SHRIMATI INDIRA GANDHI COLLEGE

(NATIONALLY ACCREDITED AT "A" GRADE (3rd CYCLE) BY NAAC)

TIRUCHIRAPPALLI-2

TUTORIAL MATERIAL

PHP & Scripting Languages

DEPARTMENT OF COMPUTER SCIENCE

php
html
sql
cms
seo
web
css
jpg
www
xhtml
ajax
pdf
http
jsp
gif
pda
svg

CORE COURSE IX

PROGRAMMING IN PHP (RCCS10CA8 / MBECS 3:3/10)

SEMESTER VI

Unit I

Essentials of PHP - Operators and Flow Control - Strings and Arrays.

Unit II

Creating Functions - Reading Data in Web Pages - PHP Browser - Handling Power.

Unit III

Object-Oriented Programming –Advanced Object-Oriented Programming .

Unit IV

File Handling –Working with Databases – Sessions, Cookies, and FTP

Unit V

Ajax – Advanced Ajax – Drawing Images on the Server.

Text Book:

1.The PHP Complete Reference, Steven Holzner, McGrawHillEducation, 2007

Reference Books:

1. PHP: A Beginner's Guide, Vikram Vaswani, McGraw Hill Education, 2008

*****

PHP & Scripting Languages

1. What is PHP?

PHP is a powerful tool for making dynamic and interactive Web pages.

PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

Before you continue you should have a basic understanding of the following:

- HTML/XHTML
- JavaScript

2. Expand PHP?

- PHP stands for **P**HP: **H**ypertext **P**reprocessor
- PHP is a server-side scripting language, like ASP
- PHP scripts are executed on the server
- PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)
- PHP is an open source software
- PHP is free to download and use

3. What is a PHP File?

- PHP files can contain text, HTML tags and scripts
- PHP files are returned to the browser as plain HTML
- PHP files have a file extension of ".php", ".php3", or ".phtml"

4. What is MySQL?

- MySQL is a database server
- MySQL is ideal for both small and large applications

- MySQL supports standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use

PHP + MySQL

- PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform)

5. Why PHP?

- PHP runs on different platforms (Windows, Linux, Unix, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP is FREE to download from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

PHP code is executed on the server, and the plain HTML result is sent to the browser.

6. Write Basic Syntax of PHP.

A PHP scripting block always starts with **<?php** and ends with **?>**. A PHP scripting block can be placed anywhere in the document.

On servers with shorthand support enabled you can start a scripting block with <? and end with ?>.

For maximum compatibility, we recommend that you use the standard form (<?php) rather than the shorthand form.

```
<?php
?>
```

A PHP file normally contains HTML tags, just like an HTML file, and some PHP scripting code.

7. Write an example of PHP.

An example of a simple PHP script which sends the text "Hello World" to the browser:

```
<html>
<body>

<?php
echo "Hello World";
?>

</body>
</html>
```

Each code line in PHP must end with a semicolon. The semicolon is a separator and is used to distinguish one set of instructions from another.

There are two basic statements to output text with PHP: **echo** and **print**. In the example above we have used the echo statement to output the text "Hello World".

8. How to put Comments in PHP?

In PHP, we use // to make a single-line comment or /* and */ to make a large comment block.

```
<html>
<body>

<?php
//This is a comment

/*
This is
```

```
a comment
block
*/
?>

</body>
</html>
```

9. Write Naming Rules for Variables.

- A variable name must start with a letter or an underscore "_"
- A variable name can only contain alpha-numeric characters and underscores (a-z, A-Z, 0-9, and _)
- A variable name should not contain spaces. If a variable name is more than one word, it should be separated with an underscore ($my_string), or with capitalization ($myString)

10. How to declare String Variables in PHP?

String variables are used for values that contain characters.

After we create a string we can manipulate it. A string can be used directly in a function or it can be stored in a variable.

Below, the PHP script assigns the text "Hello World" to a string variable called $txt:

```
<?php
$txt="Hello World";
echo $txt;
?>
```

The output of the code above will be:

```
Hello World
```

11. Write use of the Concatenation Operator.

There is only one string operator in PHP.

The concatenation operator (.) is used to put two string values together.

To concatenate two string variables together, use the concatenation operator:

```
<?php
$txt1="Hello World!";
$txt2="What a nice day!";
echo $txt1 . " " . $txt2;
?>
```

The output of the code above will be:

```
Hello World! What a nice day!
```

12. What are the modes available in files?

Modes	**Description**
R	Read only. Starts at the beginning of the file
r+	Read/Write. Starts at the beginning of the file
W	Write only. Opens and clears the contents of file; or creates a new file if it doesn't exist
w+	Read/Write. Opens and clears the contents of file; or creates a new file if it doesn't exist
A	Append. Opens and writes to the end of the file or creates a new file if it doesn't exist
a+	Read/Append. Preserves file content by writing to the end of the file

X	Write only. Creates a new file. Returns FALSE and an error if file already exists
x+	Read/Write. Creates a new file. Returns FALSE and an error if file already exists

13. How to open a file?

Opening a File

The fopen() function is used to open files in PHP.

The first parameter of this function contains the name of the file to be opened and the second parameter specifies in which mode the file should be opened:

```
<html>
<body>

<?php
$file=fopen("welcome.txt","r");
?>

</body>
</html>
```

14. How to close a file?

Closing a File

The fclose() function is used to close an open file:

```
<?php
$file = fopen("test.txt","r");
```

```
//some code to be executed

fclose($file);
?>
```

15. What is the purpose of strlen() function?

The strlen() function is used to return the length of a string.

Let's find the length of a string:

```
<?php
echo strlen("Hello world!");
?>
```

The output of the code above will be:

```
12
```

The length of a string is often used in loops or other functions, when it is important to know when the string ends. (i.e. in a loop, we would want to stop the loop after the last character in the string).

16. How to Create a Cookie?

The setcookie() function is used to set a cookie.

Note: The setcookie() function must appear BEFORE the <html> tag.

Syntax

```
setcookie(name, value, expire, path, domain);
```

17. How to Delete a Cookie?

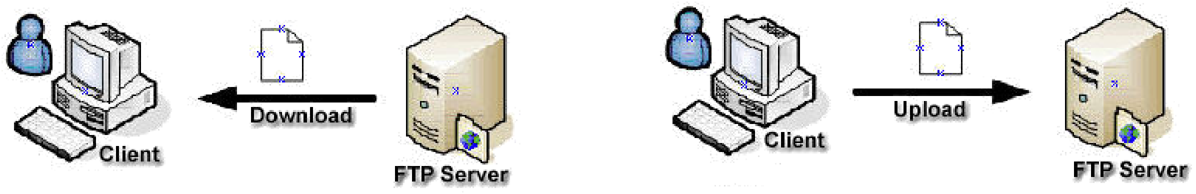
When deleting a cookie you should assure that the expiration date is in the past.

Delete example:

```
<?php
// set the expiration date to one hour ago
setcookie("user", "", time()-3600);
?>
```

18. What is FTP?

- File Transfer Protocol
- Transfers files between server and client
- Two Basic operations
    - Downloading
    - Uploading



19. What is AJAX?

AJAX = Asynchronous JavaScript and XML.

AJAX is a technique for creating fast and dynamic web pages.

FIVE MARK QUESTIONS

1. How to declare Variables in PHP?

Variables are used for storing values, like text strings, numbers or arrays.

When a variable is declared, it can be used over and over again in your script.

All variables in PHP start with a $ sign symbol.

The correct way of declaring a variable in PHP:

$var_name = value;

Example:

```
<?php
$txt="Hello World!";
$x=16;
?>
```

2. What is the purpose of strops() function?

The strpos() function

The strpos() function is used to search for a character/text within a string.

If a match is found, this function will return the character position of the first match. If no match is found, it will return FALSE.

Let's see if we can find the string "world" in our string:

```
<?php
echo strpos("Hello world!","world");
?>
```

The output of the code above will be:

```
6
```

The position of the string "world" in the example above is 6. The reason that it is 6 (and not 7), is that the first character position in the string is 0, and not 1.

3. How to Create a PHP Function?

A function will be executed by a call to the function.

Syntax

function *functionName*()
{
code to be executed;
}

PHP function guidelines:

- Give the function a name that reflects what the function does
- The function name can start with a letter or underscore (not a number)

Example

A simple function that writes my name when it is called:

```
<html>
<body>

<?php
function writeName()
{
echo "Kai Jim Refsnes";
}

echo "My name is ";
writeName();
```

```
?>

</body>
</html>
```

Output:

```
My name is Kai Jim Refsnes
```

4. How to create function with return values?

Create a PHP Function

A function will be executed by a call to the function.

Syntax

function *functionName*()
{
code to be executed;
}

PHP function guidelines:

- Give the function a name that reflects what the function does
- The function name can start with a letter or underscore (not a number)

PHP Functions - Return values

To let a function return a value, use the return statement.

Example

```
<html>
<body>
<?php
```

```
function add($x,$y)
{
$total=$x+$y;
return $total;
}
echo "1 + 16 = " . add(1,16);
?>
</body>
</html>                                    Output: 1 + 16 = 17
```

5. Explain about Function with parameters with example.

PHP Functions - Adding parameters

To add more functionality to a function, we can add parameters. A parameter is just like a variable.

Parameters are specified after the function name, inside the parentheses.

Example 1

The following example will write different first names, but equal last name:

```
<html>
<body>

<?php
function writeName($fname)
{
echo $fname . " Refsnes.<br />";
}
```

```
echo "My name is ";
writeName("Kai Jim");
echo "My sister's name is ";
writeName("Hege");
echo "My brother's name is ";
writeName("Stale");
?>
</body>
</html>
```

Output:

```
My name is Kai Jim Refsnes.
My sister's name is Hege Refsnes.
My brother's name is Stale Refsnes.
```

Example 2

The following function has two parameters:

```
<html>
<body>

<?php
function writeName($fname,$punctuation)
{
echo $fname . " Refsnes" . $punctuation . "<br />";
}
echo "My name is ";
writeName("Kai Jim",".");
echo "My sister's name is ";
writeName("Hege","!");
echo "My brother's name is ";
writeName("Ståle","?");
?>
```

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

Output:

```
My name is Kai Jim Refsnes.
My sister's name is Hege Refsnes!
My brother's name is Ståle Refsnes?
```

6. Write a short note on Access Specifiers in Php.

Public, Private and Protected

As in any object-oriented language, PHP also includes the concept of Public, Private and Protected access modifiers.

• Public – With having public access the code is visible and can be modified by any code from anywhere.

• Private – With having private access the code is visible and can be modified from within the class only.

• Protected – With having protected access the code is visible and can be modified from the same class or any of its child class.

7. Discuss about Abstract class in php.

Abstract Class

In PHP, these two features of object-oriented programming are used very frequently. Abstract classes that can't be instantiated rather they can be inherited. The class that inherits an abstract class can also be another abstract class. In PHP we can create an abstract class by using the keyword – 'abstract'.

Listing 5 – Sample code for Abstract Class

```
abstract class testParentAbstract {
            public function myWrittenFunction() {
                        // body of your funciton
            }
}

class testChildAbstract extends testParentAbstract {
            public function myWrittenFunctioninChild() {
                        // body of your function
            }
}
```

In the above example, we can create an instance of the child class – testChildAbstract, but we can't create the instance of the parent class – testParentAbstract. As we see that the child class is extending the parent class, we can use the property of the parent class in the child class. We can also implement an abstract method in our child class as per our need.

8. How to Reading a File Line by Line?

The fgets() function is used to read a single line from a file.

Note: After a call to this function the file pointer has moved to the next line.

Example

The example below reads a file line by line, until the end of file is reached:

```
<?php
$file = fopen("welcome.txt", "r") or exit("Unable to open file!");
//Output a line of the file until the end is reached
while(!feof($file))
  {
```

```
  echo fgets($file). "<br />";
  }
fclose($file);
?>
```

9. How to Retrieve a Cookie Value?

The PHP $_COOKIE variable is used to retrieve a cookie value.

In the example below, we retrieve the value of the cookie named "user" and display it on a page:

```
<?php
// Print a cookie
echo $_COOKIE["user"];

// A way to view all cookies
print_r($_COOKIE);
?>
```

In the following example we use the isset() function to find out if a cookie has been set:

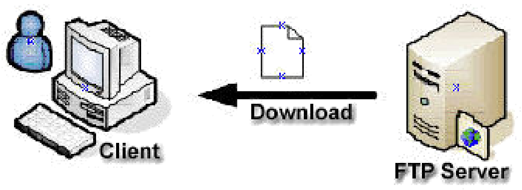
```
<html>
<body>

<?php
if (isset($_COOKIE["user"]))
  echo "Welcome " . $_COOKIE["user"] . "!<br />";
else
  echo "Welcome guest!<br />";
?>
```

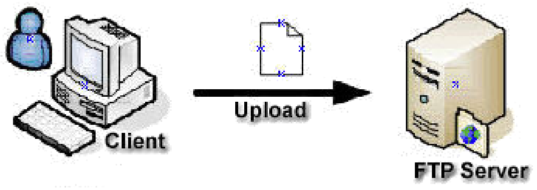
```
</body>
</html>
```

10. Describe about FTP.

What is FTP?

- File Transfer Protocol
- Transfers files between server and client
- Two Basic operations
    - Downloading
    - Uploading





How to use FTP

- Requires username and password
    - Anonymous – not secure
- Easily accessible using web browser
    - Different protocol identifier e.g. HTTP or FTP
    - *ftp://username:password@webaddress.com*
- Command Line
    - Using and 'Get' and 'Put'
- FTP Programs
    - Provide a GUI
    - CuteFTP, FileZilla, SmartFTP

11. Write a short note on AJAX?

What is AJAX?

AJAX = Asynchronous JavaScript and XML.

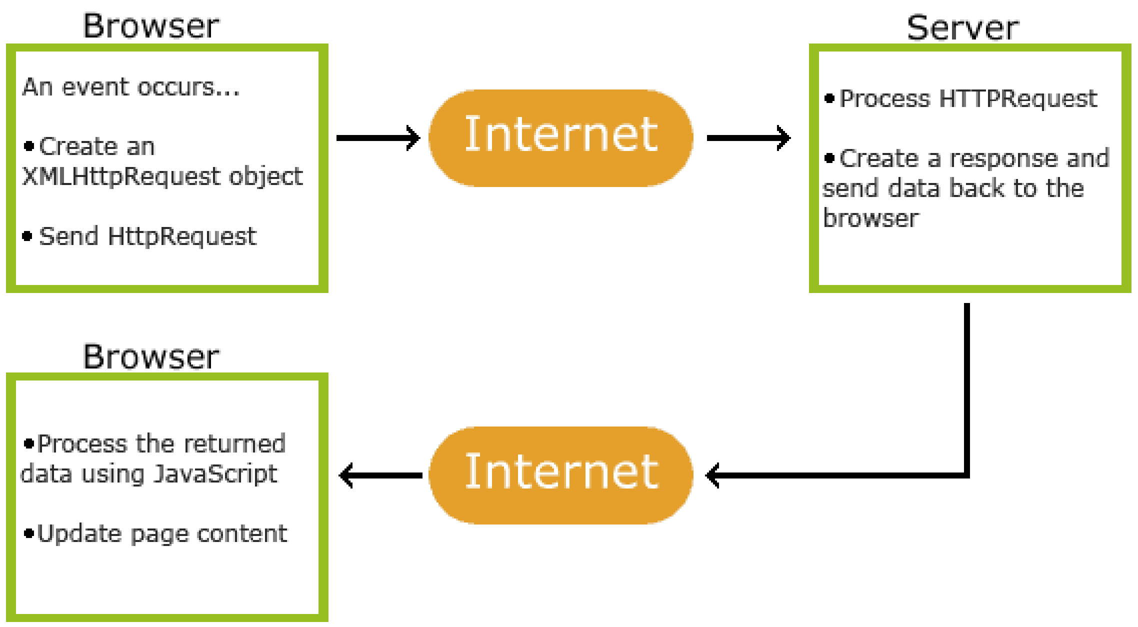
AJAX is a technique for creating fast and dynamic web pages.

AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.

Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook tabs.

How AJAX Works


Browser
An event occurs...
• Create an XMLHttpRequest object
• Send HttpRequest
Internet
Server
• Process HTTPRequest
• Create a response and send data back to the browser
Browser
• Process the returned data using JavaScript
• Update page content
Internet

TEN MARK QUESTIONS

1. Explain in detail about PHP Operators.

Operators are used to operate on values.

This section lists the different operators used in PHP.

Arithmetic Operators

Operator	Description	Example	Result
+	Addition	x=2 x+2	4
-	Subtraction	x=2 5-x	3

*	Multiplication	x=4 x*5	20
/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

Assignment Operators

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=x.y
%=	x%=y	x=x%y

Comparison Operators

Operator	Description	Example
==	is equal to	5==8 returns false
!=	is not equal	5!=8 returns true

<>	is not equal	5<>8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

Logical Operators

Operator	Description	Example

&&	and	x=6
		y=3
		
		(x < 10 && y > 1) returns true

||	or	x=6
		y=3
		
		(x==5 || y==5) returns false

!	not	x=6
		y=3
		
		!(x==y) returns true

2. Discuss about Conditional Statements in PHP.

Conditional statements are used to perform different actions based on different conditions.

Very often when you write code, you want to perform different actions for different decisions.

You can use conditional statements in your code to do this.

In PHP we have the following conditional statements:

- **if statement** - use this statement to execute some code only if a specified condition is true
- **if...else statement** - use this statement to execute some code if a condition is true and another code if the condition is false
- **if...elseif....else statement** - use this statement to select one of several blocks of code to be executed
- **switch statement** - use this statement to select one of many blocks of code to be executed

---

The if Statement

Use the if statement to execute some code only if a specified condition is true.

Syntax

if (*condition*) *code to be executed if condition is true;*

The following example will output "Have a nice weekend!" if the current day is Friday:

```
<html>
<body>

<?php
$d=date("D");
if ($d=="Fri") echo "Have a nice weekend!";
?>
```

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

Notice that there is no ..else.. in this syntax. The code is executed **only if the specified condition is true**.

The if...else Statement

Use the if....else statement to execute some code if a condition is true and another code if a condition is false.

Syntax

if (*condition*)
  code to be executed if condition is true;
else
  code to be executed if condition is false;

Example

The following example will output "Have a nice weekend!" if the current day is Friday, otherwise it will output "Have a nice day!":

```
<html>
<body>

<?php
$d=date("D");
if ($d=="Fri")
  echo "Have a nice weekend!";
else
  echo "Have a nice day!";
?>

</body>
</html>
```

If more than one line should be executed if a condition is true/false, the lines should be enclosed within curly braces:

```
<html>
<body>

<?php
$d=date("D");
if ($d=="Fri")
  {
  echo "Hello!<br />";
  echo "Have a nice weekend!";
  echo "See you on Monday!";
  }
?></body>
</html>
```

The if...elseif....else Statement

Use the if....elseif...else statement to select one of several blocks of code to be executed.

Syntax

```
if (condition)
  code to be executed if condition is true;
elseif (condition)
  code to be executed if condition is true;
else
  code to be executed if condition is false;
```

Example

The following example will output "Have a nice weekend!" if the current day is Friday, and "Have a nice Sunday!" if the current day is Sunday. Otherwise it will output "Have a nice day!":

```
<html>
<body>

<?php
$d=date("D");
if ($d=="Fri")
  echo "Have a nice weekend!";
elseif ($d=="Sun")
  echo "Have a nice Sunday!";
else
  echo "Have a nice day!";
?>

</body>
</html>
```

Conditional statements are used to perform different actions based on different conditions.

The PHP Switch Statement

Use the switch statement to select one of many blocks of code to be executed.

Syntax

```
switch (n)
{
case label1:
  code to be executed if n=label1;
  break;
case label2:
  code to be executed if n=label2;
  break;
default:
  code to be executed if n is different from both label1 and label2;
}
```

This is how it works: First we have a single expression n (most often a variable), that is evaluated once. The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed. Use **break** to prevent the code from running into the next case automatically. The default statement is used if no match is found.

Example

```
<html>
<body>
<?php
switch ($x)
{
case 1:
  echo "Number 1";
  break;
case 2:
  echo "Number 2";
  break;
case 3:
  echo "Number 3";
  break;
default:
  echo "No number between 1 and 3";
}
?>
</body>
</html>
```

3. Briefly explain about ARRAY in PHP.

An array stores multiple values in one single variable.

A variable is a storage area holding a number or text. The problem is, a variable will hold only one value.

An array is a special variable, which can store multiple values in one single variable.

If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

```
$cars1="Saab";
$cars2="Volvo";
$cars3="BMW";
```

However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?

The best solution here is to use an array!

An array can hold all your variable values under a single name. And you can access the values by referring to the array name.

Each element in the array has its own index so that it can be easily accessed.

In PHP, there are three kind of arrays:

- **Numeric array** - An array with a numeric index
- **Associative array** - An array where each ID key is associated with a value
- **Multidimensional array** - An array containing one or more arrays

---

Numeric Arrays

A numeric array stores each array element with a numeric index.

There are two methods to create a numeric array.

1. In the following example the index are automatically assigned (the index starts at 0):

```
$cars=array("Saab","Volvo","BMW","Toyota");
```

2. In the following example we assign the index manually:

```
$cars[0]="Saab";
$cars[1]="Volvo";
$cars[2]="BMW";
$cars[3]="Toyota";
```

Example

In the following example you access the variable values by referring to the array name and index:

```
<?php
$cars[0]="Saab";
$cars[1]="Volvo";
$cars[2]="BMW";
$cars[3]="Toyota";
echo $cars[0] . " and " . $cars[1] . " are Swedish cars.";
?>
```

The code above will output:

```
Saab and Volvo are Swedish cars.
```

Associative Arrays

An associative array, each ID key is associated with a value.

When storing data about specific named values, a numerical array is not always the best way to do it.

With associative arrays we can use the values as keys and assign values to them.

Example 1

In this example we use an array to assign ages to the different persons:

```
$ages = array("Peter"=>32, "Quagmire"=>30, "Joe"=>34);
```

Example 2

This example is the same as example 1, but shows a different way of creating the array:

```
$ages['Peter'] = "32";
$ages['Quagmire'] = "30";
$ages['Joe'] = "34";
```

The ID keys can be used in a script:

```
<?php
$ages['Peter'] = "32";
$ages['Quagmire'] = "30";
$ages['Joe'] = "34";

echo "Peter is " . $ages['Peter'] . " years old.";
?>
```

The code above will output:

```
Peter is 32 years old.
```

Multidimensional Arrays

In a multidimensional array, each element in the main array can also be an array. And each element in the sub-array can be an array, and so on.

Example

In this example we create a multidimensional array, with automatically assigned ID keys:

```
$families = array
  (
  "Griffin"=>array
  (
  "Peter",
  "Lois",
  "Megan"
  ),
  "Quagmire"=>array
  (
  "Glenn"
  ),
  "Brown"=>array
  (
  "Cleveland",
  "Loretta",
  "Junior"
  )
  );
```

The array above would look like this if written to the output:

```
Array
(
[Griffin] => Array
  (
  [0] => Peter
  [1] => Lois
  [2] => Megan
  )
[Quagmire] => Array
  (
  [0] => Glenn
  )
[Brown] => Array
  (
  [0] => Cleveland
  [1] => Loretta
  [2] => Junior
  )
)
```

Example 2

Lets try displaying a single value from the array above:

```
echo "Is " . $families['Griffin'][2] .
" a part of the Griffin family?";
```

The code above will output:

```
Is Megan a part of the Griffin family?
```

---

4. Explain in detail about PHP Loops Statements.

Often when you write code, you want the same block of code to run over and over again in a row. Instead of adding several almost equal lines in a script we can use loops to perform a task like this.

In PHP, we have the following looping statements:

- **while** - loops through a block of code while a specified condition is true
- **do...while** - loops through a block of code once, and then repeats the loop as long as a specified condition is true
- **for** - loops through a block of code a specified number of times
- **for each** - loops through a block of code for each element in an array

---

The while Loop

The while loop executes a block of code while a condition is true.

Syntax

```
while (condition)
  {
  code to be executed;
  }
```

Example

The example below defines a loop that starts with i=1. The loop will continue to run as long as i is less than, or equal to 5. i will increase by 1 each time the loop runs:

```
<html>
<body>

<?php
$i=1;
while($i<=5)
  {
  echo "The number is " . $i . "<br />";
  $i++;
  }
?>
```

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

Output:

```
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
```

The do...while Statement

The do...while statement will always execute the block of code once, it will then check the condition, and repeat the loop while the condition is true.

Syntax

```
do
  {
  code to be executed;
  }
while (condition);
```

Example

The example below defines a loop that starts with i=1. It will then increment i with 1, and write some output. Then the condition is checked, and the loop will continue to run as long as i is less than, or equal to 5:

```
<html>
<body>

<?php
$i=1;
do
```

```
  {
  $i++;
  echo "The number is " . $i . "<br />";
  }
while ($i<=5);
?>

</body>
</html>
```

Output:

```
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
```

The for Loop

The for loop is used when you know in advance how many times the script should run.

Syntax

for (*init; condition; increment*)
  {
  code to be executed;
  }

Parameters:

- *init*: Mostly used to set a counter (but can be any code to be executed once at the beginning of the loop)
- *Condition*: Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.

- *increment*: Mostly used to increment a counter (but can be any code to be executed at the end of the loop)

Note: Each of the parameters above can be empty, or have multiple expressions (separated by commas).

Example

The example below defines a loop that starts with i=1. The loop will continue to run as long as i is less than, or equal to 5. i will increase by 1 each time the loop runs:

```
<html>
<body>

<?php
for ($i=1; $i<=5; $i++)
  {
  echo "The number is " . $i . "<br />";
  }
?>

</body>
</html>
```

Output:

```
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
```

The foreach Loop

The foreach loop is used to loop through arrays.

Syntax

```
foreach ($array as $value)
  {
  code to be executed;
  }
```

For every loop iteration, the value of the current array element is assigned to $value (and the array pointer is moved by one) - so on the next loop iteration, you'll be looking at the next array value.

Example

The following example demonstrates a loop that will print the values of the given array:

```
<html>
<body>

<?php
$x=array("one","two","three");
foreach ($x as $value)
  {
  echo $value . "<br />";
  }
?>

</body>
</html>
```

Output:

```
one
two
three
```

5. Briefly explain about PHP Date function.

The PHP Date() Function

The PHP date() function formats a timestamp to a more readable date and time.

💡A timestamp is a sequence of characters, denoting the date and/or time at which a certain event occurred.

Syntax
date(*format*,*timestamp*)

Parameter	**Description**
format	Required. Specifies the format of the timestamp
timestamp	Optional. Specifies a timestamp. Default is the current date and time

PHP Date() - Format the Date

The required *format* parameter in the date() function specifies how to format the date/time.

Here are some characters that can be used:

- d - Represents the day of the month (01 to 31)
- m - Represents a month (01 to 12)
- Y - Represents a year (in four digits)

Other characters, like"/", ".", or "-" can also be inserted between the letters to add additional formatting:

```
<?php
echo date("Y/m/d") . "<br />";
echo date("Y.m.d") . "<br />";
echo date("Y-m-d");
?>
```

The output of the code above could be something like this:

```
2009/05/11
2009.05.11
2009-05-11
```

PHP Date() - Adding a Timestamp

The optional *timestamp* parameter in the date() function specifies a timestamp. If you do not specify a timestamp, the current date and time will be used.

The mktime() function returns the Unix timestamp for a date.

The Unix timestamp contains the number of seconds between the Unix Epoch (January 1 1970 00:00:00 GMT) and the time specified.

Syntax for mktime()

```
mktime(hour,minute,second,month,day,year,is_dst)
```

To go one day in the future we simply add one to the day argument of mktime():

```
<?php
$tomorrow = mktime(0,0,0,date("m"),date("d")+1,date("Y"));
echo "Tomorrow is ".date("Y/m/d", $tomorrow);
?>
```

The output of the code above could be something like this:

```
Tomorrow is 2009/05/12
```

6. Briefly explain about PHP include function.

PHP include() Function

The include() function takes all the content in a specified file and includes it in the current file.

If an error occurs, the include() function generates a warning, but the script will continue execution.

Example 1

Assume that you have a standard header file, called "header.php". To include the header file in a page, use the include() function:

```
<html>
<body>

<?php include("header.php"); ?>
<h1>Welcome to my home page!</h1>
<p>Some text.</p>
</body>
</html>
```

Example 2

Assume we have a standard menu file, called "menu.php", that should be used on all pages:

```
<a href="/default.php">Home</a>
<a href="/tutorials.php">Tutorials</a>
<a href="/references.php">References</a>
<a href="/examples.php">Examples</a>
```

```
<a href="/about.php">About Us</a>
<a href="/contact.php">Contact Us</a>
```

All pages in the Web site should include this menu file. Here is how it can be done:

```
<html>
<body>

<div class="leftmenu">
<?php include("menu.php"); ?>
</div>

<h1>Welcome to my home page.</h1>
<p>Some text.</p>

</body>
</html>
```

If you look at the source code of the page above (in a browser), it will look like this:

```
<html>
<body>

<div class="leftmenu">
<a href="/default.php">Home</a>
<a href="/tutorials.php">Tutorials</a>
<a href="/references.php">References</a>
<a href="/examples.php">Examples</a>
<a href="/about.php">About Us</a>
<a href="/contact.php">Contact Us</a>
</div>

<h1>Welcome to my home page!</h1>
<p>Some text.</p>
```

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

7. How to Defining and Using Variables, Constants and Functions in php?

- Three access / visibility modifiers introduced in PHP 5, which affect the scope of access to class variables and functions:
    - public : public class variables and functions can be accessed from inside and outside the class
    - protected : hides a variable or function from direct external class access + protected members are available in subclasses
    - private : hides a variable or function from direct external class access + protected members are hidden (NOT available) from all subclasses
- An access modifier has to be provided for each class instance variable
- Static class variables and functions can be declared without an access modifier → default is public

Getters and Setters

- Encapsulation : hide attributes from direct access from outside a class and provide controlled access through accessor and mutator functions
    - You can write custom getVariable() / setVariable($var) functions *or*
    - Overload the functionality with the __get() and __set() functions in PHP
- __get() and __set()
    - Prototype:

mixed __get($var);

// param represents the name of an attribute, __get returns the value of that attribute

void __set($var, $value);

// params are the name of an attribute and the value to set it to

- __get() and __set()
    - Can only be used for non-static attributes!
    - You do <u>not</u> directly call these functions;

      For an instance $acc of the BankAccount class:

      $acc->Balance = 1000;

      implicitly calls the __set() function with the value of $name set to 'Balance', and the value of $value set to 1000.

      (__get() works in a similar way)

Designing Classes

- Classes in Web development:
    - Pages
    - User-interface components
    - Shopping carts
    - Product categories
    - Customers
- TLA Consulting example revisited - a Page class, goals:
    - A consistent look and feel across the pages of the website
    - Limit the amount of HTML needed to create a new page: easily generate common parts, describe only uncommon parts
    - Easy maintainable when changes in the common parts
    - Flexible enough: ex. allow proper navigation elements in each page

Class Page

- Attributes:
    - $content → content of the page, a combination of HTML and text
    - $title → page's title, with a default title to avoid blank titles
    - $keywords → a list of keywords, to be used by search engines

    - $navigation → an associative array with keys the text for the buttons and the value the URL of the target page
- Operations:
    - __set()
    - Display() → to display a page of HTML, calls other functions to display parts of the page:
    - DisplayTitle(), DisplayKeywords(), DisplayStyles(), DisplayHeader(), DisplayMenu(), DisplayFooter() → can be overridden in a possible subclass

8. Briefly discuss about PHP Session Variables

A PHP session variable is used to store information about, or change settings for a user session. Session variables hold information about one single user, and are available to all pages in one application.

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.

Sessions work by creating a unique id (UID) for each visitor and store variables based on this UID. The UID is either stored in a cookie or is propagated in the URL.

Starting a PHP Session

Before you can store user information in your PHP session, you must first start up the session.

Note: The session_start() function must appear BEFORE the <html> tag:

```
<?php session_start(); ?>

<html>
<body>

</body>
</html>
```

The code above will register the user's session with the server, allow you to start saving user information, and assign a UID for that user's session.

Storing a Session Variable

The correct way to store and retrieve session variables is to use the PHP $_SESSION variable:

```
<?php
session_start();
// store session data
$_SESSION['views']=1;
?>

<html>
<body>

<?php
//retrieve session data
echo "Pageviews=". $_SESSION['views'];
?>
```

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

Output:

```
Pageviews=1
```

In the example below, we create a simple page-views counter. The isset() function checks if the "views" variable has already been set. If "views" has been set, we can increment our counter. If "views" doesn't exist, we create a "views" variable, and set it to 1:

```
<?php
session_start();

if(isset($_SESSION['views']))
$_SESSION['views']=$_SESSION['views']+1;
else
$_SESSION['views']=1;
echo "Views=". $_SESSION['views'];
?>
```

Destroying a Session

If you wish to delete some session data, you can use the unset() or the session_destroy() function.

The unset() function is used to free the specified session variable:

```
<?php
unset($_SESSION['views']);
?>
```

You can also completely destroy the session by calling the session_destroy() function:

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

Note: session_destroy() will reset your session and you will lose all your stored session data.

9. How to Draw Images on the Server? Explain in it.

Simple Graphics

- First, let’s see how to draw some simple graphics
- PHP 5 includes the “GD” graphics library, created by Boutell.com
- PHP 4 requires an extension be added to access this library
- PNG format used
  - Similar but superior to GIF
  - GIF patented by Unisys, expired 2003

Base image functions

- image imagecreate(width,height)
  - Creates and returns an image of specified height
- color imagecolorallocate(image, R,G,B)
  - Creates a color for image with the given Red, Green, and Blue colors
- boolean imagefill(image, x, y, color)
  - Flood fills at the given point with the given color
- header("Content-type: image/png");
  - Displays the MIME header for a PNG image
- imagepng(image)
  - Displays the data for the PNG image

Example: Green image

```
<?
 $image = imagecreate(300,200);
 $colorGreen = imagecolorallocate($image, 0, 255, 0);
 imagefill($image, 0, 0, $colorGreen);
 header("Content-type: image/png");
 imagepng($image);
```

```
?>
```

More image functions

- imagestring (image, int font, int x, int y, string s, int col)
  - Draws the string s in the image at coordinates x, y in color col. If font is 1, 2, 3, 4 or 5, a built-in font is used.
- imagefilledrectangle(image, x1,y1, x2,y2,color)
  - Draws a filled rectangle at upper left corner x1,y1 bottom right corner x2,y2
- imagefilledellipse(image, x1,y1, width,height,color)
  - Draws a filled ellipse in the bounding rectangle specified by x1,y1,x2,y2
- imagerectangle(image,x1,y1,x2,y2,color)
  - Rectangle with no fill
- imageellipse(image, x1,y1, width,height,color)
  - Ellipse with no fill

Image example

```
<?
 $image = imagecreate(300,200);
 $colorWhite = imagecolorallocate($image, 255, 255, 255);
 imagefill($image, 0, 0, $colorWhite);
 imagefilledrectangle($image, 50,50, 100, 75,
                imagecolorallocate($image,255,0,0));
 imageellipse($image, 150, 50, 100, 50,
                imagecolorallocate($image,0,0,255));
 imagestring($image, 0, 10, 10, "Hello!",
                imagecolorallocate($image,0,0,0));
 header("Content-type: image/png");
 imagepng($image);
?>
```

ImagePolygon and ImageLine

```
<?
 $image = imagecreate(300,200);
 $colorWhite = imagecolorallocate($image, 255, 255, 255);
 imagefill($image, 0, 0, $colorWhite);
 $colorBlack = imagecolorallocate($image, 0, 0, 0);
 imageLine($image, 50, 0, 200, 150, $colorBlack);
 $pointsTriangle = array(50, 10, 10, 90, 90, 90);
 imagePolygon($image, $pointsTriangle, count($pointsTriangle)/2, $colorBlack);
 header("Content-type: image/png");
 imagepng($image);
?>
```

How could you display this image multiple times in a web page?

Many more image functions

- Read jpg, gif, etc.
- Rotate, blending
- Resize
- Draw arcs
- Gamma correct
- Tiling
- PHP Reference or textbook is a great way to learn about these functions, with code samples