PHP (Day3)

Prepared by: Noha Shehab
Teaching assistant at ITI

Agenda

- String manipulation
- Regular expression
- Uploading files
- Session and Cookies

Popular functions

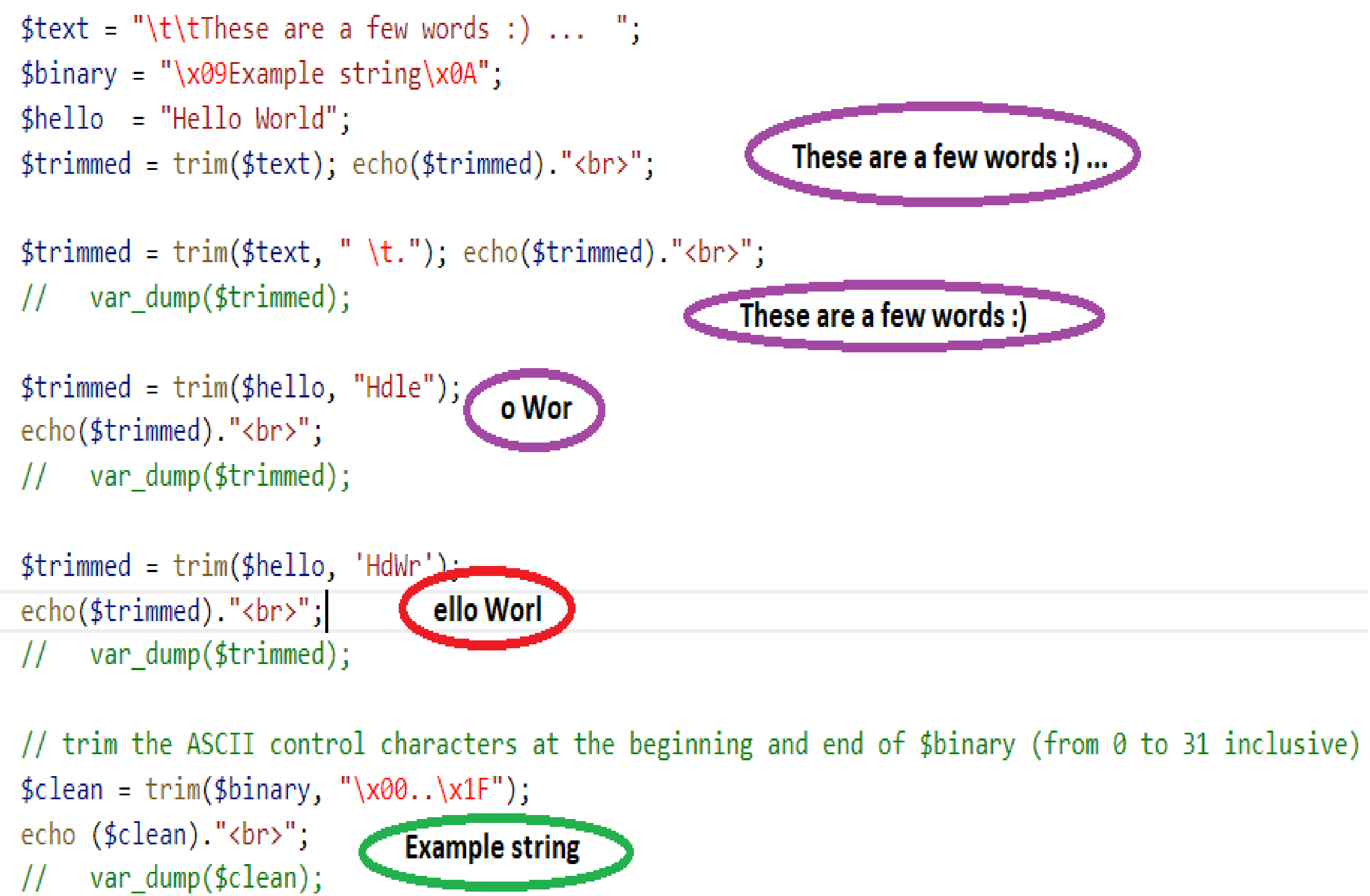
- Trim white spaces
  - trim(), ltrim(), rtrim()===chop()
- Applying some more sophisticated formatting
  - printf(), sprintf()
- Format the string into upper or lower cases
  - Strtoupper(), strtolower(), ucfirst() anducwords().
- Formatting string to storage
  - addslashes() andstripslashes()
- Join, Split strings
  - Explode, implode, join

Popular functions

- Comparing string
  - Strcomp(), strcasecmp()   #output database
- Checking string length
  - Strlen()

String Manipulation

- Trim function:
  Strip whitespace (or other characters) from the beginning and end of a string:

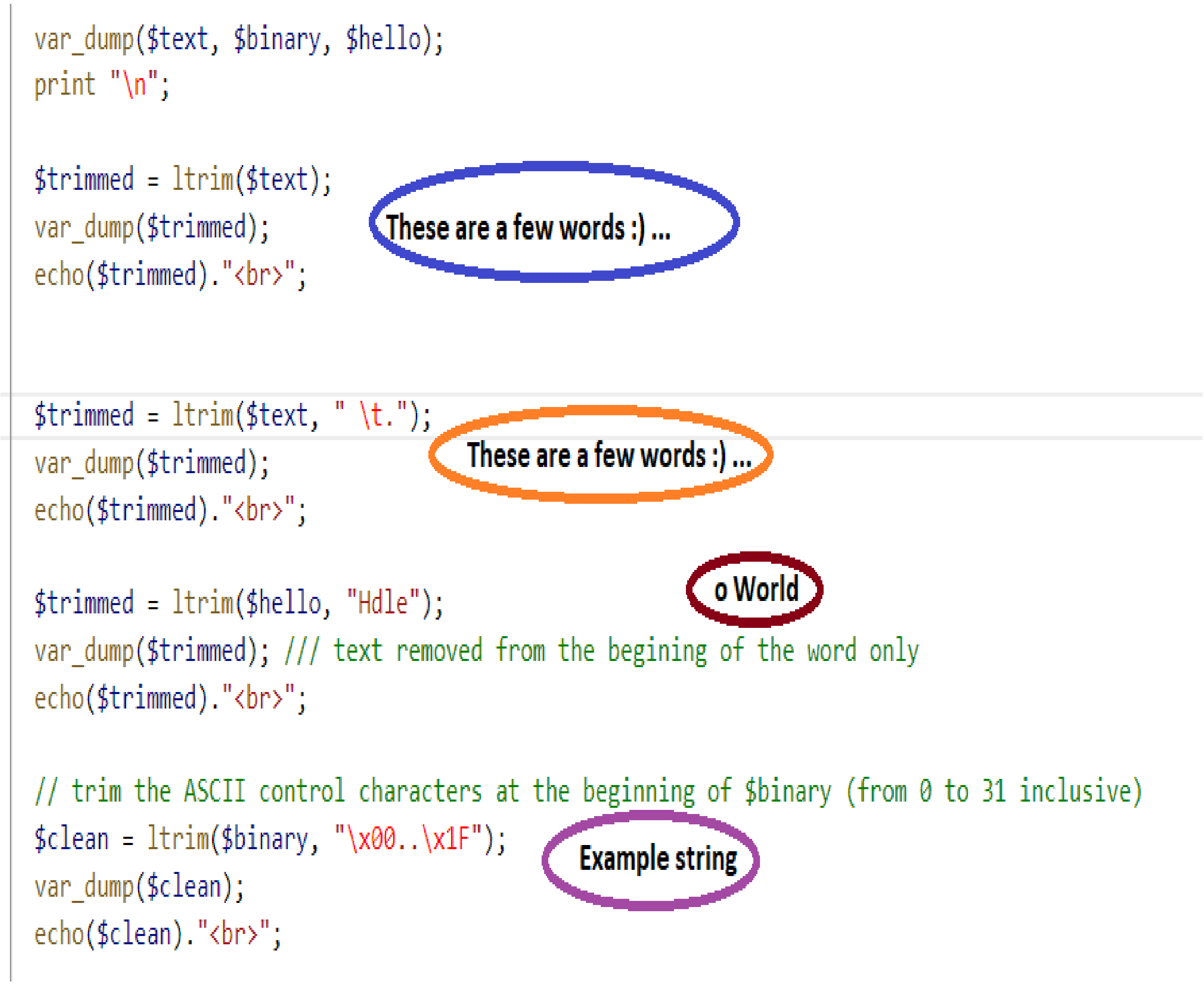


```
$text = "\t\tThese are a few words :) ...  ";
$binary = "\x09Example string\x0A";
$hello  = "Hello World";
$trimmed = trim($text); echo($trimmed)."<br>";

$trimmed = trim($text, " \t."); echo($trimmed)."<br>";
//   var_dump($trimmed);

$trimmed = trim($hello, "Hdle");
echo($trimmed)."<br>";
//   var_dump($trimmed);

$trimmed = trim($hello, 'HdWr');
echo($trimmed)."<br>";
//   var_dump($trimmed);

// trim the ASCII control characters at the beginning and end of $binary (from 0 to 31 inclusive)
$clean = trim($binary, "\x00..\x1F");
echo ($clean)."<br>";
//   var_dump($clean);
```

```
These are a few words :) ...
These are a few words :)
o Wor
ello Worl
Example string
```

String Manipulation cont.

- ltrim:
  Strip whitespace (or other characters) from the beginning of a string.



```
var_dump($text, $binary, $hello);
print "\n";

$trimmed = ltrim($text);
var_dump($trimmed);
echo($trimmed)."<br>";
```

These are a few words :) ...

```
$trimmed = ltrim($text, " \t.");
var_dump($trimmed);
echo($trimmed)."<br>";
```

These are a few words :) ...

```
$trimmed = ltrim($hello, "Hdle");
var_dump($trimmed); /// text removed from the begining of the word only
echo($trimmed)."<br>";
```

o World

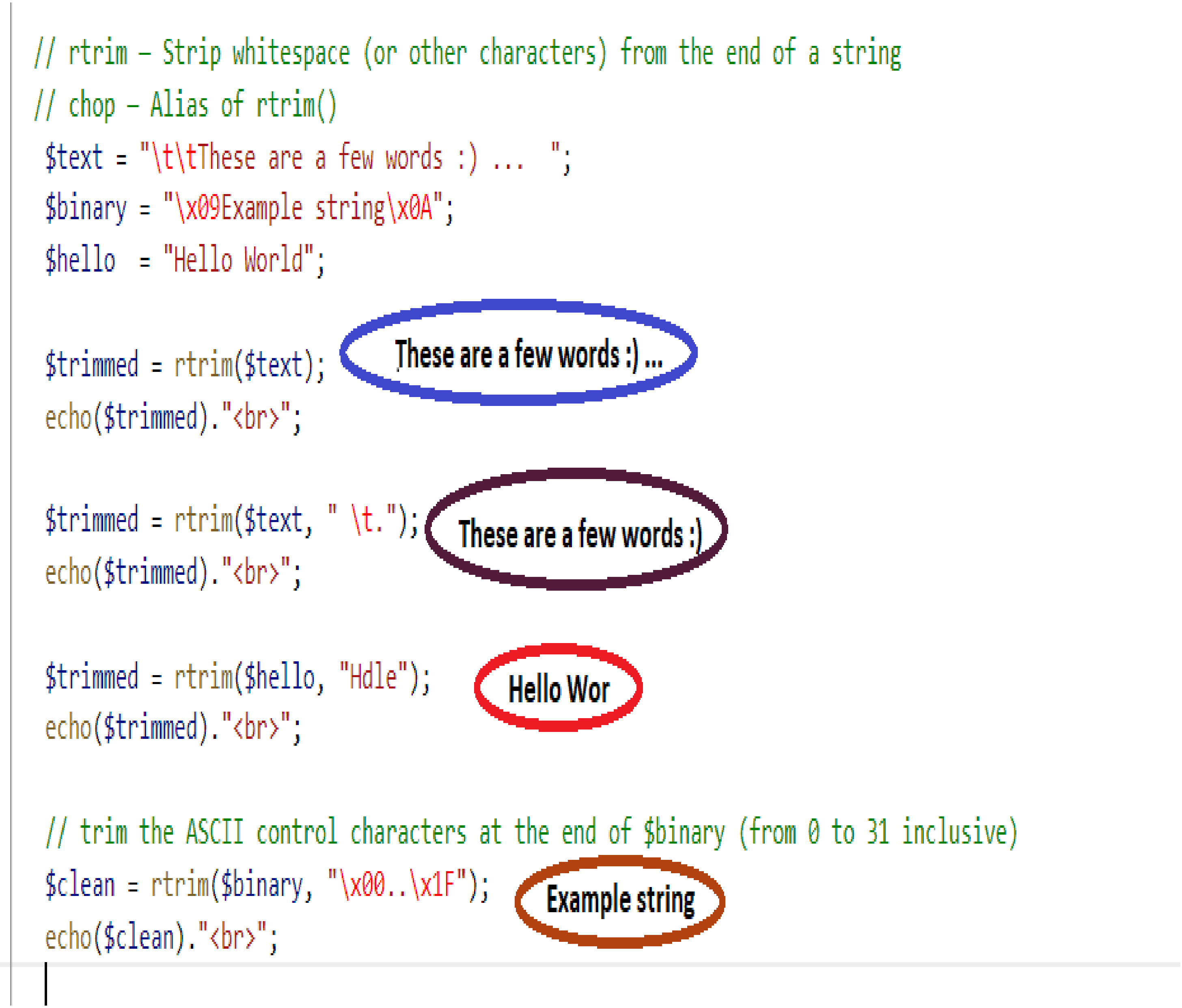
```
// trim the ASCII control characters at the beginning of $binary (from 0 to 31 inclusive)
$clean = ltrim($binary, "\x00..\x1F");
var_dump($clean);
echo($clean)."<br>";
```

Example string

String Manipulation cont.

- rtrim:
  Strip whitespace (or other characters) from the end of a string

chop —> Alias of rtrim()



```
// rtrim – Strip whitespace (or other characters) from the end of a string
// chop – Alias of rtrim()
 $text = "\t\tThese are a few words :) ...  ";
 $binary = "\x09Example string\x0A";
 $hello  = "Hello World";

 $trimmed = rtrim($text);
 echo($trimmed)."<br>";
```

```
These are a few words :) ...
```

```
 $trimmed = rtrim($text, " \t.");
 echo($trimmed)."<br>";
```

```
These are a few words :)
```

```
 $trimmed = rtrim($hello, "Hdle");
 echo($trimmed)."<br>";
```

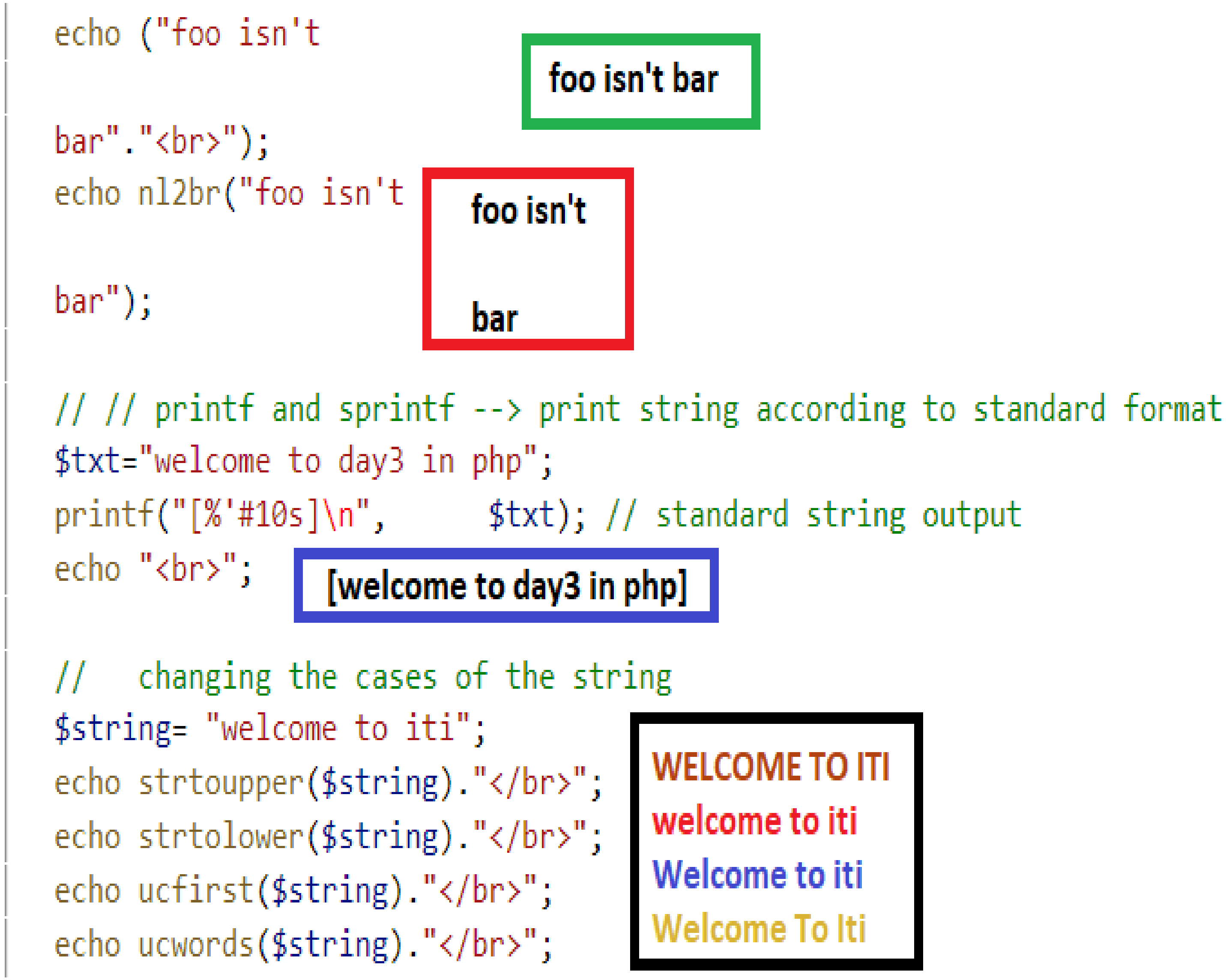
```
Hello Wor
```

```
 // trim the ASCII control characters at the end of $binary (from 0 to 31 inclusive)
 $clean = rtrim($binary, "\x00..\x1F");
 echo($clean)."<br>";
```

```
Example string
```

String Manipulation cont.

- nl2br(): Returns string with <br /> or <br> inserted before all newlines (\r\n, \n\r, \n and \r)
- Printf(), sprint()



```
echo ("foo isn't

bar"."<br>");
echo nl2br("foo isn't

bar");

// // printf and sprintf --> print string according to standard format
$txt="welcome to day3 in php";
printf("[%'#10s]\n",      $txt); // standard string output
echo "<br>";

//   changing the cases of the string
$string= "welcome to iti";
echo strtoupper($string)."</br>";
echo strtolower($string)."</br>";
echo ucfirst($string)."</br>";
echo ucwords($string)."</br>";
```

```
foo isn't bar
```

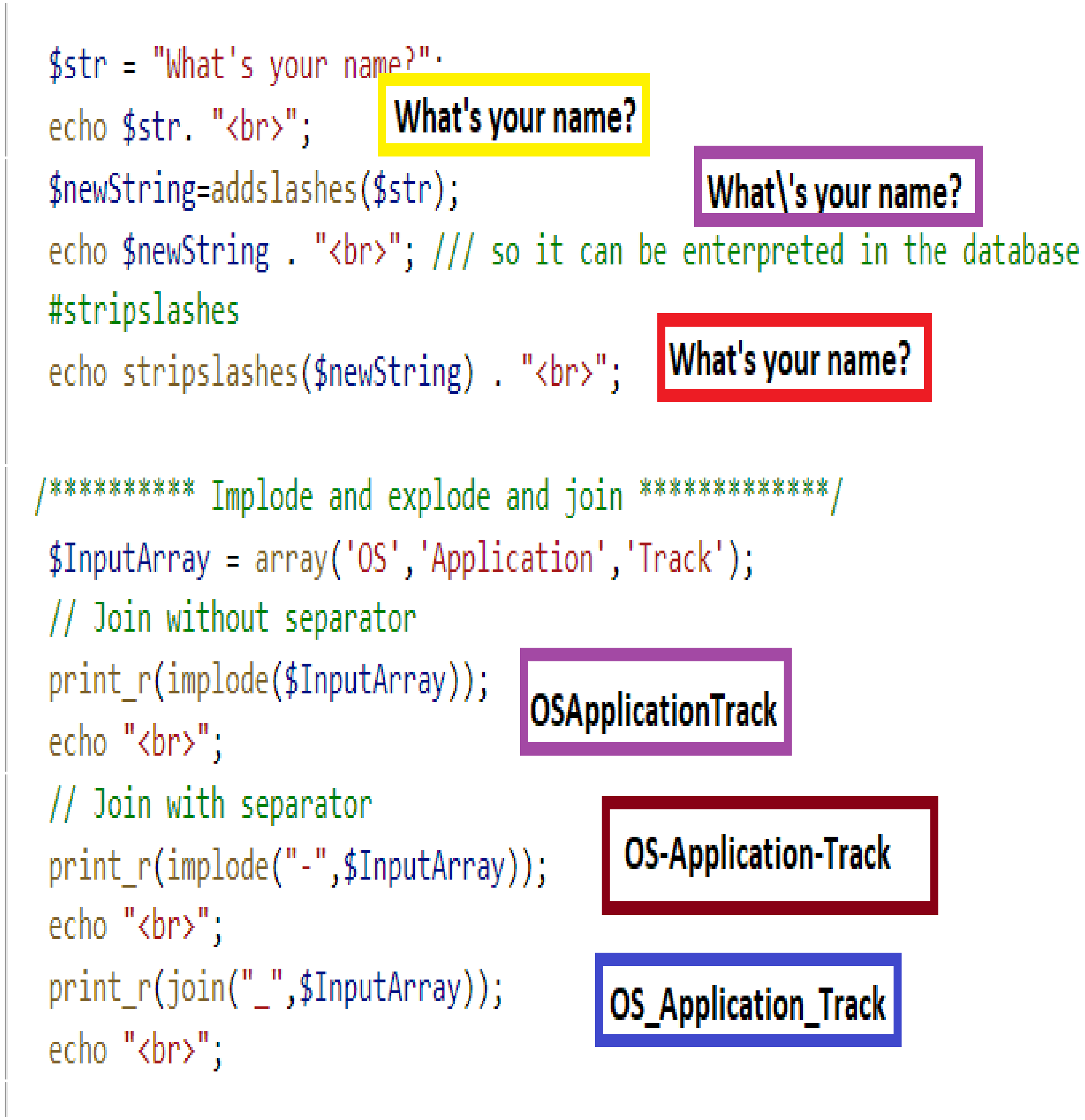
```
foo isn't

bar
```

```
[welcome to day3 in php]
```

```
WELCOME TO ITI
welcome to iti
Welcome to iti
Welcome To Iti
```

String Manipulation cont.

- formatting sting to storage:
  some strings can have quotation marks single or double, slashes an d null char so you need to escape these chars
  some dbms doesn't accept special chars
- Implode: joins the array elements into one string, sperate them with a delimiter if it is specified.

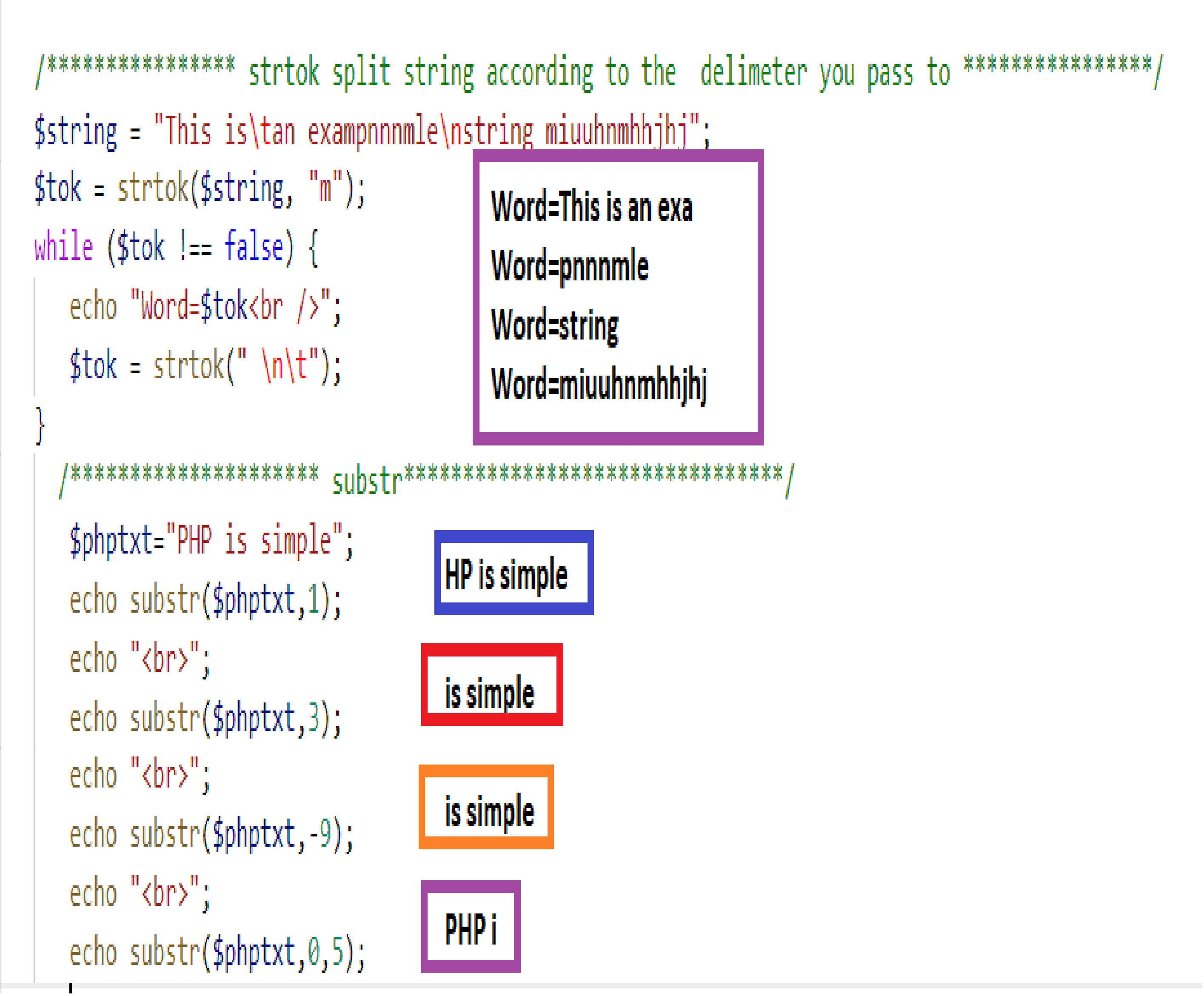


```
$str = "What's your name?";
echo $str. "<br>";
$newString=addslashes($str);
echo $newString . "<br>"; /// so it can be enterpreted in the database
#stripslashes
echo stripslashes($newString) . "<br>";

/********** Implode and explode and join *************/
$InputArray = array('OS','Application','Track');
// Join without separator
print_r(implode($InputArray));
echo "<br>";
// Join with separator
print_r(implode("-",$InputArray));
echo "<br>";
print_r(join("_",$InputArray));
echo "<br>";
```

```
What's your name?
What\'s your name?
What's your name?
OSApplicationTrack
OS-Application-Track
OS_Application_Track
```

String Manipulation cont.

- strtok split string according t o the delimiter you pass to.
- substr split string according the index



```
/**************** strtok split string according to the  delimeter you pass to ****************/
$string = "This is\tan exampnnnmle\nstring miuuhnmhhjhj";
$tok = strtok($string, "m");
while ($tok !== false) {
   echo "Word=$tok<br />";
   $tok = strtok(" \n\t");
}
 /******************** substr********************************/
   $phptxt="PHP is simple";
   echo substr($phptxt,1);
   echo "<br>";
   echo substr($phptxt,3);
   echo "<br>";
   echo substr($phptxt,-9);
   echo "<br>";
   echo substr($phptxt,0,5);
```

```
Word=This is an exa
Word=pnnnmle
Word=string
Word=miuuhnmhhjhj
```

```
HP is simple
```

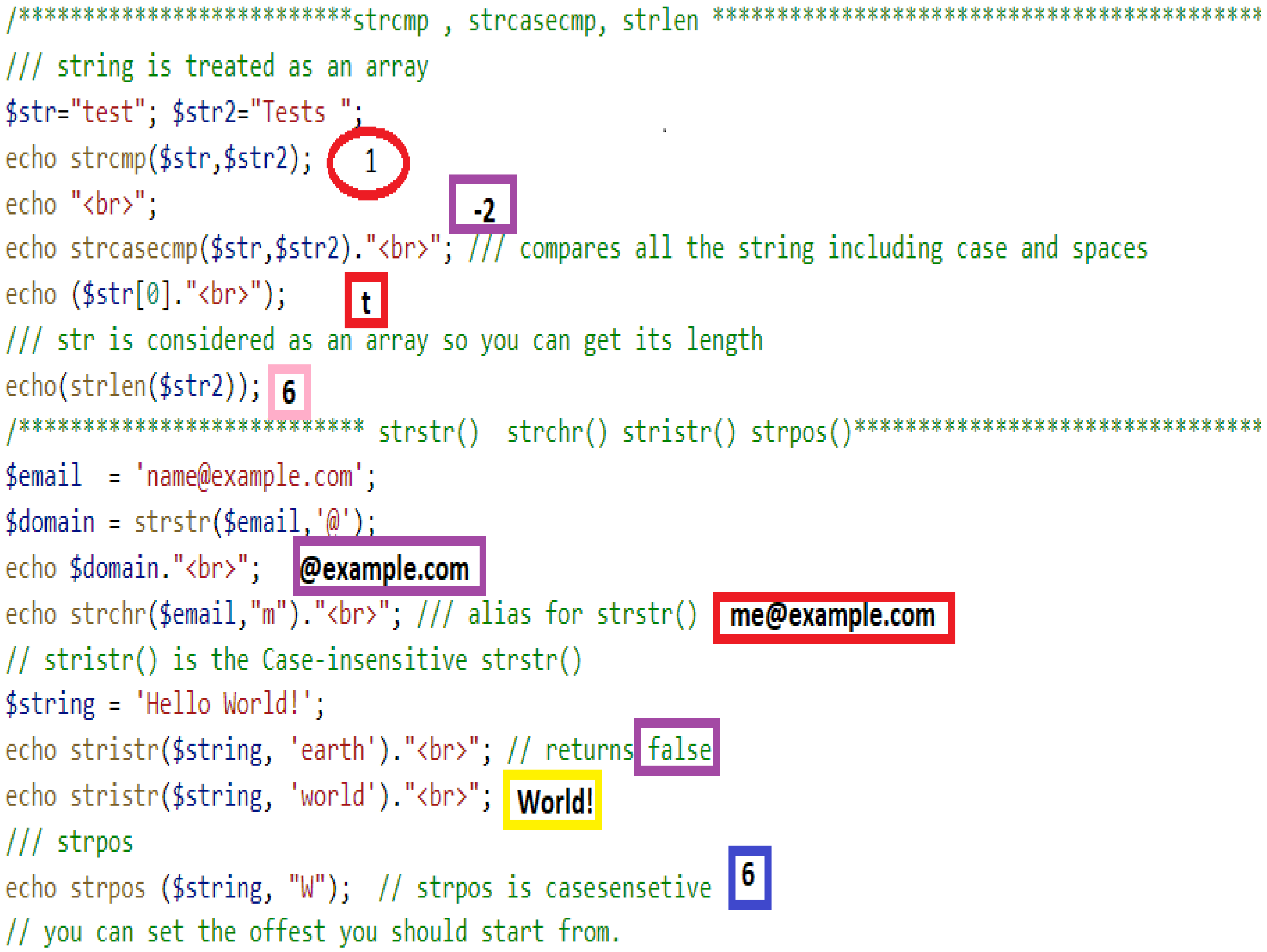
```
is simple
```

```
is simple
```

```
PHP i
```

String Manipulation cont.

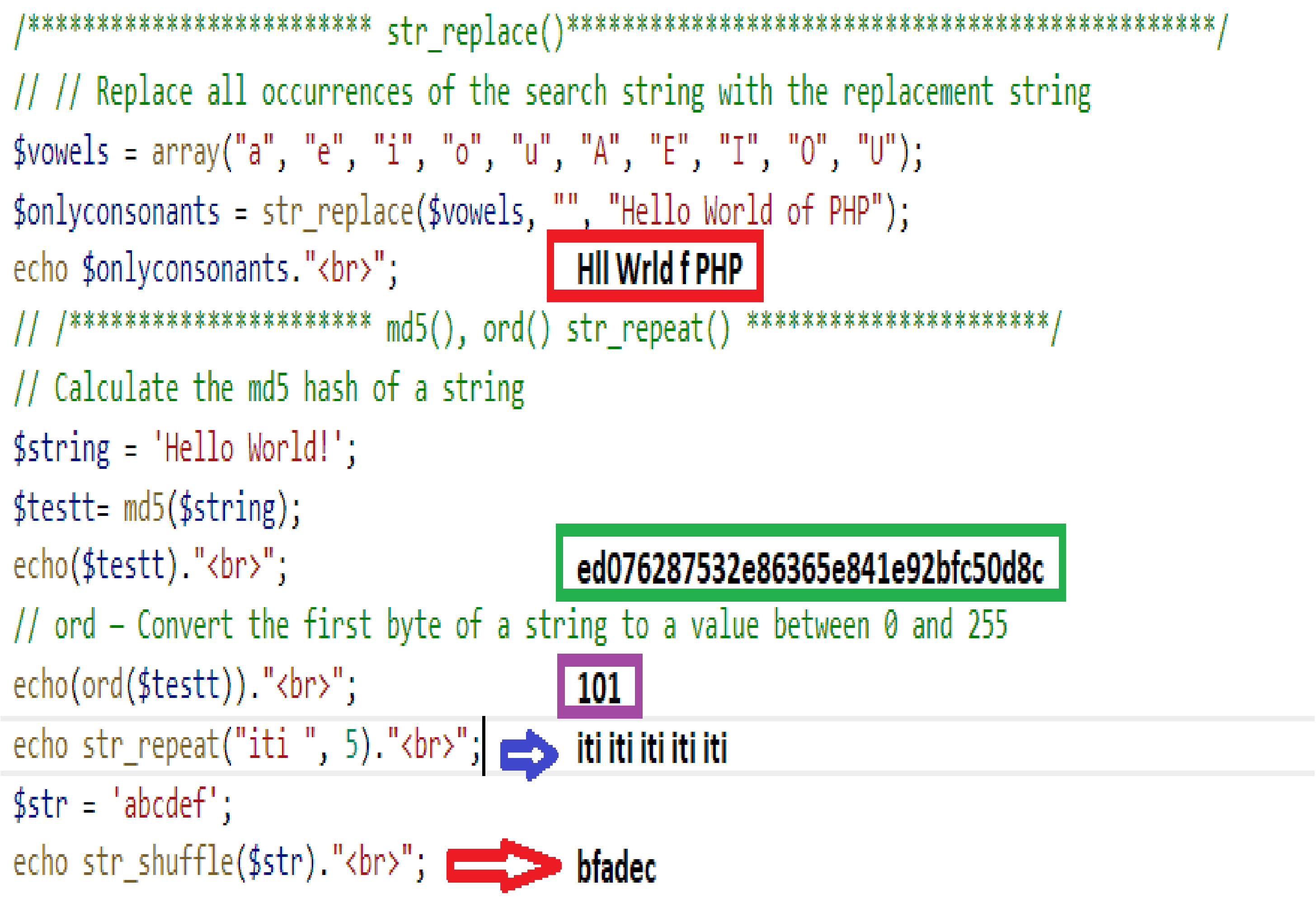
- **Strcmp() , strcasecmp(), strlen(), strstr() strchr() stristr() strpos()**



```
/**************************strcmp , strcasecmp, strlen *******************************************
/// string is treated as an array
$str="test"; $str2="Tests ";
echo strcmp($str,$str2);   1
echo "<br>";                          -2
echo strcasecmp($str,$str2)."<br>"; /// compares all the string including case and spaces
echo ($str[0]."<br>");      t
/// str is considered as an array so you can get its length
echo(strlen($str2)); 6
/*************************** strstr()  strchr() stristr() strpos()*********************************
$email  = 'name@example.com';
$domain = strstr($email,'@');
echo $domain."<br>";   @example.com
echo strchr($email,"m")."<br>"; /// alias for strstr()  me@example.com
// stristr() is the Case-insensitive strstr()
$string = 'Hello World!';
echo stristr($string, 'earth')."<br>"; // returns false
echo stristr($string, 'world')."<br>";  World!
/// strpos
echo strpos ($string, "W");  // strpos is casesensetive  6
// you can set the offest you should start from.
```

String Manipulation cont.

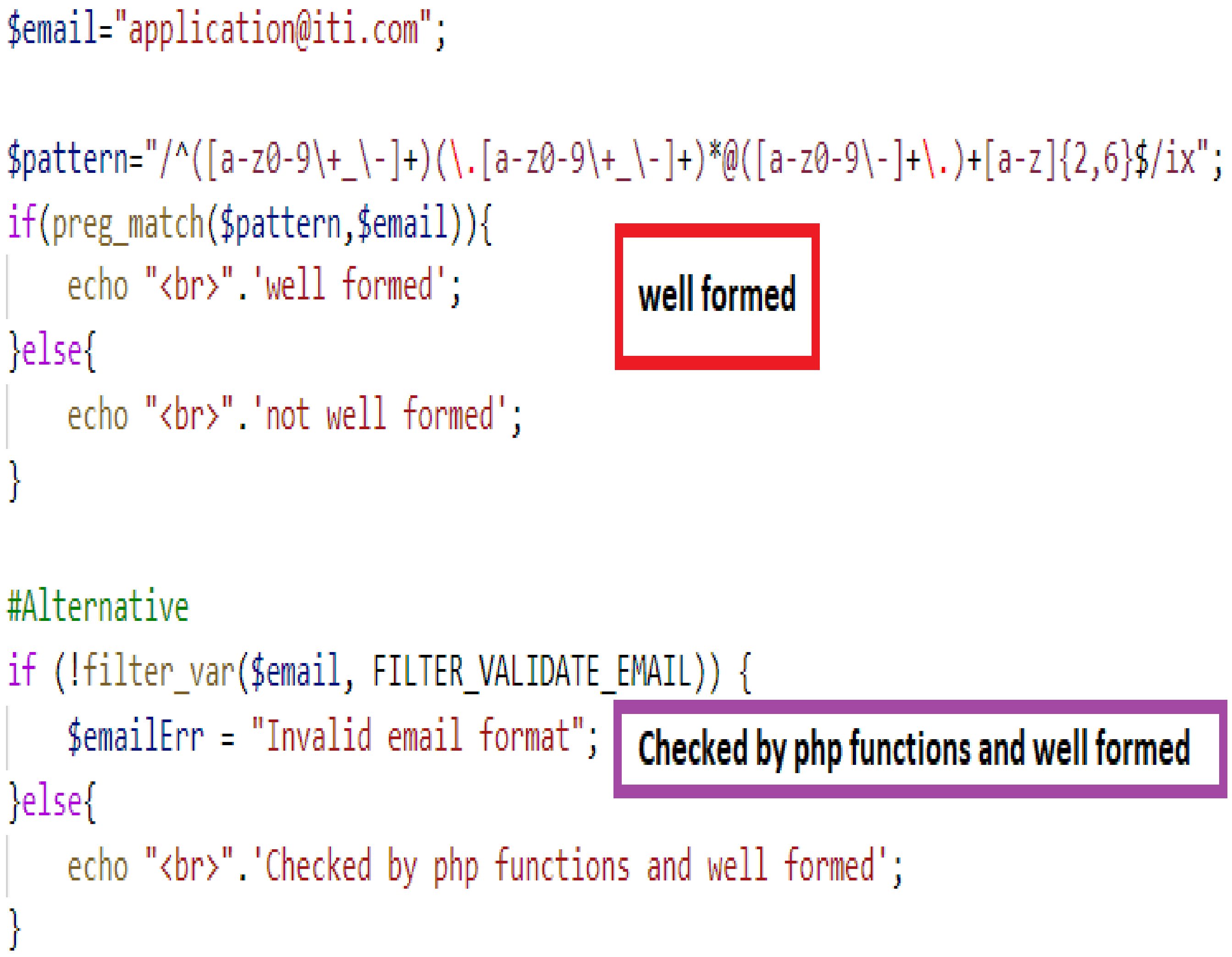
- **str_replace(), md5(), ord() str_repeat()**



```
/************************* str_replace()***********************************************/
// // Replace all occurrences of the search string with the replacement string
$vowels = array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U");
$onlyconsonants = str_replace($vowels, "", "Hello World of PHP");
echo $onlyconsonants."<br>";                 Hll Wrld f PHP
// /********************** md5(), ord() str_repeat() **********************/
// Calculate the md5 hash of a string
$string = 'Hello World!';
$testt= md5($string);
echo($testt)."<br>";                         ed076287532e86365e841e92bfc50d8c
// ord - Convert the first byte of a string to a value between 0 and 255
echo(ord($testt))."<br>";                    101
echo str_repeat("iti ", 5)."<br>";    ⇨     iti iti iti iti iti
$str = 'abcdef';
echo str_shuffle($str)."<br>";      ⇨      bfadec
```

Regular expression

- To validate strings, like emails, passwords, user_names and others.



```
$email="application@iti.com";

$pattern="/^([a-z0-9\+_\-]+)(\.[a-z0-9\+_\-]+)*@([a-z0-9\-]+\.)+[a-z]{2,6}$/ix";
if(preg_match($pattern,$email)){
    echo "<br>".'well formed';
}else{
    echo "<br>".'not well formed';
}

#Alternative
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $emailErr = "Invalid email format";
}else{
    echo "<br>".'Checked by php functions and well formed';
}
```

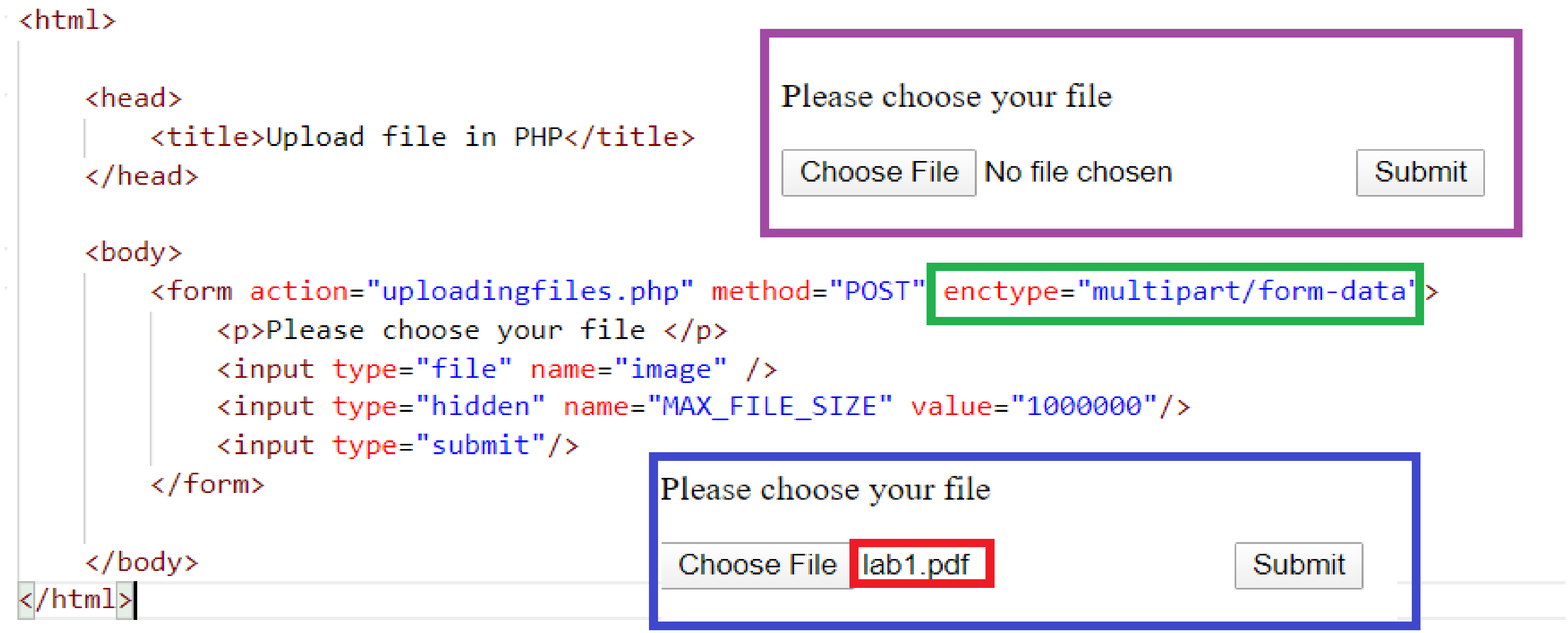
```
well formed
```

```
Checked by php functions and well formed
```

Uploading files

- First ensure that you add the enctype property to the form tag in the html file
- Add input element with type file to browse the files on your machine
- File info is stored in $_FILES super global array
- In the $_FILES array you can find all the information related the file
- Use function called move_uploaded_file() to complete the uploading process

Uploading files

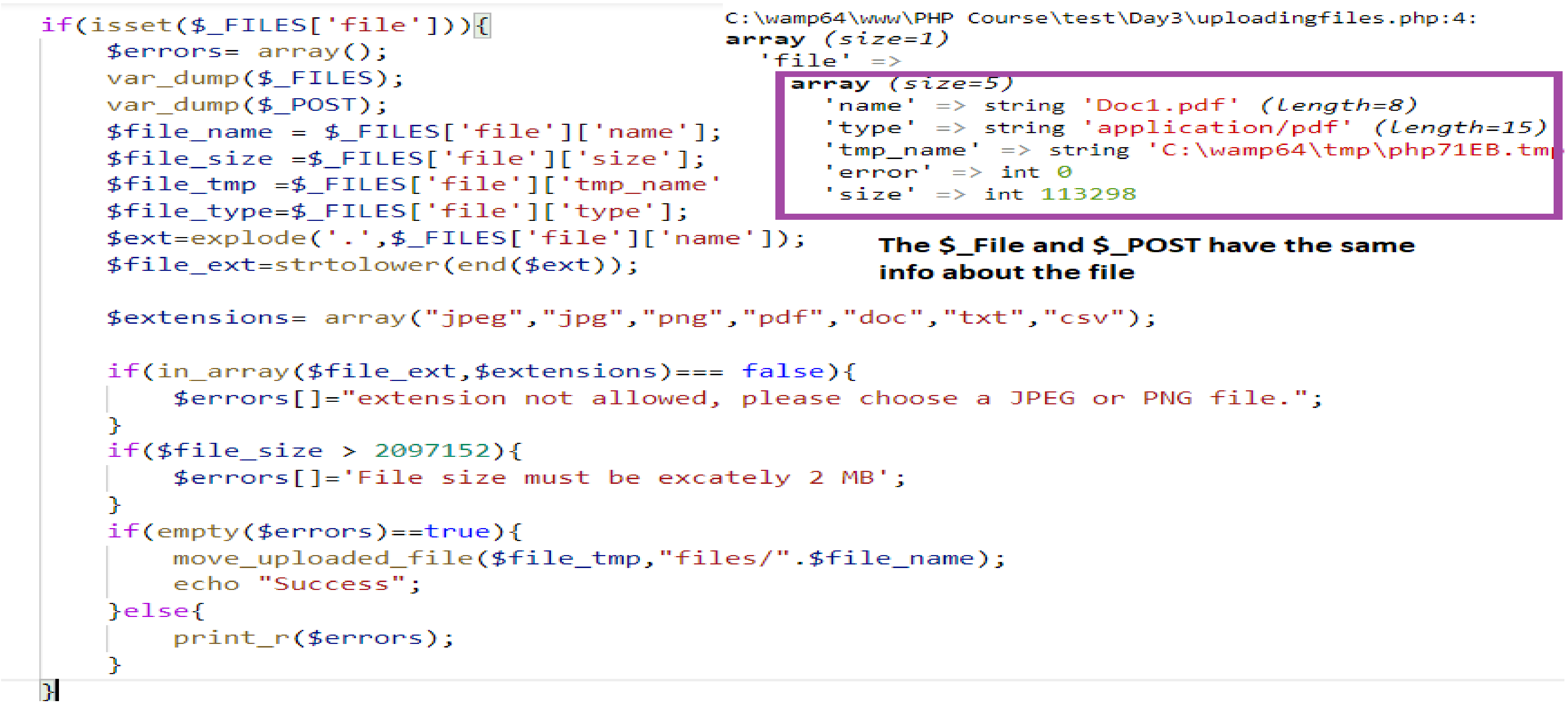


```
<html>

    <head>
        <title>Upload file in PHP</title>
    </head>

    <body>
        <form action="uploadingfiles.php" method="POST" enctype="multipart/form-data">
            <p>Please choose your file </p>
            <input type="file" name="image" />
            <input type="hidden" name="MAX_FILE_SIZE" value="1000000"/>
            <input type="submit"/>
        </form>

    </body>
</html>
```

Please choose your file

Please choose your file

PHP part:



```php
if(isset($_FILES['file'])){
    $errors= array();
    var_dump($_FILES);
    var_dump($_POST);
    $file_name = $_FILES['file']['name'];
    $file_size =$_FILES['file']['size'];
    $file_tmp =$_FILES['file']['tmp_name'
    $file_type=$_FILES['file']['type'];
    $ext=explode('.',$_FILES['file']['name']);
    $file_ext=strtolower(end($ext));

    $extensions= array("jpeg","jpg","png","pdf","doc","txt","csv");

    if(in_array($file_ext,$extensions)=== false){
        $errors[]="extension not allowed, please choose a JPEG or PNG file.";
    }
    if($file_size > 2097152){
        $errors[]='File size must be excately 2 MB';
    }
    if(empty($errors)==true){
        move_uploaded_file($file_tmp,"files/".$file_name);
        echo "Success";
    }else{
        print_r($errors);
    }
}
```

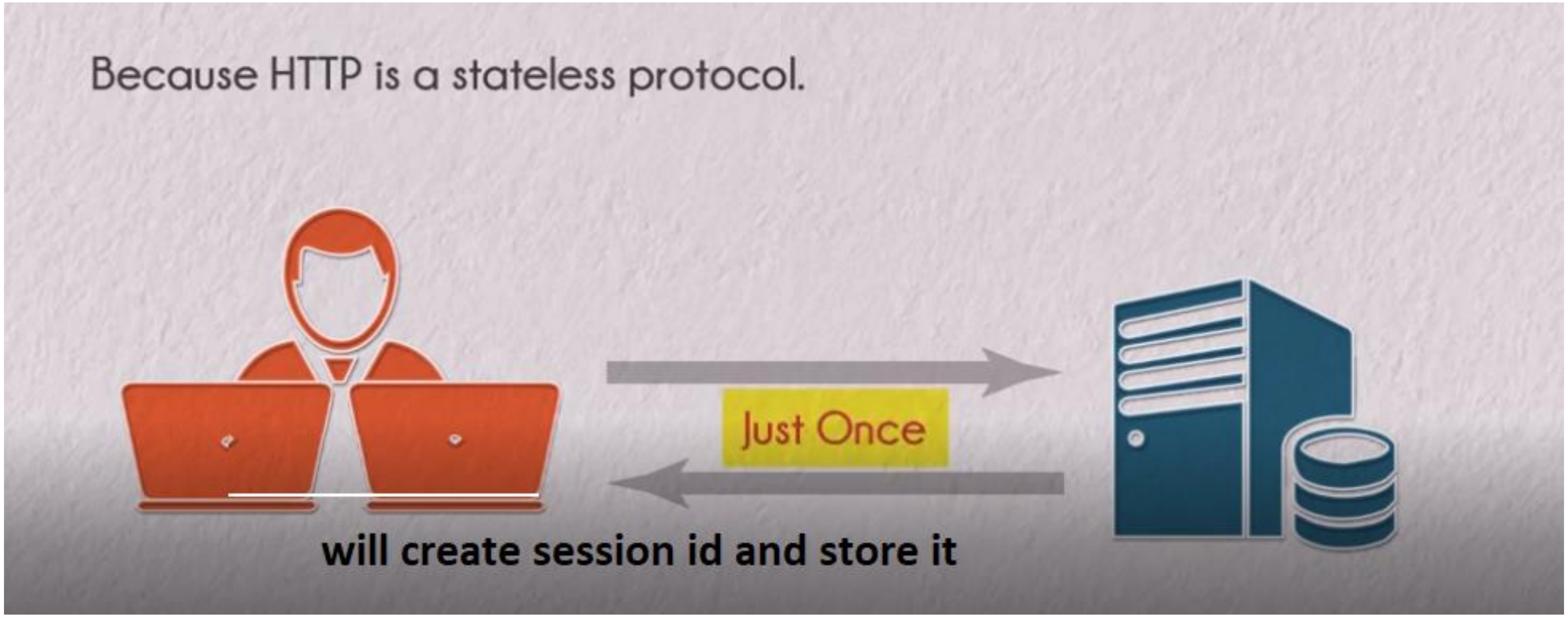
```
C:\wamp64\www\PHP Course\test\Day3\uploadingfiles.php:4:
array (size=1)
  'file' =>
    array (size=5)
      'name' => string 'Doc1.pdf' (length=8)
      'type' => string 'application/pdf' (length=15)
      'tmp_name' => string 'C:\wamp64\tmp\php71EB.tm
      'error' => int 0
      'size' => int 113298
```

The $_File and $_POST have the same info about the file

Session and cookies

- Session is a place on server to store info about user, and cookie is a place on the user machine that can hold info to identify the user
- Any sensitive info cannot be saved on the cookie - saved on the server to be more secure-
- HTTP is a stateless protocol you cannot track the working inside
- The idea of session control is to be able to track a user during a single session on a website. If you can do this, you can easily
- Support logging in a user and showing content according to her authorization level or personal preferences.
- Session is represented by session_id and stored on the server for the lifetime of the session if you didn’t specify the life time when creating session
- Session_id can be stored it in a cookie.

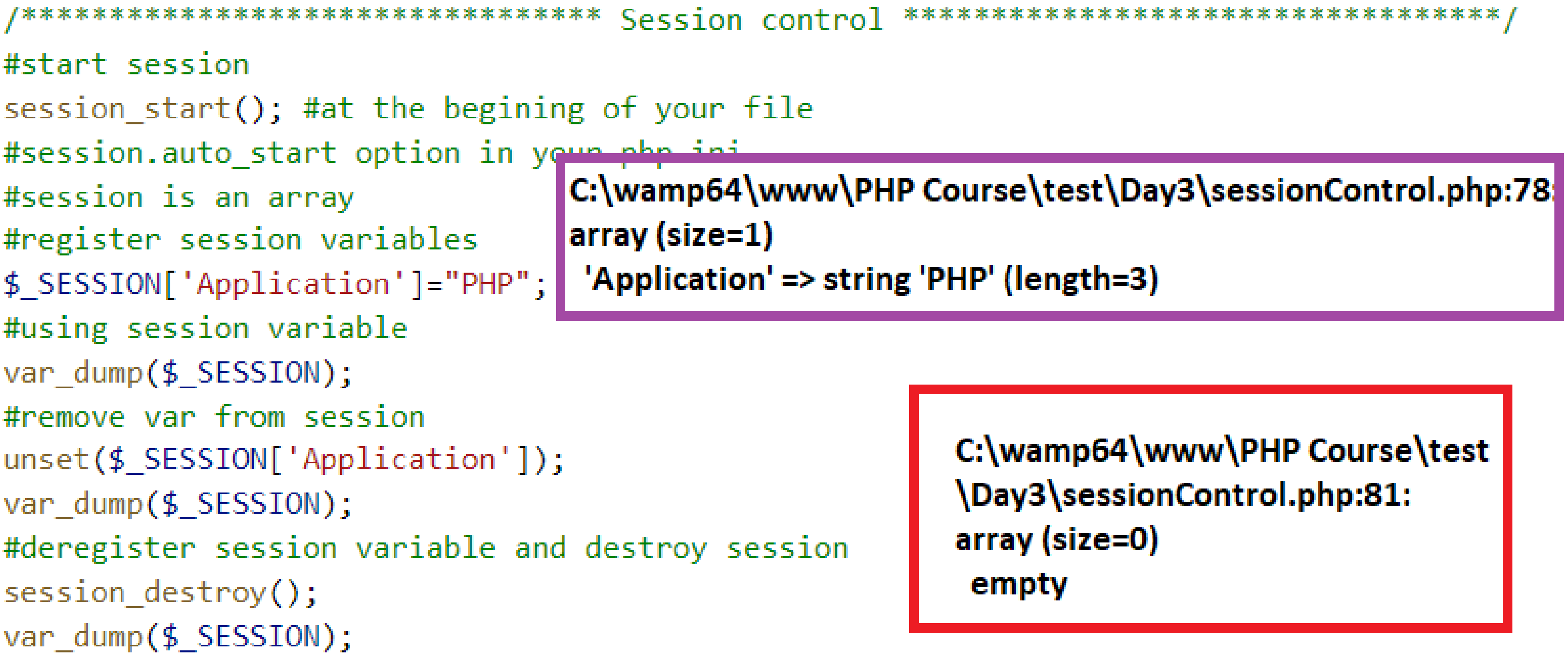
Session



Because HTTP is a stateless protocol.

will create session id and store it

Session starting syntax



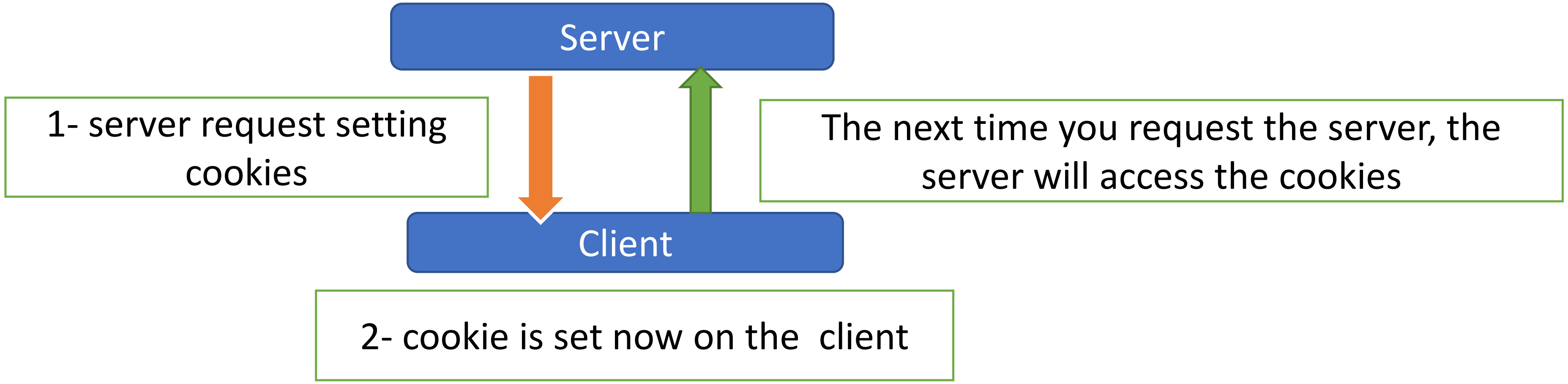
```
/********************************** Session control **********************************/
#start session
session_start(); #at the begining of your file
#session.auto_start option in your php.ini
#session is an array
#register session variables
$_SESSION['Application']="PHP";
#using session variable
var_dump($_SESSION);
#remove var from session
unset($_SESSION['Application']);
var_dump($_SESSION);
#deregister session variable and destroy session
session_destroy();
var_dump($_SESSION);
```

```
C:\wamp64\www\PHP Course\test\Day3\sessionControl.php:78
array (size=1)
  'Application' => string 'PHP' (length=3)
```

```
C:\wamp64\www\PHP Course\test
\Day3\sessionControl.php:81:
array (size=0)
  empty
```

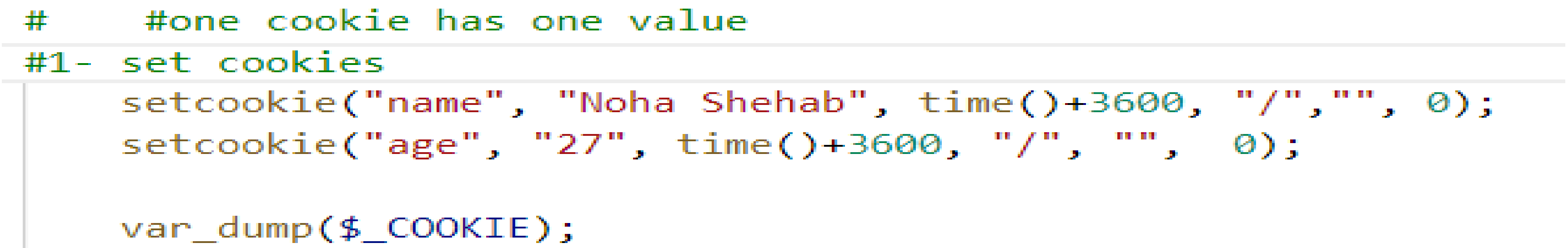
Cookie life cycle

- There are no cookies when the browser connects to server for first time
- When the request is made to the PHP script, the script makes a call to the setcookie() function
- This causes a Set-Cookie HTTP header to be sent in the response that contains the name and value of the cookie to be set.

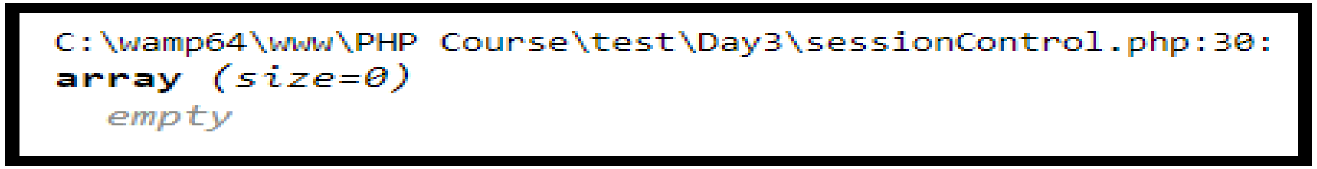


Server
1- server request setting cookies
The next time you request the server, the server will access the cookies
Client
2- cookie is set now on the client

Cookies

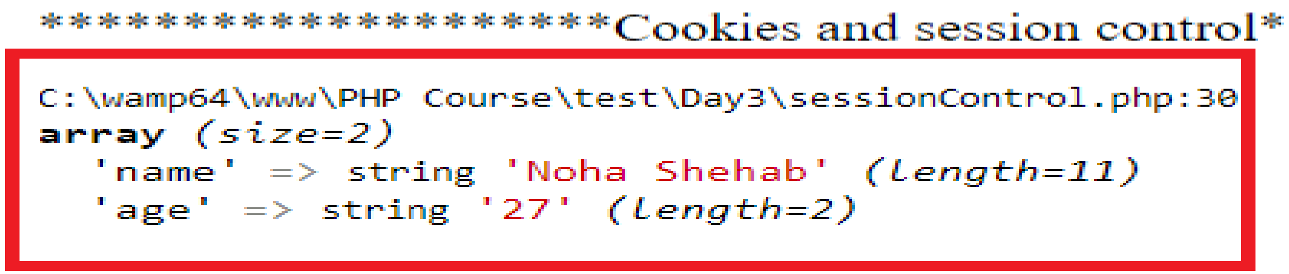


```
#     #one cookie has one value
#1- set cookies
    setcookie("name", "Noha Shehab", time()+3600, "/","", 0);
    setcookie("age", "27", time()+3600, "/", "",  0);

    var_dump($_COOKIE);
```



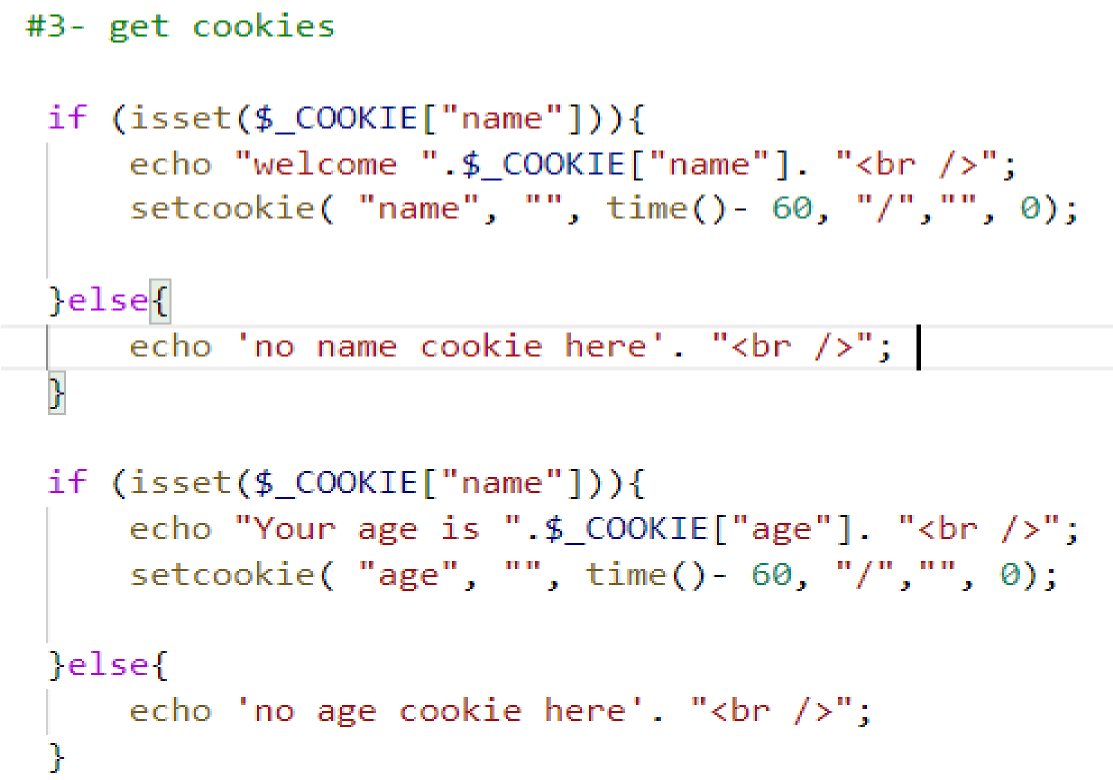
```
C:\wamp64\www\PHP Course\test\Day3\sessionControl.php:30:
array (size=0)
  empty
```

After you refresh the page you can see them

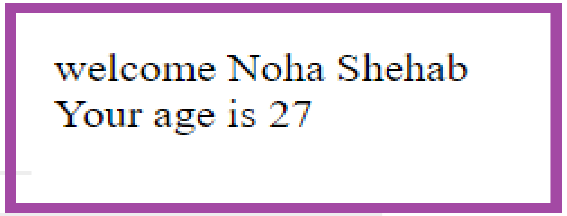


```
**********************Cookies and session control*
C:\wamp64\www\PHP Course\test\Day3\sessionControl.php:30
array (size=2)
  'name' => string 'Noha Shehab' (length=11)
  'age' => string '27' (length=2)
```

Cookie continue

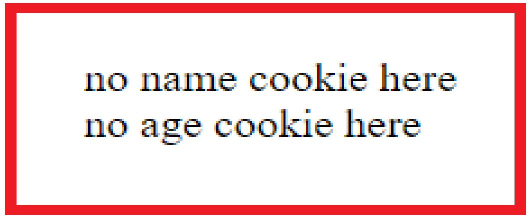


```
#3- get cookies

 if (isset($_COOKIE["name"])){
     echo "welcome ".$_COOKIE["name"]. "<br />";
     setcookie( "name", "", time()- 60, "/","", 0);

 }else{
     echo 'no name cookie here'. "<br />";
 }

 if (isset($_COOKIE["name"])){
     echo "Your age is ".$_COOKIE["age"]. "<br />";
     setcookie( "age", "", time()- 60, "/","", 0);

 }else{
     echo 'no age cookie here'. "<br />";
 }
```



```
welcome Noha Shehab
Your age is 27
```

After refreshing



```
no name cookie here
no age cookie here
```

Thank you ^^