Bharathidasan University

Centre for Differently Abled Persons
Tiruchirappalli - 620024.

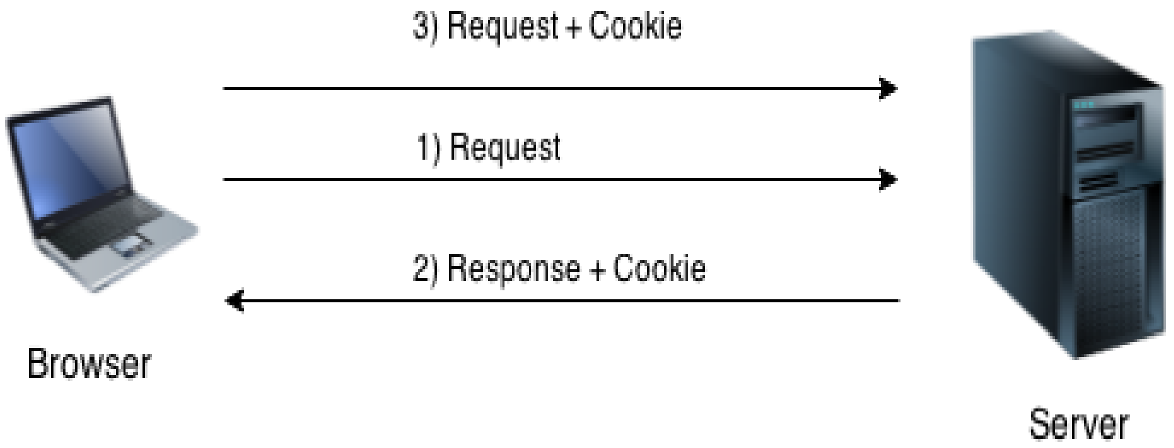
- Programme Name: Bachelor of Computer Applications
- Course Code : 20UCA6CC9
- Course Title : PHP
- Unit : Unit IV
- Compiled by : Dr. M. Prabavathy
  Associate Professor

  Ms. M. Hemalatha
  Guest Faculty

COOKIES

What are cookies?

- Cookies are text files stored on client computer
- Cookies are often used to identify user
- Cookies are created at server side and saved to client browser.
- Each time when client sends request to the server, cookie is embedded with request



SETTING COOKIES

- PHP **setcookie()** function is used to set cookie with HTTP response.
- Once cookie is set, you can access it by $_COOKIE superglobal variable.

Syntax

setcookies(name, value, expire, path, domain, secure, httponly);

- only name parameter is required
- others are optional.

```
<?php
    // Setting a cookie
    setcookie("username", "John Carter",
time()+30*24*60*60);
?>
```

Cookies will expire after 30 days

time()+30*24*60*60 →30 days

24 hours

60 minutes

60 second

USING COOKIE VARIABLES

- **$_cookie** is a superglobal variables

- It is used to retrieve a cookies value

```
<?php
// Accessing an individual cookie value
echo $_COOKIE["username"];
?>
```

- It prints the username

Sample code

```
<?php
// Setting a cookie
setcookie("username", "Abdul Kalam", time()+30*24*60*60);
echo $_COOKIE["username"];
?>
```

Output

```
Abdul Kalam
```

SESSION

- A session is a way to store information to be used across multiple page
- Information are not stored on user computer
- By default, session variables last until the user closes the browser.
- Session variables hold information about one single user, and are available to all pages in one application.

A session is started with the session_start( ) function

```
<?php
// Start the session
session_start();
?>
```

UNDERSTANDING SESSION VARIABLE

- $_SESSION super-global array is initialized with corresponding session information
- By default, it’s initialized with a blank array.
- It can store more information by using a key-value pair.

Sample code

```
<?php
// start a session
session_start();
```

```
// initialize session variables
    $_SESSION['logged_in_user_id'] = '1';
    $_SESSION['logged_in_user_name'] = 'Tutsplus';

// access session variables
    echo $_SESSION['logged_in_user_id'];
    echo $_SESSION['logged_in_user_name'];
?>
```

MANAGINIG USER PREFERENCES WITH SESSION

- Session can be managed using user preferences
- Adding user preference into session
- Now using session and control statement, can manage user preference

1.Open a notepad, start a php block and call the session_start() function

```
<?
session_start();
```

2. If…else used to check condition of font family and font size. If value is not in current session set a default value

```
if ((!$_SESSION[font_family]) || (!$_SESSION[font_size]))
{
    $font_family = "sans-serif";
    $font_size = "10";              }    Default value
    $_SESSION[font_family] = $font_family;
    $_SESSION[font_size] = $font_size;
}
```

3. Return the default value if value is not in current session

```
else
    {
        $font_family = $_SESSION[font_family];
        $font_size = $_SESSION[font_size];
    }
```

4. close the php block and start with html

```
?>
<HTML>
<HEAD>
<TITLE>My Display Preferences</TITLE>
```

5. Add a stylesheet block and add styles to tags

6. Close the style tag and use POST for form method

7.Create set of radio buttons for font family and font size

8. Add the submit button. Close the form and html tags. Save the file

ENTIRE CODE

```
<?
//start a session
session_start();
//check for stored values and register defaults
if ((!$_SESSION[font_family]) ||
(!$_SESSION[font_size])) {
    $font_family = "sans-serif";
    $font_size = "10";
    $_SESSION[font_family] = $font_family;
    $_SESSION[font_size] = $font_size;
}
```

```
else {
    //extract from $_SESSION superglobal if exist
    $font_family = $_SESSION[font_family];
    $font_size = $_SESSION[font_size];}
?>
```

```
<HTML>
<HEAD>
<TITLE>My Display Preferences</TITLE>
<STYLE type="text/css">
BODY, P, A {font-family:<? echo "$font_family"; ?>;
font-size:<? echo "$font_size"; ?>pt;font-weight:normal;}
H1 {font-family:<? echo "$font_family"; ?>;
font-size:<? echo $font_size + 4; ?>pt;font-weight:bold;}
</STYLE>
</HEAD>
</HEAD>
```

ADDING STYLES TO TAGS

```
<BODY>
<H1>Set Your Display Preferences</H1>
<FORM METHOD="POST" ACTION="session02.php">
// RADIO BUTTON
<P>Pick a Font Family:<br>
<input type="radio" name="sel_font_family" value="serif"> serif
<input type="radio" name="sel_font_family" value="sans-serif"
checked> sans-serif
<input type="radio" name="sel_font_family" value="Courier">
Courier
<input type="radio" name="sel_font_family" value="Wingdings">
Wingdings</p>
```

Form method

// RADIO BUTTON

```
<P>Pick a Base Font Size:<br>
<input type="radio" name="sel_font_size" value="8"> 8pt
<input type="radio" name="sel_font_size" value="10" checked> 10pt
<input type="radio" name="sel_font_size" value="12"> 12pt
<input type="radio" name="sel_font_size" value="14"> 14pt
</p>
```

//BUTTON

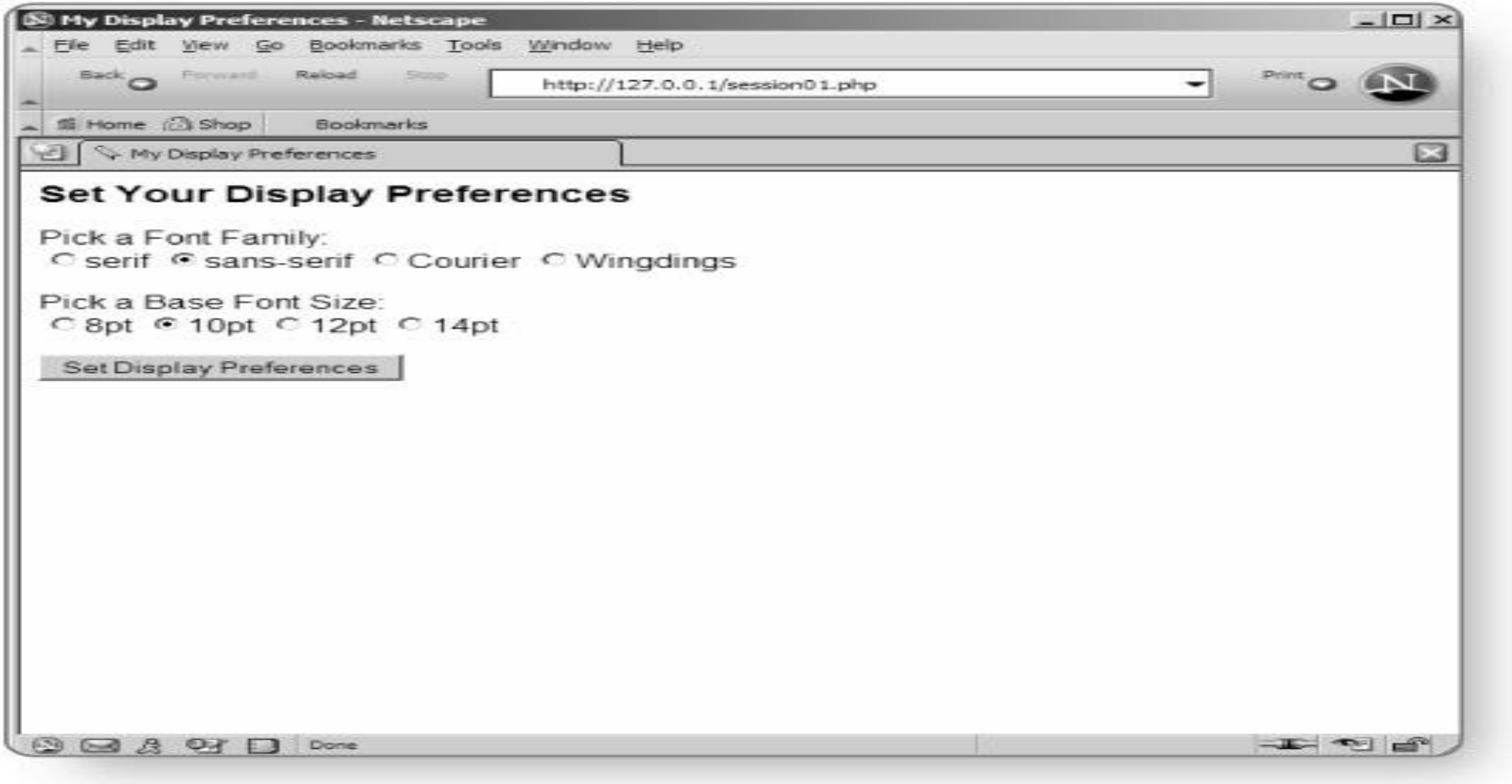
```
<P><input type="submit" name="submit" value="Set Display Preferences"></p>

</FORM>

</BODY>

</HTML>
```

OUTPUT:


My Display Preferences - Netscape
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
Print
Home Shop Bookmarks
My Display Preferences
Done

http://127.0.0.1/session01.php

Set Your Display Preferences

Pick a Font Family:
○ serif ● sans-serif ○ Courier ○ Wingdings

Pick a Base Font Size:
○ 8pt ● 10pt ○ 12pt ○ 14pt

Set Display Preferences

GRAPHICS

GRAPHICS

- Images appear in the form of logos, buttons, photographs, charts, advertisements, and icons
- PHP supports graphics creation with the GD and Imlib2 extensions.
- GD -Graphics Draw
- Imlib2 - a graphics library for fast file loading, saving, rendering and manipulation.

DRAWING FUNCTIONS

1. To draw lines

- **imageline($image, $x1, $y1, $x2, $y2, $color)** function is used to draw a straight line
- $image is a parameter from a function **imagecreatetruecolor()** or **imagecreatefromjpeg()**
- For horizontal line $y1 = $y2
- For vertical line $x1 = $x2

2. To draw arc

- The function **imagearc($image, $cx, $cy, $width, $height, $start, $end, $color)** can draw circular arcs
- $cx and $cy → center
- $width and $height → Size of an arc on different axes
- $start and $end → parameters specify the starting and ending angle of the arc in degrees

3. To draw circle

- The function **imageellipse($image, $cx, $cy, $width, $height, $color)** is used to draw complete arc from 0 to 360 degree

4. To draw rectangle

- The function **imagerectangle($image, $x1, $y1, $x2, $y2, $color)**
- $x1 and $y1 → top-left corner of the rectangle
- $x2 and $y2 →bottom-right corner of the rectangle

5. To draw polygon

- The function **imagepolygon($image, $points, $num_points, $color)** can create a polygon with any number of sides or points.
- **$points** parameter is an array where two elements are paired together

CREATING AND DRAWING IMAGE

TO CREATE AN IMAGE

```
<?php
create_image();
print "<img src=image.png?".date("U").">";
function  create_image(){
        $im = @imagecreate(200, 200) or die("Cannot Initialize new GD
image stream");
        $background_color = imagecolorallocate($im, 255, 255, 0);  //
yellow
        imagepng($im,"image.png");
        imagedestroy($im);  }  ?>
```

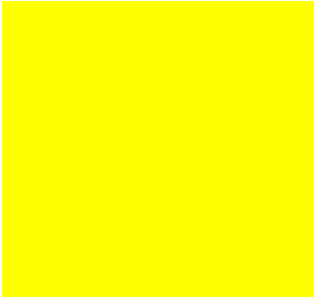
1. **create_image()** – request to execute function
2. **?** – end of an image
3. **date("U")** – avoid to show image in cache of the browser
4. **$im** – variable
5. **@ -** used before image to avoid displaying error
6. **imagecreate(200,200)** – will start the creation process and 200 is width, 200 is height
7. **imagecolorallocate() -** command is used to define colors

imagecolorallocate($im, 255, 255, 0)

- amount of red: 255 (the maximum).
- amount of green: 255 (the maximum).
- amount of blue: 0 (the minimum).

imagepng – will save image in $im

imagedestroy() – to free a memory

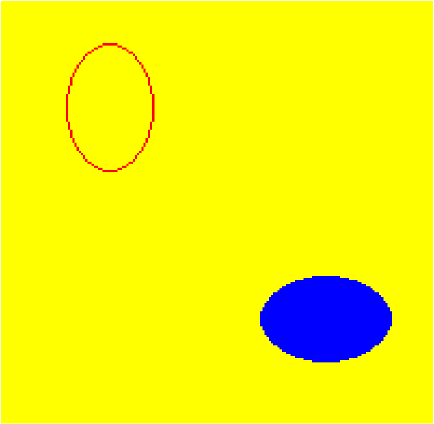


TO DRAW A CIRCLE IN AN IMAGE

```
<?php
create_image();
print "<img src=image.png?".date("U").">";
function  create_image(){
        $im = @imagecreate(200, 200) or die("Cannot Initialize new GD image stream");
        $background_color = imagecolorallocate($im, 255, 255, 0);   // yellow
       $red = imagecolorallocate($im, 255, 0, 0);           // red
        $blue = imagecolorallocate($im, 0, 0, 255);         // blue
        imageellipse($im, 50, 50, 40, 60, $red);
        imagefilledellipse($im, 150, 150, 60, 40, $blue);
imagepng($im,"image.png");
        imagedestroy($im);
}
?>
```

imageellipse($im, 50, 50, 40, 60, $red) – red circle

imagefilledellipse($im, 150, 150, 60, 40, $blue) – circle with blue color filled .



THANK YOU