



UNIVERSITETI I PRISHTINËS "HASAN PRISHTINA"
FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE

Lënda: Programimi në Internet

Ligjëruese: Dr. Ing. Lule Ahmedi

Ushtrimet: MSc. Korab Rrmoku

Email: (korab.rrmoku@uni-pr.edu)

Web: korabrrmoku.tk

Salla: 616

Dita: E Enjte

Ora: Grupi i I-rë – prej 17:00 deri në 18:30

Pjesëmarrja është OBLIGATIVE!



UNIVERSITETI I PRISHTINËS “HASAN PRISHTINA”

FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE

Java: I (e parë) - Ushtrime

Tema: HTML 5 – Hyper Text Markup Language

Përmbajtja

- HTML5 New Features
- HTML5 Canvas
- HTML5 SVG
- HTML5 Drag/Drop
- HTML5 Video/Audio
- HTML5 Form Elements/Attributes
- HTML5 Semantic/Web Storage/App Cache/Web Workers/SSE



What is HTML5?

- HTML5 is the new standard for HTML.
- HTML5 is designed to deliver almost everything you want to do online without requiring additional plugins. It does everything from animation to apps, music to movies, and can also be used to build complicated applications that run in your browser.
- The HTML 5 working group includes AOL, Apple, Google, IBM, Microsoft, Mozilla, Nokia, Opera, and hundreds of other vendors.
- HTML5 is still a work in progress. However, all major browsers support many of the new HTML5 elements and APIs.

What is HTML5?

- In HTML5 there is only one `<!doctype>` declaration, and it is very simple:

```
<!DOCTYPE html>
```

- A simple HTML5 document, with the minimum of required tags:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
</head>

<body>
Content of the document.....
</body>

</html>
```

HTML5 – New Features

Some of the most interesting new features in HTML5:

- The <canvas> element for 2D drawing
- The <video> and <audio> elements for media playback
- Support for local storage
- New content-specific elements, like <article>, <footer>, <header>, <nav>, <section>
- New form controls, like calendar, date, time, email, url, search

Complete reference list of HTML5 tags/global attributes...

<http://www.w3schools.com/tags/default.asp>

HTML5 Canvas

- A canvas is a rectangular area on an HTML page, and it is specified with the `<canvas>` element.
- HTML5 `<canvas>` element is used to draw graphics, on the fly, via scripting (usually JavaScript).
- It is only a container for graphics. You must use a script to actually draw the graphics.
- The markup looks like this:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

Note: Always specify an id attribute (to be referred to in a script), and a width and height attribute to define the size of the canvas.

Ushtrimi > (Canvas - Det 1).

HTML5 Canvas (2)

- **Canvas Coordinates**

2 dimensional grid, where the upper-left corner has the coordinate (0,0)

- **Canvas - Paths**

To draw straight lines we will use the following two methods: `moveTo(x,y)`, `lineTo(x,y)`. (Canvas - Det 2)

To draw a circle on a canvas, we will use the following method: `arc(x,y,r,start,stop)` (Canvas – Det 3)

- **Canvas – Text**

the most important property and methods are:

`font` - defines the font properties for text

`fillText(text,x,y)` - Draws "filled" text on the canvas
(Canvas – Det 4)

`strokeText(text,x,y)` - Draws text on the canvas (no fill)
(Canvas – Det 5)

HTML5 Canvas (3)

- **Canvas – Gradients**

two different types of gradients:

`createLinearGradient(x,y,x1,y1)`

`createRadialGradient(x,y,r,x1,y1,r1)` (Canvas – Det 6)

- **Canvas - Images**

To draw an image on a canvas, we will use the following method: `drawImage(image,x,y)` (Canvas – Det 7)

HTML SVG (Scalable Vector Graphics)

- SVG is used to define vector-based graphics for the Web
- SVG defines the graphics in XML format
- SVG graphics do NOT lose any quality if they are zoomed or resized
- Every element and every attribute in SVG files can be animated

Differences Between SVG and Canvas

- SVG is a language for describing 2D graphics in XML.
- Canvas draws 2D graphics, on the fly (with a JavaScript).
- In SVG, each drawn shape is remembered as an **object**. If attributes of an SVG object are changed, the browser can automatically re-render the shape.

HTML SVG (2)

- In HTML5, you can embed SVG elements directly into your HTML page:
- Use SVG tag.

```
<html>
<body>

<svg width="300" height="200">
  <polygon points="100,10 40,180 190,60 10,60 160,180"
    style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />
</svg>

</body>
</html>
```

HTML Drag and Drop

- In HTML5, drag and drop is part of the standard, and any element can be draggable.
- To make an element draggable, set the draggable attribute to true:

```
<img draggable="true">
```
- Specify what should happen when element is Dragged
 - *ondragstart* attribute specifies what function to call when an element is dragged.
 - *dataTransfer.setData* set the data type and the id of the dragged data (element).
- Specify where the data can be dropped
 - ondragover* event specifies where the dragged data can be dropped.
- Finally, drop the element.
 - ondrop* attribute specifies what function to call when the data is dropped.

HTML Video

- Until now, there has not been a standard for showing a video/movie on a web page.
- HTML5 defines a new element which specifies a standard way to embed a video/movie on a web page: the <video> element.

To show a video in HTML5, all you need is:

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogv" type="video/ogg">  
Your browser does not support the video tag.  
</video>
```

HTML Video

- Video Formats and Browser Support

Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	NO Update: Firefox 21 running on Windows 7, Windows 8, Windows Vista, and Android now supports MP4	YES	YES
Safari	YES	NO	NO
Opera	NO	YES	YES

- Create simple play/pause + resize controls for a video:
- Ushtrime> [Video.html](#)

HTML Audio

- Until now there has not been a standard for playing audio files on a web page.
- HTML5 defines a new element which specifies a standard way to embed an audio file on a web page: the `<audio>` element.

To play an audio in HTML5, all you need is:

```
<audio controls>  
  <source src="horse.ogg" type="audio/ogg">  
  <source src="horse.mp3" type="audio/mpeg">  
  Your browser does not support the audio element.  
</audio>
```

HTML Audio

- Audio Formats and Browser Support

Browser	MP3	Wav	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	NO Update: Firefox 21 running on Windows 7, Windows 8, Windows Vista, and Android now supports MP3	YES	YES
Safari	YES	YES	NO
Opera	NO	YES	YES

- Create an audio control:
- Ushtrime> [Audio.html](#)

HTML5 Input Types

- HTML5 has several new input types for forms. These new features allow better input control and validation.

Some of them are:

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

HTML5 Form Elements

- HTML5 has the following new form elements:

`<datalist>`, `<keygen>`, `<output>`

The `<datalist>` element specifies a list of pre-defined options for an `<input>` element.

The `<datalist>` element is used to provide an "autocomplete" feature on `<input>` elements. Users will see a drop-down list of pre-defined options as they input data.

(Ushtrime> New Form Element - Detyra 1)

The `<keygen>` element is to provide a secure way to authenticate users.

The `<keygen>` tag specifies a key-pair generator field in a form. The private key is stored locally, and the public key is sent to the server.

When the form is submitted, two keys are generated, one private and one public. The public key could be used to generate a client certificate to authenticate the user in the future.

The `<output>` element represents the result of a calculation (like one performed by a script). *(Ushtrime> New Form Element – Detyra 2)*

HTML5 New Form Attributes

New attributes for <form>:

-autocomplete (*Ushtrime*> *8- New Form Attribute - Detyra 1*)

-novalidate (*Ushtrime*> *8- New Form Attribute - Detyra 2*)

New attributes for <input>:

autocomplete

form

formnovalidate

list

pattern (regexp)

Step

autofocus

formaction

formtarget

min and max

placeholder

formmethod

formenctype

height and width

multiple

required

(*Ushtrime*> *W3Schools*)

HTML Semantics (2)

Tag	Description
<code><article></code>	Defines an article
<code><aside></code>	Defines content aside from the page content
<code><figcaption></code>	Defines a caption for a <code><figure></code> element
<code><figure></code>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<code><footer></code>	Defines a footer for a document or section
<code><header></code>	Specifies a header for a document or section
<code><mark></code>	Defines marked/highlighted text
<code><nav></code>	Defines navigation links
<code><section></code>	Defines a section in a document
<code><time></code>	Defines a date/time

HTML5 +

- Web Storage
- App Cache
- Web Workers
- SSE

HTML5 + (2)

What is HTML5 Web Storage?

- With HTML5, web pages can store data locally within the user's browser.
- Earlier, this was done with cookies. However, Web Storage is more secure and faster. The data is not included with every server request, but used ONLY when asked for. It is also possible to store large amounts of data, without affecting the website's performance.
- The data is stored in key/value pairs, and a web page can only access data stored by itself.

HTML5 + (3)

localStorage and sessionStorage

- There are two new objects for storing data on the client:

localStorage - stores data with no expiration date

sessionStorage - stores data for one session

HTML5 + (4)

The localStorage Object

- The localStorage object stores the data with no expiration date. The data will not be deleted when the browser is closed, and will be available the next day, week, or year.

Example

```
localStorage.lastname="Smith";  
document.getElementById("result").innerHTML="Last name: "  
+ localStorage.lastname;
```

Example explained:

Create a localStorage key/value pair with key="lastname" and value="Smith"

Retrieve the value of the "lastname" key and insert it into the element with id="result"

HTML5 + (5)

The sessionStorage Object

- The sessionStorage object is equal to the localStorage object, **except** that it stores the data for only one session. The data is deleted when the user closes the browser window.

Example

```
if (sessionStorage.clickcount)
{
    sessionStorage.clickcount=Number(sessionStorage.clickcount)+1;
}
else
{
    sessionStorage.clickcount=1;
}
document.getElementById("result").innerHTML="You have clicked the
button " + sessionStorage.clickcount + " time(s) in this session.";
```

HTML5 + (6)

What is Application Cache?

- HTML5 introduces application cache, which means that a web application is cached, and accessible without an internet connection.
- Application cache gives an application three advantages:
 - Offline browsing - users can use the application when they're offline
 - Speed - cached resources load faster
 - Reduced server load - the browser will only download updated/changed resources from the server

HTML5 + (7)

What is Application Cache?

- The example below shows an HTML document with a cache manifest (for offline browsing):

Example

```
<!DOCTYPE HTML>
```

```
<html manifest="demo.appcache">
```

```
<body>
```

```
The content of the document.....
```

```
</body>
```

```
</html>
```

CACHE MANIFEST

/theme.css

/logo.gif

/main.js

HTML5 + (8)

What is a Web Worker?

- When executing scripts in an HTML page, the page becomes unresponsive until the script is finished.
- A web worker is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page. You can continue to do whatever you want: clicking, selecting things, etc., while the web worker runs in the background.

Server-Sent Events - One Way Messaging

- A server-sent event is when a web page automatically gets updates from a server.
- This was also possible before, but the web page would have to ask if any updates were available. With server-sent events, the updates come automatically.
- Examples: Facebook/Twitter updates, stock price updates, news feeds, sport results, etc.

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Programimi në Internet



Java 2: Hyrje në konceptet themelore të PHP-së

Ushtrimi 1: Të definohen elementet bazë të PHP-së, si sintaksa themelore, variablat si dhe manipulimi me variabla.

```
<?php
echo "Pershendetje!";
?>
```

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

```
<?php
//Case in-sensitivity
ECHO "Pershendetje!<br>";
echo "Pershendetje!<br>";
EcHo "Pershendetje!<br>";
?>
```

```
<?php
$ngjyra="verdhe";
echo "Lapsi ka ngjyre te " . $ngjyra . "<br>";
echo "Lapsi ka ngjyre te " . $NGJYRA . "<br>";
echo "Lapsi ka ngjyre te " . $ngjYRA . "<br>";
?>
```

Ushtrimi 2: a) Të përdoren php variablat për llogaritje të tjeshtë, b) Të bëhet dallimi i përdorimit të variablave **lokale** dhe **globale** në php, c) Shembull i qasjes në variabla me **GLOBAL** arrays, d) Përdorimi dhe ruajtja e vlerës së variablave edhe pas përfundimit të ekzekutimit në funksion (përdorimi i **static** variablave)

```
a) <?php
$x=5;
$y=6;
$z=$x+$y;
$v=$x*$y;
$w=$x/$y;
echo $z;
echo "<br>";
```

```
echo $v;  
echo "<br>";  
echo $w;  
?>
```

```
<?php  
$x=10;  
$y=6;
```

```
echo ($x + $y);  
echo "<br>";  
echo ($x - $y);  
echo "<br>";  
echo ($x * $y);  
echo "<br>";  
echo ($x / $y);  
echo "<br>";  
echo ($x % $y);  
?>
```

```
b)  
<?php  
$x=5; // global scope
```

```
function myTest()  
{  
    $y=10; // local scope  
    echo "<p>Test variablat brenda funksionit:<p>";  
    echo "Variabla x eshte: $x";  
    echo "<br>";  
    echo "Variabla y eshte: $y";  
}
```

```
myTest();
```

```
echo "<p>Test variablat jashte funksionit:<p>";  
echo "Variabla x eshte: $x";  
echo "<br>";  
echo "Variabla y eshte: $y";  
?>
```

```
-----  
<?php  
$x=5;  
$y=10;
```

```
function myTest1()  
{  
    global $x,$y;
```

```
$y=$x+$y;  
}
```

```
myTest1();  
echo $y; // outputs 15  
?>
```

```
c)  
<!--Qasje ne Variabla me GLOBAL arrays-->  
<?php  
$x=5;  
$y=10;
```

```
function myTest2()  
{  
$GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];  
}
```

```
myTest2();  
echo $y; // outputs 15  
?>
```

```
d)  
<?php  
function myTest3()  
{  
static $x=0;  
echo $x;  
$x++;  
}
```

```
myTest3();  
myTest3();  
myTest3();
```

```
?>  
</body>  
</html>
```

Ushtrimi 3: a) Të përdoret funksioni `var_dump()` i cili kthen tipin e te dhënave dhe vlerën për numra të caktuar.

b) Të përdoret `var_dump()` për numrat me presje dhjetore.

```
a)  
<?php  
$x = 5985;  
var_dump($x);  
echo "<br>";  
$x = -345; // numrat negativ
```



```
var_dump($x);  
echo "<br>";  
$x = 0x8C; // numrat hexadecimal  
var_dump($x);  
echo "<br>";  
$x = 047; // numrat oktal  
var_dump($x);  
echo "<br>";  
?>
```

```
b)  
<?php  
$x = 10.365;  
var_dump($x);  
echo "<br>";  
$x = 2.4e3;  
var_dump($x);  
echo "<br>";  
$x = 8E-5;  
var_dump($x);  
echo "<br>";  
echo "<br>";  
?>
```

```
<?php  
$cars=array("Volvo","BMW","Toyota");  
var_dump($cars);  
echo "<br>";  
?>
```

Ushtrimi 4: Të krijohet një klasë e thjeshtë dhe një funksion i thjeshtë nga i cili përcaktohet ngjyra e një objekti (në rastin tonë p.sh. vetura)

```
<?php  
class Vetura  
{  
    var $ngjyra;  
    function Vetura($ngjyra="gjelber")  
    {  
        $this->ngjyra = $ngjyra;  
    }  
    function cfare_ngjyre()  
    {  
        return $this->ngjyra;  
    }  
}  
  
function printo_var($obj)
```

```
{
    foreach (get_object_vars($obj) as $prop => $val)
    {
        echo "\t$prop = $val\n";
    }
}

echo "<br>";
// instantizimi i nje objekti
$minivetura = new Vetura("bardhe");
echo "<br>";
// tregimi i vetive te vetures mini
echo "\minivetura: Vetite\n";
printo_var($minivetura);
echo "<br>";
echo "<br>";
?>
```

Ushtrimi 5: Përdorimi i disa string funksioneve me kryesore për të vërejtur ndikimin e tyre.

```
<?php
echo strlen("Pershendetje tung!");
echo "<br>";
echo "<br>";
?>
```

```
<?php
echo strpos("Pershendetje tung!", "tung");
echo "<br>";
echo "<br>";
?>
```

```
<?php
$str = addslashes("Pershendetje Tung!", "T");
echo($str);
echo "<br>";
echo "<br>";
```

```
$str = "Miresevini ne faqen time Personale!";
echo $str."<br>";
echo addslashes($str,'A..Z')."<br>";
echo addslashes($str,'a..z')."<br>";
echo addslashes($str,'a..g');
echo "<br>";
echo "<br>";

?>
```

```
<?php
$str = "Pershendetje!\n\n";
echo $str;
echo chop($str);
echo "<br>";
echo "<br>";
?>
```

```
<?php
$str = 'nje,dy,tre,kater';

// zero limit
print_r(explode(',',$str,0));
```

```
// positive limit
print_r(explode(',',$str,2));
```

```
// negative limit
print_r(explode(',',$str,-1));
echo "<br>";
echo "<br>";
?>
```

```
<?php
$find = array("Pershendetje","tung");
$replace = array("B");
$arr = array("Pershendetje","tung","!");
print_r(str_replace($find,$replace,$arr));
?>
```

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

Ushtrimi 6: Definimi dhe përdorimi i konstanteve në PHP. a) dallimi në mes të përdorimit të parametrave të tretë (true, false) për të dhënë kuptim për case – (in)sensitive. b) Përdorimi i konstanteve dhe vërtetimi se ato nuk mund të ndryshohen pasi që të përdoren njëherë.

a)

```
<?php
// define a case-insensitive constant
define("TUNG", "Mireseerdhet ne ushtrime!", true);
echo TUNG;
echo "<br>";
// will also output the value of the constant
echo tung;
?>
```

b)

```
<?php
define("Cmimi_PC",750);
define("Cmimi_Printer", 125.5);
define("Cmimi_HD", 75.05);
define("Cmimi_PC",25); /*nuk mund të nderrohet vlera e const Cmimi_PC */
echo "<br>";
echo "<br>";

echo "PC ". Cmimi_PC ."<br />";
echo "Printer ". Cmimi_Printer ."<br />";
echo "HD ". Cmimi_HD ."<br />";
echo "<br>";
?>
```

Ushtrimi 7: Përdorimi i operatorëve të shoqërimit në PHP. b) Përdorimi i i operatorëve të bashkimit, c) Aplikimi i inkrementit dhe dekrementit në PHP variablat, d) përdorimi i operatorëve krahasues, e) përdorimi i operatorëve në Array.

<!--//OPERATORET-->

```
<?php
$x=10;
$y=6;

echo ($x + $y);
echo "<br>";
echo "<br>";
echo "<br>";
echo ($x - $y);
echo "<br>";
echo "<br>";
echo "<br>";
echo ($x * $y);
echo "<br>";
echo "<br>";
echo "<br>";
echo ($x / $y);
echo "<br>";
echo "<br>";
echo "<br>";
echo ($x % $y);
?>
```

b) <!--//assignment operatore

-->

```
<?php
$x=10;
echo $x;
```

```
echo "<br>";
echo "<br>";
$y=20;
$y += 100;
echo $y;
echo "<br>";
echo "<br>";
$z=50;
$z -= 25;
echo $z;
echo "<br>";
echo "<br>";
$i=5;
$i *= 6;
echo $i;
echo "<br>";
echo "<br>";
$j=10;
$j /= 5;
echo $j;
echo "<br>";
echo "<br>";
$k=15;
$k %= 4;
echo $k;
echo "<br>";
echo "<br>";
?>
```

c) <!--//string operatoret e bashkimit-->

```
<?php
$a = "Hello";
$b = $a . " world!";
echo $b; // outputs Pershendetje!
echo "<br>";
echo "<br>";
```

```
$x="Hello";
$x .= " world!";
echo $x; // outputs Pershendetje!
echo "<br>";echo "<br>";
?>
```

d) <!--//inkrementi dhe dekrementi-->

```
<?php
$x=10;
```

```
echo ++$x;  
echo "<br>";  
echo "<br>";
```

```
$y=10;  
echo $y++;  
echo "<br>";  
echo "<br>";
```

```
$z=5;  
echo --$z;  
echo "<br>";  
echo "<br>";  
$i=5;  
echo $i--;  
echo "<br>";echo "<br>";  
?>
```

d) <!--//php operatoret krahasues-->

```
<?php  
$x=100;  
$y="100";
```

```
var_dump($x == $y); // returns true because values are equal  
echo "<br>";  
echo "<br>";  
var_dump($x === $y); // returns false because types are not equal  
echo "<br>";  
echo "<br>";  
var_dump($x != $y); // returns false because values are equal  
echo "<br>";  
echo "<br>";  
var_dump($x !== $y); // returns true because types are not equal  
echo "<br>";  
echo "<br>";
```

```
$a=50;  
$b=90;
```

```
var_dump($a > $b);  
echo "<br>";  
echo "<br>";  
var_dump($a < $b);  
?>
```

e) <!--//PHP array operatoret-->

```
<?php
$x = array("a" => "red", "b" => "green");
$y = array("c" => "blue", "d" => "yellow");
$z = $x + $y; // union of $x and $y
var_dump($z);
echo "<br>";
var_dump($x == $y);echo "<br>";
var_dump($x === $y);echo "<br>";
var_dump($x != $y);echo "<br>";
var_dump($x <> $y);echo "<br>";
var_dump($x !== $y);echo "<br>";echo "<br>";
?>

</body>
</html>
```

Ushtrimi 8: Të ipen dy data të caktuara dhe pastaj të gjindet diferenca në mes të atyre datave në vite, muaj dhe dite, duke shfrytëzuar funksione dhe parametra të PHP-së.

Ushtrimi 9: Të shkruhet funksioni në PHP i cili bën shkrimin e të gjitha shkronjave prej a deri ne zh.

Ushtrimi 10: Të shkruhet funksioni në PHP i cili i largon të gjitha karakteret përveç: a-z A-Z 0-9 or " "

Ushtrimi 11: Të shkruhet PHP skripta e cila e gjen karakterin e parë dhe pozitën e tij që është ndryshe në mes të dy stringjeve me gjatësi të barabartë.

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Programimi në Internet



Java 3: 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)`.

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Programimi në Internet



Java 4: PHP Klasat dhe objektet – PHP në OO

Konceptet themelore të OOP:

Object	Properties	Methods
	<code>car.name = Fiat</code> <code>car.model = 500</code> <code>car.weight = 850kg</code> <code>car.color = white</code>	<code>car.start()</code> <code>car.drive()</code> <code>car.brake()</code>

- **Class:** This is a programmer-defined datatype, which includes local functions as well as local data. You can think of a class as a template for making many instances of the same kind (or class) of object.
- **Object:** An individual instance of the data structure defined by a class. You define a class once and then make many objects that belong to it. Objects are also known as instance.
- **Member Variable:** These are the variables defined inside a class. This data will be invisible to the outside of the class and can be accessed via member functions. These variables are called attribute of the object once an object is created.
- **Member function:** These are the function defined inside a class and are used to access object data.
- **INHERITANCE:** When a class is defined by inheriting existing function of a parent class then it is called inheritance. Here child class will inherit all or few member functions and variables of a parent class.
- **Parent class:** A class that is inherited from by another class. This is also called a base class or super class.
- **Child Class:** A class that inherits from another class. This is also called a subclass or derived class.
- **POLYMORPHISM:** This is an object oriented concept where same function can be used for different purposes. For example function name will remain same but it make take different number of arguments and can do different task.
- **Overloading:** a type of polymorphism in which some or all of operators have different implementations depending on the types of their arguments. Similarly functions can also be overloaded with different implementation.
- **DATA ABSTRACTION:** Any representation of data in which the implementation details are hidden (abstracted).
- **ENCAPSULATION:** refers to a concept where we encapsulate all the data and member functions together to form an object.

- **Constructor:** refers to a special type of function which will be called automatically whenever there is an object formation from a class.
- **Destructors:** refers to a special type of function which will be called automatically whenever an object is deleted or goes out of scope.

Ushtrimi 1: Të definohet një klasë në PHP, e cila së pari i definon vetitë, pastaj metodat dhe pas krijimit të objekteve i përdorë këto metoda.

```
<?php
```

```
class Ariu {  
    //definojme vetite  
  
    public $emri;  
    public $pesha;  
    public $mosha;  
    public $gjinia;  
    public $ngjyra;  
  
    //definojme metodat  
  
    public function ngrenia()  
    {  
        echo $this->emri. " eshte duke ngrene ...\n";  
    }  
  
    public function vrapimi()  
    {  
        echo $this->emri. " eshte duke vrapuar ...\n";  
    }  
  
    public function mbyt()  
    {  
        echo $this->emri. " eshte duke mbytur prene ...\n";  
    }  
  
    public function flen()  
    {  
        echo $this->emri. " eshte duke fjetur ...\n";  
    }  
}
```

```
$babai = new Ariu;
```

```
$babai->emri = "Ariu Baba";  
$babai->mosha = 8;  
$babai->gjinia = "Mashkull";  
$babai->ngjyra = "i zi";  
$babai->pesha = 300;
```

```
$nena = new Ariu;
```

```
$nena->emri = "Ariu Nene";  
$nena->mosha = 7;  
$nena->gjinia = "Femer";  
$nena->ngjyra = "e bardhe";  
$nena->pesha = 300;
```

```
$femiu = new Ariu;
```

```
$femiu->emri = "Ariu Femi";  
$femiu->mosha = 2;  
$femiu->gjinia = "Mashkull";  
$femiu->ngjyra = "i zi";  
$femiu->pesha = 180;
```

```
print ("<h2>Shembuj me PHP te OO</h2>");
```

```
$babai->mbyt();  
echo "</br>";
```

```
$nena->ngrenia();  
echo "</br>";
```

```
$femiu->ngrenia();  
echo "</br>";
```

```
$nena->flen();  
echo "</br>";
```

```
$babai->flen();  
echo "</br>";
```

```
$femiu->flen();  
echo "</br>";
```

```
?>
```

Ushtrimi 2: Të krijohet një klasë e cila përshin variabla të krijuara në të, dhe si të tilla këto variabla përfshihen në funksione, të cilat më pas thirren nga objektet e krijuara.

```
<?php
class Librat{
    /* Variablat */
    var $cmimi;
    var $titulli;
    /* Funksionet */
    function setcmimi($par){
        $this->cmimi = $par;
    }
    function getcmimi(){
        echo $this->cmimi . "<br/>";
    }
    function settitulli($par){
        $this->titulli = $par;
    }
    function gettitulli(){
        echo $this->titulli . " <br/>";
    }
}

//krijimi i objekteve
$fizika = new Librat;
$matematike = new Librat;
$kimia = new Librat;

//thirrja e funksioneve
$fizika->settitulli( "Fizika per shkollen e mesme" );
$kimia->settitulli( "Kimia e avancuar" );
$matematike->settitulli( "Algjebra" );

$fizika->setcmimi( 10 );
$kimia->setcmimi( 15 );
$matematike->setcmimi( 7 );

//tani thirren funk me vlerat qe i jane caktuar me heret

$fizika->gettitulli();
$kimia->gettitulli();
$matematike->gettitulli();
$fizika->getcmimi();
$kimia->getcmimi();
$matematike->getcmimi();?>
```

Ushtrimi 3: Të definohet një konstruktor, dhe pastaj të përdoren variablat brenda tij për të kryer funksionin e caktuar.

```
<?php
class Ariu {
    //definojme vetite

    public $emri;
    public $pesha;
    public $ngjyra;

    public function __construct()
    {
        $this->mosha = 0;
        $this->pesha = 100;
        $this->ngjyra = "i zi";
    }
}
?>

<?php
$femiu = new Ariu;

$femiu->emri = "Ariu Femi";

echo $femiu->emri." është ".$femiu->ngjyra." dhe peshon ". $femiu->pesha. " kg në lindje.";

?>
```

E kundërta e konstruktorit është destruktori. Këta ekzekutohen para se klasa të shkatërrohet, gjë e cila ndodh automatikisht kur të gjitha referencat në klase bëhen unset.

Ngjashëm emërohen edhe destrukturët `__destruct()`. Destruktorët nuk mund të kenë parametra.

Ushtrimi 4: Përdorimi i attributeve të klasës si dhe përdorimi i funksioneve get() dhe set().

```
class classname
{
    var $attribute;
    function operation($param)
    {
        $this->attribute = $param
        echo $this->attribute;
    }
}
```

Ky shembull nuk e kufizon qasjen në attribute, andaj ju mund ti qaseni atyre nga jashtë klasës :

```
class classname
{
    var $attribute;
}
$a = new classname();
$a->attribute = 'value';
echo $a->attribute;
```

```
class klasa
{
    var $attribute;
    function __get($name)
    {
        return $this->$name;
    }
    function __set ($name, $value)
    {
        $this->$name = $value;
    }
}

?>
```

```
<?php
class person
{
    var $emri, $mbiemri;
    function __get($name)
    {
        return $this->name;
    }
    function __set($name,$value)
    {
```

```
$this->name = $value;
}
}
$personi = new person();
$personi->emri = "Artan";
print $personi->emri;

?>
```

Ushtrimi 5: Kontrollimi i qasjes me modifikatorët **private** dhe **public**

- **public** scope to make that variable/function available from anywhere, other classes and instances of the object.
- **private** scope when you want your variable/function to be visible in its own class only.
- **protected** scope when you want to make your variable/function visible in all classes that extend current class including the parent class.

<?php

```
class Studenti
{
    public $NrIndeksit = "12345";
    var $emri= "Arta";
    private $mbiemri = "Gashi";
    protected $adresa = "Prishtine";

    public function printo()
    {
        echo $this->mbiemri;
    }
}

$objStudenti = new Studenti();
print $objStudenti->NrIndeksit;
print '<br />';
print $objStudenti->emri;
print '<br />';
//print $objStudenti->printo();
print $objStudenti->mbiemri;
print '<br />';
print $objStudenti->adresa;
print '<br />';
print '<br />';
```

?>

```
<?php
    class Studenti2
    {
        public function funk1()
        {
            return "Eshte thirrur funksioni funk1";
            echo "</br>";
        }
        function funk2()
        {
            return "Eshte thirrur funksioni funk2";
            echo "</br>";
        }

        private function funk3()
        {
            return "Eshte thirrur funksioni funk3";
            echo "</br>";
        }
    }
    $ObjStudenti = new Studenti2();
    print $ObjStudenti->funk1();
    print '</br>';
    print $ObjStudenti->funk2();
    print '</br>';
    print $ObjStudenti->funk3();
    print '</br>';

?>
```

Ushtrimi 6: Të ipet një shembull ku shihet implementimi i trashëgimisë në PHP.

```
<?php
    class A
    {
        var $atributi_A;
        function funk_A()
        {
            print "<br/>". $this->atributi_A;
        }
    }
    class B extends A
    {
```

```
var $atributi_B;  
function funk_B()  
{  
    print "<br/>". $this->atributi_B;  
}  
}  
$Obj_B = new B();  
$Obj_B->atributi_A =2;  
$Obj_B->funk_A();  
$Obj_B->atributi_B=3;  
$Obj_B->funk_B();  
?>
```

b) Të bëhet Kontrollimi I dukshmërisë përgjatë trashëgimit me **private** dhe **protected**

```
<?php  
  
class A  
{  
    private function operation1()  
    {  
        echo "operation1 called";  
    }  
    protected function operation2()  
    {  
        echo "operation2 called";  
    }  
    public function operation3()  
    {  
        echo "operation3 called";  
    }  
}  
  
class B extends A  
{  
    function __construct()  
    {  
        $this->operation1();  
        print "<br/>";  
        $this->operation2();  
        print "<br/>";  
        $this->operation3();  
        print "<br/>";  
    }  
}
```

```
$b = new B;  
?>
```

Trashëgimia e shumëfishtë

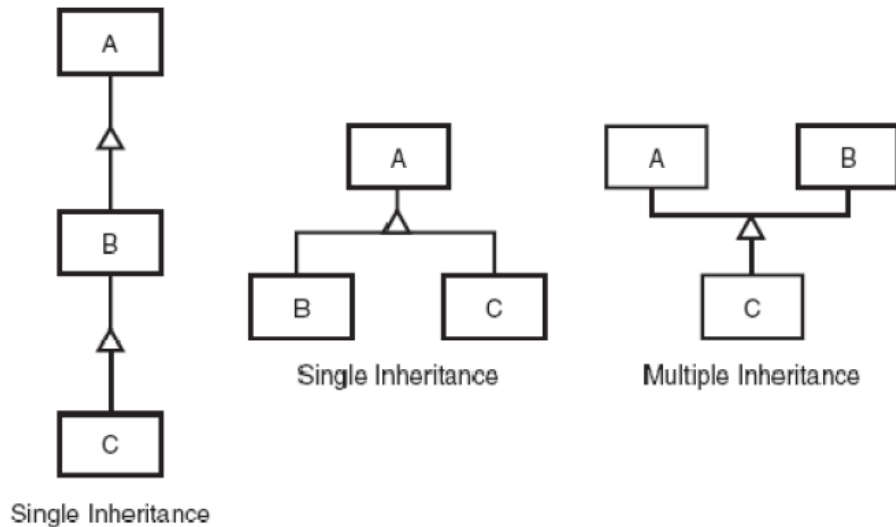


Figure 6.1 PHP does not support multiple inheritance.

Parandalimi i trashëgimit dhe mbishkruajtjes me final (diskutim)

```
class A  
{  
    var $attribute = 'default value';  
    final function operation()  
    {  
        echo 'Something<br />';  
        echo "The value of \$attribute is $this->attribute<br />";  
    }  
}
```

Mundeni gjithashtu të përdorni fjalën kyçe final që të parandaloni një klasë nga krijimet e të gjitha nënklasave. Të parandaloni klasën A nga ekzistenca e nënklasave, ju mund të shtoni këtë si në vazhdim:

```
final class A  
{...}
```

Nëse tentoni të trashëgoni nga A, do të merrni këtë error :
Fatal error: Class B may not inherit from final class (A)

Ushtrimi 7: Te shkruhet nje klase qe e paraqet mesazhin "Pershendetje, une jam Petriti!, ku "Petriti" eshte nje vlere e argumentit te nje metode brenda klasës.

```
<?php
class user_message {
    public $message = 'Pershendetje, une jam ';

    public function introduce($name)
    {
        return $this->message.$name."<br />";
    }
}
$message = New user_message();
echo $message->introduce('Petriti!');
?>
```

b) Mbledhje, zbritje, pjesëtim, shumëzim.

```
<?php
class Kalkulatori {
    private $_fval, $_sval;
    public function __construct( $fval, $sval ) {
        $this->_fval = $fval;
        $this->_sval = $sval;
    }
    public function mledh() {
        return $this->_fval + $this->_sval;
    }
    public function zbrit() {
        return $this->_fval - $this->_sval;
    }
    public function shumezo() {
        return $this->_fval * $this->_sval;
    }
    public function pjesto() {
        return $this->_fval / $this->_sval;
    }
}
$mycalc = new Kalkulatori(12, 6);
echo $mycalc->mledh(); // Tregon 18
echo '<br />'.$mycalc->shumezo(); // Tregon 72
```

```
echo '<br />'. $mycalc->zbrit(); // Tregon 6  
echo '<br />'. $mycalc->pjesto(); // Tregon 2  
?>
```

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Programimi në Internet



Java 5: PHP Regular Expressions– PHP Shprehjet e rregullta

- Rregulla te përgjithshme [regular expressions php.txt](#)

Regular Expression	Will match...
foo	The string "foo"
^foo	"foo" at the start of a string
foo\$	"foo" at the end of a string
^foo\$	"foo" when it is alone on a string
[abc]	a, b, or c
[a-z]	Any lowercase letter
[^A-Z]	Any character that is not a uppercase letter
(gif jpg)	Matches either "gif" or "jpeg"
[a-z]+	One or more lowercase letters
[0-9\.\-]	Any number, dot, or minus sign
^[a-zA-Z0-9_]{1,}\$	Any word of at least one letter, number or _
((wx))([yz])	wy, wz, xy, or xz
[^A-Za-z0-9]	Any symbol (not a number or a letter)
(([A-Z]{3}) ([0-9]{4}))	Matches three letters or four numbers

Ushtrimi 1: Të ipet shembulli ku kontrollohet dhe validohet përmbajtja e shkrimit të një email adrese, përmes shprehjeve të rregullta në PHP.

```
<?php
$email1 = "test@hotmail.com";
print "Email-i testues: '$email1'";
print "<br>";
if (preg_match('/^[a-zA-Z0-9_\-\.]+@[a-zA-Z0-9\-\]+\.[a-zA-Z0-9\-\\.]+$/ ',
$email1))
{
    echo "Email adresa1 është valide.";
    echo "<br>";
    echo "<br>";
}
```



```
}
else
{
    echo "Email adresa1 nuk është valide.";
    echo "<br>";
    echo "<br>";
}
?>

<?php
$email2 = "emri.mbiemri@aaa.bbb.com";
print "Email-i testues: '$email2'";
echo "<br>";

$regexp = "/^[^0-9][A-z0-9_]+([.][A-z0-9_]+)*[@][A-z0-9_]+([.][A-z0-9_]+)*[.][A-
z]{2,4}$/";

if (preg_match($regexp, $email2)) {
    echo "Email adresa2 është valide.";
    echo "<br>";
    echo "<br>";
} else {
    echo "Email adresa2 <u>nuk</u> është valide.";
    echo "<br>";
    echo "<br>";
}
?>
```

Ushtrimi 2: Të shkruhet një RegEx që kontrollon për format të caktuar të datës.

```
<?php

$date = "2014-04-13";
print $date;
print "<br>";
if (preg_match ('/([0-9]{4})-([0-9]{1,2})-([0-9]{1,2})/', $date, $regs)) {
    echo "$regs[3].$regs[2].$regs[1]";
} else {
    echo "Format invalid i dates: $date";
}
?>
```

Ushtrimi 3: Të shkruhen PHP RegEx që manipulojnë dhe kontrollojnë shkrim të caktuar të numrave si p.sh. IP adresë, ID numër.

```
<?php
$var = "127.0.0.1";
if (preg_match('/^([0-9]{1,3}\.){3}[0-9]{1,3}$/',$var))
{
    print "IP adresa '$var' është valide.";
}
else
{
    print "IP adresa '$var' nuk është valide.";
}
?>
```

```
-----
<?php

$var = "213-25-5674";

if (preg_match('/^([0-9]{3}\-[0-9]{2}\-[0-9]{4})$|(^([0-9]{9})$)/',$var))
{
    print "SSN '$var' është valid.";
}
else
{
    print "SSN '$var' nuk është valid.";
}
?>
```

Ushtrimi 4: Të shkruhet një PHP RegEx, e cila kontrollon varësisht nga kushtet e RegEx, nëse plotësohen apo jo.

```
<?php
$var = "#FFFFFFE";

if(preg_match('/^#?([a-f]|[A-F]|[0-9]){3}(((a-f)|[A-F]|[0-9]){3})?$/',$var))
{
    print "'$var' është valide.";
}
else
{
    print "'$var' nuk është valide.";
}
?>
----
```

```
<?php
$var = "filan123";

if (preg_match('/^[a-zA-Z0-9_]{3,16}$/', $var))
{
    print "User-i '$var' është valid.";
}
else
{
    print "User-i '$var' nuk është valid.";
}
?>
```

Ushtrimi 5: Të shkruhen RegEx të cilat kontrollojnë për rregulla të caktuar për manipulim me numra, p.sh. nr. te tel etj.

```
<?php
$var = "1-608-234-1234";

if( !preg_match("/^([1-])?[0-9]{3}-[0-9]{3}-[0-9]{4}$/i", $var) ) {
    echo 'Ju lutem shkruani numrin e telefonit ne forme valide.';
}
else
{ echo 'Numri '. $var .' eshte shkruar ne forme valide!';}
?>
```

```
-----
<?php
$var = "-15.405";
if (preg_match("/^[+-]?[0-9]*\.[0-9]+$/", $var))
{
    print "Numri i dhënë '$var' është valid.";
}
else
{
    print "Numri i dhënë '$var' nuk është valid.";
}
?>
```

Ushtrimi 6: Të shkruhen PHP RegEx për të kontrolluar përmbajtje të caktuara të stringjeve.

```
<?php
$string = 'XYZ';
if (eregi('z', $string))
{
    echo "Stringu '$string' permban ndonje 'z' or 'Z'!";
}
else
{
    echo "Stringu '$string' nuk permban ndonje 'z' or 'Z'!";
}
?>
-----
<?php
$string = 'test';
if (eregi("[[:alpha:]]", $string))
{
    echo "Stringu '$string' permban ndonje shkronje!";
}
else
{
    echo "Stringu '$string' nuk permban ndonje shkronje!";
}
?>
-----
<?php
$string = '123';
if (eregi("[[:alnum:]]", $string))
{
    echo "Stringu '$string' permban ndonje numer ose shkronje!";
}
else
{
    echo "Stringu '$string' nuk permban ndonje numer ose shkronje!";
}
?>
```

Ushtrimi 7: Shembuljt në vazhdim tregojnë përdorimin e funksionit preg_replace()

```
<?php
$string = "Ky eshte nje test!";
print $string;
print "<br />";
echo str_replace(" eshte", " ishte", $string);
print "<br />";
echo preg_replace("/ eshte/", " ishte", $string);
echo "<br>";
    echo "<br>";
?>
```

```
<?php
/* Nuk do të funksionoj, psh ne POSIX, por tani eshte ne rregull me Perl RegEx. */
$num = 4;
$string = "Ky string ka kater fjale.";
$string = preg_replace('/kater/', $num, $string);
echo $string;
print "<br />";
/* Kjo do të funksionoj. */
$num = '4';
$string = "Ky string ka kater fjale.";
$string = preg_replace('/kater/', $num, $string);
echo $string;
?>
```

Pyetje: Cilat nga shprehjet e rregullta në vijim përputh stringjet “zoot” dhe “zot”?

- A. z(oo)+t
- B. zo*t\$
- C. \$zo*t
- D. ^(zo)+t

Ushtrimi 8: Të tregohet një shembull se si punohet me RegEx në PHP

```
<?php
$str = "Sot eshte dite me diell.";

print( "Test stringu eshte: '$str'<br /><br />" );
// thirrja e funksionit preg_match te kerkimi pEr modelin 'Sot' nE variabelEn str
if (preg_match( "/Sot/", $str ) )
print( "Stringu 'Sot' eshte gjetur.<br />" );

// kerkimi pEr shprehjen 'Sot' nE fillim tE stringut
if ( preg_match( "/^Sot/", $str ) )
print( "Stringu 'Sot' eshte gjetur nE fillim te rreshtit.<br />" );

// kerkimi pEr modelin 'Sot' tek fundi i stringut
if (preg_match( "/Sot$/", $str ) )
print( "Stringu 'Sot' eshte ne fund te rreshtit.<br />" );

// marrim emrin e hostit prej URL
preg_match('@^(?:http://)?(?:[^/]+)@i',
"http://www.php.net/index.html", $matches);
$host = $matches[1];

// marrim 2 segmentet e fundit te host name
preg_match('/[^\.]+\.[^\.]+$/', $host, $matches);
echo "Emri i domain-it eshte: {$matches[0]}\n";
?>
```

Ushtrimi 9: Të shkruhet RegEx në PHP e cila kontrollon nëse një password është “strong” apo jo.

```
<?php
$password = "Fyfjk34sdfjfsjq7";

if (preg_match("/^.*(?:={8,})(?=[0-9])(?=[a-z])(?=[A-Z]).*$/", $password)) {
    echo "Passwordi eshte 'strong'.";
} else {
    echo "Passwordi eshte i 'dobet'.";
}
?>
Shpjegimi:
```

The "==" combination means "the next text must be like this"

The first grouping is `(?=.*{8,})`. This checks if there are at least 8 characters in the string.

The next grouping `(?=.*[0-9])` means "any alphanumeric character can happen zero or more times, then any digit can happen"

The next groupings `(?=.*[a-z])` and `(?=.*[A-Z])` are looking for the lower case and upper case letter accordingly anywhere in the string.

9b) Përdorimi i funksionit `preg_split()`

```
<?php
$keywords = preg_split("/[\s,]+/", "PHP, shprehjet e rregullta!");
print_r( $keywords );
?>
```

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike



Lënda: Programimi në Internet

Java 6: PHP Sessions and Cookies

The idea of session control is to be able to track a user during a single session on a Web site. When you are working with an application, you open it, do some changes and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are and what you do because the HTTP address doesn't maintain state.

A PHP session solves this problem by allowing you to store user information on the server for later use (i.e. username, shopping items, etc.). However, session information is temporary and will be deleted after the user has left the website. If you need a permanent storage you may want to store the data in a database.

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

```
bool setcookie ( string $name [, string $value [, int $expire =  
0 [, string $path [, string$domain [, bool $secure =  
false [, bool $httponly = false ]]]]] )
```

All the arguments except the name argument are optional.

name

The name of the cookie.

value

The value of the cookie. This value is stored on the clients computer; do not store sensitive information. Assuming the **name** is 'cookienam', this value is retrieved through `$ _COOKIE['cookienam']`

expire

The time the cookie expires. This is a Unix timestamp so is in number of seconds since the epoch. In other words, you'll most likely set this with the [time\(\)](#) function plus the number of seconds before you want it to expire.

path

The path on the server in which the cookie will be available on.

domain

The domain that the cookie is available to.

secure

Indicates that the cookie should only be transmitted over a secure HTTPS connection from the client. When set to `TRUE`, the cookie will only be set if a secure connection exists. On the server-side, it's on the programmer to send this kind of cookie only on secure connection (e.g. with respect to `$ _SERVER["HTTPS"]`).

httponly

When `TRUE` the cookie will be made accessible only through the HTTP protocol. This means that the cookie won't be accessible by scripting languages, such as JavaScript (ex. Due to XSS attacks)

Ushtrimi 1: Të përdoret një shembull për të treguar shfrytëzimin e cookie-s.

```
<?php
$value = 'vlere e cfaredoshme';

setcookie("CookieTest", $value);
setcookie("CookieTest", $value, time()+3600); /* skadon pas 1 ore */
setcookie("CookieTest", $value, time()+3600, "/~rasmus/", "example.com", 1);
?>

<?php
// Printimi i nje cookie individual
echo $_COOKIE["CookieTest"];
echo "<br>";
echo "<br>";

// Another way to debug/test is to view all cookies
print_r($_COOKIE);

?>
```

Ushtrimi 2: Të tregohet me një shembull se si mund të fshihet në cookie...

```
<?php
// set the expiration date to one hour ago
```

```
setcookie ("TestCookie", "", time() - 3600);  
setcookie ("TestCookie", "", time() - 3600, "/~rasmus/", "example.com", 1);  
  
echo $_COOKIE["TestCookie"];  
echo "<br>";  
?>
```

Ushtrimi 3: Të merret një shembull ku tregohet vendosja dhe ruajtja e cookie-ve në varg (eng. Array).

```
<?php  
// vendosja e cookie-ve  
setcookie("cookie[tre]", "cookietre");  
setcookie("cookie[dy]", "cookiedy");  
setcookie("cookie[nje]", "cookienje");  
  
// pasi faqja te rifreskohet, behet printimi i tyre  
  
if (isset($_COOKIE['cookie'])) {  
    foreach ($_COOKIE['cookie'] as $name => $value) {  
        $name = htmlspecialchars($name);  
        $value = htmlspecialchars($value);  
        echo "$name : $value <br />\n";  
    }  
}  
//setcookie ("cookie[two]", "", time() - 3600);  
?>
```

Ushtrimi 4: Shembull i përdorimit të cookie-ve në PHP, ku tregohet si si ndryshohet përmbajtja varësisht nga cookie-t.

```
<?php  
#kontrollojme nese forma eshte bere submitted  
if (isset($_POST['submitted'])) {  
    #nese po, vendosi vlerat nga POST array ne variabla  
    $newbgColor=$_POST['bgColor'];  
    $newtxtColor=$_POST['txtColor'];  
    #set cookies  
    setcookie("bgColor",$newbgColor,time()+3600);  
    setcookie("txtColor",$newtxtColor,time()+3600);  
  
}
```

```
#ne rast se shfrytezuesi vjen per here te pare dhe cookies nuk jane vendosur,
atehere
if ((!isset($_COOKIE['bgColor']) ) && (!isset($_COOKIE['txtColor']))) {
    $bgColor = "Black";
    $txtColor="White";
}
#nese cookie-t jane vendosur, atehere perdori ato cookie
else {
    $bgColor = $_COOKIE['bgColor'];
    $txtColor = $_COOKIE['txtColor'];
}
?>
<!-- HTML Faqja-->
<html>
<body bgcolor="<?php echo $bgColor ?>" text="<?php echo $txtColor ?>">
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
<p>Ngjyra e prapavise:</p>
<select name=bgColor>
<option value="Red">E kuqe</option>
<option value="Green" selected>E gjelber</option>
<option value="Blue">E kalter</option>
<option value="Yellow">E verdhe</option>
<option value="Black">E zeze</option>
<option value="Brown">Kafe</option>
<option value="White">E bardhe</option>
</select>
<p>Ngjyra e tekstit:</p>
<select name=txtColor>
<option value="Red">E kuqe</option>
<option value="Green" selected>E gjelber</option>
<option value="Blue">E kalter</option>
<option value="Yellow">E verdhe</option>
<option value="Black">E zeze</option>
<option value="Brown">Kafe</option>
<option value="White">E bardhe</option>
</select>
<input type="hidden" name="submitted" value="true"></br>
<input type="submit" value="Kujto vlerat">
</form>
</body>
</html>
```

session_start() creates a session or resumes the current one based on a session identifier passed via a GET or POST request, or passed via a cookie.

Ushtrimi 5: Të merret një shembull me PHP sesione i cili tregon përfshirjen dhe përdorimin e sesioneve në PHP, si p.sh. krijimi i tyre, ruajtja e vlerave dhe pastaj edhe leximi i vlerave të këtyre funksioneve.

```
<?php
// ushtrimi5.php

session_start();

echo 'Miresevini ne faqen #1';

$_SESSION['favcolor'] = 'e gjelber';
$_SESSION['kafsha'] = 'mace';
$_SESSION['koha'] = time();

// Punon vetem nese session cookie eshte pranuar
echo '<br /><a href="ushtrimi6">page 2</a>';

// Ose ndoshta te ja kalojme session id, nese i nevojitet
echo '<br /><a href="ushtrimi6.php?" . SID . "">page 2</a>';
?>
-----
Ushtrimi6.php:

<?php
// ushtrimi6.php

session_start();

echo 'Miresevini ne faqen #2<br />';

echo $_SESSION['favcolor']; // e gjelber
echo "\n";
echo $_SESSION['kafsha']; // mace
echo "\n";
echo date('Y m d H:i:s', $_SESSION['koha']);
echo "\n";

// Mund te perdorni SID ketu, sic u vepruar edhe ne ushtrimi5.php
echo '<br /><a href="ushtrimi5.php">page 1</a>';
?>
```

Ushtrimi 6: Përdorimi i PHP sesioneve për të numëruar vizitat në një ueb faqe, si dhe përdorimi i funksionit `isset()`. Në fund të tregohet se si mund të fshihet komplet të gjitha sesionet apo vetëm ndonjë sesion.

```
<?php
    session_start();
    if( isset( $_SESSION['numeruesi'] ) )
    {
        $_SESSION['numeruesi'] += 1;
    }
    else
    {
        $_SESSION['numeruesi'] = 1;
    }
    $msg = "Ju keni vizituar kete faqe ". $_SESSION['numeruesi'];
    $msg .= " here ne kete sesion.";
?>

<html>
<head>
<title>Vendosja e nje PHP sesioni</title>
</head>
<body>
<?php echo ( $msg ); ?>
----
<?php /*?><?php
    unset($_SESSION['numeruesi']);
?><?php */?>

<?php /*?><?php
    session_destroy();
?><?php */?>
```

Ushtrimi 7: Të tregohet me një shembull nëse mund të ruhet një funksion brenda një PHP sesioni?

```
<?php
session_start();
// vendosim ne varibel te sesionit te permbaje nje funksion
$_SESSION['variabla'] = '
<?php
function test(){
echo "...Jam ruajtur ne funksionin e quajtur test!";}
?>';
```

```
?>
<a href="Ushtrimi10.php">Faqja 2</a>
-----
Ushtrimi10.php
<?php
    session_start();
    // vleresojme kodin brenda variables se sesionit
    eval('?'>.'$_SESSION['variabla']);
    // ekzekutojme funksionin
    test();
?>
```

Ushtrimi 8: Të tregohet me një shembull manipulimi me PHP sesione, si dhe të paraqitet rast konkret ku shohim ID-në e sesionit dhe të vërejmë si ndryshon kjo ID.

```
<?php
    session_start();
    if ( ! isset($_SESSION['value']) ) {
        echo("<p>Sesioni eshte i zbrazet</p>\n");
        $_SESSION['value'] = 0;
    } else if ( $_SESSION['value'] < 3 ) {
        $_SESSION['value'] = $_SESSION['value'] + 1;
        echo("<p>U shtua nje...</p>\n");
    } else {
        session_destroy();
        session_start();
        echo("<p>Sesioni u ristartua</p>\n");
    }
?>
<p><a href="Ushtrimi11.php">Kliko!</a></p>
<p>ID e Sesionit tone eshte: <?php echo(session_id()); ?></p>
<pre>
<?php print_r($_SESSION); ?>
</pre>
```

Referencë shtesë: <http://www.webstepbook.com/supplements-2ed/slides/chapter14-cookies-sessions.shtml#>

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Programimi në Internet



Java 7: PHP dhe MySQL

- Anatomia e SQL pyetësorëve
- Lidhja me Bazën e të dhënave
- Krijimi i Bazës së të dhënave
- Insertimi i informatave në Bazë të dhënave
- Leximi i të dhënave nga MySQL dhe shfaqja e tyre
- Azhurimi/Fshirja e të dhë dhënave në MySQL

Anatomia e SQL pyetësoreve bazë:

```
SELECT [columns]  
FROM [table1],[table2]...  
WHERE [conditionals on columns]  
GROUP BY [columns to aggregate]  
HAVING [conditionals on aggregated values]  
ORDER BY [columns on which to sort]
```

```
UPDATE [table]  
SET [columns]  
WHERE [conditinal on columns]
```

```
DELETE FROM [table]  
WHERE [conditionals on columns]
```

```
INSERT INTO [table] (columns)  
VALUES (values)
```

Ushtrimi 1: Të ipet shembulli nëpërmes të cilit hapet dhe mbyllet një konexion në bazë të dhënave.

```
<?php
    $dbhost = 'localhost:3306';
    $dbuser = 'root';
    $dbpass = '12345678';
    $conn = mysql_connect($dbhost, $dbuser, $dbpass);

    if(! $conn )
    {
        die('Could not connect: ' . mysql_error());
    }
    echo 'Connected successfully';
    mysql_close($conn);
?>
```

Ushtrimi 2: Të shkruhet kodi i cili krijon dhe fshinë një bazë të dhënave dhe një tabelë në MySQL serverin lokal.

```
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = '12345678';
    $conn = mysql_connect($dbhost, $dbuser, $dbpass);
    if(! $conn )
    {
        die('Could not connect: ' . mysql_error());
    }
    echo 'Connected successfully';
    $sql = 'CREATE Database Ushtrime';
    $retval = mysql_query( $sql, $conn );
    if(! $retval )
    {
        die('Could not create database: ' . mysql_error());
    }
    echo "Database Ushtrime created successfully\n";
    mysql_close($conn);
?>
```

```
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = '12345678';
    $conn = mysql_connect($dbhost, $dbuser, $dbpass);
    if(! $conn )
    {
        die('Could not connect: ' . mysql_error());
    }
}
```



```
echo 'Connected successfully';
$sql = 'DROP Database Ushtrime';
$retval = mysql_query( $sql, $conn );
if( ! $retval )
{
    die('Could not drop database: ' . mysql_error());
}
echo "Database Ushtrime dropped successfully\n";
mysql_close($conn);
?>
```

Ushtrimi 3: Të krijohet table studentet me fushat ID, emri, mbiemri dhe data_regjistrimit ne instancën lokale te MySQL serverit.

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '12345678';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);
if( ! $conn )
{
    die('Could not connect: ' . mysql_error());
}
$sql = 'CREATE TABLE studentet (
        'id integer',
        'emri varchar(20)',
        'mbiemri varchar(20)',
        'data_regjistrimit date) ';

mysql_select_db('Ushtrime');
$retval = mysql_query( $sql, $conn );
if( ! $retval )
{
    die('Could not create table: ' . mysql_error());
}
echo "Table studentet was created successfully\n";
mysql_close($conn);
?>
```

Ushtrimi 4: Të insertohet një rresht në tabelën e sapokrijuar (studentët) nga shembulli paraprak.

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '12345678';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);
if( ! $conn )
{
    die('Could not connect: ' . mysql_error());
}
$sql = 'INSERT INTO studentet ';
```

```
'(id,emri, mbiemri, data_regjistrimit) '.
'VALUES ( 15243, "Filan", "Fisteku", NOW() )';

mysql_select_db('Ushtrime');
$retval = mysql_query( $sql, $conn );
if( ! $retval )
{
    die('Could not enter data: ' . mysql_error());
}
echo "Entered data successfully\n";
mysql_close($conn);

?>
```

Ushtrimi 5: Të jipet një shembull i insertimit të shënimeve të studentit nëpërmes një forme.

```
<html>
<head>
<title>Shto një studente të re në Database</title>
</head>
<body>
<?php
    if(isset($_POST['add']))
    {
        $dbhost = 'localhost';
        $dbuser = 'root';
        $dbpass = '12345678';
        $conn = mysql_connect($dbhost, $dbuser, $dbpass);
        if( ! $conn )
        {
            die('Could not connect: ' . mysql_error());
        }

        if( ! get_magic_quotes_gpc() )
        {
            $ID = addslashes ( $_POST['ID'] );
            $emri = addslashes ( $_POST['Emri'] );
        }
        else
        {
            $ID = $_POST['ID'];
            $emri = $_POST['Emri'];
        }
        $mbiemri = $_POST['Mbiemri'];

        $sql = "INSERT INTO studentet ".
            "(ID, emri, mbiemri, data_regjistrimit) ".
            "VALUES('$ID','$emri','$mbiemri', NOW())";
        mysql_select_db('Ushtrime');
        $retval = mysql_query( $sql, $conn );
        if( ! $retval )
        {
            die('Could not enter data: ' . mysql_error());
        }
    }
}
```

```
        }
        echo "Te dhenat u regjistruan me sukses\n";
        mysql_close($conn);
    }
    else
    {
?>
<form method="post" action="<?php $_PHP_SELF ?>">
<table width="400" border="0" cellspacing="1" cellpadding="2">
<tr>
    <td width="100">ID</td>
    <td><input name="ID" type="text" id="ID"></td>
</tr>
<tr>
    <td width="100">Emri</td>
    <td><input name="Emri" type="text" id="emri"></td>
</tr>
<tr>
    <td width="100">Mbiemri</td>
    <td><input name="Mbiemri" type="text" id="mbiemri"></td>
</tr>
<tr>
    <td width="100"></td>
    <td></td>
</tr>
<tr>
    <td width="100"></td>
    <td>
        <input name="add" type="submit" id="add" value="Shto Studentin">
    </td>
</tr>
</table>
</form>
<?php
}
?>
</body>
</html>
```

Ushtrimi 5: Të azhurohet emri i një studenti duke u bazuar në Id-në e dhënë nëpërmes të formës.

```
<html>
<head>
<title>Azhuro emrin e studentit!</title>
</head>
<body>

<?php
if(isset($_POST['update']))
{
    $dbhost = 'localhost';
```

```
$dbuser = 'root';
$dbpass = '12345678';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $conn )
{
    die('Could not connect: ' . mysql_error());
}

$id = $_POST['ID'];
$emri = $_POST['Emri'];

$sql = "UPDATE studentet " .
        "SET emri = '$emri', " .
        "data_regjistrimit=NOW()" .
        "WHERE ID = $id" ;

mysql_select_db('Ushtrime');
$retval = mysql_query( $sql, $conn );
if(! $retval )
{
    die('E pamundur te azhurohen te dhenat: ' . mysql_error());
}
echo "Te dhenat e studentit u azhuruan me sukses \n";
mysql_close($conn);
}
else
{
    ?>
<form method="post" action="<?php $_PHP_SELF ?>">
<table width="400" border="0" cellspacing="1" cellpadding="2">
<tr>
<td width="150">ID</td>
<td><input name="ID" type="text" id="ID"></td>
</tr>
<tr>
<td width="150">Emri i ri i studentit</td>
<td><input name="Emri" type="text" id="Emri"></td>
</tr>
<tr>
<td width="100"></td>
<td></td>
</tr>
<tr>
<td width="100"></td>
<td>
<input name="update" type="submit" id="update" value="Update">
</td>
</tr>
</table>
</form>
<?php
}
?>
```

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

Ushtrimi 6: Të fshihen të gjitha rekordet e insertuara në ditën e sotme në tabelën studenti.

```
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = '12345678';
    $conn = mysql_connect($dbhost, $dbuser, $dbpass);
    if(! $conn )
    {
        die('Could not connect: ' . mysql_error());
    }
    $sql = "DELETE FROM studentet WHERE data_regjistrimit=DATE_FORMAT(NOW(), '%Y-%m-%d')";
    echo $sql;
    mysql_select_db('Ushtrime');
    $retval = mysql_query( $sql, $conn );
    if(! $retval )
    {
        die('E pamundur të fshihen të dhenat: ' . mysql_error());
    }
    echo "Fshirja e te dhenave u realizua me sukses\n";
    mysql_close($conn);
?>
```

Ushtrimi 7: Vendorsni detalet e koneksionit me database ne nje fajll te vetem dhe perfshine ate fajll ne te gjithë fajllat ku nevojitet konektimi ne ate baze te dhenave.

Fajlli db_conn_Ushtrime.php

```
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = '12345678';
    $conn = mysql_connect($dbhost, $dbuser, $dbpass);
    mysql_select_db('Ushtrime');
?>

<?php
    require_once "db_conn_Ushtrime.php";

    if(! $conn )
    {
        die('Could not connect: ' . mysql_error());
    }

    $sql = 'SELECT ID, Emri, Mbiemri, Data_regjistrimit FROM studentet';

    $retval = mysql_query( $sql, $conn );

    if(! $retval )
```

```
{
    die('Could not get data: ' . mysql_error());
}
while($row = mysql_fetch_array($retval, MYSQL_ASSOC))
{
    echo "Student ID :{$row['ID']} <br> ".
        "Emri : {$row['Emri']} <br> ".
        "Mbiemri : {$row['Mbiemri']} <br> ".
        "Data e Regjistrimit : {$row['Data_regjistrimit']} <br> ".
        "-----<br>";
}
echo "Fetched data successfully\n";
mysql_close($conn);?>
```

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Programimi në Internet

Java 8: Koncepte të Avancuara në PHP (Pointerët dhe SQL Injection)



Pointerët

Referencat në PHP na mundësojnë qasjen e përbajtjes së njëjtë të variables prej emrave të ndryshëm të variablave. Nuk janë analog me pointerat në C, p.sh, nuk mund të aplikojmë aritmetik në ta, pasi referencat në PHP nuk paraqesin adresa të memories.

Me anë të referencave mund të performojmë tri operacione bazike: *shoqërimi nëpërmes referencës*, *përcjellja nëpërmes referencës*, dhe *kthimi nëpërmes referencës*.

Shoqërimi nëpërmes Referencës:

Shembulli 1: Dy variabla të cilat tregojnë në të njëjtën përbajtje:

```
<?php
    $a =&$b;
    /*me kete rast dy variablat jane te barabarta,
    te dy variablat tregojne ne te njejten permbajtje
    dmth a nuk pointon ne b apo anasjelltas.*/
    $a=10;
    echo "Vlera e variables b:". $b;
?>
```

Detyra 1: Të tregohet impakti i referencës në vlerën e një variable.

```
<?php
    $ref = 0;
    $row =&$ref;
    echo "Vlerat e anetareve te vektorit<br>";
    //Vlerat e anetareve te vargut
    foreach (array(1, 2, 3) as $row) {
        echo $row."<br>";
    }
    echo "-----<br>";
    echo "Vlera e variables ref:". $ref; // elementi i fundit nga iterimi i vargut?>
```

Detyra 2: Vendosja e referencave në mes të variablave dhe anëtarëve të vargut.

```
<?php
    $a = 1;
    $b = array(2, 3);
    //lidhja e anetareve te vektorit me reference
    $arr = array(&$a, &$b[0], &$b[1]);
    $arr[0]++;
```

```
$arr[1]++;  
$arr[2]++;  
/* $a == 2, $b == array(3, 4); */  
echo "Vlera e variables a:". $a. "<br>";  
echo "Vlera e aneterit te pare te vektorit b:". $b[0] . "<br>";  
echo "Vlera e aneterit te dyte te vektorit b:". $b[1] . "<br>";  
?>
```

Detyra 3:

Jepni një shembull me shoqërime të kombinuara (me dhe pa reference).

```
<?php  
$a = 1;          /* Shoqerim me vlere */  
$b =&$a;         /* Shoqerim me reference */  
$c = $b;         /* Shoqerim me vlere – me kete rast do i percillet  
                  vetem vlere ne te cilen tregon b, jo edhe referencia */  
$c = 7;          /* Pasi $c nuk eshte reference; ku shoqerim nuk do te kete  
                  efekt ne $a apo $b */  
  
echo "Vlera e variables b=". $b. "<br>";  
echo "Vlera e variables c=". $c. "<br>";  
  
/* Shoqerimi i array variablave */  
$arr = array(1);  
$a =&$arr[0];    /* $a dhe $arr[0] ndajne te njejten reference */  
$arr2 = $arr;    /* shoqerim me vlere */  
$arr2[0]++;      /* $a == 2, $arr == array(2) */  
/* Edhe pse $arr nuk eshte reference, ne rastin e vargjeve permbajtja ndryshon! */  
echo "Vlera e anetarit te pare te vektorit arr[0]=". $arr[0]. "<br>";  
echo "Vlera e anetarit te pare te vektorit arr2[0]=". $arr2[0]. "<br>";?>
```

Përcjellja nëpërmes Referencës

Realizohet duke ndikuar në variable lokale në një funksion të referencojë në të njëjtën përmbajtje me një variabël jashtë funksionit.

Detyra 4: Jepni një shembull të përcjelljes së vlerës me referencë.

```
<?php  
function foo(&$var){  
    $var++;  
}  
$a=5;  
foo($a);  
echo "Vlera e variables a:". $a;  
?>
```

Detyra 5: Jepni një shembull të funksioneve të cilat pranojnë referencë:

```
<?php  
function foo(&$var){
```



```
        $var++;
        echo "Vlera e variables var:". $var;
    }
    function &bar(){
        $a = 5;
        return $a;
    }
    foo(bar());
?>
```

Shembulli2: Shembuj të përcjelljes invalide të referencës në funksione të cilat pranojnë referenca:

```
<?php
    function foo(&$var){
        $var++;
    }
    function bar()    // Vereni mungesen e &
    {
        $a = 5;
        return $a;
    }
    foo(bar());      // Paraqet fatal error qysh prejPHP 5.0.5
    foo($a = 5);     // Shprehje, jo variable
    foo(5);          // Paraqet fatal error
?>
```

Kthimi nëpërmes Referencës

Detyra 6:

```
<?php
    $var = 1;
    $num = NULL;

    function &blah(){
        $var =& $GLOBALS["var"]; # i qasemi variables globale $var;
        $var++;
        return $var;
    }

    $num = &blah();
    echo $num."<br>"; # 2

    blah();
    echo $num; # 3
?>
```

Largimi i PHP Referencës

Detyra 7: Me largimin e referencës, ne këpusim lidhjen ndërmjet emrit të variablës dhe përmbajtjes së saj. Kjo nuk nënkupton që përmbajtja do të largohet.

```
<?php
    $a = 1;
    $b =& $a;
    unset($a);      //kjo komande nuk do te beje unset b vetem a
    echo $b;
    echo $a;        //do të ktheje error, pasi emri i variables a nuk ekziston më
?>
```

Shembulli3. Përshkrim detal mekanizmit të referencave në PHP.

```
<?php
/* Imagine this is memory map
```

pointer	value	variable	
1	NULL	---	
2	NULL	---	
3	NULL	---	
4	NULL	---	
5	NULL	---	

```
Create some variables */
$a=10;
$b=20;
$c=array ('one'=>array (1, 2, 3));
/* Look at memory
```

pointer	value	variable's	
1	10	\$a	
2	20	\$b	
3	1	\$c['one'][0]	
4	2	\$c['one'][1]	
5	3	\$c['one'][2]	

```
do */
$a=&$c['one'][2];
/* Look at memory
```

pointer	value	variable's	
1	NULL	---	//value of \$a is destroyed and pointer is free
2	20	\$b	
3	1	\$c['one'][0]	
4	2	\$c['one'][1]	
5	3	\$c['one'][2] , \$a	// \$a is now here

```
do */
$b=&$a; // or $b=&$c['one'][2]; result is same as both "$c['one'][2]" and "$a" is at same pointer.
```

/* Look at memory

pointer	value	variable's	
1	NULL	---	
2	NULL	---	//value of \$b is destroyed and pointer is free
3	1	\$c['one'][0]	
4	2	\$c['one'][1]	
5	3	\$c['one'][2] , \$a , \$b	// \$b is now here

next do */

unset(\$c['one'][2]);

/* Look at memory

pointer	value	variable's	
1	NULL	---	
2	NULL	---	
3	1	\$c['one'][0]	
4	2	\$c['one'][1]	
5	3	\$a , \$b	// \$c['one'][2] is destroyed not in memory, not in array

next do */

\$c['one'][2]=500; //now it is in array

/* Look at memory

pointer	value	variable's	
1	500	\$c['one'][2]	//created it lands on any(next) free pointer in memory
2	NULL	---	
3	1	\$c['one'][0]	
4	2	\$c['one'][1]	
5	3	\$a , \$b	//this pointer is in use

lets try to return \$c['one'][2] at old pointer and remove reference \$a,\$b.*/

\$c['one'][2]=&\$a;

unset(\$a);

unset(\$b);

/* look at memory

pointer	value	variable's	
1	NULL	---	
2	NULL	---	
3	1	\$c['one'][0]	
4	2	\$c['one'][1]	
5	3	\$c['one'][2]	// \$c['one'][2] is returned, \$a,\$b is destroyed

----- ?>

SQL Injection

Rëndom, qëllimi i SQL injection sulmi, është ekspozimi apo dëmtimi i të dhënave të një ueb-i. Nëpërmes të SQL Injection një sulmues mund të përfitojë nga qasja e paautorizuar në bazë të shënimeve dhe të krijojë, ndryshojë apo fshijë të dhënat. Tani për tani shumica e SQL bazat e të dhënave si MySQL, Oracle, MSSQL Server, PostgreSQL janë të rrezikuara nga SQL Injeksion sulmet.

Më poshtë do japim disa prej SQL Injection sulmeve të mundshme në bazë të shënimeve.

Ushtrimi 1: Demonstrimi i SQL Injection nepermes nje log-in Forme.

Fajlli Login.html

```
<!doctype html>
<html>
<head><title>SQL Injection</title> </head>
<body>
    <form action="action.php" method="POST">
        <label>User ID:</label>
        <input type="text" id="uid" name="uid" placeholder="Student ID" required>
        <label>Password:</label>
        <input type="password" id="passid" name="passid" required>
        <input type="submit" value="Submit" />
    </form>
</body>
</html>
```

Fajlli action.php i cili do të thirret “on submit” të forms.

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '12345678';
$db_name='Ushtrime';
$conn = mysql_connect($dbhost, $dbuser, $dbpass)
        or die("cannot connect");

mysql_select_db($db_name)
        or die("cannot select DB");

$sid = $_POST['sid'];
$pid = $_POST['passid'];

$sql = "select * from studentet where ID = $sid and password = '$pid'";

$result = mysql_query( $sql, $conn );
```

```
if(mysql_num_rows($result)>0)
{
    while ($row=mysql_fetch_row($result)){
        echo "<h4>". "-- Informatat e studentit me ID: ".$row[0]. "</h4>";
        echo "<p>". "Emri: ".$row[1]. "</p>";
        echo "<p>". "Mbiemri: ".$row[2]. "</p>";
        echo "<p>". "Password: ".$row[3]. "</p>";
        echo "<p>". "Data e regjistrimit: ".$row[4]. "</p>";
        echo "-----";
    }
} else echo "Invalid user id or password";
?>
```

Struktura e tabelës: Studentet(Id, emri, mbiemri, password, data_regjistrimit)

Të dhënat:

sid	emri	mbiemri	password	data_regjistrimit
1	Filan	Fisteku	Ffisteku	2014-04-24
2	Barak	Obama	bobama	2014-04-24
3	Fransua	Hollond	fhollond	2014-04-24

Shembulli 1: Leximi i të gjithë shfrytëzuesve nga tabela Studentet nëpërmes SQL Injection në fushat e Login formës së krijuar më lartë.

Duke i shtuar një ' OR '1'='1 në fushën e passwordit atëhere query nga detyra jonë do marrë këtë formë:

"select * from studentet where ID = \$sid and password = " OR '1'='1"

Ky query do të kthejë TRUE për të gjitha rreshtat, për shkak se kushti ose 1=1 do të ndikojë që WHERE të kthejë gjithmonë TRUE.

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Programimi në Internet

Java 9: Koncepte të Avancuara në PHP (File & IO, Error Handling Email)



Funksioni include()

Shembulli 1. Krijo një file *menu.php* me përmbajtjen e mëposhtme.

menu.php

```
<a href="http://www.tutorialspoint.com/index.htm">Home</a> -  
<a href="http://www.tutorialspoint.com/ebxml">ebXML</a> -  
<a href="http://www.tutorialspoint.com/ajax">AJAX</a> -  
<a href="http://www.tutorialspoint.com/perl">PERL</a> <br />
```

Tani përfshini fajllin e posa krijuar nëpërmes të funksionit include.

test.php

```
<html>  
<body>  
<?php include("menu.php"); ?>  
<p>This is an example to show how to include PHP file!</p>  
</body>  
</html>
```

Funksioni require()

Funksioni require() dhe include() dallojnë në mënyrën si i trajtojnë gabimet (errorat). Funksioni require() ndalon ekzekutimin e skriptës në rast se fajlli të cilit i referohet mungon, për dallim prej funksionit include() i cili nuk ndalon ekzekutimin e skriptës.

Shembulli 2. Të jepet shembulli i përcjelljes së emrit të një fajlli inekzistent funksioneve include() dhe require().

test_req1.php

```
<html>  
<body>  
<?php include("xxmenu.php"); ?>  
<p>This is an example to show how to include wrong PHP file!</p>  
</body>  
</html>
```

Kodi i mësipërm do na japi rezultatin e mëposhtëm

```
This is an example to show how to include wrong PHP file!
```

Në rast se provojmë të thërrasim fajllin e njejtë (inekzistent) me funksionin `require()` si më poshtë atëherë PHP do kthejë gabim.

test_req2.php

```
<html>
<body>
<?php require("xxmenu.php"); ?>
<p>This is an example to show how to include wrong PHP file!</p>
</body>
</html>
```

PHP Fajllat dhe I/O

Hapja dhe mbyllja e fajllave:

Funksioni **fopen()** përdoret për hapjen e fajllave. Kërkon dy argumente: emrin e fajllit dhe modin në të cilin do të operohet me atë fajll:

Kemi 6 opsione të kombinimeve të modeve:

Mode	Purpose
r	Opens the file for reading only. Places the file pointer at the beginning of the file.
r+	Opens the file for reading and writing. Places the file pointer at the beginning of the file.
w	Opens the file for writing only. Places the file pointer at the beginning of the file. and truncates the file to zero length. If files does not exist then it attempts to create a file.
w+	Opens the file for reading and writing only. Places the file pointer at the beginning of the file. and truncates the file to zero length. If files does not exist then it attempts to create a file.
a	Opens the file for writing only. Places the file pointer at the end of the file. If files does not exist then it attempts to create a file.
a+	Opens the file for reading and writing only. Places the file pointer at the end of the file. If files does not exist then it attempts to create a file.

Në rast se një përpjekje për të hapur një fajll dështon atëherë funksioni **fopen** kthen **false**, përndryshe kthen një fajll pointer i cili përdoret për leximin dhe shkrim në atë fajll.

Mbyllja e një fajlli të hapur bëhet me funksionin **fclose()**. Funksioni **fclose()** kërkon që ti përcjellët një fajll pointer për argument.

Leximi i fajllit

Pas hapjes së fajllit me funksionin **fopen()** nëpërmjet funksionit **fread()** mund ta lexojm atë. Funksioni **fread()** kërkon dy argumente. Fajll pointerin dhe madhësinë e fajllit i cili do të lexohet i shprehur në bajt. Madhësin e fajllit mund ta gjejmë nëpërmes të funksionit **filesize()**. Më poshtë janë paraqitur hapat të cilët duhet të ndiqen për leximin e një fajlli:

- Hapim fajllin me funksionin **fopen()**.
- Marrim madhësin e fajllit me funksionin **filesize()**.
- Lexojm përmbajtjen e fajllit me funksionin **fread()**.
- Mbyllim fajllin me funksionin **fclose()**.

Detyra 1. Të jipet një shembull i shoqërimit të përmbajtjes së një tekst fajlli një variable dhe pastaj të shfaqet përmbajtja e saj në web faqe.

file.php

```
<html>
<head>
<title>Reading a file using PHP</title>
</head>
<body>

<?php
$filename = "C:/xampp2/htdocs/ushtirimet/Java9/tmp.txt";
$file = fopen( $filename, "r" );
if( $file == false )
{
    echo ( "Error in opening file" );
    exit();
}
$filesize = filesize( $filename );
$filetext = fread( $file, $filesize );

fclose( $file );

echo ( "File size : $filesize bytes" );
echo ( "<pre>$filetext</pre>" );
?>

</body>
</html>
```

Shkrimi në fajll

Nëpërmes të funksionit **fwrite()** në PHP mund të shkruajmë në një fajll të ri apo të shtojmë përmbajtje në një fajll ekzistues.

Shembulli3. Të jipet një shembull i krijimit të një fajlli nëpërmes funksionit **fwrite()**.

Filewrite.php

```
<?php
$filename = "C:/xampp2/htdocs/ushtimet/Java9/tmp2.txt";
$file = fopen( $filename, "w" );
if( $file == false )
{
    echo ( "Error in opening new file" );
    exit();
}
fwrite( $file, "This is a simple test\n" );
fclose( $file );
?>

<html>
<head>
<title>Writing a file using PHP</title>
</head>
<body>

<?php
if( file_exists( $filename ) )
{
    $filesize = filesize( $filename );
    $msg = "File created with name $filename ";
    $msg .= "containing $filesize bytes";
    echo ( $msg );
}
else
{
    echo ( "File $filename does not exist" );
}
?>
</body>
</html>
```

PHP Gabimet dhe trajtimi i tyre

Përdorimi i funksionit die():

Gjatë kodimit të faqes në PHP duhet të kemi parasysh gabimet të cilat mund të paraqiten dhe të trajtohen ato.

Shembulli4: Shkruani kodin i cili hapë një fajll, mirëpo së pari kontrollon për ekzistencën e tij. Në rast se ekziston, atëherë vazhdohe me hapje të tij, përndryshe programi ndalet dhe kthehet mesazhi nëpërmjet funksionit die()).

die.php

```
<?php
if(!file_exists("/tmp/test.txt"))
{
```

```
die("File not found");
}
else
{
    $file=fopen("/tmp/test.txt","r");
    print "Opend file sucessfully";
}
// Test of the code here.
?>
```

Definimi i Funksionit për trajtimin e Gabimeve:

PHP ofron kornizën për definimin e funksioneve për trajtimin e gabimeve.

Funksioni në fjalë duhet të jetë në gjendje të trajtoj, minimum dy parametra (*error level* dhe *error message*) por mund të pranoj deri në pesë parametra (opsionalisht: *file*, *line-number*, dhe *error context*):

Syntax

`error_handler`

A callback with the following signature. `NULL` may be passed instead, to reset this handler to its default state.

```
bool handler ( int $errno , string $errstr [, string $errfile [
, int $errline [, array $errcontext ]]] )
```

`errno`

The first parameter, `errno`, contains the level of the error raised, as an integer.

`errstr`

The second parameter, `errstr`, contains the error message, as a string.

`errfile`

The third parameter is optional, `errfile`, which contains the filename that the error was raised in, as a string.

`errline`

The fourth parameter is optional, `errline`, which contains the line number the error was raised at, as an integer.

`errcontext`

The fifth parameter is optional, `errcontext`, which is an array that points to the active symbol table at the point the error occurred. In other words, `errcontext` will contain an array of every variable that existed in the scope the error was triggered in. User error handler must not modify error context.

If the function returns `FALSE` then the normal error handler continues.

`error_types`

Can be used to mask the triggering of the `error_handler` function just like the `error_reporting` ini setting controls which errors are shown. Without this mask set the `error_handler` will be called for every error regardless to the setting of the `error_reporting` setting.

Shembulli5. me nivel me kompleks, por qarteson te gjithë procedure e error handling:

`error_handling.php`

```
<?php
// error handler funksioni
function myErrorHandler($errno, $errstr, $errfile, $errline)
{
    if (!(error_reporting() & $errno)) {
        // Ky error code nuk eshte i perfshire ne error_reporting
        return;
    }

    switch ($errno) {
        case E_USER_ERROR:
            echo "<b>GABIMI (ERROR) IM</b> [$errno] $errstr<br />\n";
            echo "    Gabim fatal ne rreshtin $errline ne file-n $errfile";
            echo ", PHP " . PHP_VERSION . " (" . PHP_OS . ")<br />\n";
            echo "Deshtim...<br />\n";
            exit(1);
            break;

        case E_USER_WARNING:
            echo "<b>PARALAJMERIMI (WARNING) IM</b> [$errno] $errstr<br />\n";
            break;

        case E_USER_NOTICE:
            echo "<b>NJOFTIMI (NOTICE) IM</b> [$errno] $errstr<br />\n";
            break;

        default:
            echo "Tip i gabimit i panjoftur: [$errno] $errstr<br />\n";
            break;
    }

    /* Mos ekzekuto PHP internal error handler */
    return true;
}

// funksioni per testimin e error handling
function scale_by_log($vect, $scale)
{
    if (!is_numeric($scale) || $scale <= 0) {
        trigger_error("log(x) per x <= 0 eshte i padefinuar, ju keni perdorur: scale = $scale", E_USER_ERROR);
    }
}
```

```
    if (!is_array($vect)) {
        trigger_error("Vlere hyrese jokorrekte, pritet varg i vlerave",
E_USER_WARNING);
        return null;
    }

    $temp = array();
    foreach($vect as $pos => $value) {
        if (!is_numeric($value)) {
            trigger_error("Vlera ne poziten $pos nuk eshte numer, duke perdorur 0
(zero)", E_USER_NOTICE);
            $value = 0;
        }
        $temp[$pos] = log($scale) * $value;
    }

    return $temp;
}

// vendost te perdoruesi per te definuar error handler
$old_error_handler = set_error_handler("myErrorHandler");
echo "<br>";
echo "<br>";

// Nxit (trigger) disa error-a, se pari defino nje mixed varg (array) me disa prej
anetareve vlera jo numerike
echo "vector a\n";
echo "<br>";
echo "<br>";
$a = array(2, 3, "foo", 5.5, 43.3, 21.11);
print_r($a);
echo "<br>";
echo "<br>";

// tani gjenero vargun e dyte
echo "----\nvector b - a notice (b = log(PI) * a)\n";
echo "<br>";
echo "<br>";
/* Vlera ne poziten $pos nuk eshte numer, duke perdorur 0 (zero)*/
$b = scale_by_log($a, M_PI);
print_r($b);
echo "<br>";
echo "<br>";

// kjo tani eshte problem, ne i kaluam nje string ne vend te vargut
echo "----\nvector c - a warning\n";
echo "<br>";
echo "<br>";
/* Vlere hyrese jokorrekte, pritet varg i vlerave */
$c = scale_by_log("not array", 2.3);
var_dump($c); // NULL
echo "<br>";
echo "<br>";

// Ky eshte critical error, log i zero-s per nr. negativ eshte i padefinuar
echo "----\nvector d - fatal error\n";
echo "<br>";
echo "<br>";
/* log(x) per x <= 0 eshte i padefinuar, ju keni perdorur: scale = $scale*/
$d = scale_by_log($a, -2.5);
var_dump($d); // Nuk arrihet asnjehere
echo "<br>";
```

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

Trajtimi i përjashtimeve (Exceptions):

- **Try** – Funkcioni (apo trupi i kodit) i cili testohet për ndonjë përjashtim duhet të vendoset në këtë bllok. Në rast se në kodin e vendosur në bllok nuk ka shkrepje të ndonjë përjashtimi, kodi do vazhdojë së egzekutuari, përndryshe do të kthehet (thrown) përjashtimi (exception).
- **Throw** – Secili throw duhet të ketë nga një **"catch"**.
- **Catch** - - Bloku "catch" pranon përjashtimin dhe krijon një object i cili përmban informatat e përjashtimit (exception).

Detyra 2. Të jipet një shembull i përdorimit të blloqeve Try, Throw dhe Catch.

try.php

```
<?php  
try {  
    $error = 'Always throw this error';  
    throw new Exception($error);  
  
    // Code following an exception is not executed.  
    echo 'Never executed';  
  
} catch (Exception $e) {  
    echo 'Caught exception: ', $e->getMessage(), "\n";  
}  
  
// Continue execution  
echo 'Hello World';  
?>
```

Dergimi i Email-it përmes PHP-së

```
bool mail ( string $to , string $subject , string $message [, string $additional_headers [,string $additional_parameters ]] )
```

to

Receiver, or receivers of the mail.

subject

Subject of the email to be sent.

message

Message to be sent.

Each line should be separated with a CRLF (\r\n). Lines should not be larger than 70 characters.

`additional_headers` (optional)

String to be inserted at the end of the email header.

This is typically used to add extra headers (From, Cc, and Bcc). Multiple extra headers should be separated with a CRLF (\r\n). If outside data are used to compose this header, the data should be sanitized so that no unwanted headers could be injected.

Note:

When sending mail, the mail *must* contain a *From* header. This can be set with the `additional_headers` parameter, or a default can be set in `php.ini`.

Failing to do this will result in an error message similar to *Warning: mail(): "sendmail_from" not set in php.ini or custom "From:" header missing*.

The *From* header sets also *Return-Path* under Windows.

Note:

If messages are not received, try using a LF (\n) only. Some Unix mail transfer agents (most notably » [qmail](#)) replace LF by CRLF automatically (which leads to doubling CR if CRLF is used). This should be a last resort, as it does not comply with » [RFC 2822](#).

`additional_parameters` (optional)

The `additional_parameters` parameter can be used to pass additional flags as command line options to the program configured to be used when sending mail, as defined by the `sendmail_path` configuration setting. For example, this can be used to set the envelope sender address when using sendmail with the `-f` sendmail option.

This parameter is escaped by `escapeshellcmd()` internally to prevent command execution. `escapeshellcmd()` prevents command execution, but allows to add additional parameters. For security reasons, it is recommended for the user to

sanitize this parameter to avoid adding unwanted parameters to the shell command.

Since `escapeshellcmd()` is applied automatically, some characters that are allowed as email addresses by internet RFCs cannot be used. `mail()` can not allow such characters, so in programs where the use of such characters is required, alternative means of sending emails (such as using a framework or a library) is recommended.

The user that the webserver runs as should be added as a trusted user to the sendmail configuration to prevent a 'X-Warning' header from being added to the message when the envelope sender (-f) is set using this method. For sendmail users, this file is `/etc/mail/trusted-users`.

EmailForm.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Forma per dergim te emailit</title>

</head>

<body>
<form name="formsendmail" action="email2.php" method="post">

<label for="_emri">Emri: </label><br/>
<input type="text" name="emri" id="emri" /><br/>

<label for="_email">Email-i juaj: </label><br/>
<input type="email" name="email" id="email" /><br/>

<label for="_msg">Mesazhi: </label><br/>
<textarea name="msg" id="msg" cols="30" rows="6"></textarea><br/>

<input type="submit" id="btndergo" value="Dergo Email"/>
</form>
</body>
</html>
```

Email1.php

```
<?php
$name = $_POST["emri"];
$email = $_POST["email"];
$msg = $_POST["msg"];
$to = "feedback@example.com";
$subject = "Feedback-u nga ueb sajti! ";
```

```
$mailcontent = 'Emri i dërguesit: '. $name . "\n"
.'Email-i i dërguesit: '. $email . "\n"
.'Komenti i dërguesit: \n'.$msg . "\n";
$headers = 'from: webserver@example.com';
if(mail($to, $subject, $coments, $headers)) {
echo 'Email-i është dërguar me sukses!';
}
else {
echo 'Dërgimi i email-it ka dështuar!';
} ?>
```

Ndryshimi detyrës nëse dëshirojmë të dërgojm këtë email me anë të një shfrytëzuesi në gmail:

Email2.php

```
<?php

$name = $_POST["emri"];
$email = $_POST["email"];
$msg = $_POST["msg"];
$from = "korab.rrmoku@gmail.com";
$to = "rrkorab@gmail.com";
$subject = "Feedback-u nga ueb sajti! ";
$mailcontent = 'Emri i dërguesit: '. $name . "\n"
.'Email-i i dërguesit: '. $email . "\n"
.'Komenti i dërguesit: \n'.$msg . "\n";
$host = "ssl://smtp.gmail.com";
$port = "465";
$username = "korab.rrmoku@gmail.com";
$password = "SHENOHET PASSWORDI I ADRESES SE DERGUESIT";
$headers = array ('From' => $from,
'To' => $to,
'Subject' => $subject);
$smtp = Mail::factory('smtp',
array ('host' => $host,
'port' => $port,
'auth' => true,
'username' => $username,
'password' => $password));
$mail = $smtp->send($to, $headers, $mailcontent);
if($mail) {
echo 'Email-i është dërguar me sukses!';
}
else {
echo 'Dërgimi i email-it ka dështuar!';
}
?>
```

Diskutim: PHP stopping email injection – Menyra më e mire për siguri është validimi i të dhënave hyrëse.

Kjo kontrollë aplikohet para fushes “from”, dmth validohet dërguesi

- FILTER_SANITIZE_EMAIL filteri largon të gjithë karakterat ilegal të email-it nga stringu
- The FILTER_VALIDATE_EMAIL filteri validon një vlerë nëse paraqet email address apo jo

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Programimi në Internet

Java 10: AJAX



PHP & AJAX

AJAX akronimi qëndron për Asynchronous JavaScript and XML. Është teknologji e implementuar në browser si e tillë e pamvarur prej ueb server-it.

Në uebsajtat konvencional informatat transmetohen tek serveri dhe prapa duke përdorur kërkesa sinkrone e cila nënkupton: pas mbushjes së formës dhe procesimit të shënimeve (submit), fokusi ridrejtohet tek një faqe e re në server.

Në uebsajtat të cilët implementojnë AJAX teknologjinë, në momentin që shënimet të procesohen (submit në formë), JavaScript do të bëjë kërkesë serverit, pastaj do të interpretojë rezultatet dhe do të azhurojë pamjen e faqës aktuale.

Komponenta qendërore e AJAX teknologjisë është:

XMLHttpRequest- JavaScript objekt i cili performon interaksionin asinkron me serverin

Hapat e një AJAX Operacioni

1. Një ngjarje shkrepet në klient
2. Një XMLHttpRequest objekt krijohet
3. XMLHttpRequest objekti konfigurohet
4. XMLHttpRequest objekti bën një kërkesë asinkrone ndaj Uebserverit.
5. Uebserveri kthen rezultatin i cili përmban një XML dokument.
6. XMLHttpRequest objekti thërret callback() funksionin dhe proceson rezultatin.
7. Azhurohet HTML DOM.

Shembulli 1. Të jipet një shembull i ruajtjes dhe leximit të të dhënave nga një PHP skriptë nëpërmes teknologjisë AJAX.

[KerkoLendet.php](#)
[jsLendet.js](#)
[LendetServer.php](#)

KerkoLendet.php:

```
<html>
  <head>
    <title>Ajax dhe PHP</title>
    <script type="text/javascript" src="jsLendet.js">
    </script>
  </head>
  <body>
    Lenda:<input type="text" id="txtLenda" onKeyUp="kerkoLendet(this.value)"/>
    <p>
      <br/>
      <br/>
    </p>
  </body>
</html>
```

```
        <br/>
        Rezultati: <div id="txtRez"></div>
    </p>
</body>
</html>
```

jsLendet.js

```
var xmlhttp;

//Funksioni kerkoLendet() ekzekutohet cdo herë që karakteri futet në "txtLenda".
function kerkoLendet(str){
    //Thërret f-onin createRequest() të krijoj një object XMLHttpRequest
    xmlhttp = createRequest();
    if(xmlhttp==null){
        alert("Nuk mund te krijoni kerkesen");
        return;
    }

    //Definon URL (emrin e fajllit) të dërgoj në server
    var url = "http://localhost/Ushtimet/Java10/LendetServer.php";

    //Shton një parametër (q) tek URL me përmbajtje të fushës hyrëse
    url = url+"?q="+str;

    //Hap objektin XMLHttpRequest me URL-në e dhënë
    xmlhttp.open("GET",url,true);

    //Dërgon një kërkesë HTTP tek server
    xmlhttp.send(null);

    //Sa herë që ndryshon vetia readyState, f-oni stateChange() do të ekzekutohet
    xmlhttp.onreadystatechange = displayRes;
}

//Ky funksion ekzekutohet sa herë që gjendja e objektit XMLHttpRequest ndryshon.
function displayRes(){

    //alert(xmlhttp.readyState);
    //Kur gjendja ndryshon në 4("komplet") dhe status=200,
    //përmbajtja e txtRez mbushet me tekstin përgjigje.
    if(xmlhttp.readyState==4){
        if(xmlhttp.status==200){
            document.getElementById("txtRez").innerHTML = xmlhttp.responseText;
        }
    }
}

//ky funksion krijon objekte të ndryshme XMLHttpRequest për browser të ndryshëm
function createRequest()
{
    try{
        // kodi per IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();
    }
    catch(ex)
    {
        try
        {
            // kodi per IE6, IE5
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch(ex){
            xmlhttp = null;
        }
    }
}
```

```
    }  
    }  
    return xmlhttp;  
}
```

LendetServer.php

```
<?php  
  
//Mbush vargun me emra  
$a[]="Gjuhe Programuese";  
$a[]="Bazat e te Dhenave";  
$a[]="Programimi ne Internet";  
$a[]="Programimi i Orientuar ne Objekte";  
$a[]="Programimi i avancuar";  
$a[]="Matematika 1";  
$a[]="Matematika 2";  
$a[]="Matematika 3";  
//merr parametrin q nga URL-ja  
$q=$_GET["q"];  
  
//shiko per te gjitha informatat nga vargu nese gjatesi e 'q>0'  
if (strlen($q) > 0)  
{  
    $informata="";  
    for($i=0; $i<count($a); $i++)  
    {  
        if (strtolower($q)==strtolower(substr($a[$i],0,strlen($q))))  
        {  
            if ($informata=="")  
            {  
                $informata=$a[$i];  
            }  
            else  
            {  
                $informata=$informata.", ".$a[$i];  
            }  
        }  
    }  
}  
  
if ($informata == "")  
{  
    $rezultati=" ska propozime...";  
}  
else  
{  
    $rezultati=$informata;  
}  
  
//paraqit rezultatin  
echo $rezultati;  
?>
```

Shembulli 2. Shembulli me AJAX dhe MySQL.

KerkoStudentin.php

isStudentet.js

Studentet.php

KerkoStudentin.php

```
<html>
  <head>
    <title>Ajax dhe PHP</title>
    <script type="text/javascript" src="jsStudentet.js">
    </script>
  </head>
  <body>
    NrIndeksit:<input type="text" id="txtNrIndeksit"
                      onKeyUp="kerkoStudentin(this.value)"/>

    <p>
      <br/>
      <br/>
      <br/>
      Rezultati: <div id="txtRez"></div>
    </p>
  </body>
</html>
```

jsStudentet.js

```
var xmlhttp;

//Funksioni kerkoLendet() ekzekutohet cdo herë që karakteri futet në "txtLenda".
function kerkoStudentin(str)
{
    //Thërret f-onin createRequest() të krijoj një object XMLHttpRequest
    xmlhttp = createRequest();

    if(xmlhttp==null){
        alert("Nuk mund te krijoni kerkesen");
        return;
    }

    //Definon URL (emrin e fajllit) të dërgoj në server
    var url = "http://localhost/ushtimet/Javal0/studentet.php";

    //Shton një parametër (q) tek URL me përmbajtje të fushës hyrëse
    url = url+"?q="+str;

    //Hap objektin XMLHttpRequest me URL-në e dhënë
    xmlhttp.open("GET",url,true);

    //Dërgon një kërkesë HTTP tek server
    xmlhttp.send(null);

    //Sa herë që ndryshon vetia readyState, f-oni stateChange() do të ekzekutohet
    xmlhttp.onreadystatechange = displayRes;
}

//Ky funksion ekzekutohet sa herë që gjendja e objektit XMLHttpRequest ndryshon.
function displayRes()
{
    //alert(xmlhttp.readyState);
    //Kur gjendja ndryshon në 4("komplet") dhe status=200,
    //përmbajtja e txtRez mbushet me tekstin përgjigje.
    if(xmlhttp.readyState==4){
        if(xmlhttp.status==200){
            document.getElementById("txtRez").innerHTML = xmlhttp.responseText;
        }
    }
}
```

```
}

//ky funksion krijon objekte të ndryshme XMLHTTP për browser të ndryshëm
function createRequest()
{
    try
    {
        // kodi për IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();
    }
    catch(ex)
    {
        try
        {
            // kodi për IE6, IE5
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch(ex)
        {
            xmlhttp = null;
        }
    }
    return xmlhttp;
}
```

Studentet.php

```
<?php

$con = mysql_connect("localhost","root","");
mysql_select_db("ushtrime",$con);

$sql_Select = "select NrIndeksit, Emri, Mbiemri, Adresa from studentet
               where NrIndeksit like '$_GET[q]%'";

print $sql_Select;

$res_Select = mysql_query($sql_Select);
$nrStd = mysql_num_rows($res_Select);

if($nrStd) {
    echo "<table>";
    echo "<tr style='color:white; background-color:#7AC234;'>
        <th color='white'>NrIndeksit</th>
        <th>Emri</th>
        <th>Mbiemri</th>
        <th>Adresa</th>
    </tr>";

    while($row = mysql_fetch_array($res_Select)){
        echo "<tr>
            <td>$row[NrIndeksit]</td>
            <td>$row[Emri]</td>
            <td>$row[Mbiemri]</td>
            <td>$row[Adresa]</td>
        </tr>";
    }

    echo "</table>";
}
echo '<br/>';
mysql_close();
```

?>

Shembulli 3. Shembulli me AJAX dhe MySQL.

Forma.php

MerrPerdoruesin.php

Forma.php

```
<html>
<head>
<script>
function showUser(str){
    if (str=="")
    {
        document.getElementById("txtHint").innerHTML="";
        return;
    }

    if (window.XMLHttpRequest)
    {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    } else {
        // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }

    xmlhttp.onreadystatechange=function() {

        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            document.getElementById("txtHint").innerHTML=xmlhttp.responseText;
        }

    }
    xmlhttp.open("GET","MerrPerdoruesin.php?q="+str,true);
    xmlhttp.send();
}
</script>
</head>
<body>
    <form>
        <select name="users" onChange="showUser(this.value)">
            <option value="">Zgjedh Personin:</option>
            <option value="1">Filan Fisteku</option>
            <option value="2">Korab Rrmoku</option>
            <option value="3">Labinot Bajraktari</option>
            <option value="4">Artan Gashi</option>
        </select>
    </form>
    <br>
<div id="txtHint"><b>Person info will be listed here.</b></div>

</body>
</html>
```

MerrPerdoruesin.php

```
<?php
$q = $_GET["q"];
```

```
$con = mysqli_connect('localhost','root','','ushtrime');
if (!$con) {
    die('Could not connect: ' . mysqli_error($con));
}

mysqli_select_db($con,"ajax_demo");
$sql="SELECT * FROM personat WHERE id = '". $q. "'";
$result = mysqli_query($con,$sql);

echo "<table border='1'>
<tr>
<th>Emri</th>
<th>Mbiemri</th>
<th>Moshë</th>
<th>Vendbanimi</th>
<th>Profesioni</th>
</tr>";

while($row = mysqli_fetch_array($result)) {
    echo "<tr>";
    echo "<td>" . $row['Emri'] . "</td>";
    echo "<td>" . $row['Mbiemri'] . "</td>";
    echo "<td>" . $row['Moshë'] . "</td>";
    echo "<td>" . $row['Vendbanimi'] . "</td>";
    echo "<td>" . $row['Profesioni'] . "</td>";
    echo "</tr>";
}
echo "</table>";

mysqli_close($con);
?>
```

Shembulli 4. Shembulli me AJAX dhe XML.

[cd.html](#)

[getcd.php](#)

[cd_katalog.xml](#)

[cd.html](#)

```
<html>
<head>
<script>
function showCD(str) {
    if (str=="") {
        document.getElementById("txtHint").innerHTML="";
        return;
    }
    if (window.XMLHttpRequest) {
        // kodi për IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }else {
        // kodi për IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function() {
        if (xmlhttp.readyState==4 && xmlhttp.status==200) {
            document.getElementById("txtHint").innerHTML=xmlhttp.responseText;
        }
    }
    xmlhttp.open("GET","getcd.php?str="+str,true);
    xmlhttp.send();
}
```

```
        }
    }
    xmlhttp.open("GET","getcd.php?q="+str,true);
    xmlhttp.send();
}
</script>
</head>
<body>

<form>
Zgjedh CD-ne:
<br>
<br>
<select name="cds" onchange="showCD(this.value)">
<option value="">Zgjedh nje CD:</option>
<option value="Bob Dylan">Bob Dylan</option>
<option value="Bonnie Tyler">Bonnie Tyler</option>
<option value="Dolly Parton">Dolly Parton</option>
</select>
</form>
<div id="txtHint"><b>Informatat e CD-se do te listohen ketu...</b></div>

</body>
<!--Funksioni showCD() ben keto pune:

Kontrollon nese CD eshte zgjedhur (selected)
Krijon nje XMLHttpRequest objekt
Krijon funksionin qe do te ekzekutohet kur pergjigjja e serverit eshte gati
Dergon kerkesen tek fajlli ne server
Verjeme qe parametri (q) shtohet ne URL (se bashku me permbajtjen e dropdown listes)--
>
</html>
```

getcd.php

```
<?php
$q=$_GET["q"];

$xmlDoc = new DOMDocument();
$xmlDoc->load("cd_catalog.xml");

$x=$xmlDoc->getElementsByTagName('ARTIST');

for ($i=0; $i<=$x->length-1; $i++)
{
    //Proceson vetem nyjet qe jane elemente
    if ($x->item($i)->nodeType==1)
    {
        if ($x->item($i)->childNodes->item(0)->nodeValue == $q)
        {
            $y=($x->item($i)->parentNode);
        }
    }
}

$cd=($y->childNodes);

for ($i=0;$i<$cd->length;$i++) {
    //Proceson vetem nyjet qe jane elemente
    if ($cd->item($i)->nodeType==1) {
        echo("<b>" . $cd->item($i)->nodeName . " :</b> ");
    }
}
```



```
        echo($cd->item($i)->childNodes->item(0)->nodeValue);  
        echo("<br>");  
    }  
?>
```

cd_katalog.xml

```
<CATALOG>  
<CD>  
    <TITLE>Empire Burlesque</TITLE>  
    <ARTIST>Bob Dylan</ARTIST>  
    <COUNTRY>USA</COUNTRY>  
    <COMPANY>Columbia</COMPANY>  
    <PRICE>10.90</PRICE>  
    <YEAR>1985</YEAR>  
</CD>  
<CD>  
    <TITLE>Hide your heart</TITLE>  
    <ARTIST>Bonnie Tyler</ARTIST>  
    <COUNTRY>UK</COUNTRY>  
    <COMPANY>CBS Records</COMPANY>  
    <PRICE>9.90</PRICE>  
    <YEAR>1988</YEAR>  
</CD>  
<CD>  
    <TITLE>Greatest Hits</TITLE>  
    <ARTIST>Dolly Parton</ARTIST>  
    <COUNTRY>USA</COUNTRY>  
    <COMPANY>RCA</COMPANY>  
    <PRICE>9.90</PRICE>  
    <YEAR>1982</YEAR>  
</CD>  
<CD>  
    <TITLE>Still got the blues</TITLE>  
    <ARTIST>Gary Moore</ARTIST>  
    <COUNTRY>UK</COUNTRY>  
    <COMPANY>Virgin records</COMPANY>  
    <PRICE>10.20</PRICE>  
    <YEAR>1990</YEAR>  
</CD>  
<CD>  
    <TITLE>Eros</TITLE>  
    <ARTIST>Eros Ramazzotti</ARTIST>  
    <COUNTRY>EU</COUNTRY>  
    <COMPANY>BMG</COMPANY>  
    <PRICE>9.90</PRICE>  
    <YEAR>1997</YEAR>  
</CD>  
<CD>  
    <TITLE>One night only</TITLE>  
    <ARTIST>Bee Gees</ARTIST>  
    <COUNTRY>UK</COUNTRY>  
    <COMPANY>Polydor</COMPANY>  
    <PRICE>10.90</PRICE>  
    <YEAR>1998</YEAR>  
</CD>  
<CD>  
    <TITLE>Sylvias Mother</TITLE>  
    <ARTIST>Dr.Hook</ARTIST>  
    <COUNTRY>UK</COUNTRY>  
    <COMPANY>CBS</COMPANY>  
    <PRICE>8.10</PRICE>  
    <YEAR>1973</YEAR>
```

```
</CD>
<CD>
    <TITLE>Maggie May</TITLE>
    <ARTIST>Rod Stewart</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>Pickwick</COMPANY>
    <PRICE>8.50</PRICE>
    <YEAR>1990</YEAR>
</CD>
<CD>
    <TITLE>Unchain my heart</TITLE>
    <ARTIST>Joe Cocker</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>EMI</COMPANY>
    <PRICE>8.20</PRICE>
    <YEAR>1987</YEAR>
</CD>
</CATALOG>
```