

## **PHP (Personal Home Page), a V.4.0 je nazvan Hypertext Preprocessor (pretprocesor teksta)**

Skriptni "on server side" programski jezik, u kombinaciji sa open source sistemima za RDBMS, kao MySQL, PostgreSQL ili Interbase – snažna i besplatna razvojna platforma,

Primitivno stanje u razvoju 1994 god.,

Autor PHP (originalni) – Rasmus Lerdorf.

1995 predstavljen javnosti,

Nakon ugrađivanja u programe za interpretaciju podataka iz HTML formi (FI) rođen PHP/FI

Nakon dodavanja podrške za mSQL (još jedan besplatan sistem za upravljanje bazama podataka), PHP/FI postao u potpunosti funkcionalan

1997 PHP mašina napisan u potpunosti iz početka

Nastala je ver. 3.0 pa Ver. PHP 4.0 i 4.1

Najveći dio dodataka je besplatan

Neki dodaci su komercijalni

Posjeduje "titulu" najviše korištenog Apache modula, instaliran na 46 % Apache servera.

Najbliži konkurent je Perl 20 %.

Apache server pokriva 60 % svih Web sajtova

### **Prednosti**

Dinamički sadržaj se može obezbijediti na različite načine

Bogata platforma namijenjena razvoju Web aplikacija

Server-side tehnologija

Lakše se uči od Perla, ASP-a, JSP itd.

### **Osnove PHP-a**

Razvoj za Web

Web aplikacije-aplikacije koje koriste sveprisutnost i komunikaciju koju nudi Internet.

Statički i dinamički sajtovi

Statički sajtovi – ne mijenja se sadržaj stranice

Dinamički sajtovi – omogućavaju interakciju sa korisnikom

Koriste HTML za interfejs sa klijentom

Omogućavaju korisnicima preuzimanje individualnih i prilagodljivih akcija kao što su rezervacija leta, sjedišta, kupovina itd.

HTML je tehnologija koja se izvršava na strani klijenta

Posao Web servera je da obezbijedi datoteke koje se traže

HTML nije programski jezik – nema konstrukcije koje bi obezbijedile obradu podataka

PHP server – side tehnologija

PHP izvršava skript

Nema direktnu saradnju sa Web pretraživačem

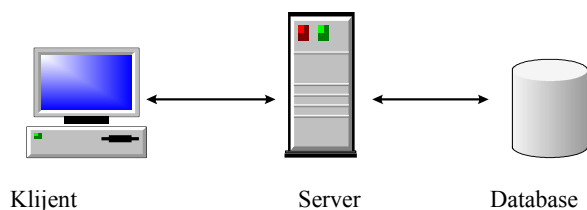
Završni proizvod PHP skripta je HTML strana koja se šalje Web pretraživaču

Ako bi pretraživač primio neobrađen PHP skript, on bi ga pokušao prikazati kao HTML dokument.

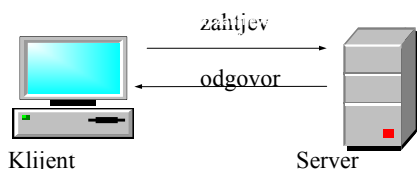
Pretraživači ne mogu izvršavati PHP skriptove

PHP kod se može jedino ubaciti u HTML stranu

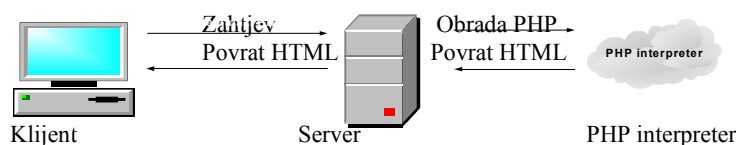
### **Klijent server paradigma**



### **Zahtjev za HTML dokument**



### **Zahtjev za PHP skriptom**



### **Osnove PHP-a – Zahtjev za PHP skriptom**

Razlika između ova dva procesa?

HTML se samo koristi za publikovanje hiperteksta i obrađuje se na strani klijenta

PHP se izvršava na strani servera i predstavlja potpuni programski jezik

## PHP sintaksa

Prije izvršavanja PHP skripta, instrukcija treba da se uputi serveru

Blok sa skriptom se uokviruje sa specijalnim oznakama koje serveru saopštavaju da je riječ o PHP kodu.

Kada server nađe na znak za otvaranje PHP-a, sve što se nalazi između te oznake i oznake za kraj, smatra se PHP-om.

### Program "Zdravo svijete" – Primjer:

#### HTML kod:

```
<html>
<head>
<title>Zdravo svijete!</title>
</head>
<body>
Zdravo svijete!
</body>
</html>
```

### Prvi PHP skript "Zdravo svijete" – Primjer:

#### PHP kod:

```
<html>
<head>
<title>Zdravo svijete!</title>
</head>
<body>
<?php print('Zdravo svijete! Kako Vam se ovo cini?'); ?>
</body>
</html>
```

### Zahtjev za PHP skriptom

Dva PHP skripta

Jedan skript sadrži HTML formu koja može da se šalje, dok je drugi aktivna strana, skript kojem se šalje forma

Kada se forma pošalje do druge strane, PHP omogućava da se sa stranice za akciju pristupi podacima iz forme.

Prvi skript    Index.php

Drugi skript   ko\_ste\_vi.php

### Slanje podataka

Dva načina: GET i POST

Metod GET dodaje podatke URL-u koji se šalje do Web servera. Dužinu URL-a ograničava klijent

Dužina URL-a je ograničena

Nije pogodan za slanje velike količine podataka

Alternativni način za slanje podatak je POST metod

Netscape ne nameće ograničenja (2048 karaktera u dijelu putanje i najviše 2083 u kompletnom URL-u uključujući i string upita)

Macromedia generator nameće ograničenje od beznačajnih 255 karaktera za dužinu URL-a.

Metod slanja se može definisati u kodu pomoću atributa method:

```
<form action="index.php">
```

```
<form action="Ko_ste_vi.php" method = "GET">
```

Sljedeće što je bitno , karakteri koji su rezervisani i koji ne mogu biti dio URL-a ako se ne kodiraju.

Kada želite da ubacite neki literal, morate tu vrijednost kodirati prema kombinaciji (Tabela)

PHP funkcija urlencode() obavlja ovaj posao umjesto korisnika:

Karakter	URL kod
Bjelina razmak)	%20
"	%22
#	%23
%	%25
&	%26
(	%28
)	%29
+	%2B
'	%2C
/	%2F
Karakter	URL kod
:	%2A
;	%2B
<	%2C

=	%2D
>	%2E
?	%2F
@	%40
\	%5C
/	%7C

### Method POST rade drugačije

Najvažniji aspekt method POST-a je da se podaci koji se šalju ne kodiraju u okviru URL-a već se šalju kao dio HTML zaglavlja

Za korisnika je ovo nevidljivo

Drugi bitan aspekt je da string nije ograničen po dužini

Kako odlučiti koji metod izabrati?

Najvažnije je da imate na umu:

Količina podataka koja se šalje

Vidljivost podataka

GET ima ograničenu dužinu putanje i stringa upita

POST može da prosljedi neograničenu količinu podataka i parova vrijednosti ime/vrijednost, ali je URL ograničen kod nekih klijenata

IE ograničava recimo putanju u URL-u i za GET i POST metod na dužinu 2048 karaktera

Ovo morate imati na umu jer je IE jedan od najpopularnijih pretraživača

Metod GET može izazvati i neželjene efekte. Osjetljivi podaci se dodaju u URL (PIN kartice npr.)

Za slanje osjetljivih podatak koristite POST

### Komentari u kodu tri stila pisanja koda

GENERALNO GOVOREĆI KOMENTAR JE DOBRA STVAR ZA PROGRAMERE RADI NAPOMENA

Koriste se u svim programskim jezicima

Postoji mišljenje da oni ne treba da stoje u kodu

Kod bi trebalo da bude napisan tako dobro da sam sebe dokumentuje

Ako se pogrešno napišu mogu dovesti do zabune

S druge strane dobri komentari su od neprocjenjive vrijednosti kasnije

Ne zaboravite da kod treba održavati, adaptirati itd.

Vrlo važno da imate konzistentan stil pisanja koda

- # Ovo je komentar, sakriva samo jednu liniju

- // ovo je komentar, sakriva samo jednu liniju

- /\* Ovo je komentar, sakriva više linija

### Primjer stila koji je TEŽAK za održavanje

```

/*****
*Svijet slike.php          \
*Autor – Amir Silic      *
*                          *
* Rutina prikaza slika    *
*                          *
*****/

```

### Primjer stila koji je POGODAN za održavanje

```

/*
Svijet slike.php
Autor – Amir Silic
Rutina prikaza slika
*/

```

### PRIMJER

```

<?php
/*
File:   who_are_you_advanced.php
Author: Jeremy Allen
Description: Who are you advanced contains multiple form
fields that require personal data from a user.
*/
?>
<html>
<head>
<title>Who are you Advanced!</title>
</head>
<body>
<form action="you_are_advanced.php" method="POST">
Please fill in the following fields:<br>

```

## Promjenljive

Promjenljive u programskom jeziku sve pamte  
Sadrže sve podatke sa kojima aplikacija radi  
PHP takode nije izuzetak  
Ima obilje tipova promjenljivih  
Svaki različit tip podataka ima svoje trikove, funkciju i memorijske zahtjeve  
Promjenljiva je jednostavno "objekat" koji sadrži malo parče podatka.  
Dio su programiranja u bilo kom jeziku pa i u PHP-u  
Suprotno od nje vrijednost koja ne može da se mijenja je literal  
Literal se ne može mijenjati i može se smatrati da je "sirov" podatak  
Identifikator promjenljivih može biti niz slova, donjih crta i cifara  
Identifikatori u PHP-u prave razliku između malih i velikih slova  
Mora početi oznakom za dolar, iza koga slijede veliko slovo (ASCII karakteri od 65 do 90 [A-Z] i od 97-122 [a-z], prošireni ASCII karakteri (VRIJEDNOSTI OD 127-255) ili donja crta.  
Može da sadrži slova od a do z i A do Z, brojeve 0 do 9 i proširene karaktere

## Promjenljive - PRIMJERI

```
$foo = "Amir Silic";  
$bar = "Husein Porobic";  
$fOO = "je razlicito od $foo"; // Case Sensitive  
$ladresa = "100 NO 5"; // ovo je wrong jer je pocetak sa brojem  
$_ladresa = "100 NO 5"; // ova sintaksa je uredi jer ne pocinje s brojem
```

Ako zadate promjenljivoj dobra imena program se lakše čita i prati  
Ime promjenljive treba da definiše na šta se odnosi promjenljiva  
Ime promjenljive treba precizno da identifikuje podatak koji ta promjenljiva predstavlja

## Promjenljive – Pravilne konvencije prilikom zadavanja imena

Lako za razumjeti: pos\_start, speed\_of\_light  
Zbunjujuće: positin\_start p\_strt, PSTART, spd\_of\_light s\_o\_l, s\_o\_lght, SQL,X  
Optimalno ime ne treba da bude suviše dugačko  
Trebaju se lako razlikovati ako ih ima više  
Ako ima više riječi, odlučiti se za neki standardni način razdvajanja

Dva su načina koji se najviše koriste:

Prvo slovo svake riječi piše se veliko slovo ili da se riječi odvajaju donjom crtom

**\$DobroOdvojenaPromjenljiva01,**  
**\$dobro\_odvojena\_promjenljiva01;**

**PHP i tipovi podataka** promjenljiva može da sadrži bilo koji tip podataka u zavisnosti od koncepta u kome se koristi

U PHP-u promjenljiva \$ može da sadrži bilo šta u bilo kom trenutku

## PHP podržava tipove podataka:

Skalarni - Boolean, integer, sa pokretnim zarezom (double), string  
Složeni – Array, object  
Specijalni – Resource, null

## Skalarni tipovi - Boolean

najjednostavnije promjenljive, može da sadrži vrijednosti tačno (True) i netačno (False), a to su ključne riječi u PHP-u

**Primjer koda:** `$display_table = true;` // kreira se Boolean i podešava se na True

Iako se kod promjenljivih pravi razlika između malih i velikih slova, PHP funkcije i ključne riječi uključujući i riječi True i False, ne prave tu razliku

**Primjer koda:**

```
if ($display_table == true  
{  
    render_table();  
}
```

- o Funkcija `render_table()` će biti pozvana ako je vrijednost promjenljive `$display_table` True.
- o Operator `==`, nakon što se procijeni vrijednost izraza, vraća vrijednost Boolean.
- o Ako je `>Boolean` vrijednost koja se vraća True, izvršava s blok sa skriptom koji se nalazi između dvije zagrade

## Skalarni tipovi - Integer

Predstavlja cio broj između 2.147.483.647 do -2.147.483.647

Mogu se definisati u decimalnoj, heksadecimalnoj i oktalnoj notaciji

**Primjer:** Integer

U ovom primjeri Integer uvijek vraća istu vrijednost jer je promjenljiva `$int_max`.

Integer ne može da sadrži brojeve sa decimalnim zarezima

Ako se PHP promjenljivoj dodijeli vrijednost 1, interno se smatra da je riječ o integeru.

## Skalarni tipovi - Brojevi sa pokretnim zarezom

Koriste se i termini `double` ili `float`, pružaju veću preciznost negi integer, PHP dozvoljava nekoliko različitih notacija koje se koriste za dodjeljivanje realnog broja promjenljivoj, Maksimalna vrijednost broja sa pokretnim zarezom je definisana IEEE 64-bitnim formatom, Na većini platformi preciznost broja sa zarezom je približno 14 decimalnih mjesta, npr: Broj sa zarezom

### Skalarni tipovi - String

String predstavlja niz karaktera koji su spojeni zajedno, Svaki karakter u stringu zauzima po 1 bajt, u PHP-u nema nikakvih ograničenja dužine stringova koji se unose, PHP dinamički zauzima memoriju za stringove onako kako string raste, u PHP-u se stringovi mogu definisati na tri načina: - pomoću jednostrukih navodnika, - pomoću dvostrukih navodnika, - pomoću heredoc sintakse

Primjer string literal definisan pomoću jednostrukih navodnika:

```
$my_string = ' Ovo je string koji je definisan sa jednostrukim navodnikom \'Single Quotes\'';
```

**Heksadecimalni i oktalni karakteri** mogu takođe da se kod stringova sa dvostrukim navodom koriste kod Escape sekvence

### Skalarni tipovi - String

ESCAPE SEKVENCA	KARAKTER KOJI PREDSTAVLJA
\n	Linefeed –LF, ili 0x0A (10) u ASCII kodu
\r	Lraj reda –CR, ili 0x0D (13) u ASCII kodu
\t	Horizontalni tabulator – HT, ili 0x09 (9) u ASCII kodu
\\	Kosa crta
\\$	Oznaka za dolar
\"	Dvostruki navodnik

**Heredoc sintaksa** se koristi za manipulaciju sa dugačkim stringovima.

### Složeni tipovi podataka

- o Skalarna vrijednost je atomska vrijednost podatka
- o Predstavlja jedno slovo ili broj
- o To je najmanja jedinica za skladištenje u PHP-u
- o Apsolutno najmanja je Boolean vrijednost koja čuva samo da ili ne
- o Složeniji tipovi podataka idu dalje
- o Nizovi su jedan od predstavnika ovakvih tipova podataka koji se često koriste

### Složeni tipovi podataka - Nizovi

- o Jedan od najfleksibilnijih i najkorisnijih tipova podataka koji postoje u PHP-u
- o Primjer za sekvencijalno numerisani niz u PHP-u: