Introduction to PHP Programming

By Nandan Kumar

What will be covered-

- **Basic Understanding About a Website**
- **A Basic static Page Using HTML and CSS**

PHP Programming

- Introduction and Installation
- The Structure of a PHP Program
- Variable and String
- Loops and Control Flow
- PHP Function
- PHP Array
- Date and Time Function
- Image Upload and File Handling

My SQL

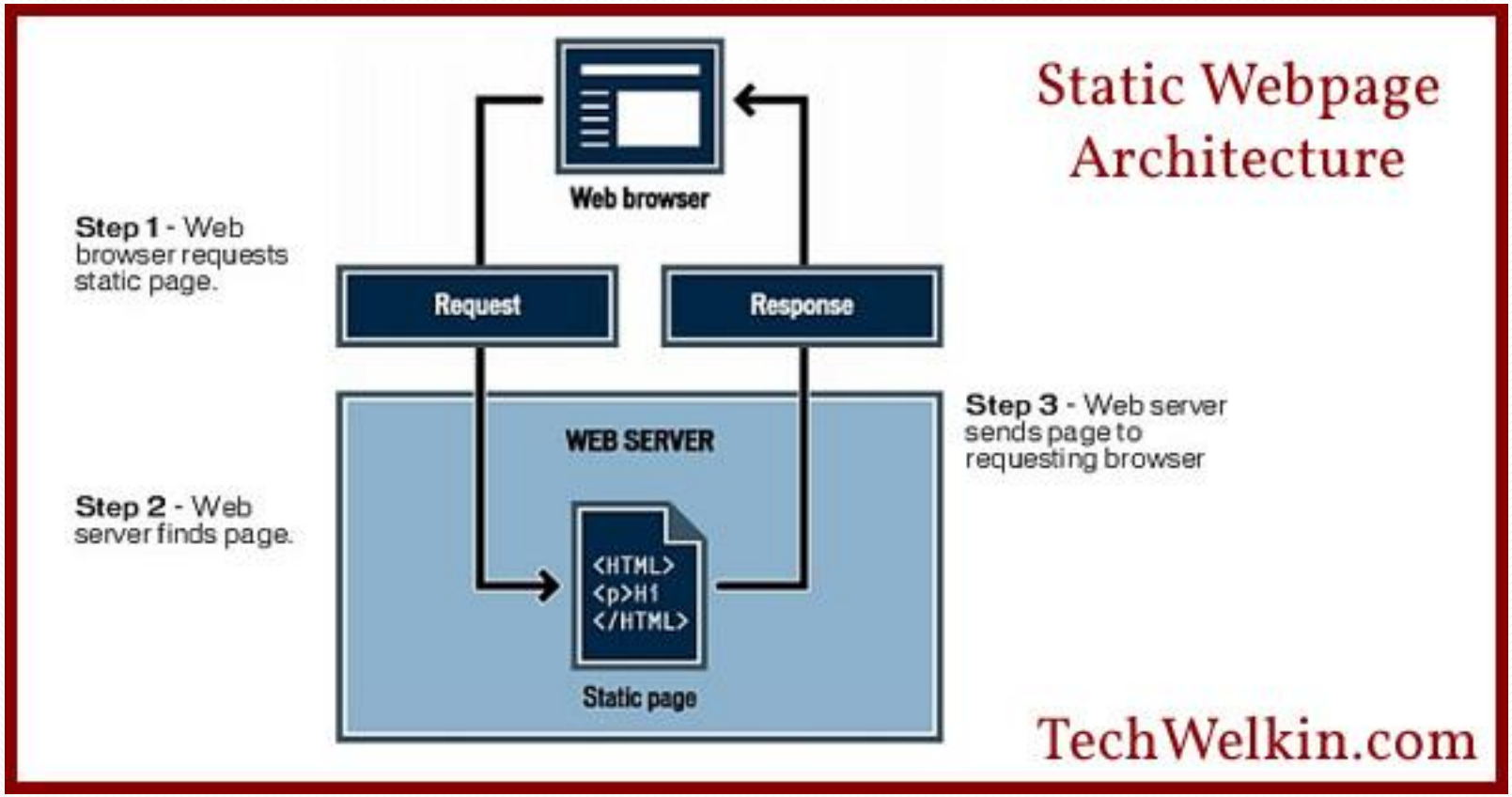
- Basic of My SQL
- Creating Database, Table,
- Alter table
- Insert , Update , Select
- Where Condition
- Limits, Distinct
- Join, Order by, Group by
- Union Import Export

Website

- Responsive non Responsive Website
- Static and Dynamic Website
- Constituents of a Dynamic Website
  - HTML, CSS, Graphic Design, PHP, MySQL, JavaScript, Ajax
- Steps Involved in Creating a Website
  - Designing Website(Coding)
  - Domain Name Registraion
  - Booking Web Space
  - Web Hosting

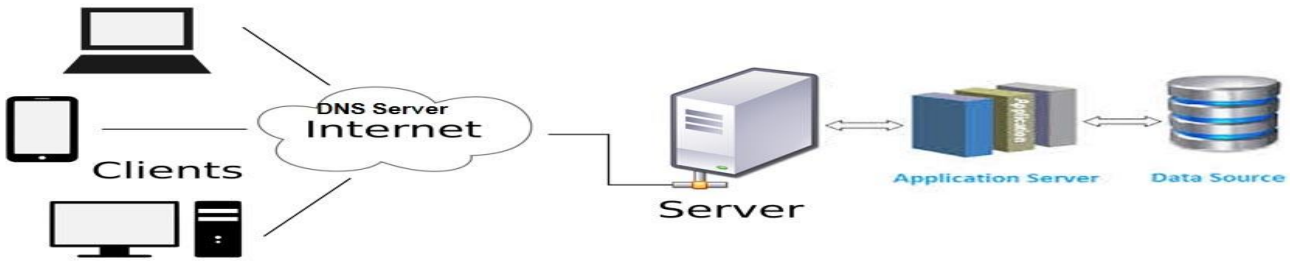
Dynamic Websites

Static Webpage

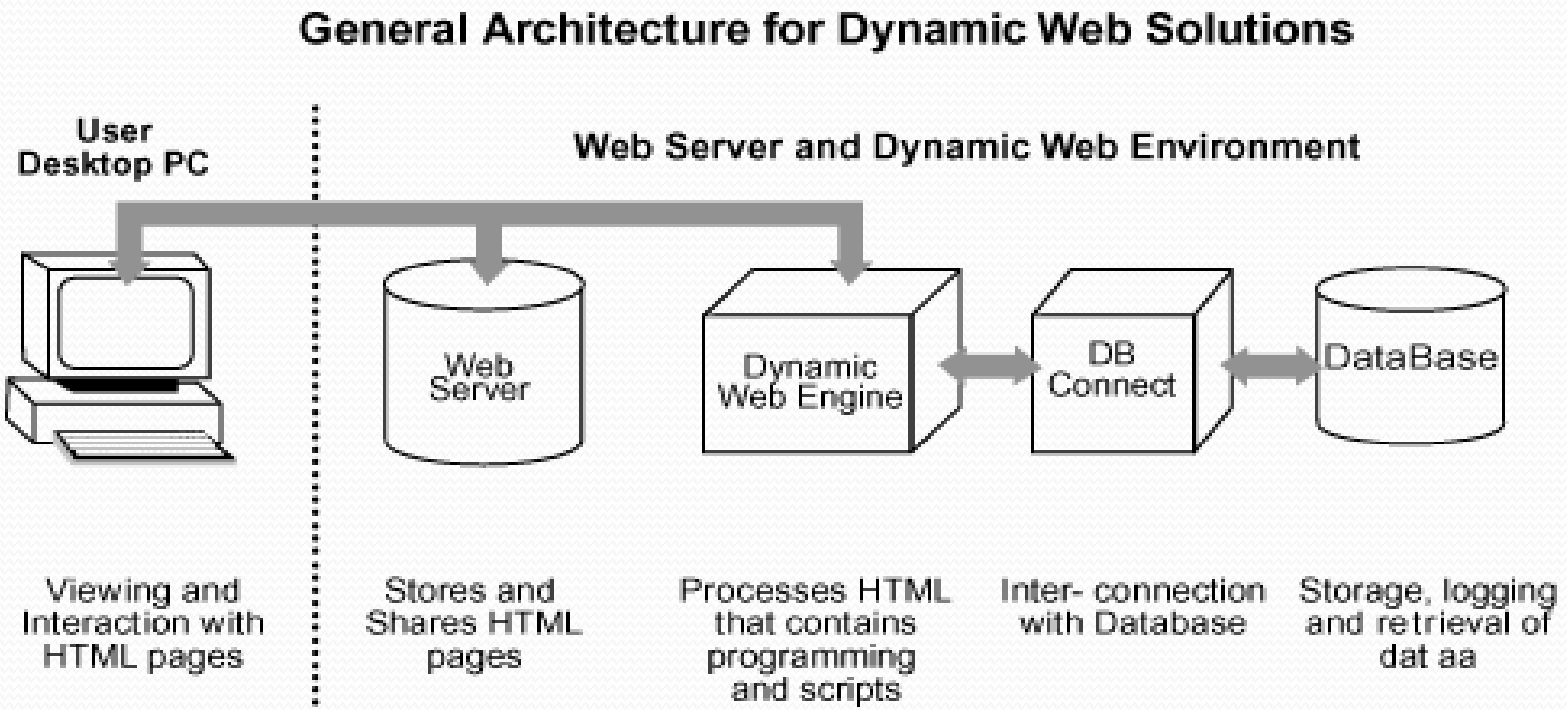


How Dynamic Website Works

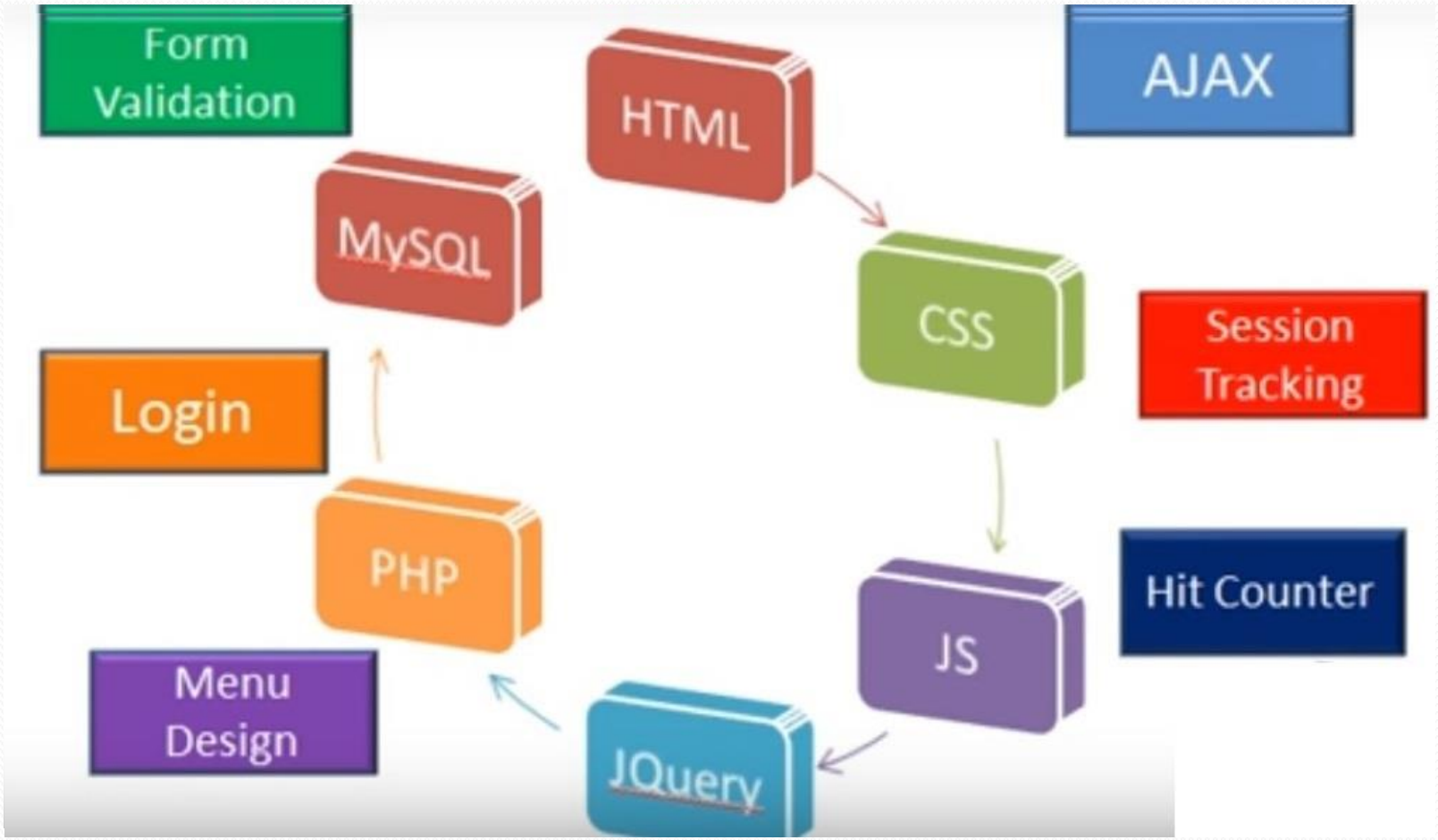
Client Server A**rchitecture**



Dynamic Web Site



Components of a Dynamic Website



HTML

- **HTML(HyperText Markup Language)** is like a skeleton of the website. It creates the structure. Like which division of the website should be placed where. It uses *tags* to define the structure.
- HTML code ensures the proper formatting of text and images in a web page so that Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, one could think of HTML as the bones (structure) of a web page.

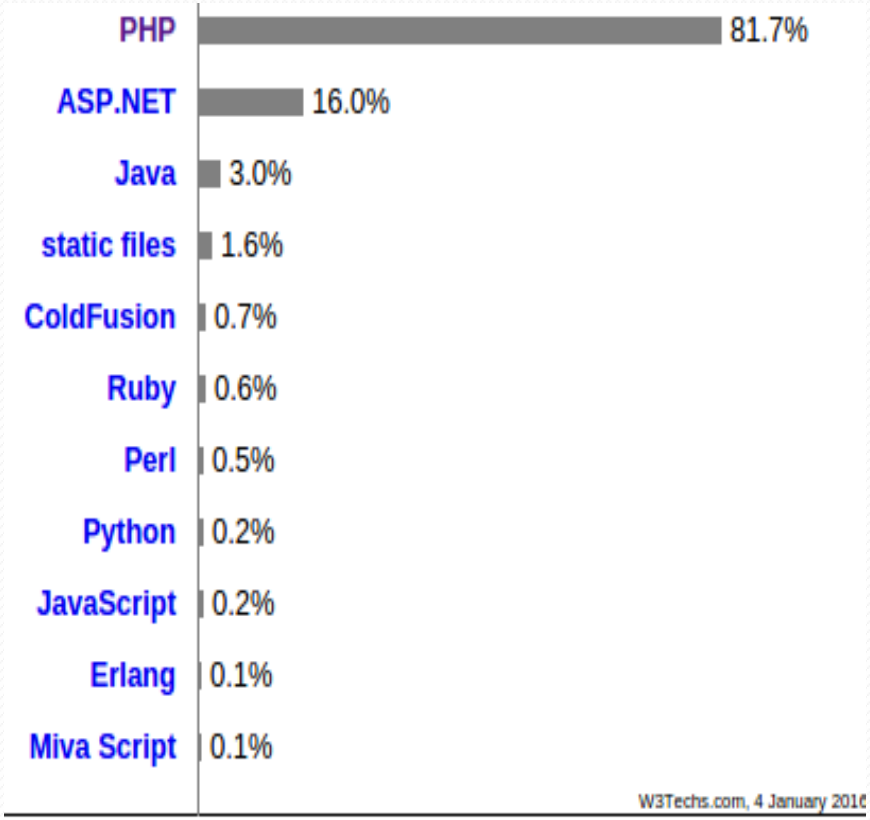
CSS

- Cascading Style sheet(CSS) - It is primarily used to add aesthetic and colour appearance the elements that are present in the Web Application. Used for styling the components. It helps us to design the layout and positions pixel perfectly to give better appearance of the elements present in the Website.
- CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the *separation of structure (or: content) from presentation*.

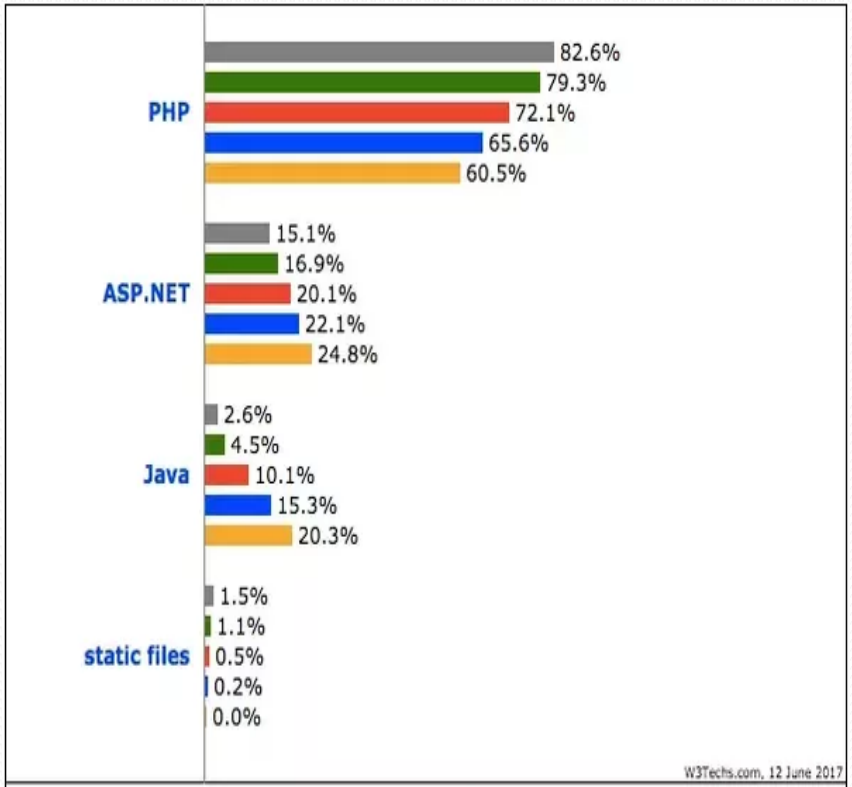
PHP

- PHP stands for Hypertext Preprocessor (no, the acronym doesn't follow the name). It is an open source, server-side, scripting language used for the development of web applications.
- PHP is mainly focused on server-side scripting, such as collect form data, generate dynamic page content, or send and receive cookies.

Scope of PHP



Percentages of websites using various server-side programming languages
Note: a website may use more than one server-side programming language



Percentages of websites using various server-side programming languages broken down by ranking
Note: a website may use more than one server-side programming language

JavaScript

- JavaScript is client-side language used to create interactive websites. It is mainly used for:
- Client-side validation
- dynamically access and update the content, structure, and style of a document using Javascript DOM.
- JavaScript works on web users' computers — even when they are offline!
- JavaScript allows you to create highly responsive interfaces. without having to wait for the server to react and show another page.
- JavaScript can load content into the document if and when the user needs it, without reloading the entire page
- JavaScript can test for what is possible in your browser and react accordingly
- JavaScript can help fix browser problems or patch holes in browser support — for example fixing CSS layout issues in certain browsers.

What is jQuery?

- jQuery is a lightweight, JavaScript library.
- The purpose of jQuery is to make it much easier to use JavaScript on your website.
- jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
- jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.
- The jQuery library contains the following features:
- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

What is MySQL?

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use
- MySQL is developed, distributed, and supported by Oracle Corporation

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.

Installing PHP Server

- **All-in-One packages -** XAMPP
- **What is XAMPP?**

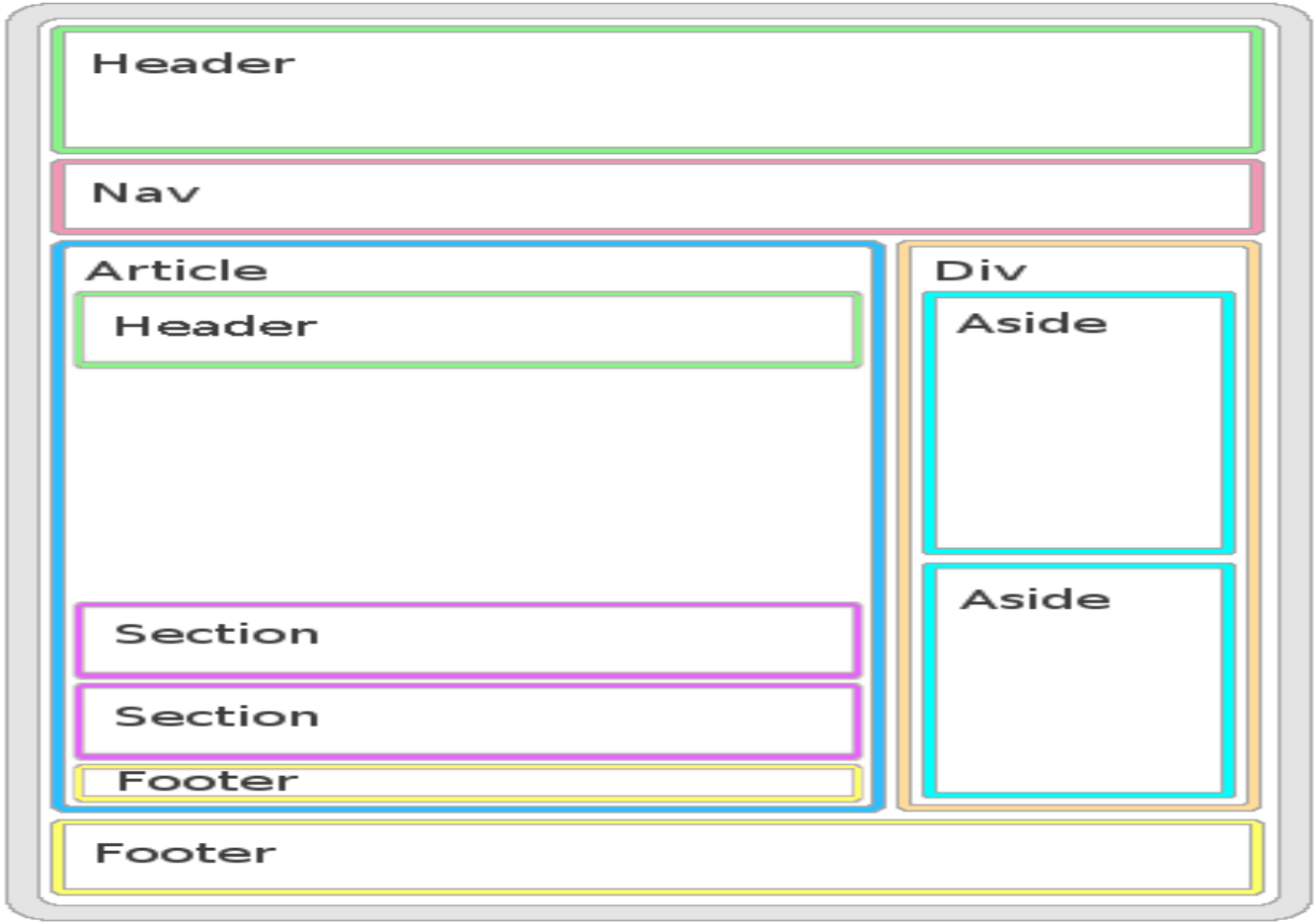
  XAMPP is the most popular, a completely free, PHP development environment

URL -- https://www.apachefriends.org/download.html

History of HTML

- In 1980, physicist Tim Berners-Lee, a contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use and share documents. Berners-Lee specified HTML and wrote the browser and server software in late 1990.
- The first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Tim Berners-Lee in late 1991.

A Typical HTML Page Structure



Coding a Typical HTML Page

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>An HTML5 Web Page</title>
</head>
<body>
<section id="wrapper">
    <header>
        <hgroup>
            <h1></h1>
            <h2></h2>
        </hgroup>
    </header>
    <nav>
    </nav>
    <article>
        <h1></h1>
        <p></p>
        <div></div>
    </article>
    <aside>
        <h1></h1>
        <p></p>
    </aside>
    <footer>
    </footer>
</section>
</body>
</html>
```

Commonly Used HTML Tags

<h1> - <h6>	Heading
<p>	Paragraph
<i>	Italic
<b>	Bold
<a>	Anchor
&	Unordered List & List Item
<blockquote>	Blockquote
<hr>	Horizontal Rule
<img>	Image
	Division

HTML Tag

- Click Me

The HTML **element** is everything from the start tag to the end tag.

HTML Tags

<!DOCTYPE> : Always add the <!DOCTYPE> declaration to your HTML documents, so that the browser knows what type of document to expect.

<!DOCTYPE html> HTML 5

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<head> Tag</head>:The following tags describe head tag metadata:

- <title> -> title of the browser tab</title>
- <style>->CSS Properties</style>
- <meta>-> used to specify page description, keywords, author of the document, last modified, and other metadata.

<meta name="viewport" content="width=device-width, initial-scale=1.0">

- <link>->used to link to external style sheets.

<link rel="stylesheet" type="text/css" href="theme.css">

- <script>-> JavaScript</script>

Html <body> Tag

HTML <h1> to <h6> Tags :The <h1> to <h6> tags are used to define HTML headings.

HTML p Tags : The <p> tag defines a paragraph.

HTML <ul> Tag : The <ul> tag defines an unordered (bulleted) list.

- Use the <ul> tag together with the <li> tag to create unordered lists.

Attributes : type Values : disc| square| circle

HTML Tables : An HTML table is defined with the **<table>** tag.

Each table row is defined with the **<tr>** tag. A table header is defined with the **<th>** tag. A table data/cell is defined with the **<td>** tag.

Attributes : bgcolor Values : rbg|xxxxxx| color name

Attributes : width Values : px/ %age

Attributes : border Values : 0/ px

Attributes : cellpadding Values : px

Attributes : cellspacing Values : px

HTML <form>

The HTML **<form>** element defines a form that is used to collect user input

```
<form action="/action_page.php" method="get" id="form1">
```

The <input> Element-

<input type="text"> Defines a one-line text input field

<input type="radio"> Defines a radio

<input type="submit"> Defines a submit button

The <select> Element : for **drop-down list**:

```
<select name="cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
</select>
```

The <button> Element

```
<button type="submit" form="form1" value="Submit">Submit</button>
```

What is Css?

- **Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language.
- CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts

Default CSS Attribute

- h1

{h1display: block;
font-size: 2em;
margin-top: 0.67em;
margin-bottom: 0.67em;
margin-left: 0;
margin-right: 0;
font-weight: bold;}

- P

{display: block;
margin-top: 1em;
margin-bottom: 1em;
margin-left: 0;
margin-right: 0;}

- a:link

{color: (internal value);
text-decoration: underline;
cursor: auto;}

- h6

{display: block;
font-size: .67em;
margin-top: 2.33em;
margin-bottom: 2.33em;
margin-left: 0;
margin-right: 0;
font-weight: bold;

PHP - Introduction

- Rasmus Lerdorf unleashed the first version of PHP way back in 1994.
- PHP originally stood for *Personal Home Page*
- PHP is a recursive acronym for "PHP: Hypertext Preprocessor". Pre Processor means all of the HyperText is processed first and then the result is send as pure HTML to the web browser.
- PHP is a server side scripting language powering over 80% of the web.
- PHP page is a file with .php extension which may contain HTML, CSS, PHP and JavaScript.
- *Some huge websites like Facebook, Yahoo, Flickr, and Wikipedia are built in PHP. Most of the major content management systems (CMS), such as WordPress, Drupal, Joomla and Magento are also built in PHP.*

Comments in PHP

- A comment in PHP code is a line that is not read/executed as part of the program. Its only purpose is to be read by someone who is looking at the code.
- Comments can be used to:Let others understand what you are doing
- Remind yourself of what you did - Most programmers have experienced coming back to their own work a year or two later and having to re-figure out what they did. Comments can remind you of what you were thinking when you wrote the code
- PHP supports several ways of commenting:

PHP Syntax

- A PHP script can be placed anywhere in the document.
- A PHP script starts with **<?php** and ends with **?>**
- The default file extension for PHP files is ".php".
- A PHP file normally contains HTML tags, and some PHP scripting code.
- PHP statements end with a semicolon (;).
- In PHP, all keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are NOT case-sensitive. ECHO, echo, EcHo are legal (and equal)
- However; all variable names are case-sensitive. $color, $COLOR, and $coLOR are different variables.

PHP 5 Variables

- Variables are "containers" for storing information.
- In PHP, a variable starts with the $ sign, followed by the name of the variable.
- When you assign a text value to a variable, put quotes around the value.
- Unlike other programming languages, PHP has no command for declaring a variable. It is created the moment you first assign a value to it.
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive ($age and $AGE are two different variables)
- PHP automatically converts the variable to the correct data type, depending on its value.

PHP Variables Scope

Global and Local Scope

- A variable declared **outside** a function has a GLOBAL SCOPE and can only be accessed outside a function.
- A variable declared **within** a function has a LOCAL SCOPE and can only be accessed within that function.
- The global keyword is used to access a global variable from within a function.
- when a function is completed/executed, all of its variables are deleted. However, sometimes we want a local variable NOT to be deleted.To do this, use the **static** keyword when you first declare the variable

PHP is a Loosely Typed Language

- We need not have to tell PHP which data type the variable is.
- PHP automatically converts the variable to the correct data type, depending on its value.
- In other languages such as C, C++, and Java, the programmer must declare the name and type of the variable before using it.

PHP 5 echo and print Statements

- echo and print are more or less the same. They are both used to output data to the screen.
- echo has no return value while print has a return value of 1 so it can be used in expressions.
- <?php

```
echo "<h2>PHP is Fun!</h2>";
    echo "Hello world!<br>";
    echo "I'm about to learn PHP!<br>";
    echo "This ", "string ", "was ", "made ", "with multiple parameters.";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
    ?>
```

PHP Print..

```
<?php
    $txt1 = "Learn PHP";
    $txt2 = "W3Schools.com";
    $x = 5;
    $y = 4;

    print "<h2>" . $txt1 . "</h2>";
    print "Study PHP at " . $txt2 . "<br>";
    print $x + $y;
    ?>
```

PHP Data Types

PHP supports the following data types:

String, Integer ,Float, Boolean ,Array, Object, NULL

Resource

PHP String

- A string is a sequence of characters, like "Hello world!".
- A string can be any text inside quotes. You can use single or double quotes

PHP Data Types

PHP Integer

- An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.
- An integer must have at least one digit
- An integer must not have a decimal point
- An integer can be either positive or negative
- The PHP var_dump() function returns the data type and value

PHP Float

- A float (floating point number) is a number with a decimal point or a number in exponential form.

PHP Data Types

PHP Boolean

- A Boolean represents two possible states: TRUE or FALSE.
- Booleans are often used in conditional testing.

PHP Array

- An array stores multiple values in one single variable.

PHP NULL Value

- Null is a special data type which can have only one value: NULL.
- A variable of data type NULL is a variable that has no value assigned to it.
- If a variable is created without a value, it is automatically assigned a value of NULL.
- Variables can also be emptied by setting the value to NULL.

PHP Data Types

PHP Object

- An object is a data type which stores data and information on how to process that data.
- In PHP, an object must be explicitly declared.
- First we must declare a class of object. For this, we use the class keyword. A class is a structure that can contain properties and methods

PHP – Operator Types

PHP language supports following type of operators.

- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators
- Assignment Operators
- Conditional (or ternary) Operators

Arithmetic Operators

Arithmetic Operators

The following arithmetic operators are supported by PHP language:

Assume variable A holds 10 and variable B holds 20 then:

Operator	Description	Example
+	Adds two operands	A + B will give 30
-	Subtracts second operand from the first	A - B will give -10
*	Multiply both operands	A * B will give 200
/	Divide the numerator by denominator	B / A will give 2
%	Modulus Operator and remainder of after an integer division	B % A will give 0
++	Increment operator, increases integer value by one	A++ will give 11
--	Decrement operator, decreases integer value by one	A-- will give 9

Comparison Operators

Operator	Description	Example
==	Checks if the value of two operands are equal or not, if yes, then condition becomes true.	(A == B) is not true.
!=	Checks if the value of two operands are equal or not, if values are not equal, then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes, then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes, then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes, then condition becomes true.	(A <= B) is true.

Logical (or Relational) Operators

Operator	Description	Example
and	Called Logical AND operator. If both the operands are true, then condition becomes true.	(A and B) is true.
or	Called Logical OR Operator. If any of the two operands are non zero, then condition becomes true.	(A or B) is true.
&&	Called Logical AND operator. If both the operands are non zero, then condition becomes true.	(A && B) is true.
||	Called Logical OR Operator. If any of the two operands are non zero, then condition becomes true.	(A || B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true, then Logical NOT operator will make false.	!(A && B) is false.

PHP – Decision Making

You can use conditional statements in your code to make your decisions. PHP supports the following three decision making statements:

if...else statement - use this statement if you want to execute a set of code when a condition is true and another if the condition is not true

elseif statement - is used with the if...else statement to execute a set of code if one of several condition are true

switch statement - is used if you want to select one of many blocks of code to be executed, use the Switch statement. The switch statement is used to avoid long blocks of if..elseif..else code.

PHP switch

```
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

PHP – Arrays

There are three different kind of arrays and each array value is accessed using an ID cwhich is called array index.

Numeric array - An array with a numeric index. Values are stored and accessed in linear fashion

Associative array - An array with strings as index. This stores element values in association with key values rather than in a strict linear index order.

Multidimensional array - An array containing one or more arrays and values are accessed using multiple indices

PHP – GET and POST Methods

The GET Method

The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the **? character.**

The GET method produces a long string that appears in your server logs, in thebrowser's Location: box.

The GET method is restricted to send up to 1024 characters only.

Never use GET method if you have password or other sensitive information to be sent to the server.

GET can't be used to send binary data, like images or word documents, to the server.

The data sent by GET method can be accessed using QUERY_STRING environment variable.

The PHP provides **$_GET** associative array to access all the sent information using GET method.

The POST Method

The POST Method

The POST method does not have any restriction on data size to be sent.

The POST method can be used to send ASCII as well as binary data.

The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.

The PHP provides $_POST associative array to access all the sent information using POST method.

The $_REQUEST variable

The PHP $_REQUEST variable can be used to get the result from form data sent with both the GET and POST methods.

PHP Inclusion

- You can include the content of a PHP file into another PHP file before the server executes it. There are two PHP functions which can be used to included one PHP file into another PHP file.
- **The include() Function**
- **The require() Function**
- This is a strong point of PHP which helps in creating functions, headers, footers, or elements that can be reused on multiple pages. This will help developers to make it easy to change the layout of complete website with minimal effort. If there is any change required, then instead of changing thousands of files just change included file.
- **The include() Function**

  The include() function takes all the text in a specified file and copies it into the file that uses the include function. If there is any problem in loading a file, then the include() function generates a warning but the script will continue execution.

<?php include("menu.php"); ?>

PHP Inclusion

The require() Function

The require() function takes all the text in a specified file and copies it into the file that uses the require function.

So there is no difference in require() and include() except they handle error conditions. It is recommended to use the require() function instead of include(), because scripts should not continue executing if files are missing or misnamed.

PHP – Loop Types

Loops in PHP are used to execute the same block of code a specified number of times. PHP supports following four loop types.

for - loops through a block of code a specified number of times.

while - loops through a block of code if and as long as a specified condition is true.

do...while - loops through a block of code once, and then repeats the loop as long as a special condition is true.

foreach - loops through a block of code for each element in an array.

PHP – Loop Types

The break statement

The PHP break keyword is used to terminate the execution of a loop prematurely. The break statement is situated inside the statement block. If gives you full control and whenever you want to exit from the loop you can come out. After coming out of a loop immediate statement to the loop will be executed.

The continue statement

The PHP continue keyword is used to halt the current iteration of a loop but it does not terminate the loop. Just like the break statement the continue statement is situated inside the statement block containing the code that the loop executes, preceded by a conditional test. For the pass encountering continue statement, rest of the loop code is skipped and next pass starts

PHP 5 Strings

Search For a Specific Text Within a String

- The PHP strpos() function searches for a specific text within a string.
- If a match is found, the function returns the character position of the first match. If no match is found, it will return FALSE.

```
echo strpos("Hello world!", "world"); // outputs 6
```

Replace Text Within a String

- The PHP str_replace() function replaces some characters with some other characters in a string.

```
echo str_replace("world", "Dolly", "Hello world!"); // outputs Hello Dolly!
```

PHP 5 Strings

Converting lowercase into Title Case

- Ucwords() is used to convert first alphabet of every word into uppercase

echo ucwords("welcome to the php world");

Converting a whole string into UPPERCASE

Strtoupper() is used to convert a whole string into uppercase.

echo strtoupper("welcome to cloudways");

Converting whole String to lowercase

Strtolower() is used to convert a string into lowercase.

echo strtolower("WELCOME TO CLOUDWAYS");

PHP 5 Strings

- A string is a sequence of characters, like "Hello world!".

PHP String Functions

- The PHP strlen() function returns the length of a string.

```
echo strlen("Hello world!"); // outputs 12
```

Count The Number of Words in a String

The PHP str_word_count() function counts the number of words in a string.

```
echo str_word_count("Hello world!"); // outputs 2
```

Reverse a String

The PHP strrev() function reverses a string

```
echo strrev("Hello world!"); // outputs !dlrow olleH
```

PHP 5 Strings

Displaying part of String

- Through substr() function you can display or extract a string from a particular position.

```
echo substr("Welcome to Cloudways",6)."<br>";
echo substr("Welcome to Cloudways",0,10)."<br>";
```

Comparing Strings

You can compare two strings by using strcmp().

If string 1 is greater than string 2 then it returns greater than zero. If string 1 is less than string 2 then it returns less than zero. It returns zero, if the strings are equal.

```
echo strcmp("cloudways","cloudways");//Both the strings are equal
```

PHP 5 Arrays

- An array stores multiple values in one single variable

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

$cars[0] = "Volvo“; $cars[1] = "BMW"; $cars[2] = "Toyota";

Types of arrays

- **Indexed arrays** - Arrays with a numeric index
- **Associative arrays** - Arrays with named keys

Associative arrays

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
$age['Peter'] = "35“;  $age['Ben'] = "37“;  $age['Joe'] = "43";
```

Looping Through Array

```
foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
  }
```

PHP 5 Functions

Built-in functions.

- The real power of PHP comes from its functions; it has more than 1000 built-in functions.

PHP User Defined Functions

- Besides the built-in PHP functions, we can create our own functions.
- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute immediately when a page loads.
- A function will be executed by a call to the function.

The PHP Date() Function

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

date(format,timestamp)

Format- Specifies the format of the timestamp

Timestamp- Optional. Specifies a timestamp.

The required *format* parameter of the date() function specifies how to format the date (or time).

- Here are some characters that are commonly used for dates:
- d - Represents the day of the month (01 to 31)
- m - Represents a month (01 to 12)
- Y - Represents a year (in four digits)
- l (lowercase 'L') - Represents the day of the week
- Other characters, like"/", ".", or "-" can also be inserted between the characters to add additional formatting.

The PHP Date() Function

- <?php
  echo "Today is " . date(“d/m/y") . "
";
  echo "Today is " . date("Y.m.d") . "
";
  echo "Today is " . date("Y-m-d") . "
";
  echo "Today is " . date("l");
  ?>

PHP mail() Function

- The mail() function allows you to send emails directly from a script.

mail(to,subject,message,headers,parameters);

To - Required. Specifies the receiver / receivers of the email

Subject - Required. Specifies the subject of the email. **Note:** This parameter cannot contain any newline characters

Message - Required. Defines the message to be sent. Each line should be separated with a LF (\n). Lines should not exceed 70 characters.

PHP mail() Function

- <?php
  $to = "somebody@example.com";
  $subject = "My subject";
  $txt = "Hello world!";
  $headers = "From: webmaster@example.com" . "\r\n" .
  "CC: somebodyelse@example.com";

  mail($to,$subject,$txt,$headers);
  ?>

PHP Operators

- PHP Arithmetic Operators
- The PHP arithmetic operators are used with numeric values to perform common arithmetical operations, such as addition, subtraction, multiplication etc.

Operator	Name	Example	Result
+	Addition	$x + $y	Sum of $x and $y
-	Subtraction	$x - $y	Difference of $x and $y
*	Multiplication	$x * $y	Product of $x and $y
/	Division	$x / $y	Quotient of $x and $y
%	Modulus	$x % $y	Remainder of $x divided by $y
**	Exponentiation	$x ** $y	Result of raising $x to the $y'th power (Introduced in PHP 5.6)

PHP Assignment Operators

- The PHP assignment operators are used with numeric values to write a value to a variable.

Assignment	Same as...	Description
x = y	x = y	The left operand gets set to the value of the expres the right
x += y	x = x + y	Addition
x -= y	x = x - y	Subtraction
x *= y	x = x * y	Multiplication
x /= y	x = x / y	Division
x %= y	x = x % y	Modulus

PHP Comparison Operators

- The PHP comparison operators are used to compare two values (number or string):

Operator	Name	Example	Result
==	Equal	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or they are not of the same type
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equa to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y

PHP Logical Operators

- The PHP logical operators are used to combine conditional statements.

Operator	Name	Example	Result
and	And	$x and $y	True if both $x and $y are true
or	Or	$x or $y	True if either $x or $y is true
xor	Xor	$x xor $y	True if either $x or $y is true, but not both
&&	And	$x && $y	True if both $x and $y are true
||	Or	$x || $y	True if either $x or $y is true
!	Not	!$x	True if $x is not true

PHP Conditional Statements

- PHP - The if Statement

The if statement executes some code if one condition is true.

- Syntax
- if (*condition*) {
  code to be executed if condition is true;
  }

The example below will output "Have a good day!" if the current time (HOUR) is less than 20:

- Example
- <?php
  $t = date("H");

  if ($t < "20") {
  echo "Have a good day!";
  }
  ?>

PHP - The if...else Statement

The if....else statement executes some code if a condition is true and another code if that condition is false.

Syntax

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

The example below will output "Have a good day!" if the current time is less than 20, and "Have a good night!" otherwise:

Example

```
<?php
$t = date("H");

if ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

PHP - The if...elseif....else Statement

The if....elseif...else statement executes different codes for more than two conditions.

Syntax

```
if (condition) {
    code to be executed if this condition is true;
} elseif (condition) {
    code to be executed if this condition is true;
} else {
    code to be executed if all conditions are false;
}
```

The example below will output "Have a good morning!" if the current time is less than 10, and "Have a good day!" if the current time is less than 20. Otherwise it will output "Have a good night!":

Example

```
<?php
$t = date("H");

if ($t < "10") {
    echo "Have a good morning!";
} elseif ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

PHP 5 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;
    case label3:
        code to be executed if n=label3;
        break;
    ...
    default:
        code to be executed if n is different from all labels;
}
```

PHP 5 switch Statement

Example

```
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

The PHP while Loop

The while loop executes a block of code as long as the specified condition is true.

Syntax

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

Example

```
<?php
$x = 1;

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

The PHP do...while Loop

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

Syntax

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

Example

```
<?php
$x = 1;

do {
    echo "The number is: $x <br>";
    $x++;
} while ($x <= 5);
?>
```

The PHP for Loop

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

Syntax

```
for (init counter; test counter; increment counter) {
    code to be executed;
}
```

Example

```
<?php
for ($x = 0; $x <= 10; $x++) {
    echo "The number is: $x <br>";
}
?>
```

The PHP foreach Loop

The foreach loop works only on arrays, and is used to loop through each key/value pair in an array.

Syntax

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

Example

```
<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
    echo "$value <br>";
}
?>
```

PHP – Arrays

There are three different kind of arrays and each array value is accessed using an ID which is called array index.

Numeric array - An array with a numeric index. Values are stored and accessed in linear fashion

Associative array - An array with strings as index. This stores element values in association with key values rather than in a strict linear index order.

Multidimensional array - An array containing one or more arrays and values are accessed using multiple indices

Numeric Array

These arrays can store numbers, strings and any object but their index will be represented by numbers. By default, the array index starts from zero.

```
<?php
/* First method to create array. */
$numbers = array( 1, 2, 3, 4, 5);
foreach( $numbers as $value )
{
echo "Value is $value <br />";
}
/* Second method to create array. */
$numbers[0] = "one";
$numbers[1] = "two";
$numbers[2] = "three";
$numbers[3] = "four";
$numbers[4] = "five";
foreach( $numbers as $value )
{
echo "Value is $value <br />";
}
?>
</body>
</html>
```

Associative Arrays

Associative array will have their index as string so that you can establish a strong association between key and values.

```
<html>
<body>
<?php
/* First method to associate create array. */
$salaries = array(
“ramesh" => 2000,
“mohan" => 1000,
“kavita" => 500
);
echo "Salary of  Ramesh is ". $salaries[‘ramesh'] . "<br />";
echo "Salary of  Mohan is ". $salaries[‘mohan']. "<br />";
echo "Salary of  Kavita is ". $salaries[‘kavita']. "<br />";
/* Second method to create array. */
$salaries[‘ramesh'] = "high";
$salaries[‘mohan'] = "medium";
$salaries[‘kavita'] = "low";
echo "Salary of  Ramesh is ". $salaries[‘ramesh'] . "<br />";
echo "Salary of   Mohan is ". $salaries[‘mohan']. "<br />";
echo "Salary of  Kavita is ". $salaries[‘kavita']. "<br />";
?>
</body>
</html>
```

Multidimensional Arrays

A multi-dimensional 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.

```
$marks =array(
"ramesh" =>array(
    'phy'=>66,
    'chem'=>88,
    'math'=>72,
    'eng'=>80,
    'bio'=>65,
    ),
    "geeta" =>array(
    'phy'=>63,
    'chem'=>81,
    'math'=>74,
    'eng'=>83,
    'bio'=>66,
    ));

    echo '<pre>'; print_r($marks); echo '</pre>';
ECHO (" Ramesh Physics Marks".$marks['ramesh']['phy']."<br/>");
ECHO ("Geeta Chemistry Marks".$marks['geeta']['chem']."<br/>");
```

Sending Data From Browser to Server

- Web browsers offer four basic mechanisms that can be used to place data into the HTTP GET or POST request the browser makes to the server:
- links
  - clicking a link triggers a GET request to be made to the server
- forms
  - submitting a form can trigger either a GET or POST request to be made to the server
- javascript
  - a Javascript technique known as AJAX can be used to trigger either a GET or POST request to the server
- cookies
  - a web server can ask the web browser to keep a piece of data (for example, an ID number) that is then repeatedly sent back to the server with every subsequent request

Sending Data…

Using links

- When a visitor clicks a link, the browser makes an HTTP GET request to the server for the linked file.
- Click me to magically send data to the server
- **Server-side code**

```
<?php
 //show the data that was received from the client along with the request
 print($_GET['wonderful_data']); //outputs 'this is meaningless'
 ?>
```

PHP – Strings

Strings are sequences of characters, like "PHP supports string operations".

Singly quoted strings are treated almost literally, whereas **doubly quoted strings** replace variables with their values as well as specially interpreting certain character sequences.

```
$variable = "name";
$literally = 'My $variable will not print!\\n';
print($literally);
$literally = "My $variable will print!\\n";
print($literally);
```

The escape-sequence replacements are:

\n is replaced by the newline character

\r is replaced by the carriage-return character

\t is replaced by the tab character

\$ is replaced by the dollar sign itself ($)

\" is replaced by a single double-quote (")

\\ is replaced by a single backslash (\)

PHP File Upload

```php
<?php
$username= $_POST['username'];
$f=$_FILES['myfile'];
echo "Hello $username";
echo "File Name: ".$f['name'];
echo "File Type: ".$f['type'];
echo "File Size: ".$f['size'];
if(file_exists("photos/".$f['name']))
{
 echo $f['name']." already exists";
}
else if($f['type']=="image/jpeg")
{
 move_uploaded_file($f['tmp_name'],"photos/".$f['name']);
}
else
 echo "File format is not valid, unable to upload";
```

What is MySQL?

- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use
- MySQL is developed, distributed, and supported by Oracle Corporation
- MySQL is named after co-founder Monty Widenius's daughter: My

MySQL database system

Database: A database is a collection of tables, with related data.

Table: A table is a matrix with data. A table in a database looks like a simple spreadsheet.

Column: One column (data element) contains data of one and the same kind, for example the column postcode.

Row: A row (tuple, entry or record) is a group of related data, for example the data of one subscription.

Redundancy: Storing data twice, redundantly to make the system faster.

Primary Key: A primary key is unique. A key value can not occur twice in one table. With a key, you can find at most one row.

Foreign Key: A foreign key is the linking pin between two tables.

Database system

- What's Database Normalization ?
  Normalization is the process where a database is designed in a way that removes redundancies, and increases the clarity in organizing data in a database. In easy English, it means take similar stuff out of a collection of data and place them into tables. Keep doing this for each new table recursively and you'll have a Normalized database.

MYSQL Commands

The query and update commands form the DML part of SQL:

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database

The most important DDL statements in SQL are:

- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

PHP Connect to MySQL

- PHP 5 and later can work with a MySQL database using:
- **MySQLi extension** (the "i" stands for improved)
- **PDO (PHP Data Objects)**
- Earlier versions of PHP used the MySQL extension. However, this extension was deprecated in 2012.

Should I Use MySQLi or PDO?

- If you need a short answer, it would be "Whatever you like".
- Both are object-oriented, but MySQLi also offers a procedural API.

Open a Connection to MySQL

- Before we can access data in the MySQL database, we need to be able to connect to the server:

Example (MySQLi Object-Oriented)

- ```
  <?php
  $servername = "localhost";
  $username = "username";
  $password = "password";

  // Create connection
  $conn = new mysqli($servername, $username, $password);

  // Check connection
  if ($conn->connect_error) {
      die("Connection failed: " . $conn->connect_error);
  }
  echo "Connected successfully";
  ?>
  ```
  ```

PHP Create a MySQL Database

- A database consists of one or more tables.
- You will need special CREATE privileges to create or to delete a MySQL database.
- ```
  <?php
  $servername = "localhost";
  $username = "username";
  $password = "password";

  // Create connection
  $conn = new mysqli($servername, $username, $password);
  // Check connection
  if ($conn->connect_error) {
      die("Connection failed: " . $conn->connect_error);
  }

  // Create database
  $sql = "CREATE DATABASE myDB";
  if ($conn->query($sql) === TRUE) {
      echo "Database created successfully";
  } else {
      echo "Error creating database: " . $conn->error;
  }

  $conn->close();
  ?>
  ```
  ```

PHP Create MySQL Tables

- The CREATE TABLE statement is used to create a table in MySQL.
- **NOT NULL** - Each row must contain a value for that column, null values are not allowed
- **DEFAULT** value - Set a default value that is added when no other value is passed
- **UNSIGNED** - Used for number types, limits the stored data to positive numbers and zero
- **AUTO INCREMENT** - MySQL automatically increases the value of the field by 1 each time a new record is added
- **PRIMARY KEY** - Used to uniquely identify the rows in a table. The column with PRIMARY KEY setting is often an ID number, and is often used with AUTO_INCREMENT
- Each table should have a primary key column (in this case: the "id" column). Its value must be unique for each record in the table.

MySQL Data Types

Text data types:

Data type	Description
CHAR(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters
VARCHAR(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters. **Note:** If you put a greater value than 255 it will be converted to a TEXT type
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT	Holds a string with a maximum length of 65,535 characters
BLOB	For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data

MySQL Data Types

- **Number data types:**
- INT(size) -2147483648 to 2147483647 normal. 0 to 4294967295 UNSIGNED*. The maximum number of digits may be specified in parenthesis
- FLOAT(size,d) A small number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter
- **Date data types:**
- DATE() A date. Format: YYYY-MM-DD**Note:**
- DATETIME() *A date and time combination. Format: YYYY-MM-DD

PHP Create MySQL Tables

- **Example (MySQLi Object-oriented)**
- ```
  <?php
  $servername = "localhost";
  $username = "username";
  $password = "password";
  $dbname = "myDB";

  // Create connection
  $conn = new mysqli($servername, $username, $password, $dbname);
  // Check connection
  if ($conn->connect_error) {
      die("Connection failed: " . $conn->connect_error);
  }

  // sql to create table
  $sql = "CREATE TABLE MyGuests (
  id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  firstname VARCHAR(30) NOT NULL,
  lastname VARCHAR(30) NOT NULL,
  email VARCHAR(50),
  reg_date TIMESTAMP
  )";

  if ($conn->query($sql) === TRUE) {
      echo "Table MyGuests created successfully";
  } else {
      echo "Error creating table: " . $conn->error;
  }

  $conn->close();
  ?>
  ```
  ```

Create Table if Not Exists

```
CREATE TABLE IF NOT EXISTS tasks (
    task_id     INT AUTO_INCREMENT PRIMARY KEY,
    subject     VARCHAR (255)       DEFAULT NULL,
    start_date  DATE                DEFAULT NULL,
    end_date    DATE                DEFAULT NULL,
    description VARCHAR (400)       DEFAULT NULL
);
```

PHP Insert Data Into MySQL

- After a database and a table have been created, we can start adding data in them.
- Here are some syntax rules to follow:
- The SQL query must be quoted in PHP
- String values inside the SQL query must be quoted
- Numeric values must not be quoted
- The word NULL must not be quoted
- The INSERT INTO statement is used to add new records to a MySQL table:
- *INSERT INTO table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...)*

PHP Insert Data Into MySQL

- **Example (MySQLi Object-oriented)**
- ```
  <?php
  $servername = "localhost";
  $username = "username";
  $password = "password";
  $dbname = "myDB";

  // Create connection
  $conn = new mysqli($servername, $username, $password, $dbname);
  // Check connection
  if ($conn->connect_error) {
      die("Connection failed: " . $conn->connect_error);
  }

  $sql = "INSERT INTO MyGuests (firstname, lastname, email)
  VALUES ('John', 'Doe', 'john@example.com')";

  if ($conn->query($sql) === TRUE) {
      echo "New record created successfully";
  } else {
      echo "Error: " . $sql . "<br>" . $conn->error;
  }

  $conn->close();
  ?>
  ```
  ```

PHP Select Data From MySQL

- The SELECT statement is used to select data from one or more tables:
- *SELECT column_name(s) FROM table_name*
- or we can use the * character to select ALL columns from a table:
- *SELECT * FROM table_name*

PHP Select Data From MySQL

- **Example (MySQLi Object-oriented)**
-
```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>
```

PHP Update Data in MySQL

- **Update Data In a MySQL Table Using MySQLi and PDO**
- The UPDATE statement is used to update existing records in a table:
- *UPDATE table_name*
  SET column1=value, column2=value2,...
  WHERE some_column=some_value
- **Notice the WHERE clause in the UPDATE syntax:** The WHERE clause specifies which record or records that should be updated. If you omit the WHERE clause, all records will be updated!

PHP Update Data in MySQL

- **Example (MySQLi Object-oriented)**
-

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn->error;
}

$conn->close();
?>
```

PHP Delete Data From MySQL

- The DELETE statement is used to delete records from a table:
- *DELETE FROM table_name*
  WHERE some_column = some_value
- **Notice the WHERE clause in the DELETE syntax:** The WHERE clause specifies which record or records that should be deleted. If you omit the WHERE clause, all records will be deleted!

PHP Delete Data From MySQL

- **Example (MySQLi Object-oriented)**

- ```
  <?php
  $servername = "localhost";
  $username = "username";
  $password = "password";
  $dbname = "myDB";

  // Create connection
  $conn = new mysqli($servername, $username, $password, $dbname);
  // Check connection
  if ($conn->connect_error) {
      die("Connection failed: " . $conn->connect_error);
  }

  // sql to delete a record
  $sql = "DELETE FROM MyGuests WHERE id=3";

  if ($conn->query($sql) === TRUE) {
      echo "Record deleted successfully";
  } else {
      echo "Error deleting record: " . $conn->error;
  }

  $conn->close();
  ?>
  ```
  ```

PHP File/Image Upload

STEP-I: Create The HTML Form

```
<form action="upload.php" method="post" enctype="multipart/form-data">
    Select image to upload:
    <input type="file" name="fileToUpload" id="fileToUpload">
    <input type="submit" value="Upload Image" name="submit">
</form>
```

Some rules to follow for the HTML form above:

Make sure that the form uses **method="post"**

The form also needs the g attribute: **enctype="multipart/form-data".** It specifies which **content-type to use** when submitting the form

Without the requirements above, the file upload will not work.

Other things to notice:

The **type="file"** attribute of the <input> tag shows the input field as a file-select control, with a "Browse" button next to the input control

File/Image Upload

STEP-II : Check for uploading error-

```
if(isset($_FILES["image"]) && $_FILES["image"]["error"] == 0){
```

STEP-III: Find extention of uploaded file:

```
$expld= explode('.',$_FILES['image']['name']); // explode file name at "."
$end= end($expld);  // getting  last item as extention from $expld array
$file_ext= strtolower($end); // converting to lower case
```

STEP-IV: **Allowing only desired extention**

```
$allowed = array("jpg" => "image/jpg", "jpeg" => "image/jpeg", "gif" =>
    "image/gif", "png" => "image/png")  //create an associative array of
    allowed extentions
if(!array_key_exists($file_ext, $allowed)) die("Error: Please select a
    valid file format."); // checking  allowed array key
```

File / Image Upload

STEP-V : Checking Maximum allowed size of Image

```
$filesize = $_FILES["image"]["size"];  //Getting Size of Uploaded File/Image
$maxsize = 1.0 * 1024 * 1024;  //Setting Maximum Size
 if($filesize > $maxsize) die("Error: File size is larger than the allowed limit.");  // Allowing Image/File with less than maxsize
```

STEP –VI : **Setting desired file name and folder name to save uploaded file:**

```
$fname=uniqid();  // crecting a unique file name using current time
$filename=$fname."".$file_ext; //Final File Name
$destination='file/'.$filename;  //Path where  uploaded file will be saved
```

File/Image Upload

STEP-VII : Moving Uploaded file to desired location

```
else{
        move_uploaded_file($_FILES["image"]["tmp_name"],
$destination);
        echo "Your file was uploaded successfully.";


    }
}                   else{
      echo "Error: There was a problem uploading your file. Please try
again.";
    }
```

mysqli_error()

- ```
  <?php
  $con=mysqli_connect("localhost","my_user","my_password","my_db");
  // Check connection
  if (mysqli_connect_errno())
    {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }

  // Perform a query, check for error
  if (!mysqli_query($con,"INSERT INTO Persons (FirstName) VALUES
  ('Glenn')"))
    {
    echo("Error description: " . mysqli_error($con));
    }

  mysqli_close($con);
  ?>
  ```
  ```

HTML 5 Form Validation

Form Validation with HTML5

HTML5 includes a fairly solid form validation mechanism powered by the following <input /> attributes: type, pattern, and require.

<input type="email" >

<input type="email" /> validates the field to ensure the entered data is in fact a valid email address. If the field contains an invalid value, the form cannot be submitted for processing until it is corrected.

Telephone numbers:

<label for="phonenum">Phone Number:</label>

<input **type="tel" pattern="**/(6|7|8|9)\d{9}$/ “>

Alpha-Numeric Values

The following matches an alpha-numeric (combination of alphabets and numbers) character.

<input type="text" **pattern="[a-zA-Z0-9]+"** >

Js Image Slider

HTML....

```
<body>
<h2>Automatic Slideshow</h2>
<div class="slideshow-container">
<div class="mySlides fade">
  <div class="numbertext">1 / 3</div>
  <img src="img_nature_wide.jpg" style="width:100%">
  <div class="text">Caption Text</div>
</div>
// Block for each Image
<div style="text-align:center">
  <span class="dot"></span>
  <span class="dot"></span>
  <span class="dot"></span>
</div>
```

Js Image Slider ….CSS

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {box-sizing: border-box;}
body {font-family: Verdana, sans-serif;}
.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}

/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}
```

Js Image Slider..CSS

```
/* Number text (1/3 etc) */
.numbertext {
  color: #f2f2f2;
  font-size: 12px;
  padding: 8px 12px;
  position: absolute;
  top: 0;
}

/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}

.active {
  background-color: #717171;
}
```

Js Image Slider … CSS

```
/* Fading animation */
.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 1.5s;
}

@-webkit-keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}
```

JS Imag Slider …. JavaScript

```
<script>
var slideIndex = 0;
showSlides();

function showSlides() {
  var i;
  var slides = document.getElementsByClassName("mySlides");
  var dots = document.getElementsByClassName("dot");
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  slideIndex++;
  if (slideIndex > slides.length) {slideIndex = 1}
  for (i = 0; i < dots.length; i++) {
     dots[i].className = dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
  setTimeout(showSlides, 2000); // Change image every 2 seconds
}
</script>
```

PHP Sending Data to Server

Web browsers offer four basic mechanisms that can be used to place data into the HTTP GET or POST request the browser makes to the server:

links

- clicking a link triggers a GET request to be made to the server

forms

- submitting a form can trigger either a GET or POST request to be made to the server

javascript

- a Javascript technique known as AJAX can be used to trigger either a GET or POST request to the server

cookies

- a web server can ask the web browser to keep a piece of data (for example, an ID number) that is then repeatedly sent back to the server with every subsequent request

CSS Transform

- CSS transforms allow you to translate, rotate, scale, and skew elements.
- A transformation is an effect that lets an **element change shape, size and position.**
- CSS supports 2D and 3D transformations.

The translate() Method

- ```
  div {
      -ms-transform: translate(50px, 100px); /* IE 9 */
      -webkit-transform: translate(50px, 100px); /* Safari */
      transform: translate(50px, 100px);
  }
  ```

The rotate() Method : The rotate() method **rotates an element** clockwise or counter-clockwise according to a given degree.

```
div {
      transform: rotate(20deg);
  }
```

Using negative values will **rotate the element counter-clockwise**.
```

CSS Transform

The scale() Method

The scale() method **increases or decreases the size of an element** (according to the parameters given for the width and height).

```
div {
        transform: scale(2, 3);
    }
```

The following example decreases the element to be half of its original width and height:

```
div {
        transform: scale(0.5, 0.5);
}
```

CSS Transitions

CSS transitions allows you to **change property values smoothly** (from one value to another), **over a given duration.**

To create a transition effect, you must specify two things:

- the **CSS property you want to add an effect to**
- the **duration of the effect**

```
div {
    width: 100px;
    height: 100px;
    background: red;
    transition: width 2s;
}
```

The transition effect will start when the specified CSS property (width) changes value.

Now, let us specify a new value for the width property when a user mouses over the element:

```
div:hover {
    width: 300px;
}
```

Notice that when the cursor mouses out of the element, it will gradually change back to its original style.

CSS Transitions

Change Several Property Values

The following example adds a transition effect for both the width and height property, with a duration of 2 seconds for the width and 4 seconds for the height:

```
div {
        transition: width 2s, height 4s;
}
```

Specify the Speed Curve of the Transition

- ease - specifies a transition effect with **a slow start, then fast, then end slowly (this is default)**
- **linear** - specifies a transition effect with the **same speed from start to end**
- **ease-in** - specifies a **transition effect with a slow start**
- **ease-out** - specifies a **transition effect with a slow end**
- **ease-in-out** - specifies a **transition effect with a slow start and end**

CSS Transition

Transition + Transformation

The following example also adds a transformation to the transition effect:

```
div {
    width: 100px;
    height: 100px;
    background: red;
     transition: width 2s, height 2s, transform 2s;
}

div:hover {
    width: 300px;
    height: 300px;
     transform: rotate(180deg);
}
```

CSS Animation

An animation lets an element gradually change. from one style to another

You can change as many CSS properties you want, as many times you want.

To use CSS animation, you **must first specify some keyframes** for the animation.

Keyframes hold what styles the element will have at certain times.

The @keyframes Rule

- When you specify CSS styles inside the **@keyframes** rule, the animation will gradually change from the current style to the new style at certain times.
- To get an animation to work, **you must bind the animation** to an **element.**

CSS Animation

- The following example binds the "**example**" **animation** to the element. The animation will last for 4 seconds, and it will gradually change the background-color of the element from "red" to "yellow":

```
/* The animation code */
@keyframes example {
    from {background-color: red;}
    to {background-color: yellow;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

Note: The animation-duration property defines how long time an animation should take to complete. If the animation-duration property is not specified, no animation will occur, because the default value is 0s (0 seconds).

CSS Animation

It is also possible to use percent. By using percent, you can add as many style changes as you like.

The following example will change the background-color of the element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

```
/* The animation code */
@keyframes example {
    0%   {background-color: red;}
    25%  {background-color: yellow;}
    50%  {background-color: blue;}
    100% {background-color: green;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

CSS Animation

- **Set How Many Times an Animation Should Run**
- The animation-iteration-count property specifies the number of times an animation should run.
- The following example will run the animation 3 times before it stops:

```
div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
    animation-iteration-count: 3;
}
```

We can also set animation-iteration-count: **infinite;** **for infinite loop**

CSS Animation

Run Animation in Reverse Direction or Alternate Cycles

The **animation-direction** property can have the following values:

normal - The animation is played as normal (forwards). This is default

reverse - The animation is played in reverse direction (backwards)

alternate - The animation is played forwards first, then backwards

alternate-reverse - The animation is played backwards first, then forwards

```
div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
    animation-direction: reverse;
}
```

CSS Animation

Specify the Speed Curve of the Animation

The animation-timing-function property can have the following values:

ease - Specifies an animation with a slow start, then fast, then end slowly (this is default)

linear - Specifies an animation with the same speed from start to end

ease-in - Specifies an animation with a slow start

ease-out - Specifies an animation with a slow end

ease-in-out - Specifies an animation with a slow start and end

cubic-bezier(n,n,n,n) - Lets you define your own values in a cubic-bezier function

```
div
{
    animation-name: example;
     animation-duration: 4s;
     animation-direction: reverse;
    animation-timing-function: linear
;}
```

Ajax Introduction

- Update a web page without reloading the page
- Request data from a server - after the page has loaded
- Receive data from a server - after the page has loaded
- Send data to a server - in the background
- <body>
-
- <h1>The XMLHttpRequest Object</h1>
- <button type="button" **onclick="loadDoc()"**>Change Content</button>
-

```
<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>
```

What is Ajax

What is AJAX?

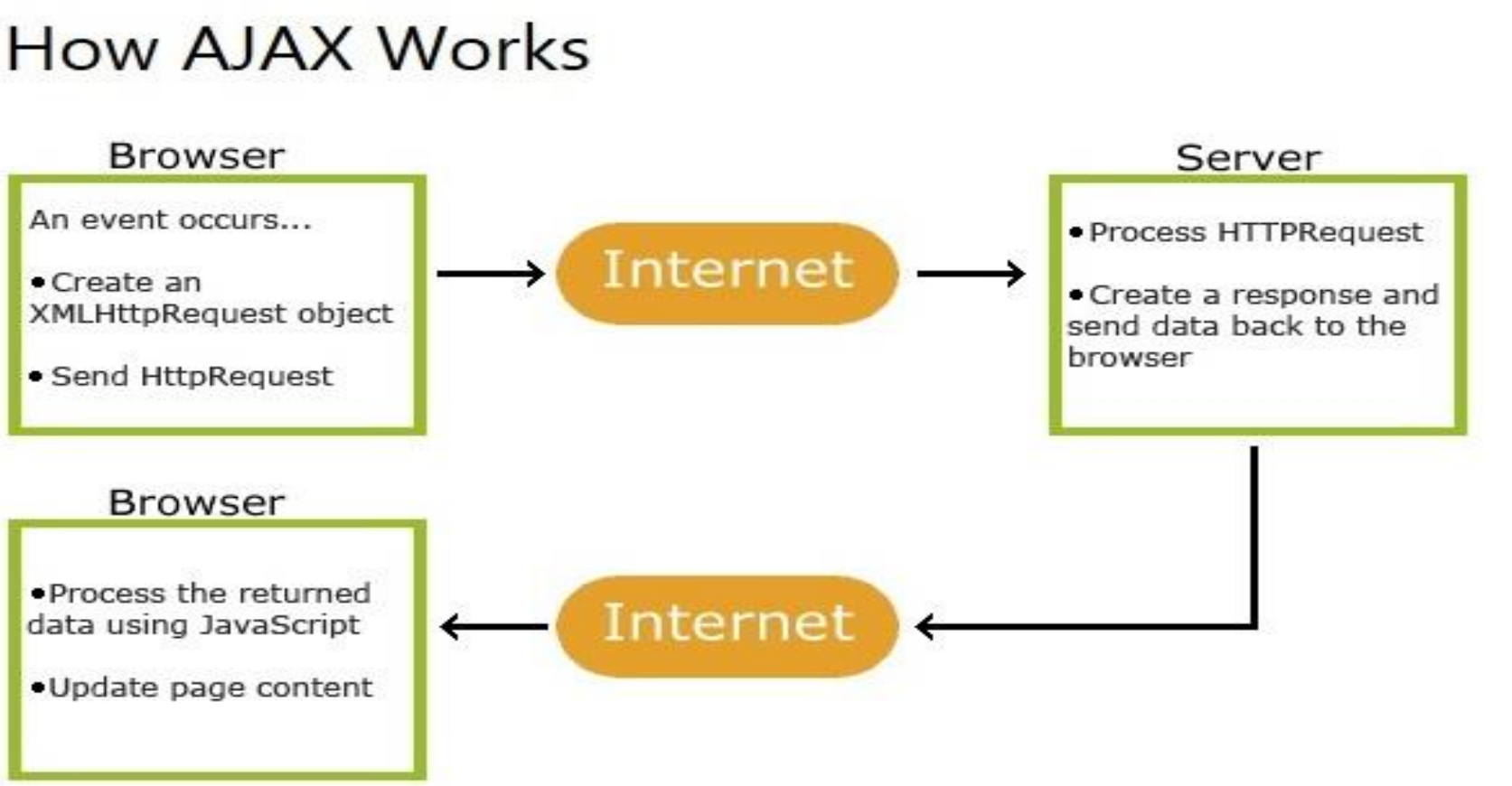
AJAX = Asynchronous JavaScript And XML.

AJAX is not a programming language.

AJAX just uses a combination of: A browser built-in **XMLHttpRequest** object (to request data from a web server) JavaScript and HTML DOM (to display or use the data)

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

How Ajax Works



Steps in Ajax

1.An event occurs in a web page (the page is loaded, a button is clicked)

2. An **XMLHttpRequest object** is created by JavaScript

3. The XMLHttpRequest object sends a request to a web server

4. The server processes the request

5. The server sends a response back to the web page

6. The response is read by JavaScript

7. Proper action (like page update) is performed by JavaScript

XMLHttpRequest Object

- **Create an XMLHttpRequest Object**
- *variable* = new XMLHttpRequest();

XMLHttpRequest Object Methods

Method	Description
new XMLHttpRequest()	Creates a new XMLHttpRequest object
abort()	Cancels the current request
getAllResponseHeaders()	Returns header information
getResponseHeader()	Returns specific header information
open(*method,url,async,user,psw*)	Specifies the request *method*: the request type GET or POST *url*: the file location *async*: true (asynchronous) or false (synchronous) *user*: optional user name *psw*: optional password
send()	Sends the request to the server Used for GET requests
send(*string*)	Sends the request to the server. Used for POST requests
setRequestHeader()	Adds a label/value pair to the header to be sent

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

PHP Session

- A session is a methods of **storing** data (using variables) so the browser can use it throughout **multiple** webpages. In contrast to **cookies**, the data is not kept on the user's system.
- Session **variables** contain data about the current user, and all pages contained in a single web application .
- The session **ends** once the window or tab in which the webpage was loaded, is closed.

Start a Session

- **Starting PHP Sessions**
- To **start** a session, you must use the function **session_start().**
- To **set** session variables, you will need to apply a global variable $_SESSION.
- **Note:** The session_start() function must be the **very first thing** in your document. Before any HTML tags.
- <?**php session_start();?>** // Start the session
- <!DOCTYPE html>
- <html>
- <body>
- <?php
- **$_SESSION["color"]= "blue";** // Set session variables
- **$_SESSION["animal"]= "dog";**
- echo "The session variable are set up.";
- ?>
- </body>
- </html>

Getting Session Veriable Value

- Notice how session data in form of variables must be **individually** retrieved (session_start()).
- In addition to that the $_SESSION variable holds all of the session data declared in your
- <?php **session_start();?>**
- <!DOCTYPE html>
- <html>
- <body>
- <?php// Echo session variables that were set on previous page
- echo "The color of my choice is " **. $_SESSION["color"]** . ".
";
- echo "The animal of my choice is " **. $_SESSION["animal"] .** ".";
- ?>
- </body>
- </html>

Display All Session Veriables

- <?**php session_start**();?>
- <!DOCTYPE html>
- <html>
- <body>
- <?php
- **print_e($_SESSION);**
- ?>
- </body>
- </html>
- **Modifying PHP Session Variables**
- <?php //you can change a session variable by asigning a new value **$_SESSION["color"] = "red";**
- print_e($_SESSION);
- ?>

Destroying Session

- Using session_unset() will remove all of the global variable, while session_destroy() will destroy the session entirely (however the effect of both is similar):
- <?**phpsession_start();?>**
- <!DOCTYPE html>
- <html>
- <body>
- <?php
- **session_unset();** //remove all session variables
- **session_destroy();** // destroy the session
- ?>
- </body>
- </html>