

PHP

UNIT - 3

Handling Form, Session
Tracking & PHP
Components & AJAX

SESSION

- Sessions is the temporary storage are where the information to be stored. The sessions are used when there is any kind of login.
- Where there is login there is session. Once the user gets login the username of the user is developed on each page until the user gets logout.
- So the entire session depends on the user.



○ To create session follow steps are shown below

○ 1. Start session.

○ 2. Set the value of the session.

- To start session Using `session_start()` .

- To set session value Using global variable `$_SESSION ['session_name'] =value.`

- Next to display session value there are two steps :

○ 3. Start session.

○ 4. Display using the echo function.



CONN..

- **To destroy session there are 3 steps to be used**
- 1. Start session.
- 2. Clear the session value.
- Destroy the session.
 - To clear session Using `unset($_SESSION['sessionname'])`.
 - To destroy session using `session_destroy()`.



○ Example :

(1) login.php

```
<?php
    $user="admin";
    $pass="ad123";
    if($user=="admin" and $pass=="ad123")
    {
        session_start();
        $_SESSION['username']='admin';
        header("next.php");
    }
?>
```



○ Example :

(1) next.php

```
<?php
    session_start();
    echo $_SESSION['username'].
    "<a href= 'logout.php'>Logout</a>";
?>
```




○ Example :

(1) logout.php

```
<?php
    session_start();
    unset($_SESSION['username']);
    session_destroy();
?>
```



COOKIE

- Cookie are the temporary storage area where the information is to be stored. The cookie are seen in the address bar of the browser.
 - The cookies which are created are present in the local machine and are located in c:\documents and settings\admin\cookies.
 - The cookies depend on the time. To destroy the cookie it has to be done manually by deleting it form the given above location.
- 

CONN..

- To create cookie the setcookie().
- To display the cookie value using \$_COOKIE ['cookie name'].

- Example:

```
<?php
```


```
    setcookie("tempCookie", "cookie will expire in  
    1 minut", time()+60);
```

```
    echo $_SESSION["tempCookie"];
```

```
?>
```



SERVER VARIABLE

- PHP provides various predefined variables.
 - PHP provides set of predefined arrays containing variables from the web server the web server environment, and user input.
 - Such new arrays are called as superglobal variables.
 - Below are super global variables which are automatically variable in every scope.
- 

Variable

Description

`$GLOBALS`

Contains a reference to every variable which is currently available within the global scope of the script.

`$_SERVER`

This is an array containing information such as header, path etc.

`$_GET`

An associative array of variables passed to the current script via the HTTP GET method.



Variable

Description

`$_POST`

An associative array of variable passed to the current script via the HTTP POST method.

`$_FILES`

An associative array of items uploaded to the current script via the HTTP POST method.

`$_REQUEST`

An associative array consisting of the contents of `$_GET`, `$_POST`, and `$_COOKIE`.

Variable

Description


\$_COOKIE	An associative array of variables passed to the current script via HTTP cookies.
\$_SESSION	An associative array containing session variables available to the current script.





PHP Components

PHP GRAPHICAL DISPLAY

- PHP provides a feature to draw any images using various kinds of shapes. The area where the shapes are to be drawn is known as **canvas**.
- There are different versions of Graphical display in PHP :
 - Gd-1.6 which supports GIF images and not PNG.
 - Ge-2.0.28 support all kind of images including PNG.
 - Generally, there are **steps** for the graphics image to be used in PHP. 

o 1) Create Canvas :

imagecreate() :

- o it is first step to draw image. This function is used for creating the canvas.giv height and width.

```
<?php
```

```
    $canvas=imagecreate(500,500);
```

```
?>
```



o2) Allocate background color for canvas.

imagecolorAllocate() :

it is next step for drawing the image on the canvas. This function is used for allocating the color to background of canvas and the shapes which are drawn.

```
<?php
```

```
    $canvas=imagecreate(500,500);
```

```
    $backcolor=imagecolorallocate($canvas,0,0,0);
```


```
?>
```

o3) Allocate background color for shapes.

Imageline() :

This function is used to draw line on the canvas.

```
<?php  
$canvas=imagecreate(500,500);  
$backcolor=imagecolorallocate($canvas,0,0,0);  
$linecolor=imagecolorallocate($canvas,1,1,1);  
imageline();  
?>
```



o4) Draw shapes on the canvas.

l) `imagefillrectangle()` or `imagerectangle()` :

`imagefilledrectangle()` function is used to draw a rectangle with the color filled in it.

```
<?php
```

```
    $canvas=imagecreate(500,500);
```

```
    $backcolor=imagecolorallocate($canvas,0,  
0,0);
```

```
    $color=imagecolorallocate($canvas,1,1,1);
```

```
    imagefilledrectangle($canvas,200,200,  
250,250,$color); ?>
```

||) **imageellipse()** or **imagefilledellipse()** :

imageellipse() is used to draw simple ellipse.

imagefilledellipse() is used to draw an ellipse with color filled in it.

```
<?php
```

```
$canvas=imagecreate(500,500);
```

```
$backcolor=imagecolorallocate($canvas,0,0,0);
```

```
$color=imagecolorallocate($canvas,1,1,1);
```

```
imagefilledellipse($canvas,200,200,10,10,$color);
```

```
?>
```

||) imagepolygon() or imagefilledpolygon() :

Polygon means a shape having starting and ending point as the same.

imagepolygon() fun. Is used to draw simple polygon.

imagefilledpolygon() is used to draw poly. With color filled in it.



```
<?php
```

```
$canvas=imagecreate(500,500);
```

```
$backcolor=imagecolorallocate($canvas,  
0,0,0);
```

```
$color=imagecolorallocate($canvas,1,1,1);
```

```
$array=array(100,100,150,110,50,110,  
100,100);
```

```
imagefilledellipse($canvas,$array,3,$color);
```

```
?>
```



|V) imagearc() :

This function is used to draw an arc. It has 8 parameter into it.

```
<?php
```

```
    $canvas=imagecreate(500,500);
```

```
    $backcolor=imagecolorallocate($canvas,  
0,0,0);
```

```
    $color=imagecolorallocate($canvas,1,1,1);
```

```
    imagearc($canvas,200,200,15,5,0,  
180,$color);
```

```
?>
```



o5) Place image on web-page Function

l) `imagepng()` or `imagejpeg()` or `imagegif()` :

The last step is to place this canvas on the web-page. This image can be in any format that is either PNG, JPEG or GIF.

depending on the user the image are created.

If **PNG** (**Portable network graphics**) image is to be created, then `imagepng()` is used.

If **JPEG**(**joint photographic experts group**) image is to be created, then `imagejpeg()` is used.

If **GIF (Graphics Interchange Format)** image is to be created, then `imagegif()` is used.

All these kinds of functions take a single variable.

```
<?php
    $canvas=imagecreate(500,500);
    $backcolor=imagecolorallocate($canvas,
    0,0,0);
    $color=imagecolorallocate($canvas,1,1,1);
    imagearc($canvas,200,200,15,5,0,
    180,$color);
```

```
?>
```



PHP REGULAR EXPRESSION

- Regular expression means the formula or the general form of any string, or expression.
- It is used for pattern matching.
- It checks any characters or digits which are specified in the format. It is used generally for the validation on server-side scripting.



○ Advantage :

- It is easy to use and learn.
- It is a portable lang. as it has general format.
- It prevents data to be entered in invalid format.

○ Disadvantage :

- The validation which is provided using server side scripting is disadvantageous, as web-page is refreshed again and again.



○ **Types of Regular Expression :**

○ There are **two** types of regular expression in PHP:

○ [1] POSIX extended Regular Expression.

○ [2] Perl Compatible Regular Expression.

• **1] POSIX Extended Regular Expression :**

○ When this type of regular expression is used then the `ereg()` is required to be used for the pattern matching.

- There are Different symbols used which are given below :

Symbol	Description
^	For Starting format string
\$	For ending format string
[a-z]	Only lowercase character
[A-Z]	Only uppercase character
[a-z A-Z]	Only alphabets
[0-9]	Only digit
[a-z A-Z 0-9]	Alpha numeric character
\	Use when any special character are used in string
()	Used when more than one value to be checked
 	Use for condition
{n,m}	N for minimum value and m for maximum value

○ Above shown are the general symbols the other symbols used in the format are as shown below :

○ * => one or more times

○ + => zero or more times

○ ? => zero or one time

○ When the format which is matched is not found, then it return as false otherwise it would have returned as true value.

○ Syntax for `ereg` function is given below :

• `Boolean ereg('format', variable or value to be checked);`

- **For example :**

```
<?php
```

```
    $date="04/09/2010";
```

```
    if(ereg('^[0-9]{2}V[0-9]{2}V[0-9]{4}$', $date))
```

```
        echo "it is valid date";
```


```
    else
```

```
        echo "it is not valid date";
```

```
?>
```



- **2] Perl Compatible Regular Expression :**

- When this type of regular expression is developed using the Perl language.
 - The function which is used for the given regular expression `preg_match()` that is perl regular expression.
 - It also return the boolean value. If format is matched, then it return true value otherwise will return false value.
- 

- There are different symbols used which are given as below :

Symbol	Description
\D	For anything for number
\d	Any number
\S	Anything other than whitespace
\s	Any kind of whitespace
\W	Any word character (including special symbols)
\w	Any thing other than word character
\A	Beginning of string
\b	Word boundary
\B	Not word boundary
\z	End of string(Match before final new line or at end of string)
\Z	End of string (Match only at end of string)

- Syntax :
 - Boolean `preg_match`('format', Variable or value to be checked)

- Example :

```
<?php
```

```
    $date="04/09/2010";
```

```
    if(preg_match('^\\d{2}\\\\d{2}\\\\d{4}$',$date))
```

```
        echo "it is valid date";
```

```
    else
```

```
        echo "it is not valid date";
```

```
?>
```



FILE UPLOADING AND SENDING MAIL

○ 1] File Uploading :

- Sometimes it is required that files are to be the server. For this first step is to create HTML form as here :

- **HTML Form :**

```
<html>
```

```
<body>
```

```
<form action="uploading_file.php" method=  
"post" enctype="multipart/form-data">
```



```
<label for="file">File name</label>
```

```
<input type="file" name="file" id="file"><br>
```

```
<input type="submit" name="submit" value="Submit">
```

```
</form>
```

```
</body>
```

```
</html>
```



- **PHP Script:**

```
<?php
if($FILES["file"]["error"] > 0)
{
echo "Error:" . $FILES["file"]["error"]. "<br>";
}
else
{
echo "Upload:" . $FILES["file"]["name"]. "<br>";
echo "Type :" . $FILES["file"]["type"]. "<br>";
echo "Stored in:" . $FILES["file"]["tmp_name"].
    "<br>";
} ?>
```

○ 1] File Uploading with proper extension :

```
<?php
```

```
$allowedExts=array("gif","jpg","png");
```

```
$extension=end(explode(".",$_FILES  
["file"]["name"]));
```

```
if(((($_FILES['file']['type']==="image/gif");
```

```
|| ($_FILES['file']['type']==="image/jpeg")
```

```
|| ($_FILES['file']['type']==="image/jpg")
```

```
|| ($_FILES['file']['type']==="image/png"))
```

```
&& ($_FILES["file"]["size"] < 20000)
```

```
&& in_array($extension, $allowedExts))
```


```
{
```



```
If($_FILES["file"]["error"] > 0)
{
    echo "Return Code: ".$_FILES["file"]["error"].
    "<br>";
}
else
{
    If(file_exists("images/" . $_FILES["file"]["name"]))
    {
        Echo $_FILES["file"]["name"]. "already exists. ";
    }
}
```



```
else
{
    move_uploaded_file($_FILES["file"]["tmp_name"], "images/"). $_FILES["file"]["name"];
    echo "Uploading";
}
}
}
else
{
    Echo "Invalid file";
}
?>
```



SENDING MAIL USING MAIL()

- To activate the mail() for PHP it must be configured correctly in the php.ini file.
- Open php.ini file available in /etc/directory and find the section headed [mail function].
- Simply need to let PHP know the location of their send mail application. The path and any desired switches should be specified to the sendmail_path directives.



- PHP makes use of mail() to send email. This function requires three mandatory parameters that specify the recipient's email address, the subject of the message and the actual message additionally there are other two optional parameters.


- Syntax :

- `mail(to,subject,message,header,parameters);`

- Here, Message parameter define message to be sent. Header is optional parameter like Cc and Bcc.

○ Sending simple text email :

```
<?php
    $to="email";
    $subject="this is subject";
    $message="this is simple text message";
    $header="From:email \r\n";
    $retval= mail ($to,$subject,$message,$header);
    if($retval == true) {
        echo "Message sent successfully"; }
    else {
        echo "Message could not sent successfully"; }
?>
```



○ Sending HTML email :

```
<?php
```

```
$to="email";
```

```
$subject="this is subject";
```

```
$message="<b>this is simple text message</b>";
```

```
$message="<h1>this is Headline</h1>";
```

```
$header="From:email \r\n";
```

```
$header="Cc;email2 \r\n";
```

```
$header="MIME- Version: 1.0 \r\n";
```

```
$header="Content-type: text/html\r\n";
```



```
$retval= mail ($to,$subject,$message,$header);
```

```
if($retval == true)
```

```
{
```

```
    echo "Message sent successfully";
```

```
}
```

```
else
```

```
{
```

```
    echo "Message could not sent successfully";
```

```
}
```

```
?>
```



SENDING MAIL USING SMTP()

- Go to the link below to get the class file for configuring smtp :

<https://github.com/Synchro/PHPMailer>

- Now from three main class file are required to download :
 - (1) class.phpmailer.php
 - (2) class.pop3.php
 - (3) class.smtp.php
 - Place all this file same folder and in that folder create a new file with code below to send mail using smtp().