Universiteti i Prishtinës
Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike
Lënda: Programimi në Internet
Java 9: Koncepte 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 shebulli 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 opsine 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 attemts 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 attemts 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 attemts 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 attemts 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/ushtrimet/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()** ne 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/ushtrimet/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 exit" );
}
?>
</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.

Shembull4: Shkruani kodin i cili hapë një fajll, mirëpo së pari kontrollon për ekzistencën e tij. Në rast se ekziston, atëherë vazhdohet 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 (opcionalisht: *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 gjithe 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 –** Funksioni (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 vazhdoj 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_hea
ders [,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 addtional 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