

Laboratorijske vježbe

- PHP
- DOM

Domaća zadaća



- Proučiti načine kodiranja podataka opisane u sklopu MIME
 - Base64
 - Quoted printable
- Riješiti laboratorijsku vježbu



Nismo više *van struje*

- Rezultati vježbe – *online*, na računalu pinus ili vlastitom
- Provjerite radi li vam *account*
- Svi problemi – CIP

- Pristup pinusu
 - PuTTY / nešto drugo (SSH)
 - WinSCP / nešto drugo (SCP)



Vlastito računalno (prijenosno)



- XAMPP paket
 - Apache
 - MySQL
 - PHP
 - Perl
- Linux paketi
- Bilo što drugo
 - PHP 5.2.x



Struktura direktorija

- Vlastiti direktorij na pinusu - ~
 - /public_html
 - index.html
 - dizajn.css
 - podaci.xml
 - pretraga.php
 - funkcije.php
 - ...
- Prava pristupa – direktoriji 755, datoteke 644
 - Korisnik (user): rw(x)
 - Grupa (group): r-(x)
 - Ostali (other): r-(x)



Pristupanje stranicama

- Poslužitelj na portu 4790
- <http://pinus.cc.fer.hr:4790/~ime/>
 - Pokazuje na direktorij public_html
- Test: phpinfo()

```
<?php phpinfo(); ?>
```

Ciljevi vježbe

- Upoznavanje s PHP-om
 - Sintaksa
 - Korisne funkcije
 - Stringovi
 - Polja
- Upoznavanje s DOM-om
 - Metode
 - svojstva
- Rad sa podacima



Rezultati vježbe

- Obrazac
- XML dokument s podacima
 - XSL predložak prikaza
- Funkcionalna tražilica
 - Dvije skripte
 - pretraga.php
 - funkcije.php



Kako riješiti zadatak?



- Pretraga
 - Ručno *šetati* po DOM stablu
 - Koristiti XPath
 - `/korijen/element[contains(ime, 'a')]`
- Prikaz rezultata
 - HTML generiran PHP-om



PHP – brzi uvod

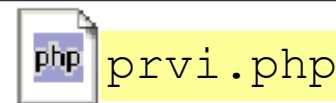
Cilj ovog uvoda u PHP

- ... nije detaljno proučavanje jezika, biblioteka ...
- Jezik se može naučiti samo njegovim korištenjem!
- Unutar predmeta PHP će biti korišten za
 - programsku manipulaciju XML dokumentima
 - stvaranje dinamičkih web stranica
 - mrežno programiranje
- Cilj uvoda je brzo upoznavanje s osnovama izrade jednostavnih programa u PHP-u kako bi brzo svladali početne prepreke i mogli samostalno nastaviti učenje

Prvi PHP program

```
-----  
<?php  
  
echo "Hello, World!";    //stoput vidjeno  
?>  
  
A da probamo nesto novo?  
<?php  
  
echo "Wilkommen, Welt!";  
echo "\n";  
/*  
echo "tu staviti na francuskom";  
*/  
?>  
-----
```

```
>php prvi.php  
-----  
Hello, World!  
A da probamo nesto novo?  
Wilkommen, Welt!  
-----  
  
>
```



PHP blokovi

```
-----  
<?php
```

```
echo "Hello, World!";    //stoput vidjeno
```

```
?>
```

```
A da probamo nesto novo?
```

```
<?php
```

```
echo "Wilkommen, Welt!";
```

```
echo "\n";
```

```
/*
```

```
echo "tu staviti na francuskom";
```

```
*/
```

```
?>  
-----
```

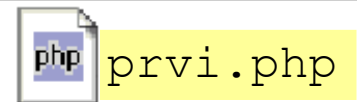
```
>php prvi.php
```

```
-----  
Hello, World!
```

```
A da probamo nesto novo?
```

```
Wilkommen, Welt!
```

```
>
```



Naredbe, odjeljivanje naredbi

```
-----  
<?php  
  
echo "Hello, World"; //stoput vidjeno  
?>  
  
A da probamo nesto novo?  
<?php  
  
echo "Wilkommen, Welt";  
echo "\r";  
/*  
echo "tu staviti na francuskom";  
*/  
?>  
-----
```

```
>php prvi.php  
-----  
Hello, World!  
A da probamo nesto novo?  
Wilkommen, Welt!  
-----  
  
>
```

Komentari

```
-----  
<?php  
  
echo "Hello, World!"; //stoput vidjeno  
?>  
  
A da probamo nesto novo?  
<?php  
  
echo "Wilkommen, Welt!";  
echo "\n";  
/*  
echo "tu staviti na francuskom";  
*/  
?>  
-----
```

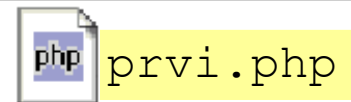
```
>php prvi.php  
-----  
Hello, World!  
A da probamo nesto novo?  
Wilkommen, Welt!  
-----  
  
>
```

Pokretanje skripta

```
-----  
<?php  
  
echo "Hello, World!";    //stoput vidjeno  
?>  
  
A da probamo nesto novo?  
<?php  
  
echo "Wilkommen, Welt!";  
echo "\n";  
/*  
echo "tu staviti na francuskom";  
*/  
?>  
-----
```

```
>php prvi.php
```

```
-----  
Hello, World!  
A da probamo nesto novo?  
Wilkommen, Welt!  
-----  
  
>
```



Varijable i tipovi

```
<?php
```

```
$var1 = TRUE; //boolean  
$var2 = 24; //integer  
$var3 = "or"; //string
```

**definicija varijabli
(ne deklarira se tip!)**

```
$res1 = $var1 + $var2;  
$res2 = $var2 + $var3;  
$res3 = ($var2 == $res2);  
$res4 = is_bool($var1);
```

**provjera tipa varijable
funkcija vraća tip boolean,
oblik: is_ (bool, int, string, float ...)**

```
echo "\$res1=$res1, \$res2=$res2, \$var1=$var1,"  
" \$res3=$res3, \$var3=$var3, \$res4=$res4 \n";
```

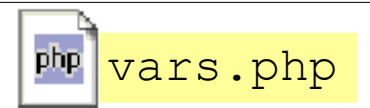
**\$varijabla se zamjenjuje
s njenom vrijednošću**

```
echo gettype($res1), ", ", gettype($res2), ", ",  
" ", gettype($var1), "  
", " ", gettype($res3), ", ", gettype($var3), ", ",  
" ", gettype($res4);  
?>
```

vraća ime tipa varijable

```
>php vars.php
```

```
$res1=25, $res2=24, $var1=1, $res3=1, $var3=or, $res4=1  
integer, integer, boolean, boolean, string, boolean
```



Stringovi

- Nizovi znakova (teoretski neograničene duljine)
- Znakovi kodirani u 8 bita, Unicode nije izravno podržan :(
- Stringove omeđuju jednostruki ili dvostruki navodnici
 - jednostruki navodnici ne podržavaju kontrolne znakove (npr. \r, \t, \n, \\$), ne zamjenjuju varijable s njihovim vrijednostima
- Pristup i promjena pojedinog elementa stringa
 - `$strvar[index]` - indeksi od 0 do `strlen($strvar)`
- Konkatencija stringova: operator `.` (točka)
 - `$str3 = "Hello, " . "World!";`

Stringovi i vrste navodnika

```
<?php
$str1 = 'Hello, ';
$str2 = "World!";

$res1 = $str1 + $str2;
$res2 = $str1 . $str2;

$res3 = $str2;
$res3[strlen($res3)-1] = '?';

echo 0$res1 | $res2 | $res3\n0;
echo "$res1 | $res2 | $res3\n";
?>
```

```
>php strings.php
$res1 | $res2 | $res3\n0 | Hello, World! | World?
```

Neke funkcije za rad sa stringovima

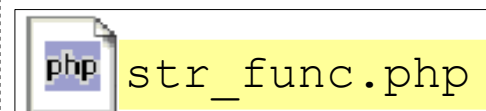
```
<?php
$str = " Hello, world! What a nice day.";
$delimiters = " ,!."; $matching = "day"; $day = 11; $month=21; $year = 1864;

printf("Today is %d-%d-%d.\n", $year, $month, $day);

echo ">" . trim($str) . "<\n";
echo substr($str, 0, strpos($str, ',')) . "\n";

$token = strtok($str, $delimiters);
while($token !== FALSE) {
    echo "|$token";
    if( strcmp($matching, $token) == 0 ) echo "*";
    $token = strtok($delimiters);
}
echo "|\n";
?>
```

```
>php str_func.php
Today is 1864-21-11.
>Hello, world! What a nice day.<
Hello
|Hello|world|What|a|nice|day*|
```



Pretvorba broj ↔ string

```
<?php
$str1 = '10 litara'; $str2 = 'deset litara'; $str3 = '10.0';
$int1 = 10; $bool1 = true; $bool2 = false; $float1 = 10.3;

$val = 0 + $str1; echo $val . ": " . gettype($val) . "\n";
$val = 0 + $str2; echo $val . ": " . gettype($val) . "\n";
$val = 0 + $str3; echo $val . ": " . gettype($val) . "\n";
echo "-----\n";
$str = strval($int1); echo $str . ": " . gettype($str) . "\n";
$str = strval($bool1); echo $str . ": " . gettype($str) . "\n";
$str = strval($bool2); echo $str . ": " . gettype($str) . "\n";
$str = strval($float1); echo $str . ": " . gettype($str) . "\n";
?>
```

```
>php str_conv.php
```

```
10: integer
```

```
0: integer
```

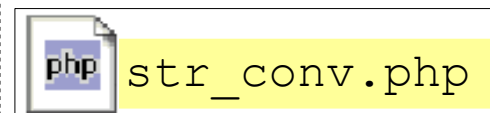
```
10: double
```

```
-----
10: string
```

```
1: string
```

```
: string
```

```
10.3: string
```



Funkcije za rad sa stringovima



The screenshot shows the PHP Manual interface. On the left, a sidebar lists various function categories, with 'String Functions' circled in red. The main content area displays a 'Table of Contents' for string functions, listing numerous functions with brief descriptions.

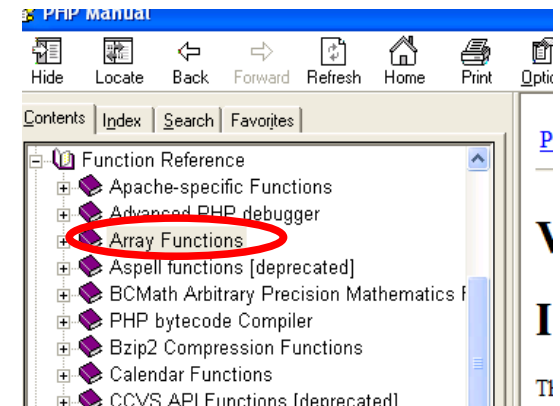
Table of Contents

- [addslashes](#) — Quote string with slashes in a C style
- [addslashes](#) — Quote string with slashes
- [bin2hex](#) — Convert binary data into hexadecimal representation
- [chop](#) — Alias of [rtrim\(\)](#)
- [chr](#) — Return a specific character
- [chunk_split](#) — Split a string into smaller chunks
- [convert_cyr_string](#) — Convert from one Cyrillic character set to another
- [convert_uuencode](#) — Decode a uuencoded string
- [convert_uuencode](#) — Uuencode a string
- [count_chars](#) — Return information about characters used in a string
- [crc32](#) — Calculates the crc32 polynomial of a string
- [crypt](#) — One-way string encryption (hashing)
- [echo](#) — Output one or more strings
- [explode](#) — Split a string by string
- [fprintf](#) — Write a formatted string to a stream
- [get_html_translation_table](#) — Returns the translation table used by [htmlspecialchars\(\)](#) and [htmlspecialchars_decode\(\)](#)
- [hebrew](#) — Convert logical Hebrew text to visual text
- [hebrevc](#) — Convert logical Hebrew text to visual text with newline conversion
- [html_entity_decode](#) — Convert all HTML entities to their applicable characters
- [htmlentities](#) — Convert all applicable characters to HTML entities
- [htmlspecialchars_decode](#) — Convert special HTML entities back to characters
- [htmlspecialchars](#) — Convert special characters to HTML entities
- [implode](#) — Join array elements with a string
- [join](#) — Alias of [implode\(\)](#)
- [levenshtein](#) — Calculate Levenshtein distance between two strings
- [localeconv](#) — Get numeric formatting information
- [ltrim](#) — Strip whitespace (or other characters) from the beginning of a string
- [md5_file](#) — Calculates the md5 hash of a given file
- [md5](#) — Calculate the md5 hash of a string
- [metaphone](#) — Calculate the metaphone key of a string
- [money_format](#) — Formats a number as a currency string
- [nl_langinfo](#) — Query language and locale information
- [nl2br](#) — Inserts HTML line breaks before all newlines in a string
- [number_format](#) — Format a number with grouped thousands
- [ord](#) — Return ASCII value of character
- [parse_str](#) — Parses the string into variables
- [print](#) — Output a string
- [printf](#) — Output a formatted string
- [quoted_printable_decode](#) — Convert a quoted-printable string to an 8 bit string
- [quotemeta](#) — Quote meta characters
- [rtrim](#) — Strip whitespace (or other characters) from the end of a string
- [setlocale](#) — Set locale information
- [sha1_file](#) — Calculate the sha1 hash of a file
- [sha1](#) — Calculate the sha1 hash of a string

Polja



- Polje (array) je u PHP-u izvedeno kao uređena mapa, mapa sadrži parove (ključ, vrijednost)
- Polje se može koristiti kao vektor, polje, stog, red, ...
- Ključevi samo stringovi i cijeli brojevi
- Vrijednosti mogu biti jednostavni i složeni tipovi (uključujući i druga polja – tvorba složenih struktura, npr. stabla)



Stvaranje polja (I)

```
<?php
$arr1 = array( 0 => "nula", 1 => 'jedan', 2 => 'dva' );
$arr2 = array( 'nula' => 0,
               'dva' => 'nije po redu',
               1 => 'ali vrijedan' );
$arr3 = array( 'pocetni' => '-----',
               'prvi' => $arr1,
               'drugi' => $arr2 );

$arr4 = array( 5, 12, 5, 'b', 'a' );

if( is array($arr1) ) echo "\$arr1 je polje!\n";

print_r($arr3);
echo "\n-----\n";
echo $arr1[0] . " " . $arr2['dva'] . " " . $arr3['prvi'][2];
echo "\n-----\n";
print_r($arr4);
?>
```

par (ključ, vrijednost)

ključ => vrijednost

ključevi nisu navedeni!

pristup elementima polja



array1.php

Stvaranje polja (II)

```
>php array1.php
$arr1 je polje!
Array
(
    [pocetni] => -----
    [prvi] => Array
        (
            [0] => nula
            [1] => jedan
            [2] => dva
        )

    [drugi] => Array
        (
            [nula] => 0
            [dva] => nije po redu
            [1] => ali vrijedan
        )
)
```

```
-----
nula nije po redu dva
-----

Array
(
    [0] => 5
    [1] => 12
    [2] => 5
    [3] => b
    [4] => a
)
```

Operacije nad elementima polja

```
<?php

$arr1 = array( 2 => 5, 12, 5, 'e1'
    => 'b', 'e2' => 'a');

print_r($arr1);

echo "\n-----\n";

$arr1[] = 7;

$arr1['e3'] = 'c';

$arr1[3] = 11;

unset($arr1['e1']);

print_r($arr1);

?>
```

```
>php array2.php
```

```
Array
```

```
(
    [2] => 5
    [3] => 12
    [4] => 5
    [e1] => b
    [e2] => a
)
```

```
-----
```

```
Array
```

```
(
    [2] => 5
    [3] => 11
    [4] => 5
    [e2] => a
    [5] => 7
    [e3] => c
)
```



array2.php

Iteriranje nad sadržajem polja

```
<?php

$arr2 = array(0, 1, 2, 'c');

$arr1 = array(5, 12, $arr2, 'el1',
             5, 'el2');

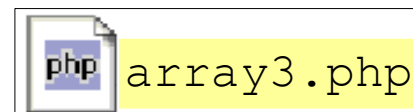
foreach($arr1 as $key => $value) {
    if( is_array($value) == TRUE )
        echo "$key => Polje\n";
    else echo "$key => $value\n";
}

echo "-----\n";

foreach($arr1 as $value)
    echo "$value\n";

?>
```

```
>php array3.php
0 => 5
1 => 12
2 => Polje
3 => el1
4 => 5
5 => el2
-----
5
12
Array
el1
5
el2
```



Operatori

- Gotovo isti kao u jeziku C/C++
 - Aritmetički: +, -, *, /, %, ...
 - Dodjeljivanja: =, +=, . =, ...
 - Usporedbe: ==, !=, ...
 - Inkrementa/dekrementa: ++, -- (pre- i post-)
 - Logički: and, or, xor, !, ...
 - Nad poljima ... (+, ==, ===, !=, <>, !==)

Upravljačke strukture

- Gotovo iste kao u jeziku C/C++
 - if (statement) ... elseif(statement) ... else
 - while (statement)
 - do ... while (statement)
 - for (; ;) ...
 - foreach – novo, uporaba kod iteriranja sadržajem polja
 - switch .. case
 - break, continue
 - return

Funkcije (I)



```
function ime($param1, $param2, ...) {
```

```
    ...
```

```
    return $rval;
```

```
}
```

samo ako postoje parametri funkcije

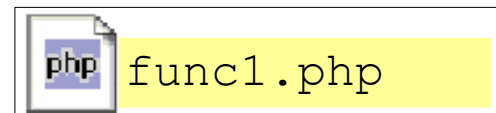
parametri su lokalnog dosega (scope)

samo ako postoji povratna vrijednost

```
$val = add($arg1, $arg2, ...);
```

funkcija ne mora biti definirana prije korištenja!

osnovni prijenos argumenata po vrijednosti (by value)



Funkcije (II)



Prijenos argumenata po referenci (by reference)

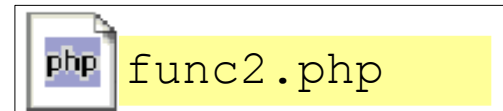
```
function add(&$var) { ...
```

Podrazumijevani argumenti

```
function add($var, $inc=1) { ...
```

```
add(14,7);
```

```
add(12); //add(12,1);
```



Razredi i objekti



- Ne zanima nas pisanje novih razreda, samo uporaba postojećih za stvaranje i korištenje objekata
- Ponovimo osnove objektno-orientiranog programiranja
 - razred je tip ili “predložak” stvaranja objekata
 - razred izvana vidljiv kao skup javnih metoda i atributa
 - instanciranje: postupak stvaranja objekta nekog razreda
 - korištenjem javnih metoda i atributa utječemo na stanje objekta

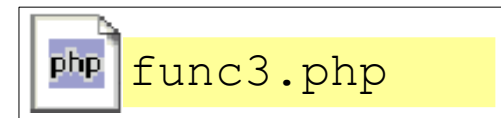
```
class Razred
{
    // atribut
    public $atr = 12;

    // deklaracija metode
    public function inc($amount) {
        $this->atr += $amount;
    }
}
```


Rekurzivni pozivi funkcije



- Ograničena dubina rekurzije zbog male veličine stoga (100-200)
- Primjer u datoteci func3.php: ispis sadržaja polja, funkcionalnost identična funkciji `print_r()` ;



Korištenje objekata



Stvaranje instance razreda

```
$obj1 = new Razred();
```

Pristup vrijednosti javnog atributa

```
$obj1->atr;
```

Poziv javne metode

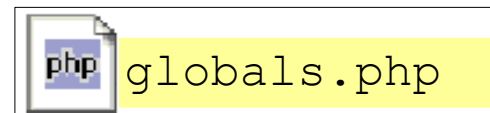
```
$obj1->inc(5);
```

Globalno definirane varijable



```
<?php  
print_r($GLOBALS);  
?>
```

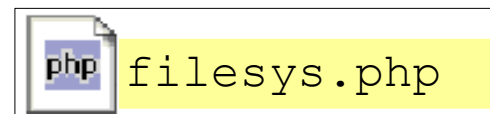
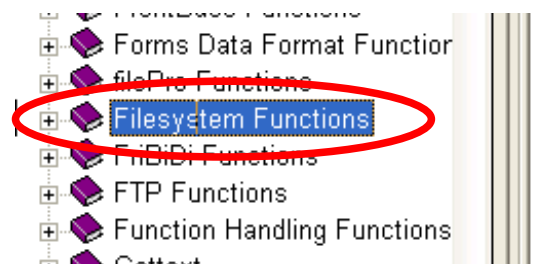
- \$GLOBALS sadrži reference na sve skripti trenutno dostupne varijable, definirane na globalnoj razini
- \$GLOBALS je tzv. jedna od superglobal varijabli (ostale vezane uz web: \$_GET, \$_POST, \$_REQUEST, \$_SERVER, ...)



Rad s datotekama



- Čitanje argumenata naredbenog retka
 - \$argc – broj argumenata
 - \$argv – polje argumenata, početni indeks 0 (ime skripte)
- Funkcije za rad s datotekama gotovo iste kao u jeziku C
 - fopen(ime_datoteke, način_rada), fclose, ...
 - feof, fseek, ...
 - fgets, fwrite ...





Primjer PHP+DOM

Podaci u XML dokumentu

<popis>

 <student upis="redovan">

 <ime>Ivan</ime>

 <prezime>Ivanov</prezime>

 </student>

...

 <student upis="izvanredan">

 <ime>Marko</ime>

 <prezime>Markov</prezime>

 </student>

</popis>

Prikaz rezultata i pretraga

- Tablični prikaz
- Pretraga po prvom slovu
 - Ne želimo obrasce
 - Slično imeniku

Svi A B C D E F G H I J K L M N O P Q R S T U V W X Y

Ime	Prezime	Status
Ivan	Ivanov	redovan
Marko	Markov	izvanredan
Boris	Borisov	redovan
Darko	Darkov	redovan
Petar	Petrakov	redovan
Tomislav	Rajko	izvanredan

Rješenje



- Izvođenje rješenja
 - <http://pinus.cc.fer.hr:4790/~marvin/xml.php>
- Kôd rješenja
 - <http://pinus.cc.fer.hr:4790/~marvin/xml.phps>
- Ulazni podaci
 - <http://pinus.cc.fer.hr:4790/~marvin/primjer.xml>



Pitanja?

- Probleme i pitanja možete i naknadno postaviti u forumu na stranicama predmeta