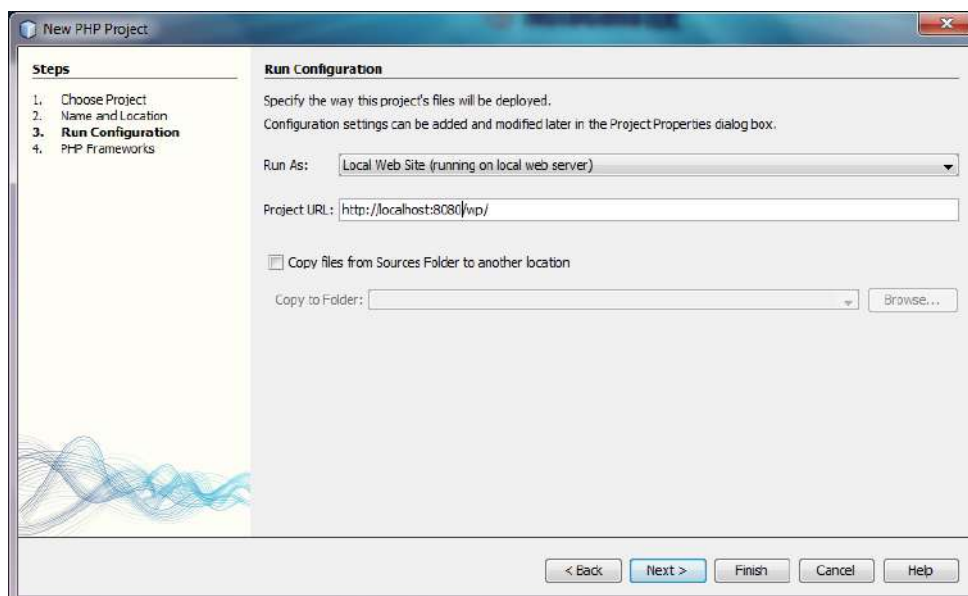
Kreiranje NetBeans projekta



Treba obratiti pažnju da se projekat locira u DOCUMENT_ROOT WAMP Apache instalacije, u prikazanom slučaju **c:\wamp\www**.

Slično važi i za XAMPP server, gde se svi projekti lociraju u folder **c:\xampp\htdocs**.

NAPOMENA: Na računaru može da postoji više web servera koji koriste port 80 (default port koji se koristi za HTTP komunikaciju). Ako na računaru već postoji web server, kao što je IIS, neophodno je promeniti port koji koristi, izbrisati ga ili deaktivirati. U suprotnom, Apache neće raditi. Da bi se promenio port koji koristi Apache, potrebno je izmeniti fajl **httpd.conf**. Najpre treba locirati linije koje sadrže *Listen 80* i *ServerName localhost:80*, a zatim promeniti vrednost 80 sa brojem porta koji se izabere (8080 je tipičan izbor za drugi web server).



Ugrađivanje PHP koda

Jedna od prednosti PHP-a je i mogućnost da se **PHP kod ugrađuje direktno u HTML strukture**. Međutim, potrebno je obratiti pažnju da se takve stranice identifikuju odgovarajućom ekstenzijom (tipično “.php”), kako bi ih web server pravilno interpretirao.

Podrazumevana sintaksa je da se ono što treba interpretirati kao PHP kod stavi između tagova:

```
<?php i ?>
```

Primer najjednostavnije dinamičke stranice u PHP-u bi glasio:

```
<h3>Naslov</h3>
<?php
    echo "<p>Dinamicki izlaz</p>";
?>
<p>Staticki izlaz</p>
```

Moguće je izaći i ući u PHP onoliko puta koliko je to potrebno na datoj stranici. Recimo, sledeći kod savršeno funkcioniše:

```
<html>
    <head>
        <title><?php echo "Dobrodosli!";?></title>
    </head>
    <body>
        <?php
            $datum = "26 jul 2007";
        ?>
        <p>Today's date is <?php echo $datum; ?></p>
    </body>
</html>
```

Komentari

Dozvoljeno je koristiti 3 vrste komentara:

C++ stil:

```
<?php
    // Title: My first PHP script
    // Author: Jason
    echo "This is a PHP program";
?>
```

C stil:

```
<?php
    /*
        Title: My PHP Program
        Author: Jason
        Date: July 26, 2007
    */
?>
```

Shell stil:

```
<?php
# Title: My PHP program
# Author: Jason
echo "This is a PHP program";
?>
```

Ispis podataka u web pretraživač

PHP poseduje nekoliko komandi za ispis teksta u web browser:

```
int print(argument)
void echo(string argument1 [, ...string argumentN])
boolean printf(string format [, mixed args])
```

Kao što se vidi iz prototipa, osnovna razlika između ove tri naredbe je u broju argumenata i povratnoj vrednosti. Evo nekoliko primera:

```
<?php
    print("<p>I love the summertime.</p>");
?>
<?php
    $season = "summertime";
    print "<p>I love the $season.</p>";
?>
<?php
    $heavyweight = "Lennox Lewis";
    $lightweight = "Floyd Mayweather";
    echo $heavyweight, " and ", $lightweight, " are great
fighters.";
?>
<?php
    printf("Bar inventory: %d bottles of tonic water.", 100);
    printf("%d bottles of tonic water cost $%f", 100, 43.20);
?>
```

Poslednji primer štampa sledeći tekst:

```
Bar inventory: 100 bottles of tonic water.
100 bottles of tonic water cost $43.20
```

Očigledno je da se koriste oznake formata identične onima poznatim iz jezika C. Sličnost sa C-om ogleda se i u prisustvu naredbe **sprintf**:

```
string sprintf(string format [, mixed arguments])
```

Evo i primera:

```
$cost = sprintf("$%.2f", 43.2); // $cost = $43.20
```

Skalarni tipovi

Logički tip (boolean)

Logički tip se u PHP-u ponaša slično kao u C-u, dakle nema stroge tipizacije kao u jeziku Java. 0 se tretira kao **false**, a sve ostale vrednosti kao **true**:

```
$alive = false; // $alive is false.
$alive = 1;     // $alive is true.
$alive = -1;    // $alive is true.
$alive = 5;     // $alive is true.
$alive = 0;     // $alive is false.
```

Celobrojni tip (Integer)

Zapisi celih brojeva mogu biti dekadni, heksadekadni i oktalni. U PHP7 uveden je i 64-bitni celobrojni tip

```
42          // decimal
-678900     // decimal
0755        // octal
0xC4E       // hexadecimal
```

Realni tip (Float)

Evo nekoliko načina zapisa:

```
4.5678
4.0
8.7e4
1.23E+11
```

String

Stringovi se u PHP-u tretiraju tradicionalno, kao nizovi karaktera. Mogu biti uokvireni jednostrukim ili dvostrukim navodnicima:

```
"PHP is a great language"
"whoop-de-do"
'*9subway\n'
"123$%^789"
```

```
$color = "maroon";
$parser = $color[2]; // Assigns 'r' to $parser
```

Znaci navođenja mogu biti dati unutar stringa ako se ispred njih stavi escape karakter “\”:

```
echo "Francuski kralj Filip IV \"lepi\"";
```

Postoji značajna razlika između uokviravanja stringova jednostrukim i dvostrukim navodnicima. Naime, string vrednosti u duplim navodnicima se automatski parsiraju u odnosu na specijalne karaktere i imena promenljivih:

```
<?php
$identity = 'James Bond';
$car = 'BMW';
// this would contain the string "James Bond drives a BMW"
$sentence = "$identity drives a $car <br />";
echo $sentence;
```

```
// this would contain the string "$identity drives a $car"
$sentence1 = '$identity drives a $car';
echo $sentence1;
?>
```

Primer korišćenja PHP operatora

Primer štampa HTML tabelu sa originalnom cenom artikla, cenom sa uračunatim popustom, itd.

```
<html>
  <head>
  </head>
  <body>

    <?php
    // set quantity
    $quantity = 1000;
    // set original and current unit price
    $origPrice = 100;
    $currPrice = 25;
    // calculate difference in price
    $diffPrice = $currPrice - $origPrice;
    // calculate percentage change in price
    $diffPricePercent = (($currPrice - $origPrice) * 100)/$origPrice;
    ?>

    <table border="1" cellpadding="5" cellspacing="0">
      <tr>
        <td>Quantity</td>
        <td>Cost price</td>
        <td>Current price</td>
        <td>Absolute change in price</td>
        <td>Percent change in price</td>
      </tr>
      <tr>
        <td><?php echo $quantity ?></td>
        <td><?php echo $origPrice ?></td>
        <td><?php echo $currPrice ?></td>
        <td><?php echo $diffPrice ?></td>
        <td><?php echo $diffPricePercent ?>%</td>
      </tr>
    </table>
  </body>
</html>
```

Primer konkatencije stringova operatorom "."

```
<?php
// set up some string variables
$a = 'the';
$b = 'games';
$c = 'begin';
$d = 'now';
// combine them using the concatenation operator
// this returns 'the games begin now<br />'
$statement = $a.' '.$b.' '.$c.' '.$d.'<br />';
print $statement;
// and this returns 'begin the games now!'
$command = $c.' '.$a.' '.$b.' '.$d.'!';
print $command;
?>
```

Obrada podataka iz formi

Izračunavanje matematičkih izraza i manipulacije stringovima su svakako korisni, ali to je nešto što se ne mora obavljati na strani servera, već se može uraditi i iz samog browser-a, pomoću jezika *JavaScript*. Serversko programiranje dolazi do izražaja onda kada je potrebno obrađivati podatke iz HTML formi, npr. unositi ih u bazu podataka.

PHP ima direktnu vezu sa podacima iz forme koji se prenose preko HTTP metoda POST ili GET:

forma.html:

```
<html>
  <head></head>
  <body>
    <form action="message.php" method="post">
      Enter your message: <input type="text" name="msg" size="30">
      <input type="submit" value="Send">
    </form>
  </body>
</html>
```

message.php

```
<html>
  <head></head>
  <body>
    <?php
      // retrieve form data
      $input = $_POST['msg'];
      // use it
      echo "You said: <i>$input</i>";
    ?>
  </body>
</html>
```

Sledeći primer se ponaša slično kao prethodni, osim što se i forma i kod nalaze u jednom jedinom PHP fajlu i kao HTTP protokol se koristi protokol GET. Prilikom korišćenja GET protokola, vrednosti polja iz forme će se naći u *Address bar*-u web pretraživača. U primeru je takođe primetna upotreba asocijativnog niza (mape) `$_SERVER`, i to člana `$_SERVER["PHP_SELF"]`, koji daje adresu PHP fajla iz koga je pozvan:

```
<html>
  <head>
    <title>Primer forme za unos</title>
  </head>
  <body>
    <form action="<?php echo $_SERVER['PHP_SELF'] ?>" method="get">
      Ukucati poruku: <input type="text" name="poruka" size="30">
      <input type="submit" name="posalji" value="Posalji">
    </form>
  </body>

  <?php
    // Ako je kliknuto na "Posalji"
    if (isset ($_GET['posalji']))
    {
      // prihvatiti poruku iz forme
      $input = $_GET['poruka'];
```

```
// Pisi poruku u italic stilu
echo "Napisali ste: <i>$input</i>";

}
?>
</html>
```