

## Universiteti i Prishtinës

### Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

#### Lënda: Programimi në Internet



### Java 2: Hyrje në konceptet themelore të PHP-së (2)

**Ushtrimi 1:** Te merren shembuj te ndryshëm me kushtëzimet IF, IF...ELSE, IF...ELSEIF...ELSE, Switch.

```
<?php
$t=date("H");
if ($t<"20")
{
    echo "Kaloni nje dite te mire!";
}
?>
```

-----

```
<?php
$t=date("H");
if ($t<"20")
{
    echo "Kaloni nje dite te mire!";
}
else
{
    echo "Nate e mire!";
    echo "<br>";
    echo "<br>";
}
?>
```

-----

```
<?php
$t=date("H");
if ($t<"10")
{
    echo "Miremengjes!";
}
elseif ($t<"20")
{
    echo "Miredita!";
}
else
{
    echo "Naten e mire!";
    echo "<br>";
    echo "<br>";
}
?>
```

-----

```
<?php
$ngjyrapelqyer="bardhe";
switch ($ngjyrapelqyer)
{
case "bardhe":
    echo "Ngjyra juaj e preferuar eshte ngjyra e bardhe!";
    break;
case "kalter":
    echo "Ngjyra juaj e preferuar eshte ngjyra e kalter!";
    break;
case "gjelber":
    echo "Ngjyra juaj e preferuar eshte ngjyra e gjelber!";
    break;
default:
    echo "Ngjyra juaj e preferuar nuk eshte asnje nga ngjyrat e bardhe, e kalter apo e gjelber!";
}
?>
```

**Ushtrimi 2:** Te merren shembuj nga përdorimi i unazave WHILE, DO...WHILE, FOR, FOREACH.

```
<?php
$a = 0;
$b = 0;

for( $i=0; $i<5; $i++ )
{
    $a += 10;
    $b += 5;
}
echo ("Ne fund te unazes a=$a dhe b=$b" );
echo "<br>";
echo "<br>";

?>
```

```
<?php
$i = 0;
$num = 50;

while( $i < 10)
{
    $num--;
    $i++;
}
echo ("Unaza ndaloi ne i = $i dhe num = $num" );
echo "<br>";
echo "<br>";

?>
```

```
<?php
$i = 0;
$num = 0;
do
{
    $i++;
}while( $i < 10 );
```

```
echo ("Unaza ndaloi ne i = $i" );
echo "<br>";
    echo "<br>";
?>

<?php
$array = array( 1, 2, 3, 4, 5);
foreach( $array as $value )
{
    echo "Vlera eshte $value <br />";

}
echo "<br>";
    echo "<br>";
?>

<?php
$i = 0;

while( $i < 10)
{
    $i++;
    if( $i == 3 )break;
}
echo ("Unaza ndaloi ne i = $i" );
echo "<br>";
    echo "<br>";
?>

<?php
$array = array( 1, 2, 3, 4, 5);
foreach( $array as $value )
{
    if( $value == 3 )continue;
    echo "Vlera eshte $value <br />";
}
?>
```

**Ushtrimi 3:** Te paraqiten forma te ndryshme te përdorimit te funksioneve ne PHP. Te përfshihen krijimi i funksionit, PHP funksionet me parametra, kalimi i argumenteve me ane te referencës, PHP funksionet me vlere kthyesë, vendosja e parametrave te nënkuptuar, thirrja dinamike e funksioneve.

```
<?php
/* Definimi i PHP funksionit */
function shkruajMesazh()
{
    echo "Shembull i thjeshte i funksionit!";
}
/* Thirrja e PHP Funksionit */
shkruajMesazh();
echo "<br>";
    echo "<br>";
?>

<?php
function FunkMbledhja($num1, $num2)
```

```
{
    $sum = $num1 + $num2;
    echo "Shuma e dy numrave eshte : $sum";
}
FunkMbledhja(10, 20);
echo "<br>";
    echo "<br>";
?>

<?php
function shtoPese(&$num)
{
    $num += 5;
}

function shtoGjashte(&$num)
{
    $num += 6;
}
$orignum = 10;
shtoPese( $orignum );
echo "Vlera origjinale eshte: $orignum<br />";
shtoGjashte( $orignum );
echo "Vlera origjinale eshte: $orignum<br />";
echo "<br>";
    echo "<br>";
?>

<?php
function FunkShto($num1, $num2)
{
    $shuma = $num1 + $num2;
    return $shuma;
}
$return_value = FunkShto(10, 20);
echo "Vlera e kthyer prej funksionit eshte: $return_value";
echo "<br>";
    echo "<br>";
?>

<?php
function printo($param = NULL)
{
    print $param;
}
printo("Ky eshte test");
printo();
echo "<br>";
    echo "<br>";
?>

<?php
function Tung()
{
    echo "Tungjatjeta!<br />";
}
```

```
$function_holder = "Tung";  
$function_holder();  
echo "<br>";  
    echo "<br>";  
?>
```

**Ushtrimi 4:** Vargjet – Arrays – Te merren shembuj te ndryshëm te përdorimit te vargjeve ne PHP. Te definohen metodat kryesore te përdorimit te vargjeve dhe te merren shembuj prej tyre.

Një varg është një strukturë e të dhënave që ruan një ose më shumë tipe të ngjashme të vlerave në një vlerë të vetme.

Ka tre llojet te ndryshme të vargjeve dhe çdo vlerë e vargut është e qasshme duke përdorur një ID c që quhet indeksi i vargut (array).

Numeric array - Një varg me indeks numerik. Vlerat janë të ruajtura dhe arrihen në mënyrë lineare  
Associative array - Një varg me string si indeks. Kjo ruan vlerat e elementit në bashkëpunim me vlerat kyçe dhe jo në një mënyrë lineare.

Multidimensional array - Një varg që përmban një ose më shumë vargje dhe vlerat arrihen duke përdorur indeksat e shumëfishta.

```
<?php  
/* Menyra e pare. */  
$numrat = array( 1, 2, 3, 4, 5);  
foreach( $numrat as $value )  
{  
    echo "Vlera eshte: $value <br />";  
}  
echo "<br>";  
    echo "<br>";  
/* Menyra e dyte. */  
$numrat[0] = "nje";  
$numrat[1] = "dy";  
$numrat[2] = "tre";  
$numrat[3] = "kater";  
$numrat[4] = "pese";  
  
foreach( $numrat as $value )  
{  
    echo "Vlera eshte: $value <br />";  
}  
echo "<br>";  
    echo "<br>";  
?>  
  
<?php  
/* Menyra e pare - associate array. */  
$pagat = array(  
    "Janar" => 2000,  
    "Shkurt" => 1000,  
    "Mars" => 500  
);  
  
echo "Paga per Janar: ". $pagat['Janar'] . "<br />";  
echo "Paga per Shkurt: ". $pagat['Shkurt'] . "<br />";  
echo "Paga per Mars: ". $pagat['Mars'] . "<br />";
```

```
echo "<br>";
echo "<br>";

/* Menyra e dyte array. */
$pagat['Janar'] = "larte";
$pagat['Shkurt'] = "mesatare";
$pagat['Mars'] = "ulet";

echo "Paga per Janar e ". $pagat['Janar'] . "<br />";
echo "Paga per Shkurt ". $pagat['Shkurt'] . "<br />";
echo "Paga per Mars e ". $pagat['Mars'] . "<br />";
echo "<br>";
echo "<br>";
?>

<?php
$nota = array(
    "Janar" => array
    (
        "Fizike" => 35,
        "Matematike" => 30,
        "Kimi" => 39
    ),
    "Shkurt" => array
    (
        "Fizike" => 30,
        "Matematike" => 32,
        "Kimi" => 29
    ),
    "Mars" => array
    (
        "Fizike" => 31,
        "Matematike" => 22,
        "Kimi" => 39
    )
);
/* Qasja ne vlerat e multi-dimensional array */
echo "Nota per Janar ne Fizike : " ;
echo $nota['Janar']['Fizike'] . "<br />";
echo "Nota per Shkurt ne Matematike : ";
echo $nota['Shkurt']['Matematike'] . "<br />";
echo "Nota per Mars ne Kimi : " ;
echo $nota['Mars']['Kimi'] . "<br />";
echo "<br>";
echo "<br>";
?>
```

**Ushtrimi 5.** Te merren shembuj me radhitjen (sort) ne PHP. Te perdoren funksione te ndryshme te sortimit dhe te shihet efekti i tyre.

- sort() - sort arrays in ascending order
- rsort() - sort arrays in descending order
- asort() - sort associative arrays in ascending order, according to the value

- ksort() - sort associative arrays in ascending order, according to the key
- arsort() - sort associative arrays in descending order, according to the value
- krsort() - sort associative arrays in descending order, according to the key

```
<?php
$veturat=array("Volvo","BMW","Toyota");
sort($veturat);
```

```
$clength=count($veturat);
for($x=0;$x<$clength;$x++)
{
    echo $veturat[$x];
    echo "<br>";
}
echo "<br>";
echo "<br>";
?>
```

```
<?php
$numrat=array(4,6,2,22,11);
sort($numrat);
```

```
$arlength=count($numrat);
for($x=0;$x<$arlength;$x++)
{
    echo $numrat[$x];
    echo "<br>";
}
echo "<br>";
echo "<br>";
?>
```

```
<?php
$veturat=array("Volvo","BMW","Toyota");
rsort($veturat);
```

```
$clength=count($veturat);
for($x=0;$x<$clength;$x++)
{
    echo $veturat[$x];
    echo "<br>";
}
echo "<br>";
echo "<br>";
?>
```

```
<?php
$numrat=array(4,6,2,22,11);
rsort($numrat);
```

```
$arlength=count($numrat);
for($x=0;$x<$arlength;$x++)
{
    echo $numrat[$x];
    echo "<br>";
}
```

```
}  
echo "<br>";  
echo "<br>";  
?>
```

5b)

```
<?php  
$mosha=array("Petrit"=>"35","Arben"=>"37","Jeton"=>"43");  
asort($mosha);
```

```
foreach($mosha as $x=>$x_value)  
{  
    echo "Celesi=" . $x . ", Vlera=" . $x_value;  
    echo "<br>";  
}  
    echo "<br>";  
echo "<br>";  
?>
```

```
<?php  
$mosha=array("Petrit"=>"35","Arben"=>"37","Jeton"=>"43");  
ksort($mosha);
```

```
foreach($mosha as $x=>$x_value)  
{  
    echo "Celesi=" . $x . ", Vlera=" . $x_value;  
    echo "<br>";  
}  
    echo "<br>";  
echo "<br>";  
?>
```

```
<?php  
$mosha=array("Petrit"=>"35","Arben"=>"37","Jeton"=>"43");  
arsort($mosha);
```

```
foreach($mosha as $x=>$x_value)  
{  
    echo "Celesi=" . $x . ", Vlera=" . $x_value;  
    echo "<br>";  
}  
    echo "<br>";  
echo "<br>";  
?>
```

```
<?php  
$mosha=array("Petrit"=>"35","Arben"=>"37","Jeton"=>"43");  
krsort($mosha);
```

```
foreach($mosha as $x=>$x_value)  
{  
    echo "Celesi=" . $x . ", Vlera=" . $x_value;  
    echo "<br>";  
}  
    echo "<br>";  
echo "<br>";
```



## Ushtrimi 6: Variablat GLOBAL te PHP-se – SUPERGLOBALS

- \$GLOBALS
- \$\_SERVER
- \$\_REQUEST
- \$\_POST
- \$\_GET
- \$\_FILES
- \$\_ENV
- \$\_COOKIE
- \$\_SESSION

### PHP \$GLOBAL

\$GLOBAL is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods). PHP stores all global variables in an array called \$GLOBALS[index].

### PHP \$\_SERVER

\$\_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.

### PHP \$\_REQUEST

PHP \$\_REQUEST is used to collect data after submitting an HTML form.

### PHP \$\_POST

PHP \$\_POST is widely used to collect form data after submitting an HTML form with method="post". \$\_POST is also widely used to pass variables.

### PHP \$\_GET

PHP \$\_GET can also be used to collect form data after submitting an HTML form with method="get". \$\_GET can also collect data sent in the URL.

Element/Code	Description
\$_SERVER['PHP_SELF']	Returns the filename of the currently executing script
\$_SERVER['GATEWAY_INTERFACE']	Returns the version of the Common Gateway Interface (CGI) the server is using
\$_SERVER['SERVER_ADDR']	Returns the IP address of the host server
\$_SERVER['SERVER_NAME']	Returns the name of the host server (such as www.w3schools.com)
\$_SERVER['SERVER_SOFTWARE']	Returns the server identification string (such as Apache/2.2.24)
\$_SERVER['SERVER_PROTOCOL']	Returns the name and revision of the information protocol (such as HTTP/1.1)
\$_SERVER['REQUEST_METHOD']	Returns the request method used to access the page (such as POST)
\$_SERVER['REQUEST_TIME']	Returns the timestamp of the start of the request (such as 1377687496)
\$_SERVER['QUERY_STRING']	Returns the query string if the page is accessed via a query string
\$_SERVER['HTTP_ACCEPT']	Returns the Accept header from the current request
\$_SERVER['HTTP_ACCEPT_CHARSET']	Returns the Accept_Charset header from the current request (such as utf-8,ISO-8859-1)
\$_SERVER['HTTP_HOST']	Returns the Host header from the current request
\$_SERVER['HTTP_REFERER']	Returns the complete URL of the current page (not reliable because not all user-agents support it)
\$_SERVER['HTTPS']	Is the script queried through a secure HTTP protocol
\$_SERVER['REMOTE_ADDR']	Returns the IP address from where the user is viewing the current page
\$_SERVER['REMOTE_HOST']	Returns the Host name from where the user is viewing the current page
\$_SERVER['REMOTE_PORT']	Returns the port being used on the user's machine to communicate with the web server
\$_SERVER['SCRIPT_FILENAME']	Returns the absolute pathname of the currently executing script
\$_SERVER['SERVER_ADMIN']	Returns the value given to the SERVER_ADMIN directive in the web server configuration file (if your script runs on a virtual host, it will be the value defined for that virtual host) (such as someone@w3schools.com)
\$_SERVER['SERVER_PORT']	Returns the port on the server machine being used by the web server for communication (such as 80)
\$_SERVER['SERVER_SIGNATURE']	Returns the server version and virtual host name which are added to server-generated pages
\$_SERVER['PATH_TRANSLATED']	Returns the file system based path to the current script
\$_SERVER['SCRIPT_NAME']	Returns the path of the current script
\$_SERVER['SCRIPT_URI']	Returns the URI of the current page

```

<?php
$x = 75;
$y = 25;

function mbledhja()
{
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}

mbledhja();

```

```
echo $z;

echo "<br>";
echo "<br>";
?>
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
echo "<br>";
echo "<br>";
?>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
Emri: <input type="text" name="fname">
<input type="submit">
</form>

<?php
$emri = $_REQUEST['fname'];
echo $emri;
?>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
Emri: <input type="text" name="fname">
<input type="submit">
</form>

<?php
$emri = $_POST['fname'];
echo $emri;
?>
```

#### fajlli.php

```
<a href="fajlli.php?subject=PHP&vendi=ushtrimet ne klase">Testo $GET</a>
<br />

<?php
echo "Meso " . $_GET['subject'] . " ne " . $_GET['vendi'];

echo "<br>";
echo "<br>";
?>
```

**Ushtrimi 7:** Te krijohet nje PHP skripte duke përdorur unazën FOR e cila ne dalje na paraqet pamjen ne vijim:



**Ushtrimi 8:** Eshte dhene vargu ne vijim: `$x = array(1, 2, 3, 4, 5);` Te largohet nje element (cilido) nga vargu i dhene. Pasi te largohet elementi, vargu te paraqitet ne forme normale.

Pritet: `array(5) { [0]=> int(1) [1]=> int(2) [2]=> int(3) [3]=> int(4) [4]=> int(5) }`  
`array(4) { [0]=> int(1) [1]=> int(2) [2]=> int(3) [3]=> int(5) }`

**Ushtrimi 9:** Te shkruhet nje PHP funksion, i cili kontrollon nëse nje fajle e dhene eshte POLINDROM.

**Ushtrimi 10:** Te shkruhet nje PHP skripte e cila përmes *preg\_grep* kërkon per nje fjale te caktuar, dhe pastaj kthen si rezultat ate qe eshte bazuar filtrimi. Te perdoret *PREG\_GREP\_INVERT*.

Ky eshte vargu i dhene: `array(0, 1, 2, 'tre', 4, 5, 'gjashte', 7, 8, 'nente', 10);`  
Ne dalje pritet:

- a) `Array ( [4] => 4 [5] => 5 )`
- b) `Array ( [3] => tre [6] => gjashte [9] => nente )`.