

# Web Technology Fundamentals

Module Number: 3

**Module Name: HTML and PHP**

**AIM:**

To get familiarize about the usage of PHP handling and HTML using Mark UP Language

## Objectives:

- Hypertext Mark up language
- Link List , tables and forms
- Create Web pages through HTML
- Principles of SQL

## Outcome:

- How to create Web pages using HTML
- Procedures of PHP Handling
- How to display table and forms
- SQL Functions and its features

# Contents

## 1. Part I - An Introduction to Markup Languages

- HyperText Markup Language – HTML
- Basics of HyperText Markup Language
- Links, Lists, Tables, Frames and Forms

## 2. Part II – PHP and MySQL

- Introduction
- PHP Basics
- PHP Form Handling
- PHP/MySQL Functions
- Displaying Queries in Tables

- Building Forms from Queries

# Part I - Mark-up Languages

What is a **Mark-up**  
**Languages?**

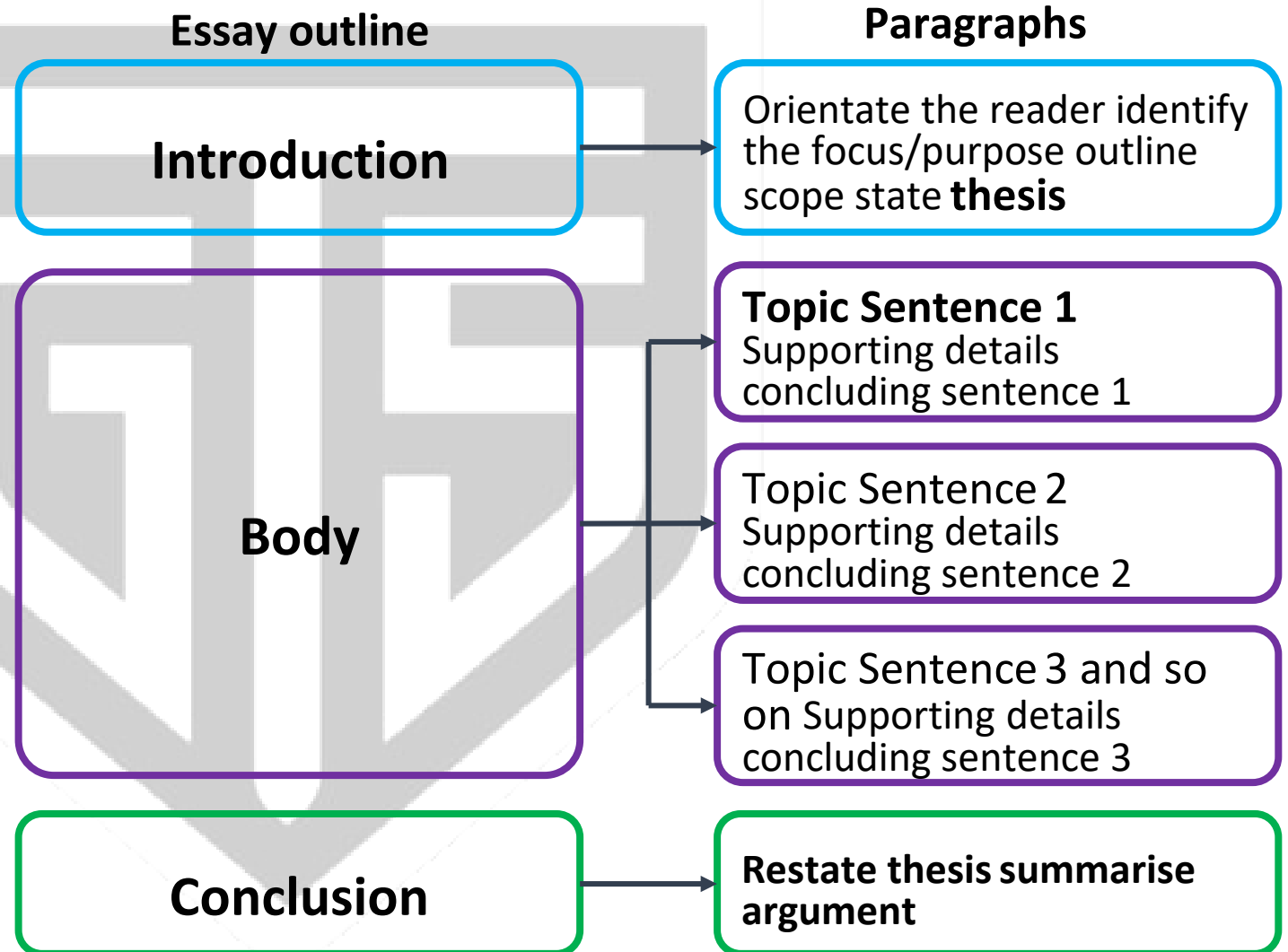


## Example – writing an essay

Consider the example of writing an essay, it is marked up as:

- Introduction Section
- Body Section
- Conclusion Section

Markup is formatting the data you have into the proper format.



**So, a markup language is designed for the processing and formatting of text.**

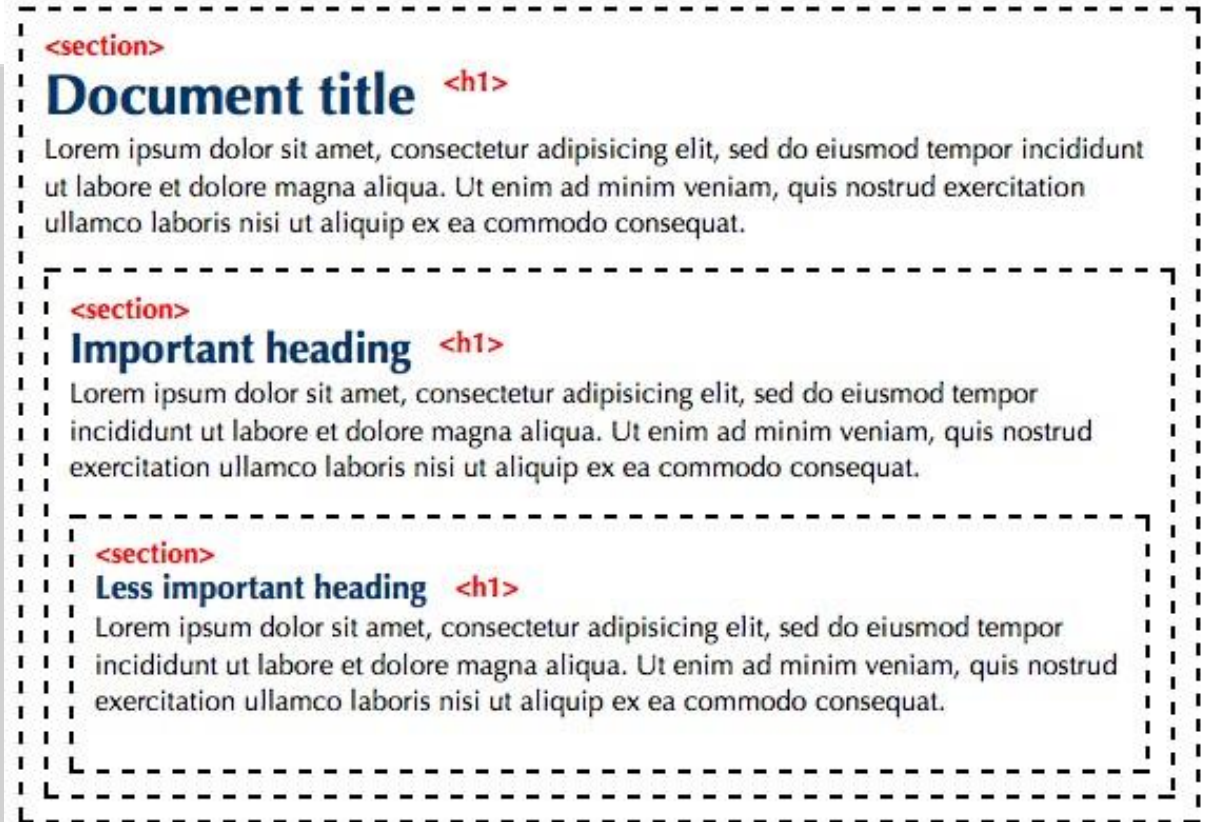


**A Markup language is?**

## So Markup languages are.

Formatting and processing text includes:

1. Putting text in paragraphs
2. Dividing it into sections
3. Giving headings and subheadings
4. Highlight the important words.
5. Creating lists and tables.





## Some Popular Markup languages

The code used to specify the formatting is called tags.



Hyper Text Markup language




eXtensible Markup Language



eXtensible Hyper Text Markup Language

## Know More

### Mark up Languages

 The link is an article about markup languages, it looks at three markup languages – HTML, XML and XHTML.

Topic	URL
History about Mark up languages	<a href="https://alistapart.com/article/a-brief-historyof-markup">https://alistapart.com/article/a-brief-historyof-markup</a>

## Self Assessment Questions

1. An example of a markup language is \_\_\_\_\_.
  - a) C++
  - b) XHTML
  - c) Java
  - d) Python

**Answer: b)**

2. Marking up a document includes:

- a) Dividing the content into various sections.
- b) Writing data in tabular form wherever appropriate.
- c) Creating images.
- d) Giving heading to each section.

**Answer: a), b), d).**

# Hyper Text Mark-up Language

HTML



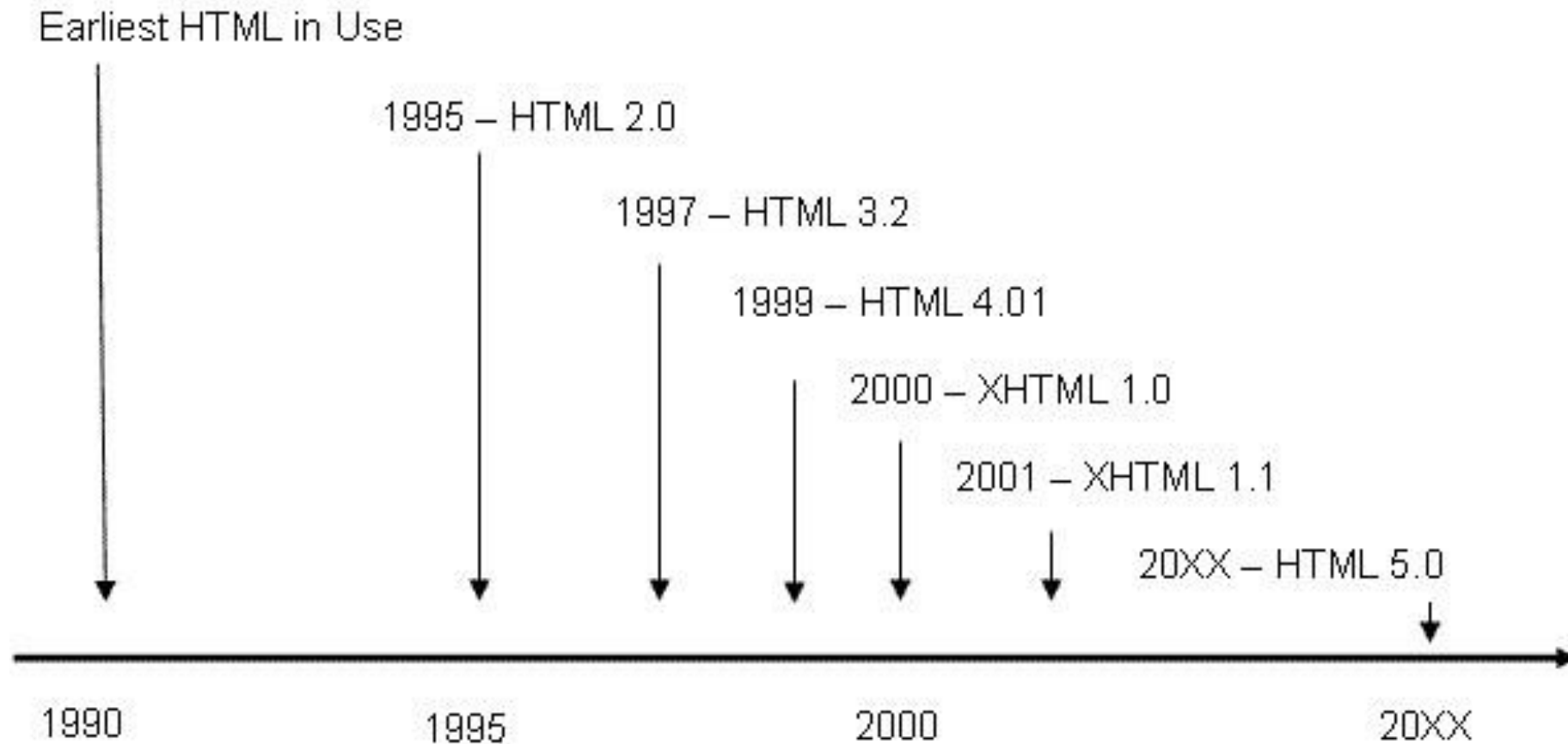


Tim Berners-Lee

**Which is the first**

**HTML editor?**

## Evolution of HTML



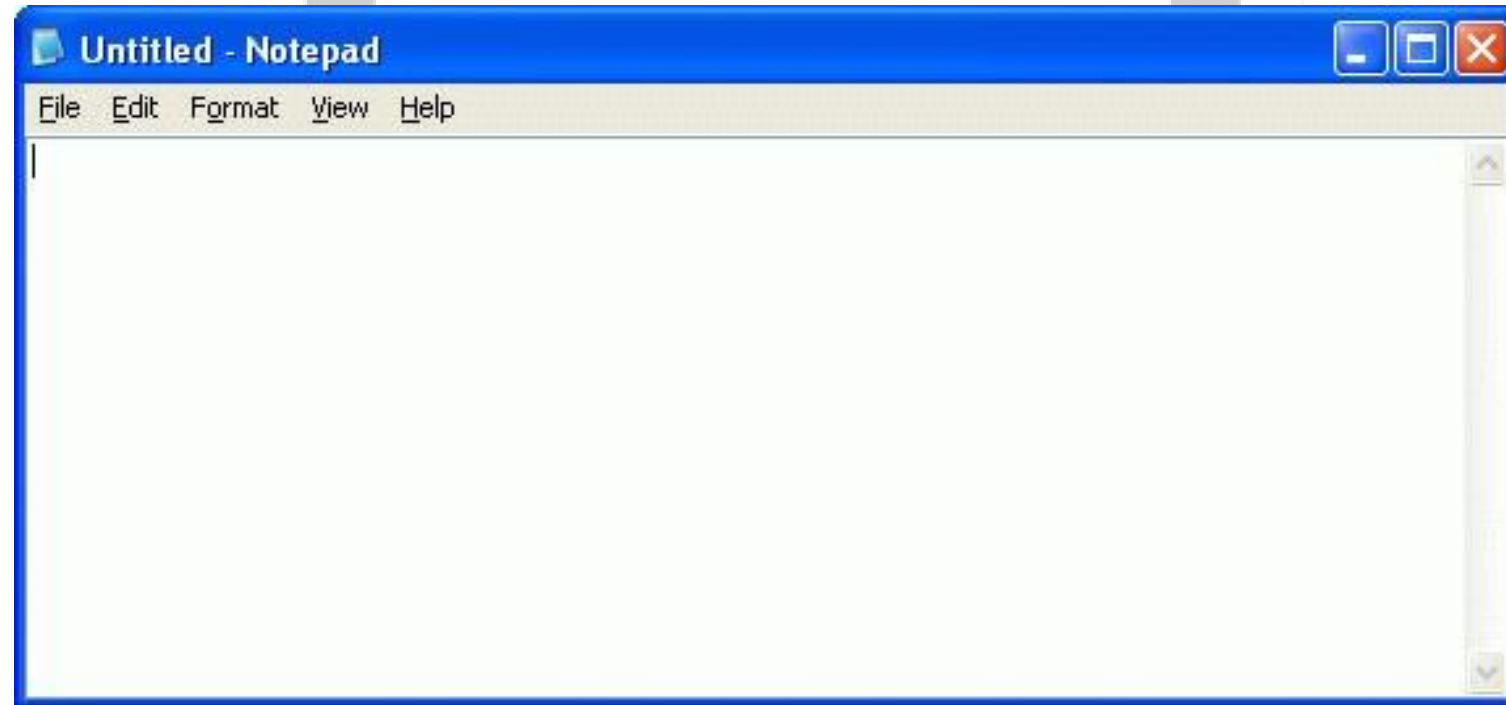
# How to create and view an HTML document ?

The following slides show the steps to create an HTML document.

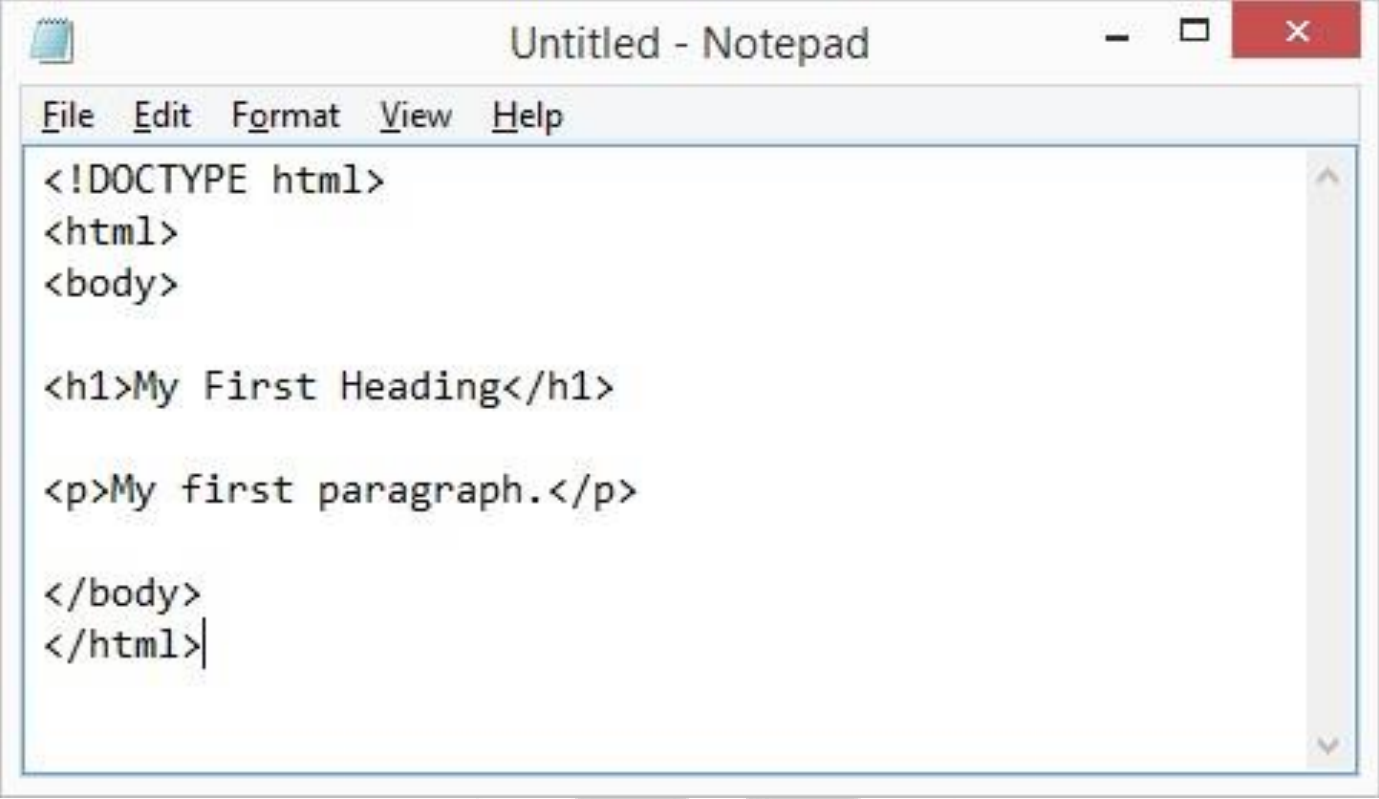




# 1. Open a text editor (e.g. Notepad)



## 2. Write HTML content in the file

A screenshot of a Notepad window titled "Untitled - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text area contains the following HTML code:

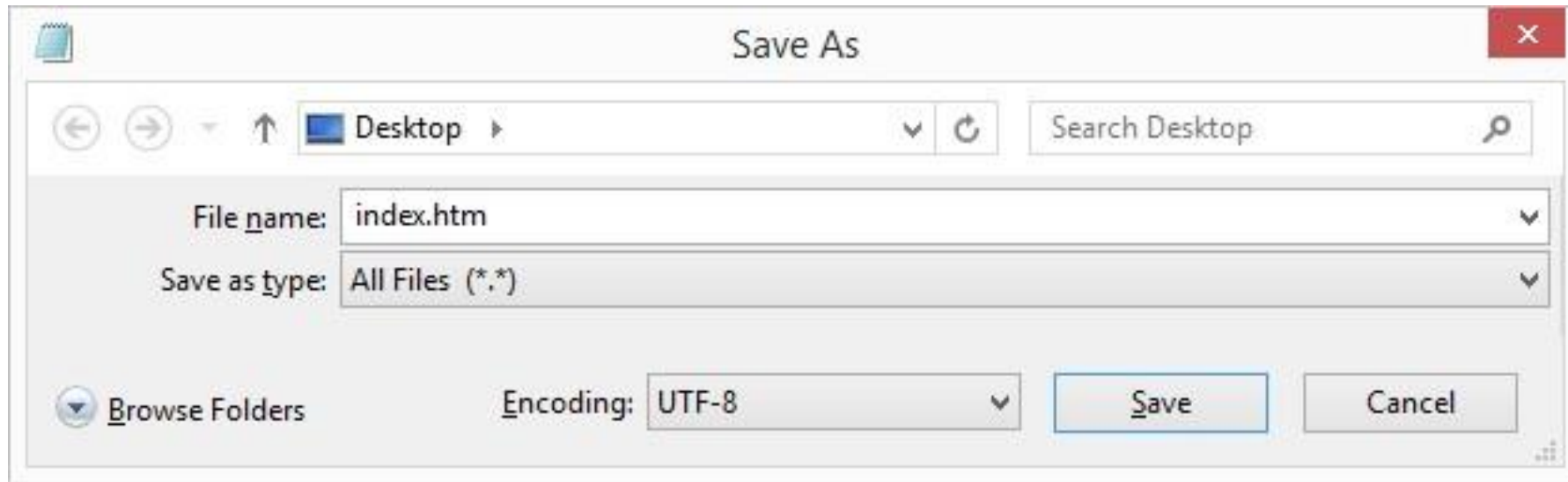
```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

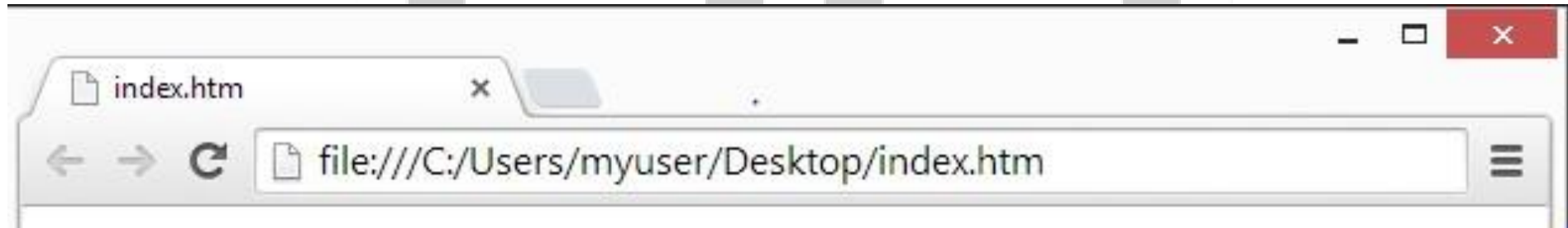
<p>My first paragraph.</p>

</body>
</html>
```

## 3. Save the file on your system by giving file extension .html or .htm.



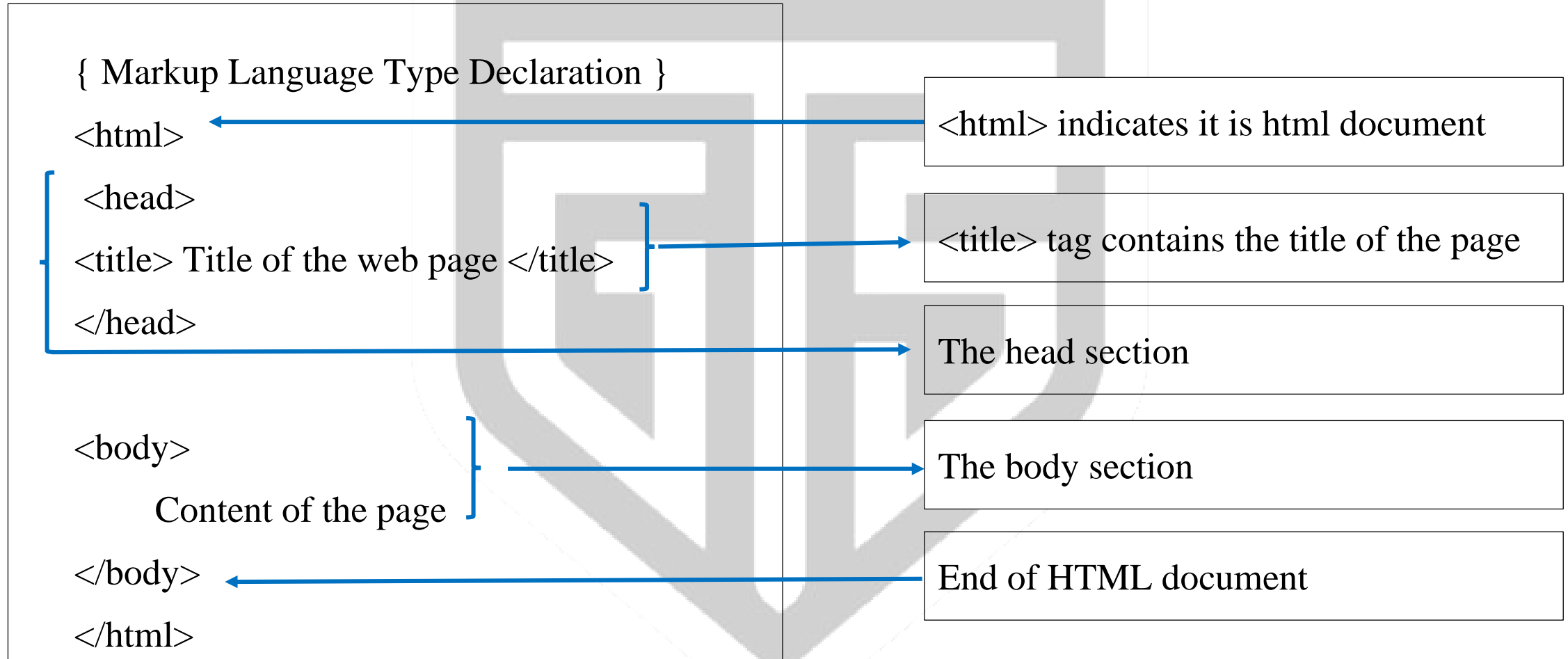
4. **Open your Web browser (Internet Explorer, Chrome or Firefox) and enter the path of the HTML file**



5. **The browser will display the web page.**



## Basic HTML Document Structure



## Structure



{ Markup Language Type  
Declaration }

`<html>`

`<head>`

`<title> { Descriptive Text Here }`

`</title>`

`</head>`

`<body>`

{ Main content here }

`</body>`

`</html>`

## Example

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title> The Page Title </title>
```

```
</head>
```

```
<body>
```

```
<h1> Page Heading </h1>
```

```
<p> This is my first HTML document </p>
```

```
</body>
```

```
</html>
```



## Document type Declarations

Syntax	<!DOCTYPE HTML PUBLIC "html version">
HTML 2 as standardized by the Internet Engineering Task Force	<! DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
HTML 3.2	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN >
HTML 4.01	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
XHTML	<DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
HTML5	<!DOCTYPE html>

GROW TECH SKILLS

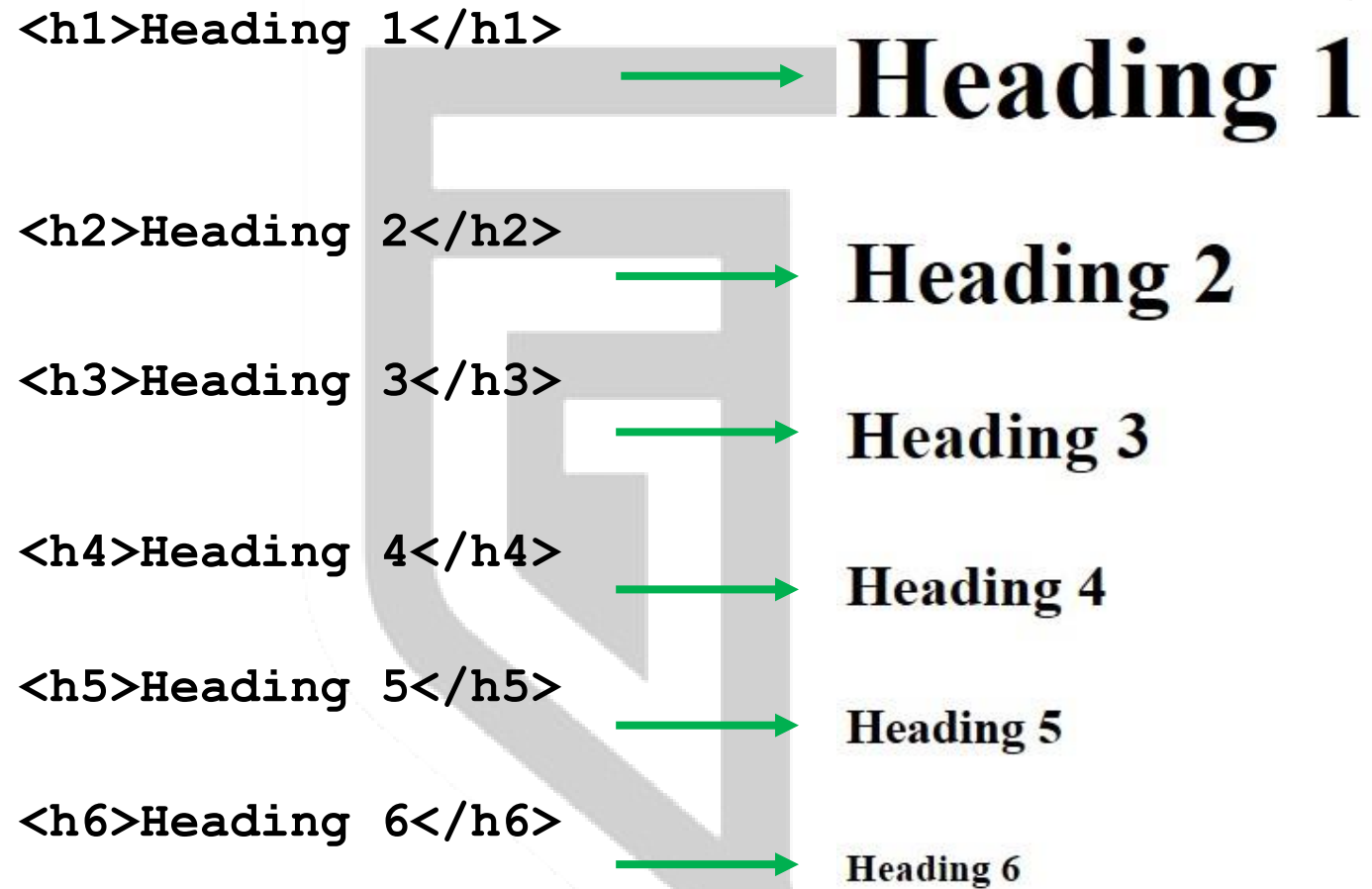
## **HTML Comments**



## Headings – Six levels

*<!-- This is a comment -->*

**Output**



## Physical Character tags

**Bold**This is

`<b> bold </b>` This is **bold**

**Italic**This is

`<i> italics </i>` This is *italic*

**Subscript**`H<sub>2</sub>O`H<sub>2</sub>O

**Superscript**`Mar`

`31<sup>st</sup>2010`Mar 31<sup>st</sup>2010

2010

**Small**This is

`<small>small</small>` This is small

**Underlined**`This is <u> underlined` `</u>`This is underlined

**Big**This is `<big> big </big>`This is big

## Logical Character tags

### 1. Citation – displays in italics

This is a short `<cite> quote </cite>`

This is a short *quote*.

### 2. Code – displays in courier font

`<code> x = y * m +2; </code>`

x = y \* m + 2

### 3. Text with a line through`<del>deleted</del>` text deleted-text

### 4. Definition – displays in italics

`<dfn>` Internet is a network of networks. `</dfn>` *Internet*  
*is a network of networks.*

**5. Emphasized text – displays in italics** This is `<em>` emphasized text `</em>`

This is *emphasized text*

**6. Program variable – displays in italics**

This is a `<var>` variable `</var>`

This is a *variable*

## Paragraphs and breaks

### Code Snippet

```
<p> This is a paragraph </p>
```

And this line follows the paragraph.

```
<p>This is a paragraph <br> and contains a  
line break </p>
```

### Output

This is a paragraph  
And this line follows the paragraph.

This is a paragraph  
And contains a line break

## Know More

HTML Basics



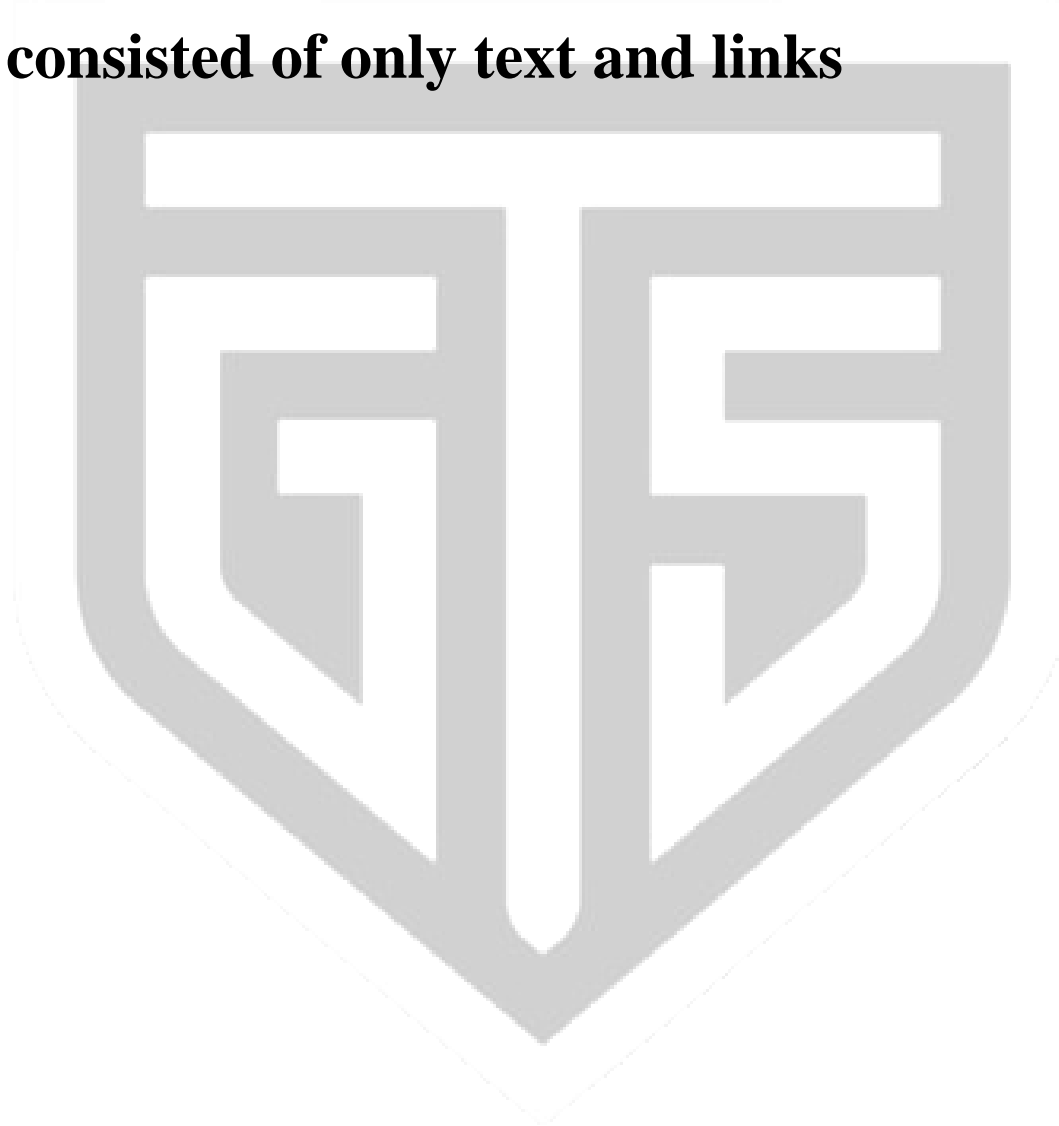


This video covers basic of HTML.

Topic	URL
Basic of HTML	<a href="https://www.youtube.com/watch?v=hrZqiCUx6kg">https://www.youtube.com/watch?v=hrZqiCUx6kg</a>

# When did the first website go online?

**The first ever website consisted of only text and links**



# World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#)

## [What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

## [Help](#)

on the browser you are using

## [Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,X11 [Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#) )

## [Technical](#)

Details of protocols, formats, program internals etc

## [Bibliography](#)

Paper documentation on W3 and references.

## [People](#)

A list of some people involved in the project.

## [History](#)

A summary of the history of the project.

## [How can I help ?](#)

If you would like to support the web..

## [Getting code](#)

Getting the code by [anonymous FTP](#) , etc.



## Creating Hyperlinks

*`<a href = “URL”> anchor text </a>`*

**Example external hyperlink**

*`<a href = “http://www.ignou.ac.in”> IGNOU </a>`*

## Example internal hyperlink

Consider the following code,

```
<!-- At the top of the page -->
```

```
<h1> <a name=“top”> Introduction </a> </h1>
```

```
<!-- Some more code goes here -->
```

```
<!-- At the end of the page -->
```

```
<a href=“#top”> back to top </a>
```

- The first line marks `<h1>` as the destination.
- The `<a>` tag displays back to top as a link which when clicked by the user will display the “Introduction section” of the page.

**Images make a web page more interesting!**





## Images

```

```

### Attributes:

- `src = "url"`

Identifies the file to be loaded into a document.

- **align** = “top”, “middle”, “bottom”, “left”, or “right” Positions the image in the document.
- **alt** = “message”

Provides the alternate message to be shown if image cannot be displayed.

## Example:

```
<img src = “kitten.jpg” alt=“Picture of a kitten” align=“center”>
```

- Loads a jpeg\* file named “kitten” into the document at the location of the element and center aligned.



- If the file is not available the message, '**Picture of a kitten**' is displayed.

## **Lists Display a list of things**

1. I am ordered
2. Second in the list
  - I am unordered
  - Second in the list

## Ordered Lists



Ordered list numbering	Description
<b>type ="1"</b>	This type of list is numbered with numbers.
<b>type ="I"</b>	This type of list is numbered with uppercase roman numbers.
<b>type ="i"</b>	This type of list is numbered with lowercase roman numbers.
<b>type ="A"</b>	This type of list is numbered with uppercase letters.
<b>type ="a"</b>	This type of list is numbered with lowercase letters.

**General form :**

`<ol type = "numbering">`

`<li> list-item </li>`

`<li> list-item </li>`

`<li> list-item </li>`

`</ol>`

## Unordered Lists



Item marker	Description
<b>type ="disc"</b>	Sets the item marker to bullets (default)
<b>type ="circle"</b>	Sets the item marker to a circle.
<b>type ="square"</b>	Sets the item marker to a square.
<b>type ="none"</b>	This is list will not be marked

**General form :**

```
<ul type = "type of bullets">
```

```
<li> list-item </li>
```

```
<li> list-item </li>
```

```
<li> list-item </li>
```

```
</ul>
```

## **Lists Example**



- **Example A - Ordered List**

```
<h4> Web Technologies I want to learn. </h4>
```

```
<OL>
```

```
  <LI> HTML </LI>
```

```
  <LI> CSS </LI>
```

```
  <LI> JAVA SCRIPT </LI>
```

```
</OL>
```

**Web Technologies I want to learn.**

1. HTML
2. CSS
3. JAVA SCRIPT

- **Example B – Unordered List**

```
<h4> Server-side technologies </h4>
```

```
<UL>
```

```
  <LI> PHP </LI>
```

```
  <LI> AJAX </LI>
```

```
  <LI> JSP </LI>
```

```
</UL>
```

**Server-side technologies**

- PHP
- AJAX
- JSP

## Know More

### HTML Lists and Images



Explains how links and images are created in HTML.

Topic	URL
How Links and Images created in HTML	<a href="https://www.youtube.com/watch?v=3my2mOOHoNU">https://www.youtube.com/watch?v=3my2mOOHoNU</a>



# Tables

**All about rows and columns**

**Tables**

	Column 1	Column 2	Column 3
Row 1	cell	cell	cell
Row 2	cell	cell	cell
Row 3	cell	cell	cell

## Tables Types



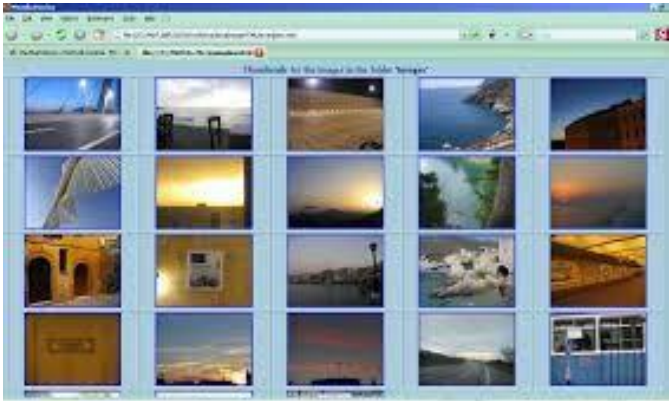


Table 1		
Table 2		
Header 1	Header 2	Header 3

Figure 3: Table within a table

Figure 1: Table with images

ID	Name	Phone	Email
24	Alexandra Nixon	(422) 644-3488	<a href="#">nec.luctus@ornarefacilis.co</a>
17	Alisa Monroe	(859) 974-4442	<a href="#">adipiscing.ligula@aretraNam</a>
10	Baker Osborn	(378) 371-0559	<a href="#">turpis.Nulla@ac.edu</a>
9	Caldwell Larson	(850) 562-3177	<a href="#">elit@dolor.com</a>
25	Charissa Manning	(438) 395-9392	<a href="#">nibh.vulputate@necelendnor</a>
48	Charity Hahn	(395) 200-9188	<a href="#">ac@Quisque.edu</a>
30	Dorian Hodge	(304) 536-8850	<a href="#">pellentesque@laoreet.org</a>
50	Eden Burks	(576) 196-6013	<a href="#">lorem@magna.com</a>
1	Ezekiel Hart	(627) 536-4760	<a href="#">tortor@est.ca</a>
12	Fletcher Briggs	(992) 962-9419	<a href="#">amet.ante@lentesque.edu</a>

Figure 2: Table with links

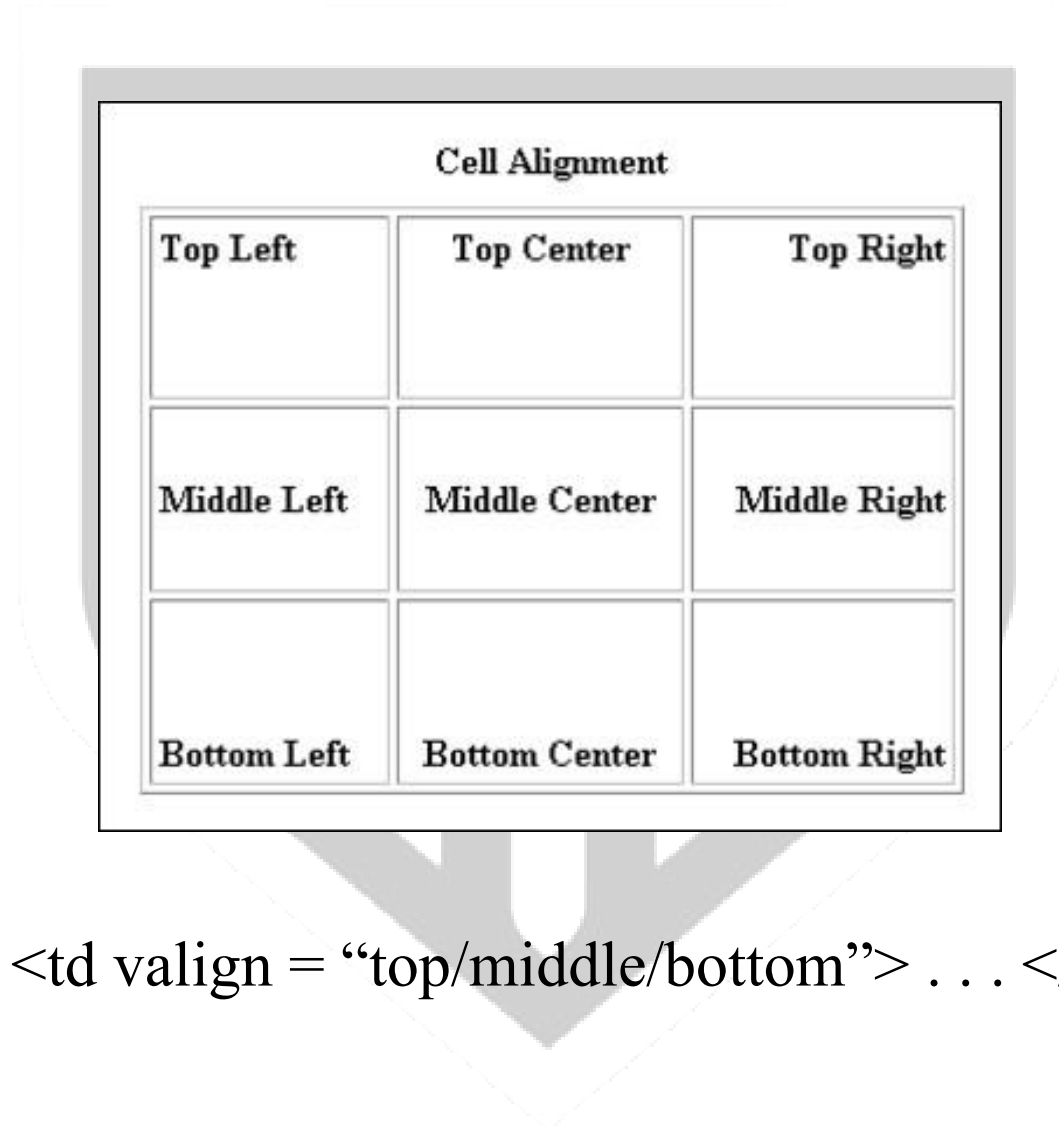
# Colspan and Rowspan Attributes

1 colspan="3"			4	5	
6 rowspan="4"	7	8	9 colspan=2 and rowspan="3" together merge cells in 2 directions.		
	12	13			
	17	18			
	22	23	24	25	

<th colspan = “x”>...</th>

<td rowspan = “y”>...</td>

# Cell Alignment



```
<td align = “left/center/right”> . . . </td>
```



## **An Example Code Snippet**





```
<table width="500px"
border="1px"
bordercolor="green"
bgcolor="orange">
```

```
<tr>
<th> Sl.No. </th>
<th> First Name </th>
<th> Last Name </th>
<th> Phone no. </th>
</tr>
```

```
<tr>
<td> 1 </td>
<td> Ramesh </td>
<td rowspan="2"> m </td>
<td> 9837472723 </td>
</tr>
<tr>
<td> 2 </td>
<td> Suresh </td>
<td> 9837472724 </td>
</tr>
<tr>
<td> 3 </td>
```

```
<td> Rajesh </td>
<td> t </td>
<td> 9837472725 </td>
</tr>
<tr>
<td colspan="2">Total Students :
3</td>
</tr>
</table>
```

## Sample Output for code in previous slide

Sl.No.	First Name	Last Name	Phone no.
1	Ramesh	m	9837472723
2	Suresh		9837472724
3	Rajesh	t	9837472725
Total Students : 3			

## Know More

HTML Tables



Explains how to create tables in HTML

Topic	URL
How to create tables in HTML	<a href="https://www.youtube.com/watch?v=BczLWImAmBk">https://www.youtube.com/watch?v=BczLWImAmBk</a>

# Forms

## Getting information from the user

Forms



# Login Form

Fill the required fields

First Name:

Last Name:

India

Welcome to the form attributes

firstname:

lastname:

email:

Text box

Submit Button

Drop Down List

Text area

## General Form

`<form action = “script URL” method = “GET|POST” >`

- `form elements`

- `</form>`

## Input Element

The input element is used to define the different controls in a form.

`<input type = “control name”>`

- **Text box** – for one-line text input field

```
<input type="text" name="firstname">
```

A simple rectangular text input field with a black border. A green arrow points from the code snippet to this field.

- **Password** – password input field

```
<input type="password" name="firstname">
```

A rectangular password input field with a black border. A green arrow points from the code snippet to this field.

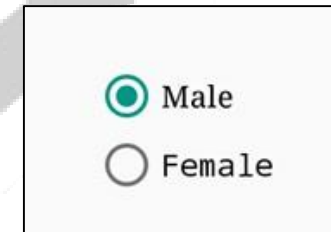
- **Radio** – for radio button

```
<select>
```

```
<option value="male">Male</OPTION>
```

```
<option value="female "> Female </OPTION>
```

```
</select>
```

A selection box containing two radio buttons. The first radio button is selected (indicated by a green dot) and is labeled "Male". The second radio button is unselected and is labeled "Female". A green arrow points from the code snippet to the "Male" option, and another green arrow points from the code snippet to the "Female" option.

- **Button** – clickable button

`<input type="button" onclick="alert('Hello!')" value="Click me">`

Click me

- **Submit** – submit button

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

Submit

- **Reset** – reset button

`<input type="reset" value="Reset">`

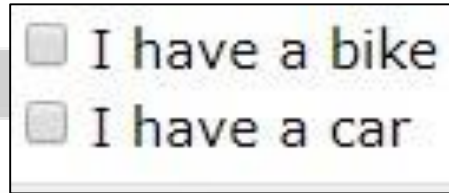
Reset

- **Checkbox** – checkbox

`<input type="checkbox" name="vehicle1" value="bike"> I have a bike <br>`

`<input type="checkbox" name="vehicle2" value="car"> I have a car`

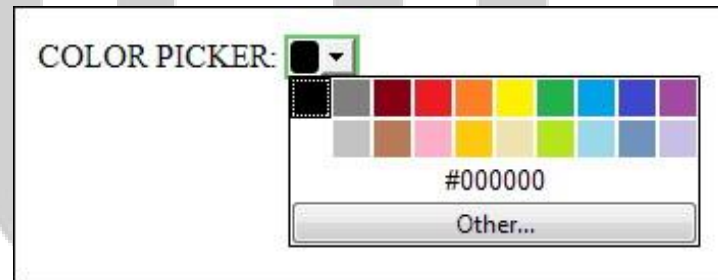




☐ I have a bike  
☐ I have a car

- **Color** – color picker is shown for selecting a color

`<input type="color" name="COLOR PICKER">`



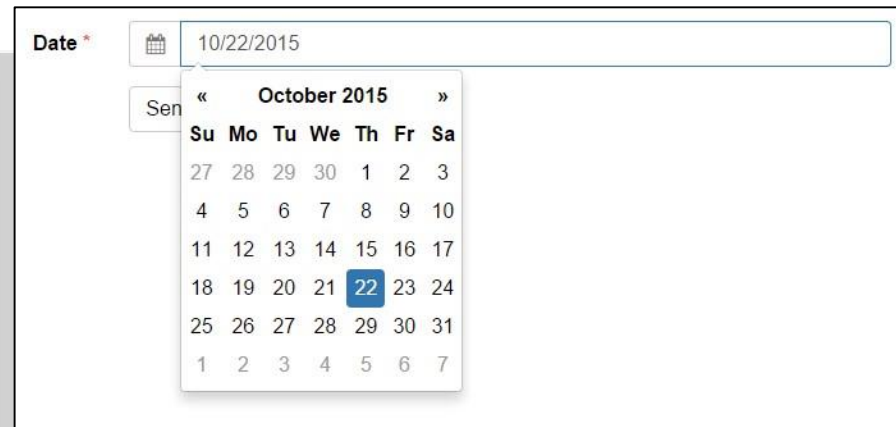
COLOR PICKER:

#000000

Other...

- **Date** – date picker is shown for selecting a date

`<input type="date" name="Date">`



The screenshot shows a web form with a 'Date' field and a 'Send' button. The 'Date' field is set to '10/22/2015' and has a calendar dropdown menu open. The calendar shows the month of October 2015, with the 22nd highlighted. Below the date field is a text input field.

October 2015						
Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

- **Email** – for field which require email as input `<input type="email" name="email">`



The screenshot shows a text area input field, which is a rectangular box for entering multiple lines of text.

- **Text area**  
`<textarea name="message" rows="10" cols="30">`  
The cat was playing in the garden.

```
</textarea>
```

The cat was playing in the garden.

Know More

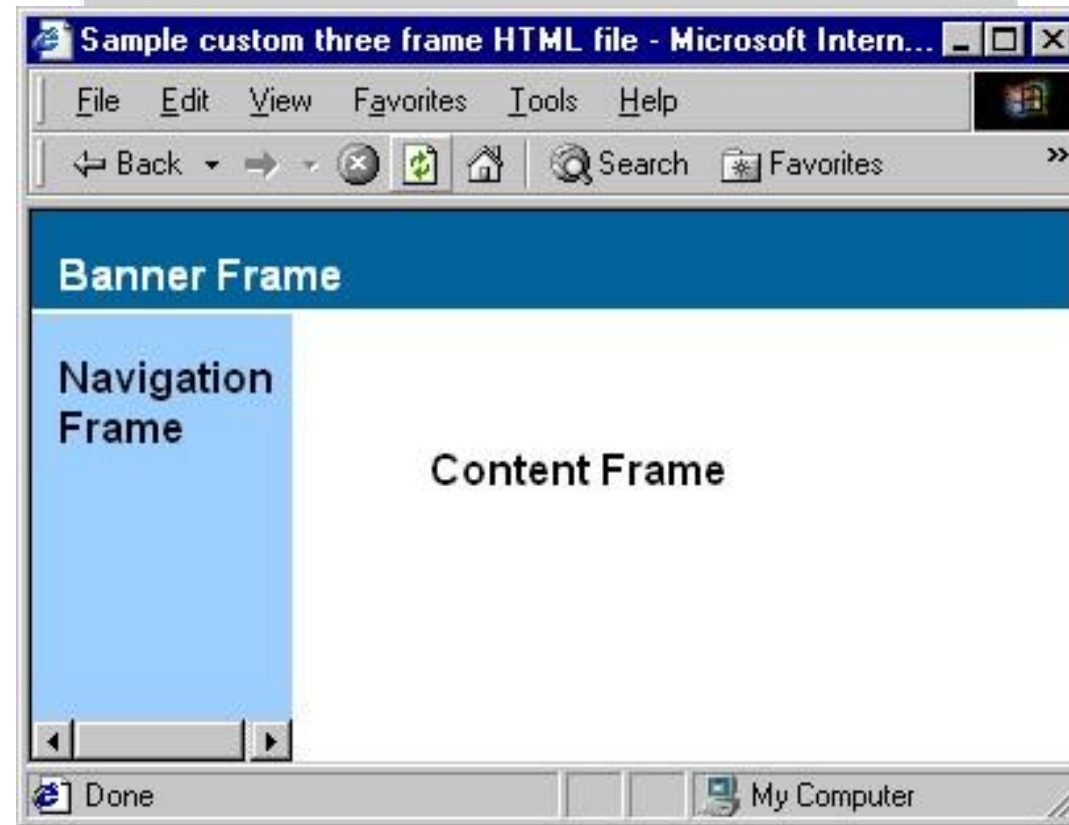
HTML Forms



It talks about how to create forms in HTML

Topic	URL
How to create Forms in HTML	<a href="https://www.youtube.com/watch?v=GNuCNQDGBnk">https://www.youtube.com/watch?v=GNuCNQDGBnk</a>

## Frames A browser window can have multiple frames



**The code snippet for creating a window similar to the one in the previous slide is as follows:**

```
<frameset rows = "20%,80%">  
  <frame name = "banner" src = "/html/top_frame.htm" />  
  <frameset cols = "30%,70%">  
    <frame name = "Navigation frame" src = "/html/main_frame.htm" />  
    <frame name = "content frame" src = "/html/bottom_frame.htm" />  
  </frameset>  
</frameset>  
<noframes>  
  <body> Your browser does not support frames. </body>  
</noframes>  
</frameset>
```

# Frames are not supported by HTML5

## Know More

HTML Frames



Explains frames and Frameset tag.

Topic	URL
Frames and Frameset tag	<a href="https://www.youtube.com/watch?v=Gi2mCS5f68A">https://www.youtube.com/watch?v=Gi2mCS5f68A</a>

## Issue No 1. Closing tags

```
<html>
  <head>
    <title> My Document </title>
  </head>
  <body>
    <h1> Heading 1 </h1>
    <p> This is paragraph whose closing tag has been forgotten by the author.
    <a href = "home.html" > Go to home page </a>
  </body>
</html>
```



## Heading 1

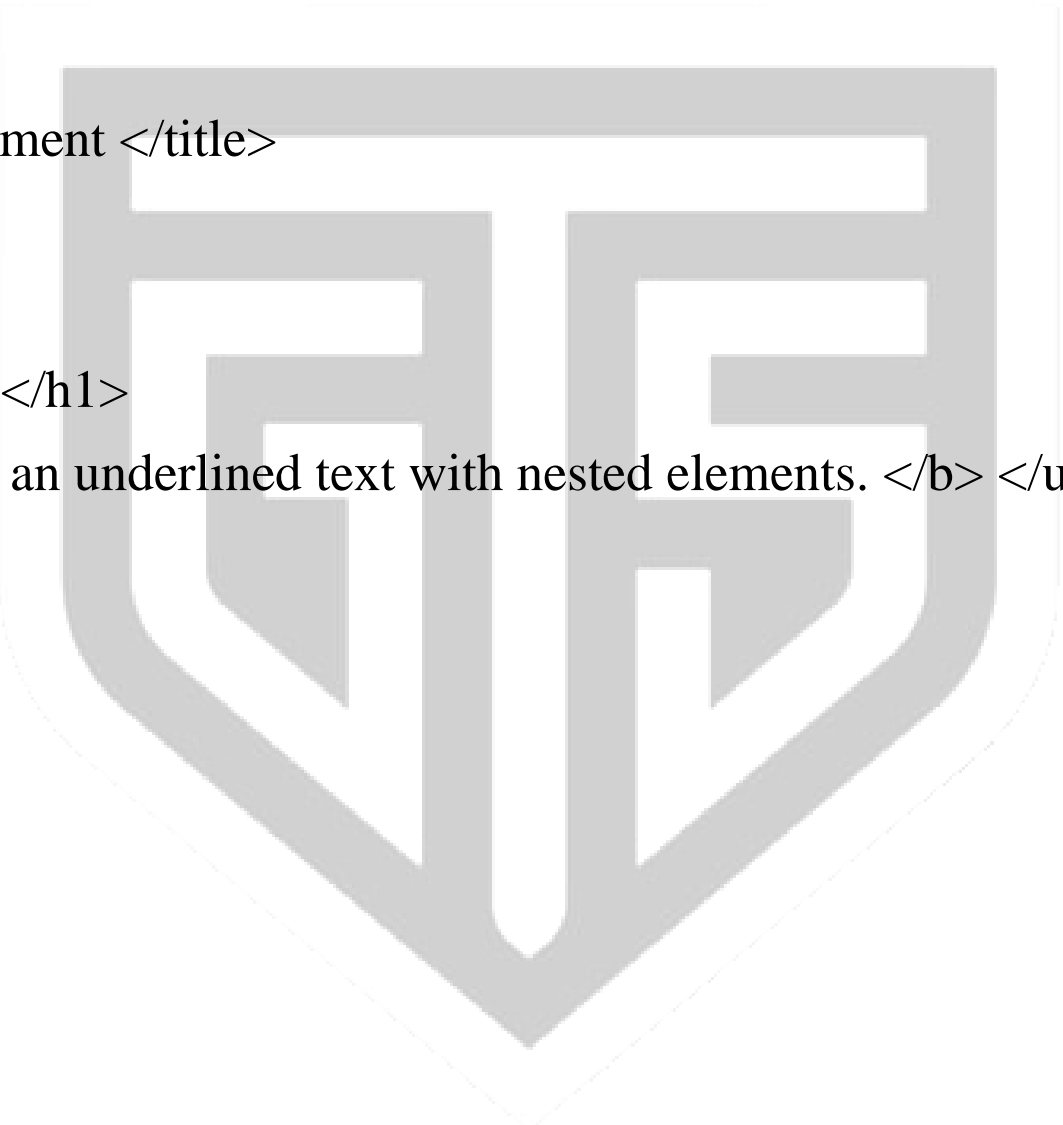
This is paragraph whose closing tag has been forgotten by the author. [Go to home page](#)

## Issue No 2. Improper nesting of tags

<html>



```
<head>  
  <title> My Document </title>  
</head>  
<body>  
  <h1> Heading 1 </h1>  
  <b> <u> This is an underlined text with nested elements. </b> </u>  
</body>  
</html>
```





# Heading 1

This is an underlined text with nested elements.



## Issue No 3. Custom tags

```
<html>
  <head>
    <title> My Document </title>
  </head>
  <body>
    <h1> My Address </h1>
    <address>
      11, stephens street, Bangalore – 560006
    </address>
  </body>
</html>
```



## Issue No 4. Human Readable, not machine readable

<p> Mr. RamSumit

<br>

No. 336, Stephens Road,

<br>

Frazer Town,

<br>

Bangalore, 560006

</p>

# Know More

## HTML

 Tutorial links for learning HTML.

Topic	URL
Tutorial of HTML	<a href="https://html.com/">https://html.com/</a>

## Self Assessment Questions

3. What will the browser do with the following nested element with mismatched end tags?

`<b> <i>this is confusing! </b></i>`

- a) Displays an error
- b) displays the text in bold and italics

**Answer: b)**

4. For Font size acceptable values are:

- a) 1 – 7 (1 for smallest and 7 for largest)
- b) Up to 40
- c) 1-10 (1 for smallest and 10 for largest)
- d) 1-17(1 for smallest and 17 for largest)

**Answer: a)**

5. What is wrong with the following code:

```
<ol type = “disc”>
```



```
<li> C </li>  
<li> C++ </li>  
<li> Java </li>  
<ol>
```

**Answer: Ordered list uses numbering. Disc is a bullet type used for unordered lists.**

6. \_\_\_\_\_ tag is used to create a header cell in a Table.

**Answer: <th> </th>**

7. Write the input tag for creating a password field.

**Answer: `<input type="password" name="psd">`**



8. Write HTML code for defining a 3x4 table as shown below,

Header1	Header2	Header3	
Data1	Data2	Data3	Data4
Data5	Data6	Data7	Data8

**Answer:** table>

```

<tr>
  <th> Header1 </th>      <th> Header2 </th>      <th colspan=2> Header3 </th>
</tr>
<tr>
  <td> Data1 </td> <td> Data2 </td> <td> Data3 </td> <td> Data 4 </td>
</tr>
<tr>
  <td> Data5 </td> <td> Data6 </td> <td> Data7 </td> <td> Data8 </td>
</tr>
</table>

```

9. Which of the following attributes of text box control allow to limit the maximum character? a) size
- b) maxlength
  - c) length
  - d) Ln

**Answer: b)**

10. To vertically align the cell data in an html table the attribute used is \_\_\_\_\_.

**Answer: Valign.**

11. The latest HTML standard is \_\_\_\_\_.

**Answer: HTML 5.1.**

12. An email link is created using:

- a) `<mail href="mail-id">`
- b) `<a href=mailto:mail-id>`
- c) `<a href="mail-d">`
- d) (B) and (C)

**Answer: b)**

13. HTML stands for \_\_\_\_\_.

**Answer: Hyper Text Markup Language**

14. HTML was created by:

- a) Robert Kahn
- b) Tim Berners Lee

**Answer: b)**

15. Tables were introduced in HTML version \_\_\_\_\_.

- a) HTML 2.0

b) HTML 3.2

**Answer: b)**

16. \_\_\_\_\_ was the first HTML editor.

- a) NeXt
- b) Notepad
- c) Nexus
- d) Eclipse

**Answer: c)**

17. HTML files must be viewed using \_\_\_\_\_ to see the output.

- a) The command prompt
- b) A browser

**Answer: b)**

18. The basic structure of an HTML document doesn't includes:

- a) Head
- b) Paragraph
- c) Body
- d) Title

**Answer: c)**

19. <hr> is used to:.



- a) Include a horizontal line.
- b) Include a line break.

**Answer: b)**

20. DOCTYPE specifies the HTML version being used.

- a) Yes
- b) NO

**Answer: b)**

21. HTML offers \_\_\_\_\_ levels of headings.

**Answer: 6 levels.**

22. The text placed between `<sup>` `</sup>` is displayed as:

- a) Strong text
- b) Superscript
- c) Subscript
- d) Super imposed text

**Answer: b)**

23. Logical tags are used:

- a) To format the characters.

- b) To indicate how the text is being used.

**Answer: b)**

24. The tag used to display text with a line through is:

- a) `<u></u>`
- b) `</del></del>`

**Answer: b)**

25. `<em> </em>` is used to:

- a) Emphasise the text.
- b) Indicate an empty variable

**Answer: a)**

26. The first ever website went online in \_\_\_\_\_.

a) 1991

b) 1993

**Answer: a)**

27. The first website included text, links and tables.

a) Yes

b) No

**Answer: b)**

28. A hyperlink:

- a) Links documents.
- b) Links tables with graphs.

**Answer: a)**

29. An internal hyperlink connects two sections in the same document.

- a) Yes
- b) No

**Answer: a)**

30. The tag for including an image is:

- a) ``
- b) ``

**Answer: b)**

31. Write the syntax for including alternate text for image in an HTML document.

**Answer: ``.**

32. Name the two types of lists.

**Answer: Ordered lists and unordered lists.**

33. `<ol>` is used to create \_\_\_\_\_ list.

**Answer: Ordered list.**

34. Write the code to create a list as shown:

- Cloud computing
- Big Data Analytics
- Machine Learning
- Internet of Things
- Information Security

**Answer:** <ul type =square>

- <li>Cloud computing </li>
- <li>Big Data Analytics</li>
- <li>Machine Learning </li>
- <li>Internet of Things </li>
- <li>Information Security </li>

</ul>

35. A table can contain:

- a) Data only
- b) Data and images
- c) Data and multimedia
- d) Data, multimedia and other tables



**Answer: d)**

36. <th> is used for:

- a) Adding header to the table
- b) Adding table data

**Answer: a)**

37. The possible values for the method attribute of the form tag are \_\_\_\_\_ and \_\_\_\_\_.

**Answer: GET and POST.**

38. The controls in a form are defined using:

- a) Control element
- b) Input element

**Answer: b)**

39. Write the difference between a text field and a password field.

**Answer:** When you type the characters in a password field are displayed as asterisk.

40. On clicking the submit button:

- a) The form handler specified as the action attribute is executed.
- b) Does nothing. **Answer: a)**

41. Write HTML code to create 3 checkboxes, displaying the options:

- a) Red
- b) Green
- c) Yellow

**Answer:** <form>

```
<input type="checkbox" name="red" value="red"> Red <br>  
<input type="checkbox" name="green" value="green"> Green<br>  
<input type="checkbox" name="yellow" value="yellow"> Yellow  
</form>
```

42. The visible width of a text area is specified using:

- a) Rows attribute
- b) Cols attribute

**Answer: a)**

43. A frame:

- a) Divides a window into multiple sections.
- b) Defines the border of an image.

**Answer: a)**

44. A collection of frames in a page is called \_\_\_\_\_.

**Answer: Frameset.**

45. HTML5 supports frames.

- a) Yes
- b) No

**Answer: b)**

46. Select the sentences that are correct.

- a) HTML is strict about closing tags.
- b) Improper nesting of tags creates an error.
- c) HTML is not extensible, hence custom tags are not allowed.
- d) HTML tags are not machine readable.

**Answer: c), d).**

47. The Doctype declaration for HTML 5 is \_\_\_\_\_.

**Answer: <DOCTYPE HTML>**

48. Which is the valid one?

- a) <!-- This is a comment -->

b) `<! This is a comment -->`

**Answer: a)**

49. Write html code to display the text “All the Best!” in bold and underlined.

**Answer: `<b><u> All the Best! </u></b>`**

50. `<cite>` displays the text in:

- a) Bold
- b) Italics

**Answer: b)**

51. The logical character tag to display a definition is \_\_\_\_\_.

- a) <dfn>
- b) <definition>

**Answer: a)**

52. Can rowspan and colspan attributes be used with <th> tag. (Yes/No) **Answer:**

**Yes.**

53. The text between <noframes> </noframes> is displayed when the browser does not support frames.

(Yes/No)

**Answer: Yes.**

# Assignment

## Assignment

1. Create a sample web page using HTML elements, include: a link, an image and a table.
2. Create an HTML document demonstrating the use of frames.
3. Using HTML form element create a student registration form with submit and reset buttons.

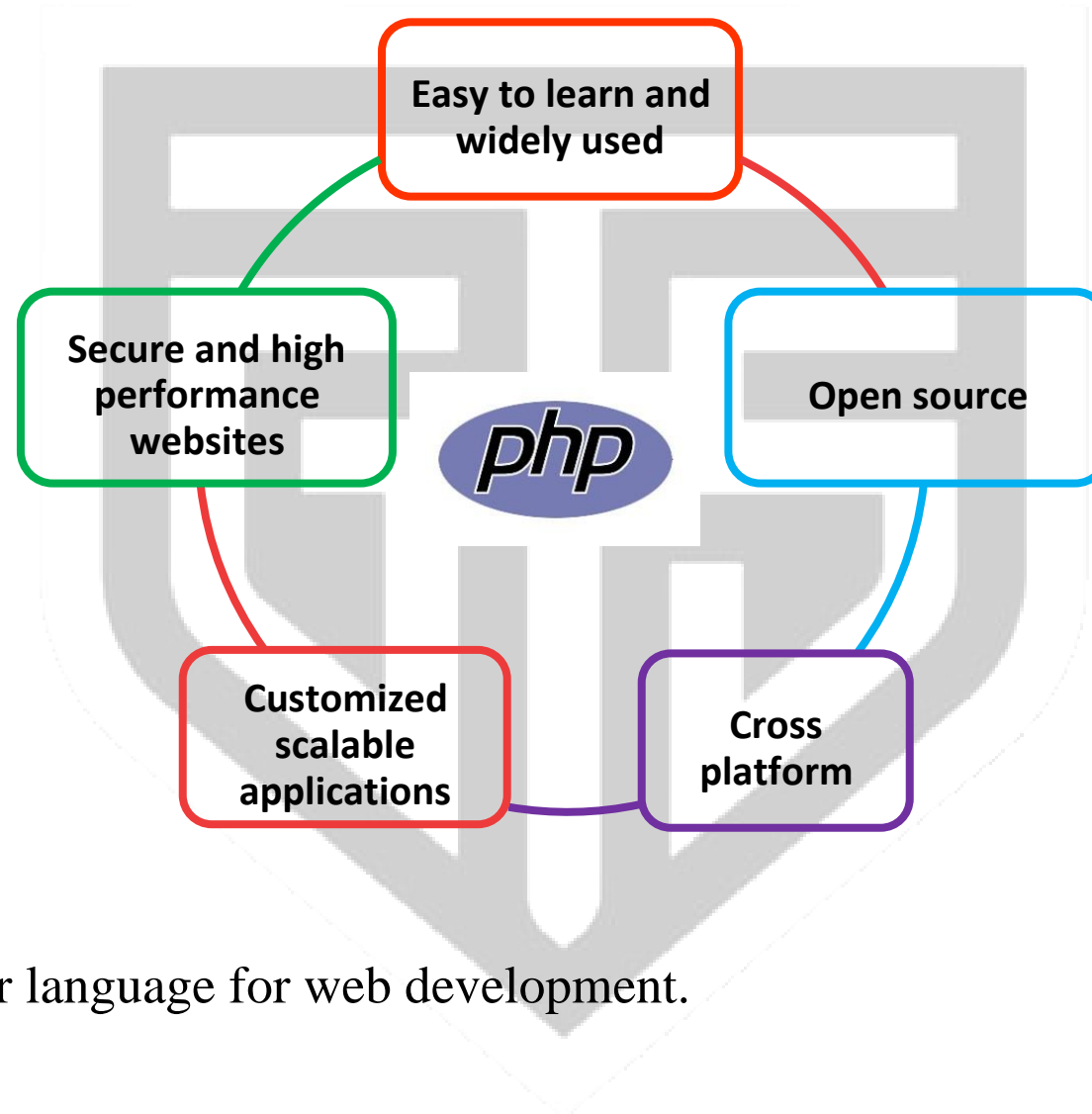


4. Prepare a document on Document Type Declarations in XHTML.

# **Part II – PHP and MySQL**

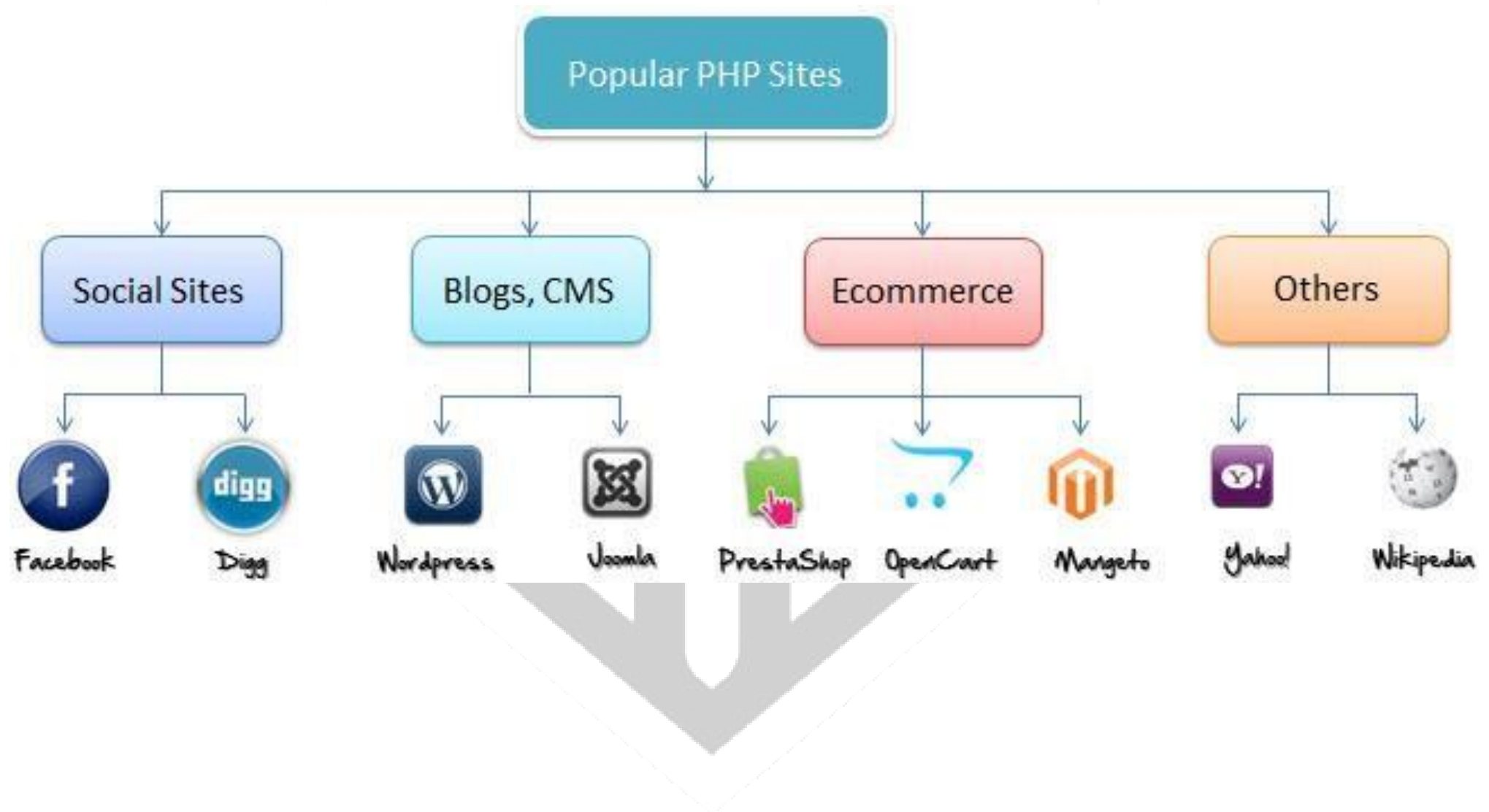
## **Why use PHP?**

PHP is the most popular language for web development.



## Popular PHP Sites

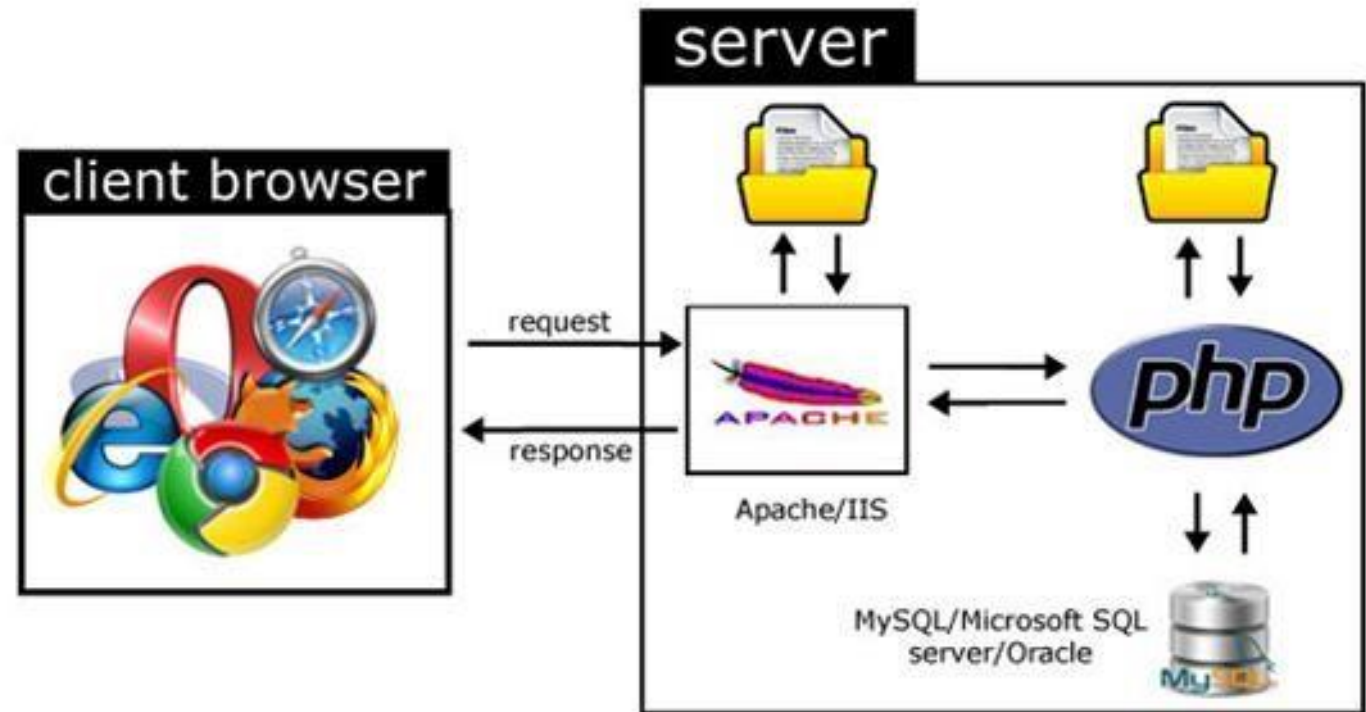
PHP is the most popular language for web development.




# How PHP Works



## Basic Structure



 = File System

 = Database

[webdevtutes.blogspot.com](http://webdevtutes.blogspot.com)

# Hypertext PreProcessor (PHP) - Basics

## PHP Syntax

**A PHP file has .php extension.**

Syntax:

```
<?php  
// PHP code goes  
here ?>
```

Example: HelloWorld.php

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

```
<h1>My first PHP page</h1>
```

```
<?php echo "Hello World!";  
?>
```

```
</body>
```

```
</html>
```

## PHP Syntax Rules

- **PHP is case sensitive.**  
Count and count are different.
- **Statements and expressions must end with a semicolon.**  
print("Hello World");
- **Curly braces are used for making blocks.**  
{

```
//Block of  
code; }
```

- **It is whitespace insensitive.** (backspace, tab, newline, carriage return) – all the given expressions are valid.

1. Sum=a+b;

2. Sum = a + b;

3. Sum = a  
+ b;

//no spaces

//spaces  
included

//newline  
included

## PHP Comments

### Single Line Comments (# and //):

1. # This is a single line comment in PHP.



2. // This too is a single line comment in PHP.

### **Multi-Lines Comments (/\*.. \*/):**

```
/* This is a multi line  
Comment in PHP.  
Uses the C style.*/
```

## **Variables**

### **Syntax:**

```
$variablename = value;
```

### **Examples:**

```
$num = 23;  
$name = "Jay";  
$Name = "Mehta";
```

```
$value = 3.4;
```

**Rules:**

- Starts with a \$ sign followed by the variable name.
- Variable name itself must begin with a letter or underscore.
- Can contain only letters (A-Z,a-z), number(0-9) and underscore(\_).
- Case sensitive.
- A string must be enclosed within double quotes.

**Echo and Print Statements**

There are two ways to display the output

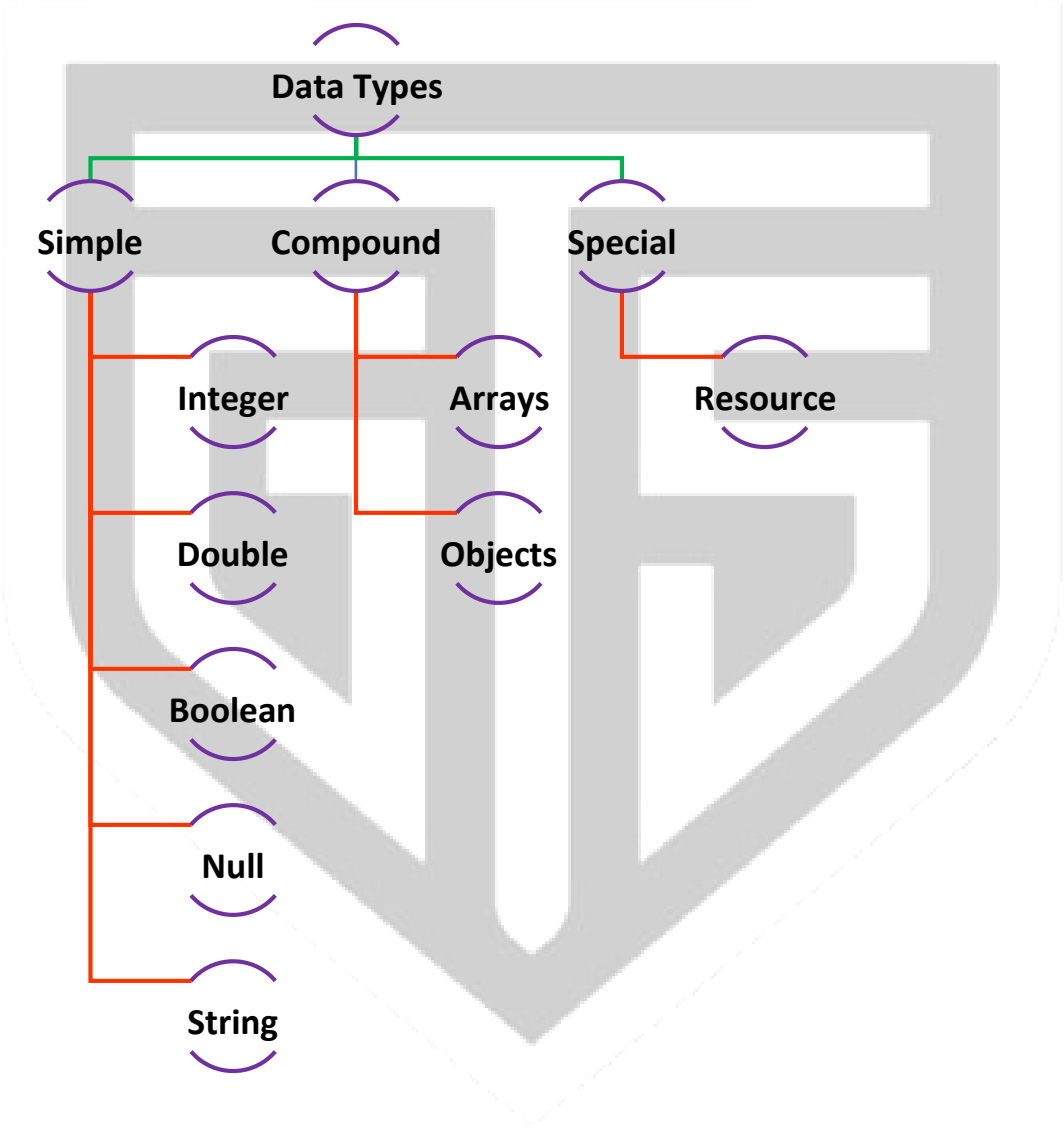
– **echo()** echo “Hello World!”; echo  
\$var;

```
print()      print("Hello  
World!"); print "Hello  
World"; print $var;
```



# Data Types in PHP





## Simple Data Types Examples

```
$n = 345;
```

```
$d = 3.14159;
```

```
$flag = true;
```

```
$str = "I like PHP";
```

```
$empty = "";
```

//Empty string

```
$no_value = NULL;
```

## Constants

**Constants are values which cannot be changed.**

**Syntax:**

The define() function is used to define a constant.

```
define ("constant name", value, case-insensitive)
```

**Example:**

```
define ("PI", 3.14); //creates a constant PI which is case-sensitive
```

```
define ("PI", 3.14, true); //creates case-insensitive constant PI
```

**Rules:**

- A constant name must start with a letter or an underscore.
- it can have alphabets (A-z,a-z), numbers (0-9) and underscore(\_).
- Once defined it cannot be changed.
- The define() function is used to define a constant.

- The constant name is used to access its value.
- The `constant()` function is used to access the constant dynamically.

## Predefined Constants

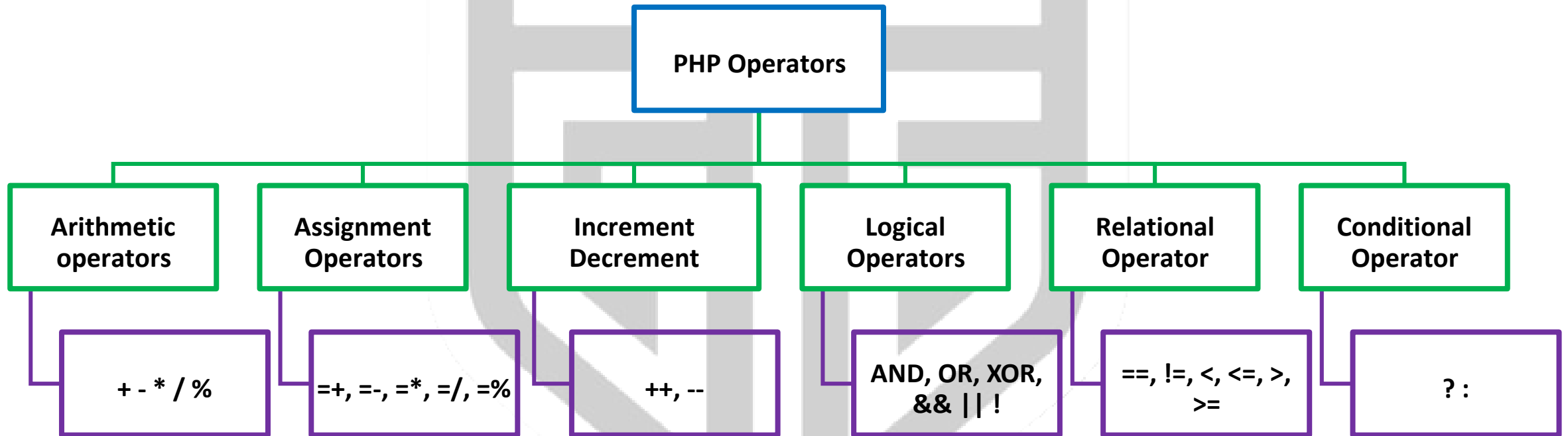
There are **five predefined constants** in PHP which **change depending on where they are used**.

Constant Name	Description
<code>_LINE_</code>	The current line number
<code>_FUNCTION_</code>	The function name



<b>_FILE_</b>	The full path and name of the file
<b>_CLASS_</b>	Name of the class
<b>_METHOD_</b>	Class method name

## Types of Operators



### Increment/Decrement Operators

Used to increment (++) or decrement (--) the value of a variable by 1.  
The operators can be used before (pre) or after (post) the variable.

**Example:**

```
$a = 9;  
echo "Pre increment: ++$a";  
echo "Post increment:$a++";  
echo "Final value of a: $a";
```

In the first echo statement a is incremented first and then the value is displayed.  
In the second echo statement the value of a is displayed and then incremented.

**Output:**

```
Pre increment: 10  
Post increment: 10  
Final Value of a: 11
```

**Logical Operators**

Used to combine conditional statements.

1. **and (\$x and \$y)** – true if both \$x and \$y are true.
2. **or (\$x or \$y)** – true if either \$x or \$y is true.
3. **Xor (\$x xor \$y)** – true if either \$x or \$y is true but not both.
4. **&& - and (\$x && \$y)** true if both \$x and \$y are true.
5. **|| - or(\$x || \$y)** true if either \$x or \$y is true.
6. **! – Not(!\$x)** true if \$x is false

## Conditional Operator (? :)

### Syntax:

If Condition is true ? Then value x : Otherwise value y

### Example:

```
($a > $b) : echo "$a is bigger"; : echo "$b is bigger";
```

Concatenation Operator (.)  
Used to concatenate strings

**Syntax:**

String1.String2

**Example:**

“Hello”.”World!”

Output : HelloWorld!

# Decision Making Statements

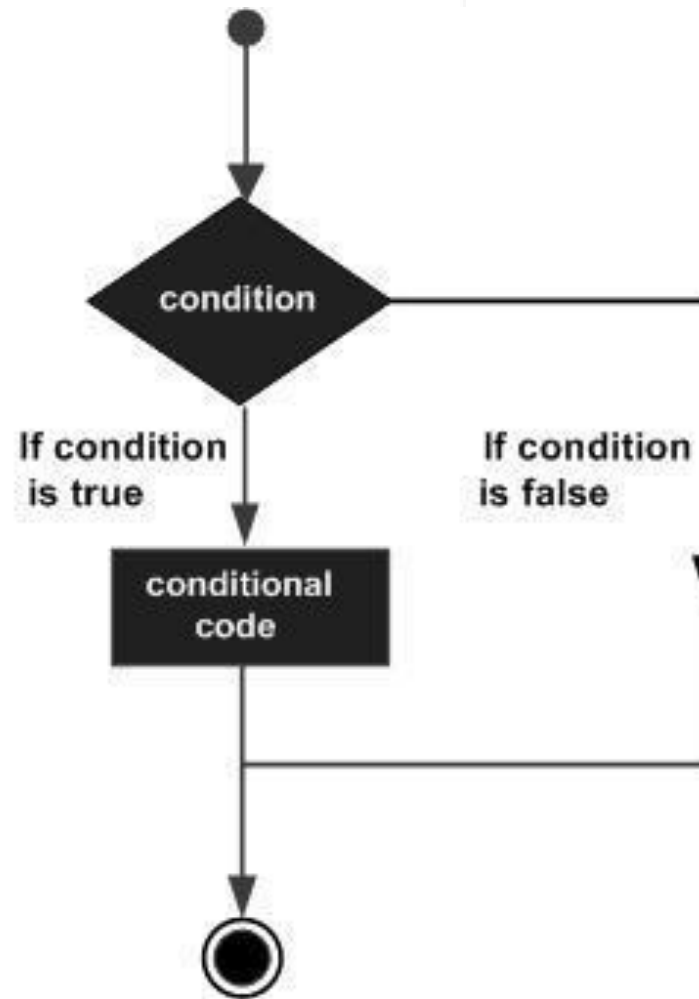
**If .. Else Statement**

**Syntax:**

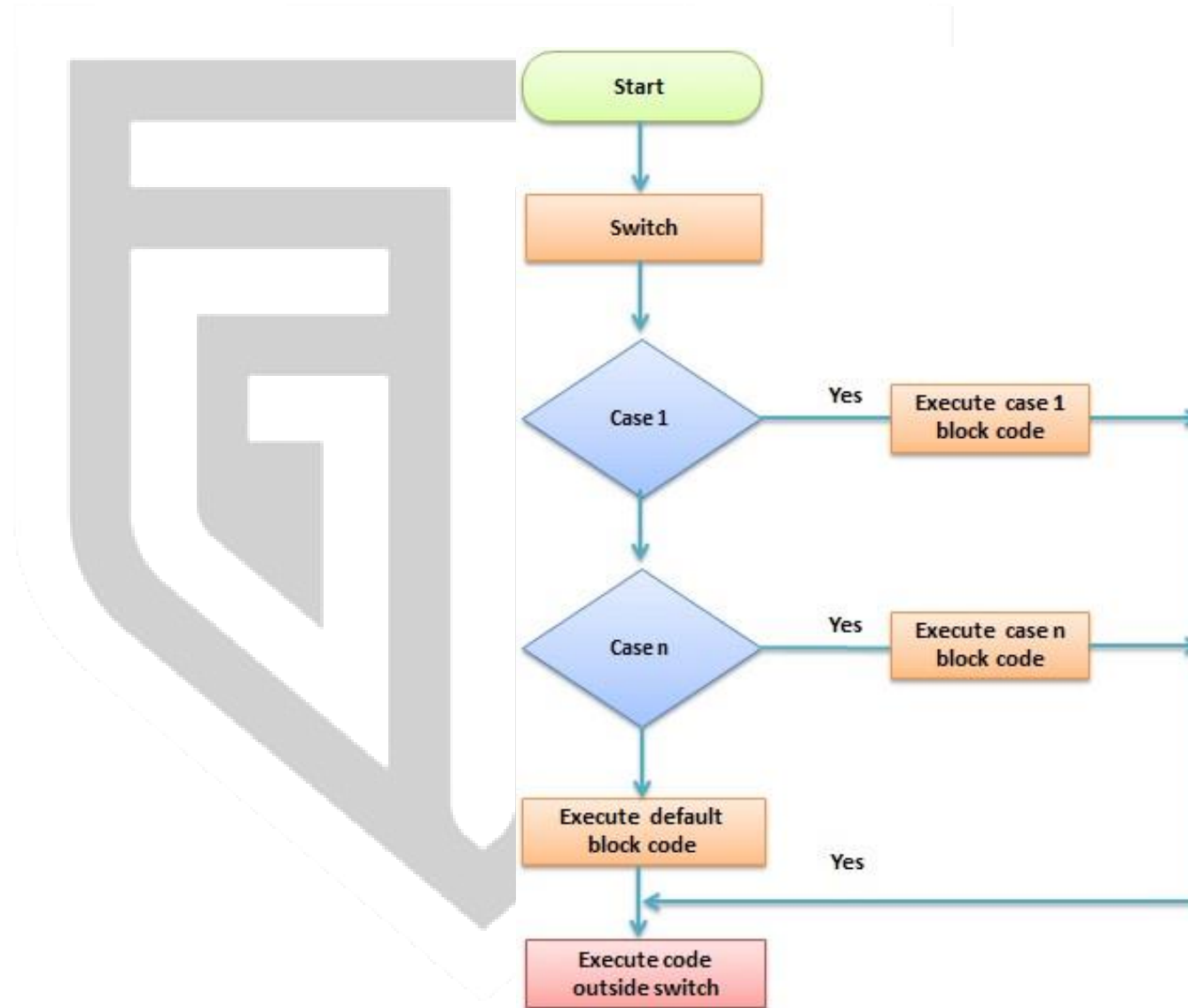
```
if (condition) {  
  //Statement block if condition is true  
} else{  
  //Statement block if condition is false  
}
```

**Example:**

```
if(a > 5)  
echo "Weekend"; Else echo "Week Day";
```



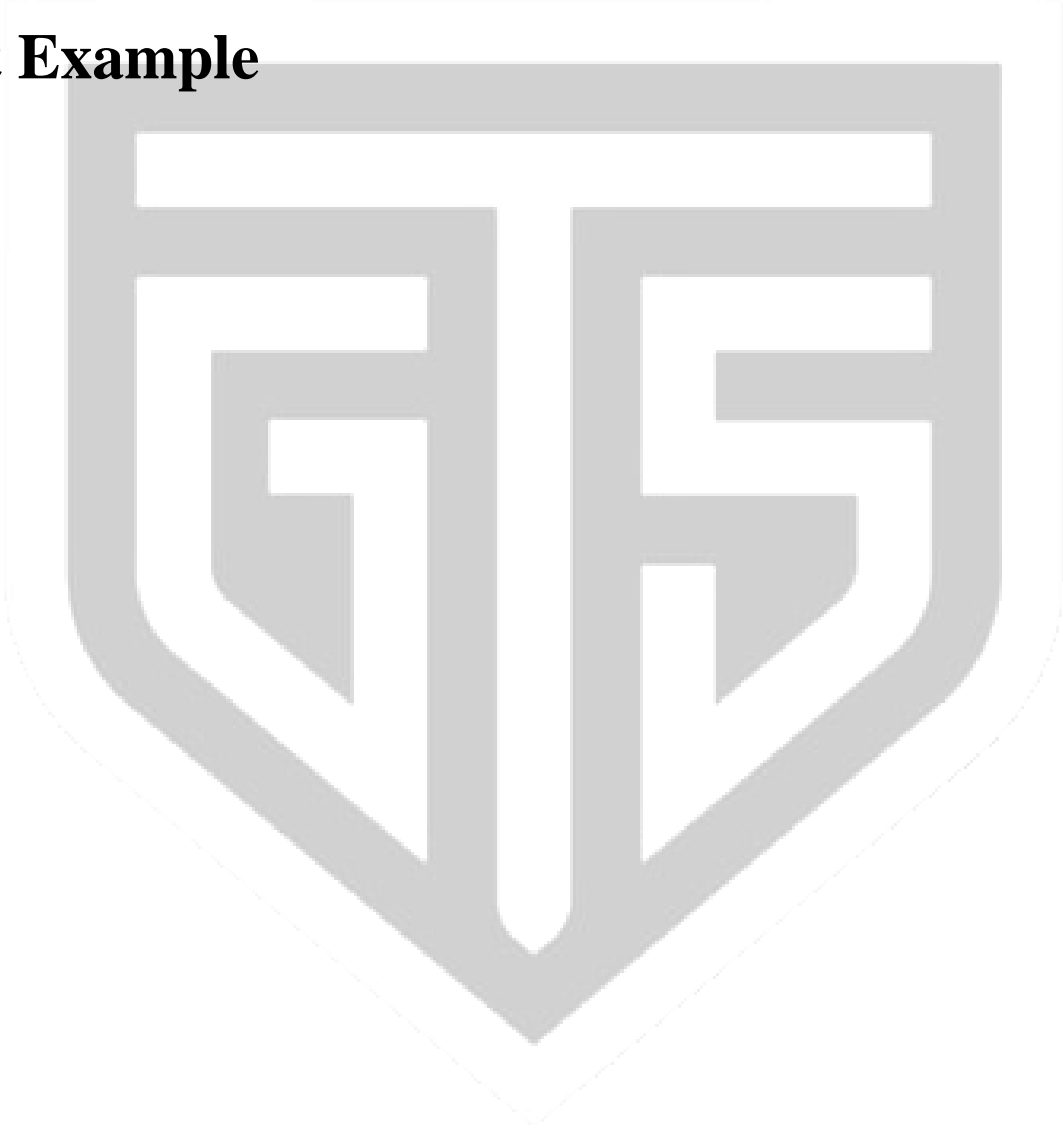
## Switch-case Statement





```
switch (expression) { case 1:  
//Statements if expression = case 1; break;  
.  
.  
. case N:  
//Statements if expression = case n; break;  
  
default:  
//Statements for if none of the value match  
}
```

# Switch-case Statement Example



```
switch (day)
{
```

```
case 1: echo
“Monday”;
break;
```

```
case 2: echo
“Tuesday”;
break;
```

```
case 3: echo
“Wednesday”;
break;
```

```
case 4: echo
“Thursday”;
break;
```

```
case 5: echo
“Friday”;
break;
```

```
case 6: echo
“Saturday”;
break;
```

```
case 7: echo
“Sunday”;
break;
```

```
default: echo
“Invalid day”;
break;
}
```

# Looping Constructs

**For Statement**

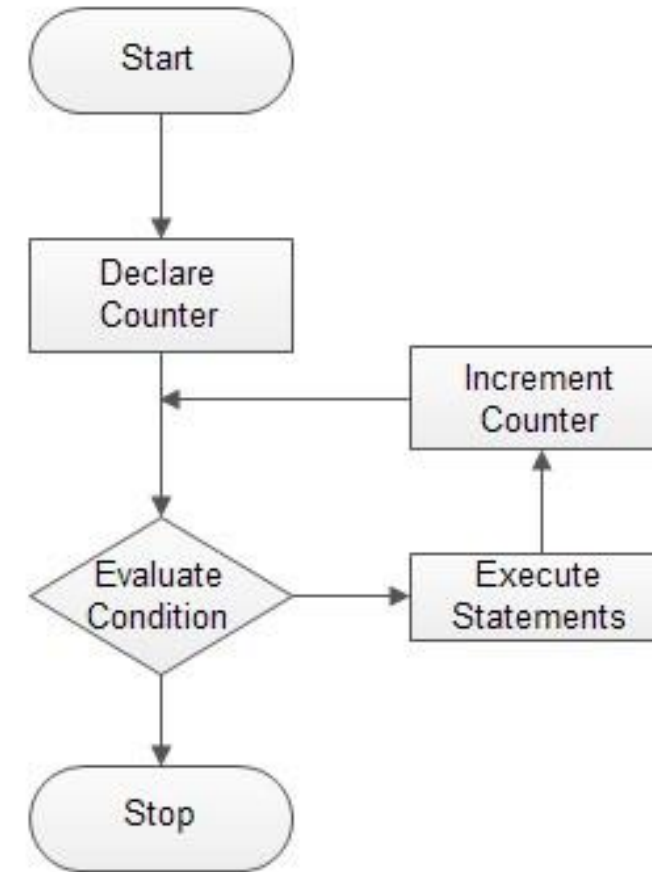


**Syntax:**

```
for (initialization; condition; increment)
{ code to be executed;
}
```

**Example:**

```
<?php
for( $i = 0; $i<5; $i++ )
{
    $a += 5;
}
?>
```

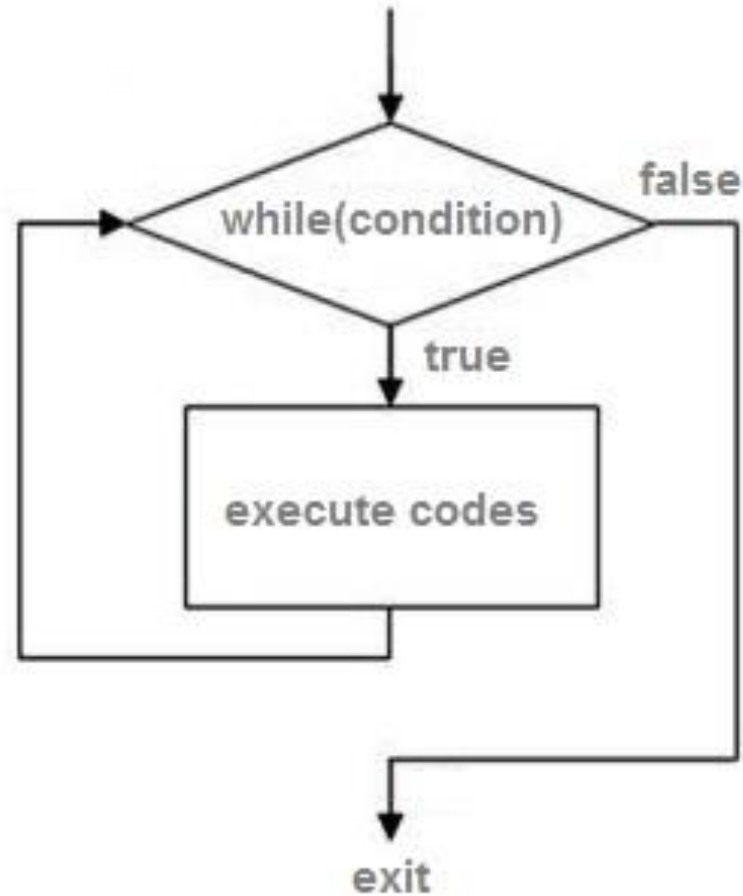


## While Statement Syntax:

```
while (condition)
{
//Statements if condition is true
}
```

### Example:

```
<?php
while( $i < 10) { echo $i;
}
?>
```

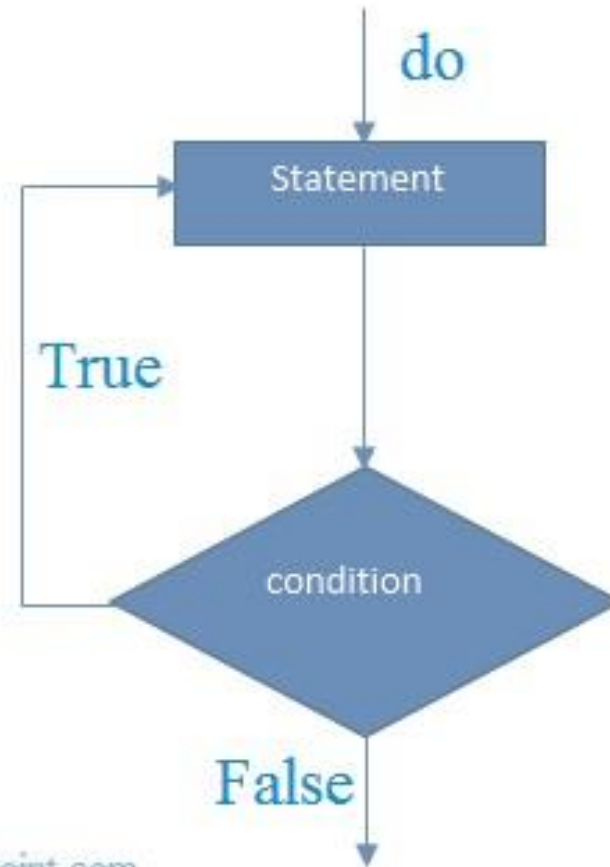


## Do-while Statement Syntax:

```
do {  
    //statements  
} while (condition);
```

### Example:

```
<?php do  
{ echo $i; } while( $i <  
10 );  
?>
```



javaTpoint.com

## Foreach Statement

Foreach loop is used to **loop through an array until it reaches the end of array.**

**Syntax:** foreach (array as  
value)  
{  
    //Statements;  
}

Here, for each pass of the loop **the current value of array is assigned to value and the array index is incremented by one.**

### Example:

```
<?php  
foreach($scores as $value)  
{ echo "score : $value";
```



}?>

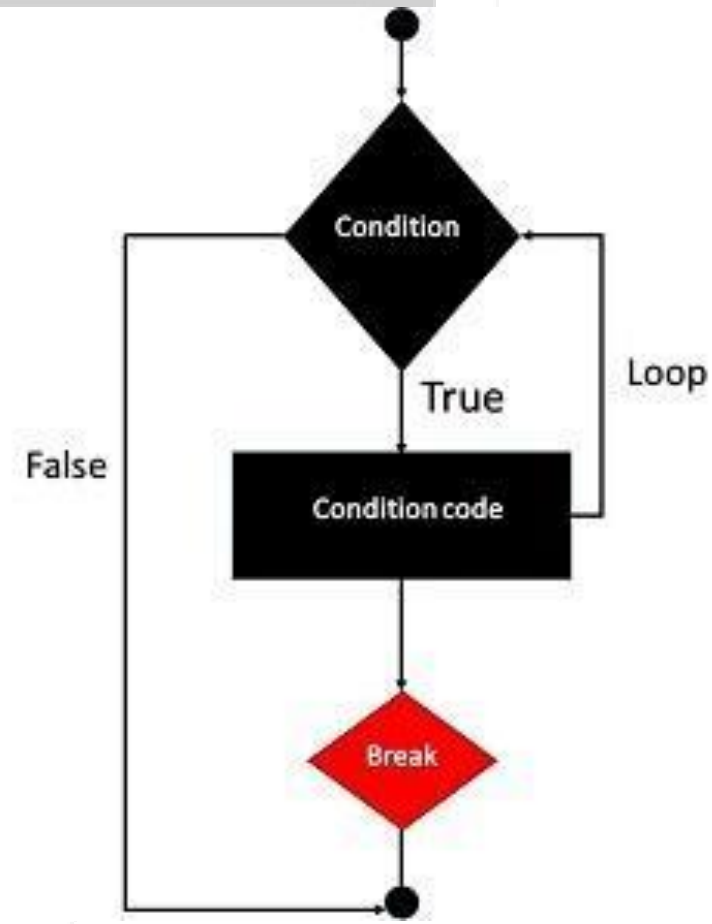


## Break Statement

**To terminate a loop before completion.**

**Syntax:** break; **Example:**

```
<?php $i = 1; while($i < 10){ echo "i  
= $i"; if($i == 5){ break;  
}  
} echo "The loop stopped at i = $i";  
?>
```



# Continue Statement

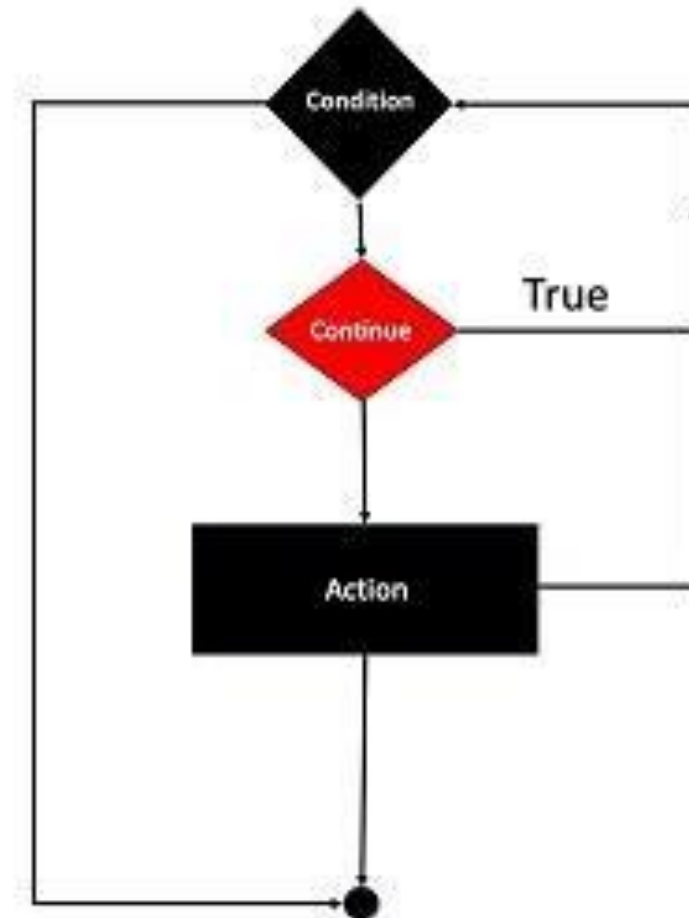
To skip an iteration of a loop.

**Syntax:** continue; **Example:**

Prints odd numbers between 1 and 10.

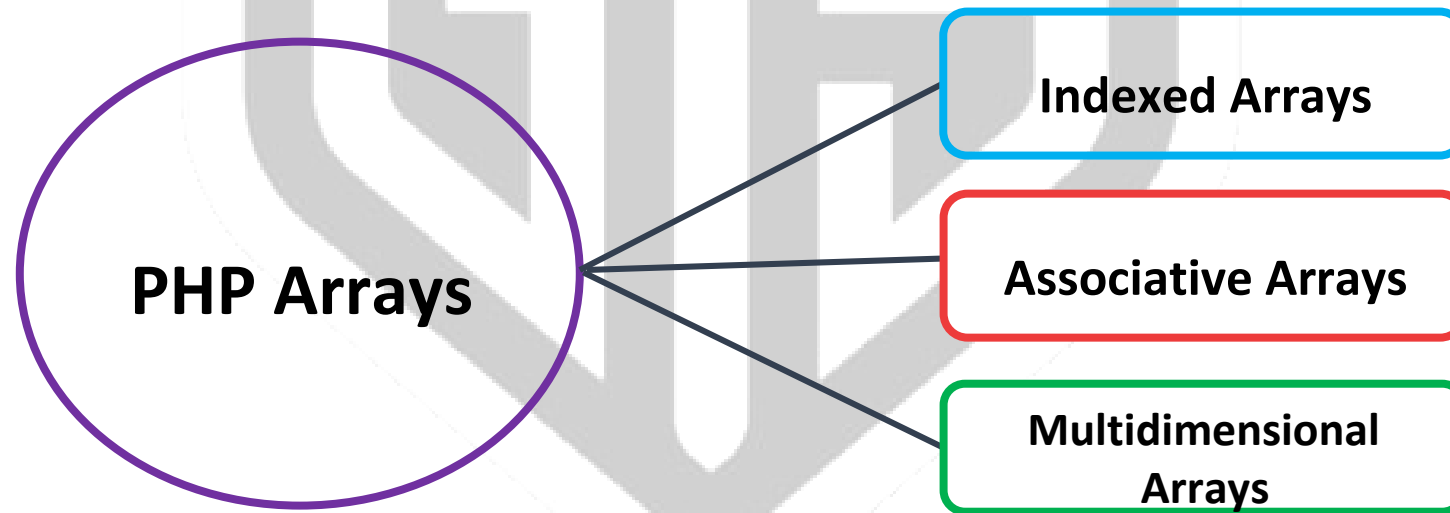
```
<?php $i = 1; while($i < 10){ if($i%2  
    == 0){ continue; } else{ echo $i;  
    }  
}  
?>
```

False



# Arrays

## Types of Arrays in PHP



## Indexed Arrays

An array with numeric index.

### Syntax:

```
$arrayname = array(value 1, value 2, . . . . , value n);
```

**value can be numbers, strings or objects.**

### Example:

```
$num = array (1, 2, 3, 4, 5);
```

Here the array is accessed as,

```
$num[0] = 1;
```

```
$num[1] = 2;
```

```
$num[2] = 3;
```

```
$num[3] = 4;
```

```
$num[4] = 5;
```

## Associative Arrays

Arrays with index as string.

**For instance,**

Marks of students - instead of having a numeric index, the student ID can be made an index.

**Syntax:**

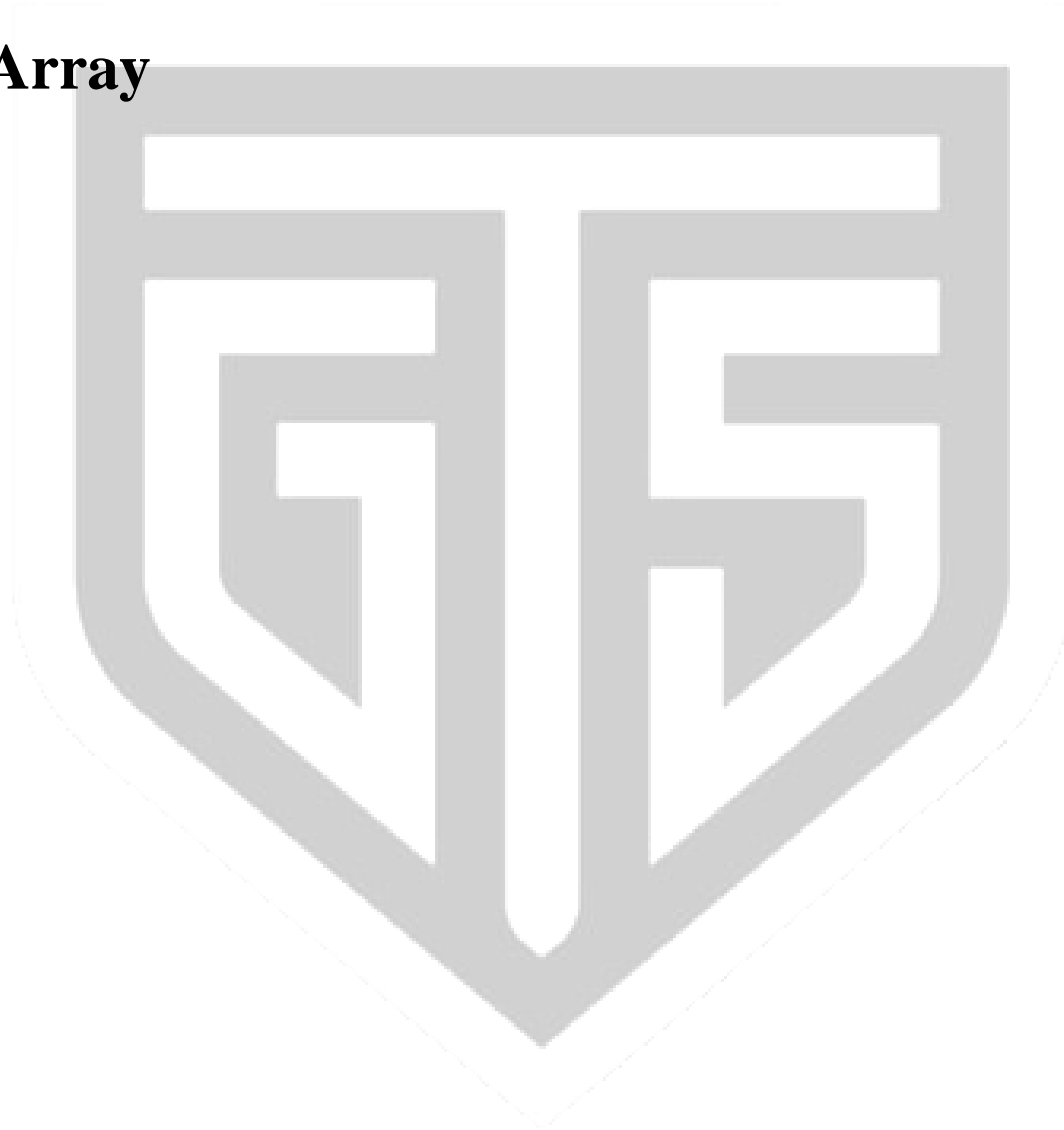
```
$arrayname = array("index 1" => value 1, .. index n" => value n);
```

**Example:**

```
$marks = array ("st18001" => 48, "st18002" => 42, "st18003" => 45);
```

To access the marks of student with ID st18003,  
echo "marks st18003:" . \$marks['st18003'];

# Multidimensional Array



## Array within an array

### Example:

```
/* Declaring a multidimensional array */
```

```
<?php
```

```
$marks = array(
    "mohammad" => array
        ( "physics" => 35,
          "maths" => 30,
          "chemistry" => 39
        ),
```

```
    "qadir" => array (
        "physics" => 30,
        "maths" => 32,
        "chemistry" => 29
    ),
```

```
    "zara" => array (
        "physics" => 31,
        "maths" => 22,
        "chemistry" => 39
    ) );
```

```
/* Accessing multi-dimensional array values */
```

```
echo "Marks for mohammad in physics : " ;
echo $marks['mohammad']['physics'] . "<br />";
```

```
echo "Marks for qadir in maths : "; echo
$marks['qadir']['maths'] . "<br />";
```

```
echo "Marks for zara in chemistry : " ;
echo $marks['zara']['chemistry'] . "<br />";
```

```
?>
```



# Strings

## Strings

A String is a sequence of characters such as,  
“Hello World” or ‘Hello World’

### String Functions:

1. **strlen()** – Returns the length of a string.

```
strlen(“Hello World!”);           // returns 12
```

2. **strpos()** – to search a string or character in a string. Returns the position of the first match, false if no match found.

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

```
strpos("Hello world!","l");//returns 2
```

```
strpos("Hello world!","hi"); //returns false
```

**3. strrev()** – reverses a given string.

```
strrev("Hello World!"); // returns !dlroW olleH
```

**4. strreplace()** – replace some characters with other characters.

```
strreplace("World", "Sunshine", "Hello world!");
```

//replaces World with Sunshine – Hello Sunshine

**5. str\_word\_count()** – counts the number of words in a given string.

```
str_word_count("I like PHP!"); // returns 3
```

**6. String Concatenation** – The dot (.) operator is used to combine two strings into one.

```
$str = "Hello World!". "I like PHP";
```

// \$str will contain “Hello World! I like PHP”

## Escape Sequences:

Some characters cannot be used in a string they must be escaped. PHP offers the following escape sequences.

- `\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 (\)

## Know More



A tutorial link for PHP. Covers the basics of PHP.



A tutorial video link for PHP. It covers the installation of PHP and PHP basics in a series of short videos .

Topic	URL
Tutorial Link of PHP	<a href="https://www.w3schools.com/html/html5_new_elements.asp">https://www.w3schools.com/html/html5_new_elements.asp</a>
New elements of HTML	<a href="https://www.youtube.com/watch?v=mdDlsqelEh0">https://www.youtube.com/watch?v=mdDlsqelEh0</a>

# PHP Form Handling

## PHP Environment

The PHP environment on your computer will allow you to edit and test your web pages before they are uploaded on the server. To setup PHP environment,

- 1. Install a web server like Apache**
- 2. Install PHP**
- 3. Install a good code editor such as Eclipse**

OR

You can also install **WAMP Server** which is a **software stack for Windows** consisting of all the above listed elements and MySQL.

All the listed software are open source.

## PHP Environment Variables

ENV variables help to create and shape the environment where the code is running.

- The PHP environment variables are listed in the `phpinfo.php` file.
- **HTTP\_USER\_AGENT** identifies the user's browser and operating system.
- **HTTP\_ACCEPT\_\_LANGUAGE** identifies the language accepted such as en-US(English – United States).
- **SERVER\_NAME** – contains the name of the server.

- The **getenv()** function can be used to access the value of all environment variables.

## GET and POST Methods

### GET Method

- The parameters are listed in URL at the end as name-value pairs.

**Example:** <http://www.foo.com/?name=Kelly&age=18> •

Used to get something without writing any data to the server.

- Used to send small amount of data.
- Sensitive data like passwords cannot be sent.

- PHP provides `$_GET` associative array to access the data.

## **POST Method**

- The data is embedded in the request message (HTTP header).
- Used to write something to the server.
- Used when large amount of data needs to be sent (eg. Files).
- Sensitive data such as passwords can be sent using POST (by using HTTP secure).
- PHP provides `$_GET` associative array to access the data.

## **Form Handling**

**`$_GET` and `$_POST` are used to collect the form data.**



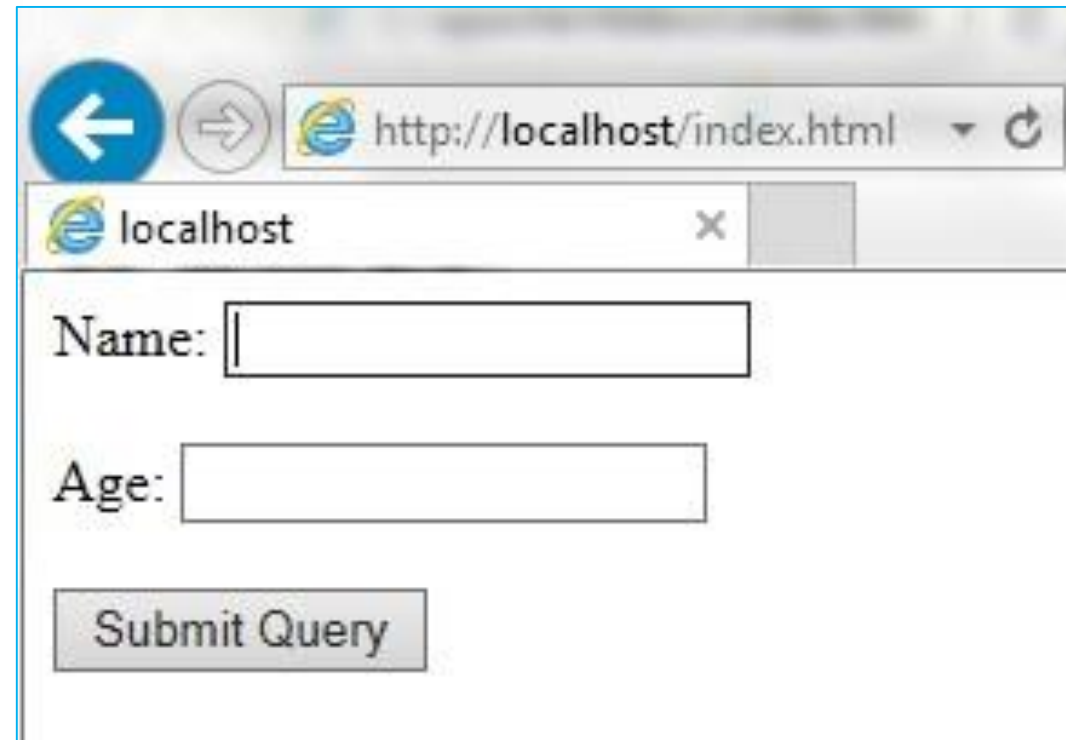
Consider this simple HTML form which uses POST – index.html

```
<html>
  <body>
    <form action="sample.php" method="post"> Name:
      <input type="text" name="name"> <br><br>
      Age: <input type="text" name="age"> <br><br>
      <input type="submit">
    </form>
  </body>
</html>
```

**Save the file under apache\htdocs.**

## **Form Handling – Output**

**Open index.html from the apache server to see the following output.**



A screenshot of a web browser window. The address bar shows 'http://localhost/index.html'. The browser tab is labeled 'localhost'. The page content includes a form with two input fields: 'Name:' and 'Age:'. Below these fields is a button labeled 'Submit Query'.

**Create the php file - sample.php**

```
html>
```

```
<body>
```

Welcome

<?php echo \$\_POST["name"]; ?> you  
are

<?php echo \$\_POST["age"]; ?> years  
old.

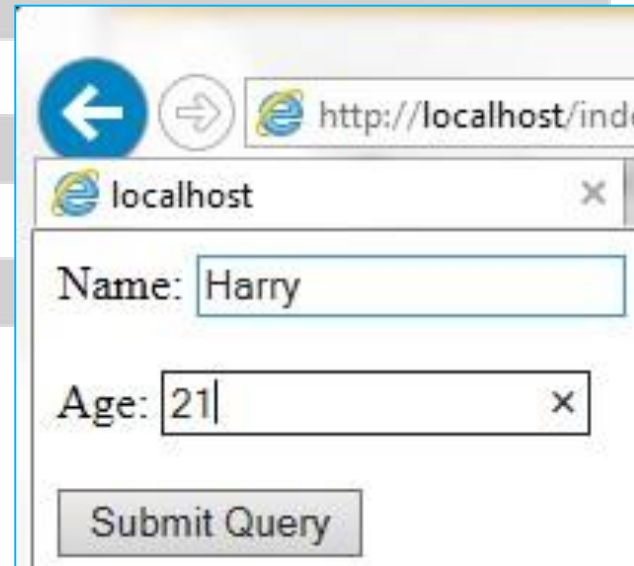
</body>

</html>

**Save the file under apache\htdocs.**

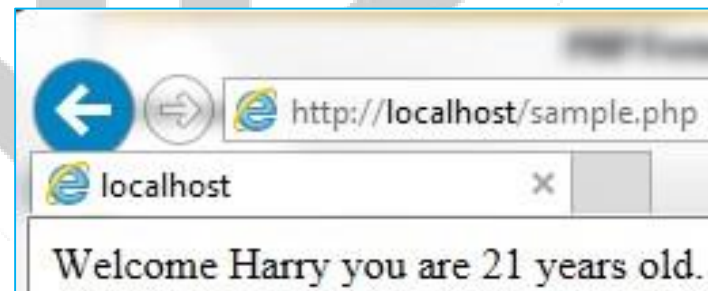
## PHP Forms - Sample Output

Enter name as Harry and age as 21.



A screenshot of a web browser window. The address bar shows 'http://localhost/index'. The page content includes a form with two input fields: 'Name: Harry' and 'Age: 21'. Below the fields is a button labeled 'Submit Query'.

On pressing the Submit Query button, we get the following output:



A screenshot of a web browser window. The address bar shows 'http://localhost/sample.php'. The page content displays the text 'Welcome Harry you are 21 years old.'

## Know More



The links explains how form data is handled in PHP using examples.



The video explains how form data is handled in PHP using examples.

Topic	URL
PHP handling in form of data	<a href="https://www.w3schools.com/php/php_examples.asp">https://www.w3schools.com/php/php_examples.asp</a>
PHP Handling and data types	<a href="https://www.youtube.com/watch?v=n2oWBRTeDyg">https://www.youtube.com/watch?v=n2oWBRTeDyg</a>

# Files

## File Inclusion

Contents of files such as PHP, HTML, text or other files can be included in a PHP file using,

- **include statement** • **require statement**

### Syntax:

```
include 'filename';
```

or

```
require 'filename';
```

**On failure,**

- Include gives a warning E\_WARNING.
- Require gives a fatal error E\_COMPILE\_ERROR.

## Files in PHP

**Opening a File: fopen()** - returns false if it fails to open a file, otherwise it **returns a file pointer.**

**Syntax:**

```
fopen("filename", "mode");
```

**Mode can be,** r – read

only r+ - read and

write w – write only

w+ - read and write

a – append a+ - read  
and append

## Closing a file

### **fclose()**

Used to close a file. Returns true if it succeeds, false otherwise.

### **Reading from a File**

To read a file,

1. Open the file in read mode using,  
`fopen("file1.txt","r");`
2. Get the  
file's length using `fsize()`,  
`fsize("file1.txt");`



3. Read the file using, `fread()`;
4. Close the file using, `fclose()`;

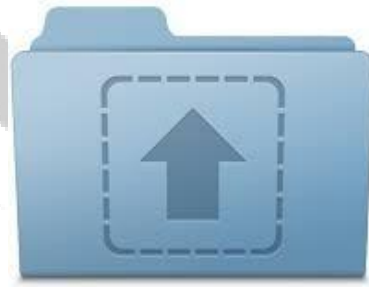
## Writing to a File

The `fwrite()` function is used to write into a file.

1. Open the file in write mode.  
`$fp = fopen("file1.txt","w")`
2. Write the data.  
`fwrite($fp,"This is my first file.");`
3. Close the file. `fclose()`;

## Uploading a File

1. Using PHP we can upload files to the server.
2. The files are initially uploaded into a temporary directory on the server and then uploaded to the destination by a PHP Script.
3. the php.ini file contains the information,
  - upload\_tmp\_dir – contains the temporary directory name used for uploads
  - upload\_max\_filesize – contains the maximum permitted size for uploading.



1. Configure the **php.ini** file to allow file uploads.

In **php.in** file, find the **file\_uploads** directive and **set it to on**.  
`file_uploads = on`

2. Create an HTML form that allows users to browse the file they want to upload.

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

- type = file will create a browse button.
- For method must be post.
- Include the attribute enctype="multipart/form-data"

3. Create the php script to upload the file.

## File Upload - HTML Form

```
<html>
  <body>

    <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>

  </body>
</html>
```

## Uploading a File – PHP Script

The PHP global variable `$_FILES` is an **associative array** which is created by PHP for storing all the information about the file to be uploaded.

So, if the input name attribute of form in HTML is given as “**file**”, PHP creates the following variables:

1. `$_FILE['file']['tmp_name']` – uploaded file in temporary directory on the server.
2. `$_FILE['file']['name']` - actual name of the uploaded file.
3. `$_FILE['file']['size']` – size in bytes.
4. `$_FILE['file']['type']` – MIME type of the file.
5. `$_FILE['file']['error']` – error code associated with the file upload.

## File Upload - PHP Script

### Checking if the file exists before uploading:

```
// Check if file already exists if  
(file_exists($file_tmp)) { echo  
"Sorry, file already exists."  
}
```

### Uploading the File:

The **move\_uploaded\_file()** function is used to move the file to its target destination from the temporary directory.

```
move_uploaded_file($file_tmp,"images/".$file_name);
```

Here the file is moved to /images directory.

### **Checking the file size:**

```
// Check file size
if ($_FILES["file_tmp"]["size"] > 2097152 )
    { echo "Sorry, your file is too large.";
    }
```

### **Checking the File Type:**

```
// Allow certain file formats if($imageFileType != "jpg" && $imageFileType != "png" &&
$imageFileType != "jpeg" &&
$imageFileType != "gif" ) { echo "Sorry, only JPG, JPEG, PNG
& GIF files are allowed."; $uploadOk = 0; }
```

## Know More



Explains how file handling is done in PHP.



Explains how file handling is done in PHP.

Topic	URL
File Handling in PHP	<a href="https://www.w3schools.com/php/php_file.asp">https://www.w3schools.com/php/php_file.asp</a>
File Handling in PHP	<a href="https://www.youtube.com/watch?v=e7NvwnWaOZw">https://www.youtube.com/watch?v=e7NvwnWaOZw</a>



# Self Assessment Questions

## Self Assessment Questions

1. What does PHP stand for:
  - a) Personal Home Page
  - b) Hypertext Preprocessor
  - c) Pretext Hypertext Processor
  - d) Preprocessor Home Page

**Answer: b)**

2. Write the PHP syntax.

**Answer: <?php**  
**php code**  
**?>**

3. Which of the given option(s) is/are used to write a single line comment in PHP?

- a) /\* \*/
- b) //
- c) #
- d) \$

**Answer: b), c).**

4. Choose the correct string declaration in PHP from the given options:

- a) `$str = "Hello World!";`
- b) `String str = "Hello World!";`
- c) `String $str = "Hello World!";`
- d) `String str[] = "Hello World!";`

**Answer: a)**

5. What will be the output of the code?

```
<?php  
$n=10; $s=20; echo  
$n . "+" . $s; ?>
```

- a) 30
- b) error
- c) 10+20
- d) 10+.20

**Answer: c)**

6. Choose the valid variable name(s) from the given options.

- a) \$var
- b) \$9jk
- c) \$\_set
- d) \$jk9

**Answer: a), d).**

7. What will be the output of the given code,

```
<?php  
$a=4; echo  
"$a"; ?>
```

- a) 4

- b) \$a
- c) \4
- d) error

**Answer: b)**

8. List the different types of arrays in PHP.

**Answer: Indexed array, Associative array, multidimensional array.**

9. Associative arrays use a string as an index.

- a) Yes
- b) No

**Answer: a)**

10. Choose the correct statement for creating an array.

- a) `$numbers[] = array (1, 2, 3, 4, 5);`
- b) `Numbers[] = {1, 2, 3, 4, 5};`
- c) `Number[0] = array (1, 2, 3, 4, 5);`
- d) `$numbers = array (1, 2, 3, 4, 5);`

**Answer: d)**

11. Consider the code,

```
$sal = array ("John" => 20000, "Ram" => 40000, "Drake" => 35000)
```

Write the echo statement to print the salary of 'Ram'.

**Answer: echo "Salary of Ram is :" . \$sal['Ram'];**

12. What will be the output of:

```
strpos("I like Web Development!", "i");
```



**Answer: o**

13. The \_\_\_\_\_ function is used to reverse a string.

**Answer: strrev()**

14. Write the statement to count the words in the string “The fox jumped over the lazy dog.”

**Answer: str\_word\_count(“The fox jumped over the lazy dog.”);**

15. \_\_\_\_\_ function is used to access all environment variables.

- a) `getenv()`
- b) `$_Env()`
- c) `$_ENVIRON`
- d) `getinfo()`

**Answer: a)**

16. `$_GET` and `$_POST` are used to collect form data. (True/False) **Answer: True.**

17. GET is used when we have to write something to the server. (True/False)

**Answer: False.**

18. Consider the URL, `http://www.test.com/`

How will the URL change if we send the name-value pairs,

‘user=ram’ and ‘status=active’

using GET.

**Answer: `http://www.test.com/?user =ram&status=active`**

19. We can send credit card number using GET. (True/False) **Answer: False.**

20. \_\_\_\_\_ and \_\_\_\_\_ statements are used to include the contents of a file in a PHP file.

**Answer: include and require.**

21. E\_COMPILE\_ERROR is a fatal error produced by:

- a) include statement
- b) require statement
- c) error statement
- d) on compilation

**Answer: b)**

22. If the file 'test' is opened as, `fopen("test","a");`

- a) The file is opened in read mode, file pointer points to the beginning.
- b) The file is opened in write mode, file pointer points to the beginning.
- c) The file is opened in write mode, file pointer points to the end of file.
- d) The file is opened in read/write mode, file pointer points to the end of file.

**Answer: c)**

23. The `filesize()` method returns the size of file in:

- a) bits
- b) kilobytes
- c) word count
- d) bytes

**Answer: d)**

24. Write the syntax of the input statement used when uploading a file to the server.

**Answer: <input type = “file” name=”filename” id=”fileid” enctype=”multipart/form-data”>**

25. All the information about the file to be uploaded is stored in the global variable \_\_\_\_\_.

**Answer: \$\_FILES**

## **Database-Use Case**

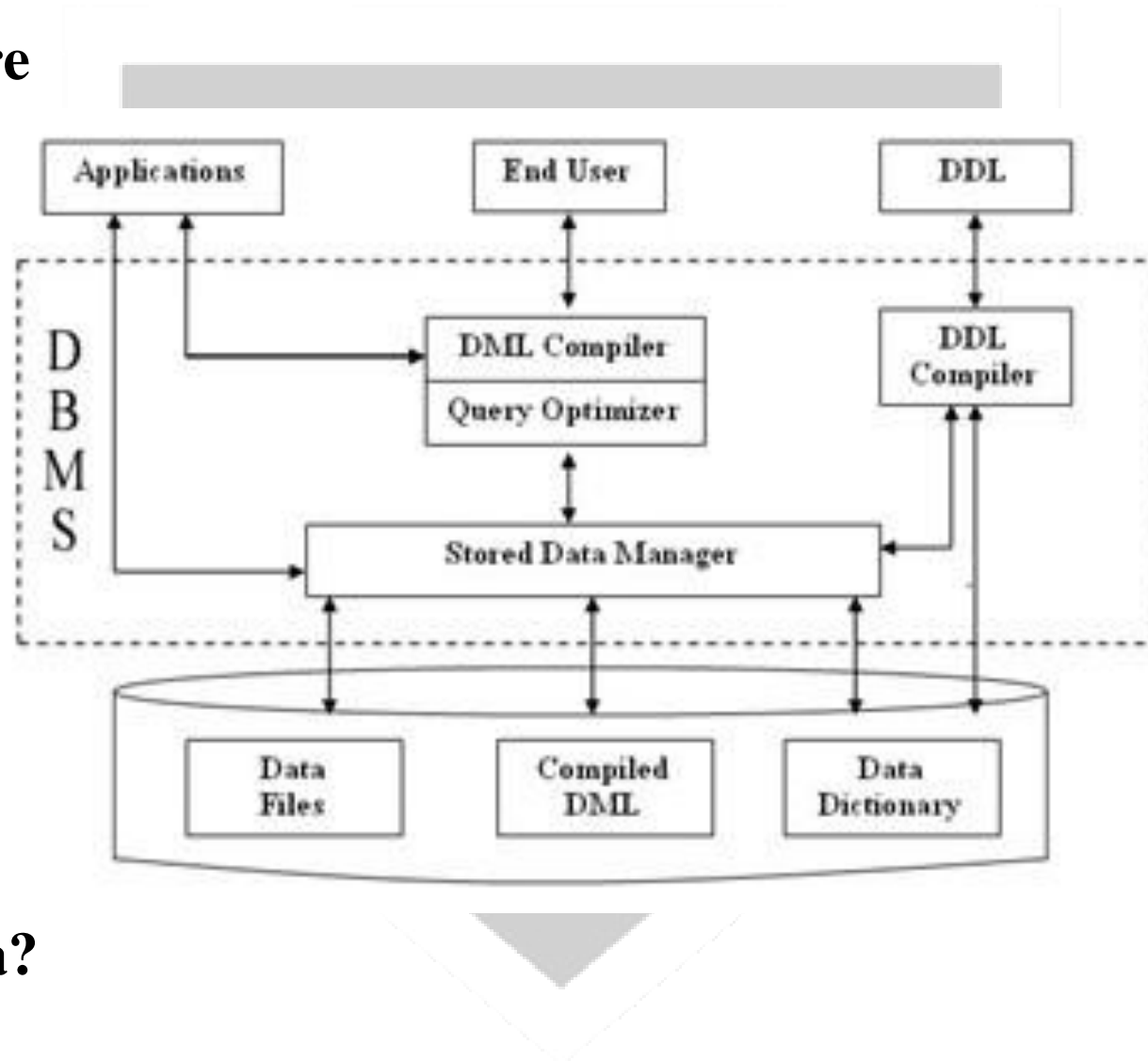
Consider a scenario, where xyz bank doing thousands of transactions a day. Through these transitions data is generated per day and ranges from MBs to GBs. Now how to store and manipulate for easy and fast access?

## What is RDBMS?

1. RDBMS stands for Relational Database Management System
2. It is a kind of software to manage and manipulate data
3. RDBMS examples
  - MySQL
  - Oracle
  - MS SQL Server



## RDBMS Architecture



## How to Manage Data?

1. All RDBMS software use internal language to manage data.
2. The language is SQL.
3. SQL stands for Structured Query Language.
4. Syntax of SQL commands in each RDBMS software varies.
5. Example create select statement is not same in Oracle and MySQL

## **SQL Types**

1. DDL
  - Data Definition Language. SQL commands are create, drop and modify.
2. DML
  - Data Manipulation Language. SQL commands are insert, update and delete.

### 3. DQL

- Data Query Language. SQL commands are select.

## **PHP Database Handling**

1. PHP works with all databases like Oracle, MySQL and MS SQL Server.
2. PHP has different API to connect database.
3. API Types.
  - mysql\_ functions .
  - These functions are deprecated from PHP 5.x.
  - mysqli\_ functions.
  - PDO objects.
4. In coming slides, we will discuss mysqli\_ functions.

## Mysqli\_ functions

1. `mysqli_connect()`.
  - Is used to connect to specific database.
2. `mysqli_close()`.
  - Is used to close the connection from database.
3. `mysqli_query()`.
  - Is used to executed SQL quireis like (insert, delete, update and select).
4. `mysqli_num_rows(mysqli_result $result)`.
  - Is used to check number of rows.
5. `mysqli_fetch_assoc(mysqli_result $result)`.

## Mysqli\_connect() Example

1. This function takes three or four parameters to connect & returns resource link.
  - resource mysqli\_connect (server, username, password).
  - resource mysqli\_connect (server, username, password, database).
2. Code snippet as follows to connect MySQL database.

```
<?php
```

```
$host = 'localhost:3306'; //database server instance
```

```
$user = 'shiva'; //database user name
```

```
$pass = 'kumar'; //database password
```

```
$conn = mysqli_connect($host, $user, $pass);
```

?>

## Mysqli\_close() example

- This function closes the connection
- `bool mysqli_close(resource $resource_link)`

<?php

```
$host = 'localhost:3306'; //database server instance
```

```
$user = 'shiva'; //database user name
```

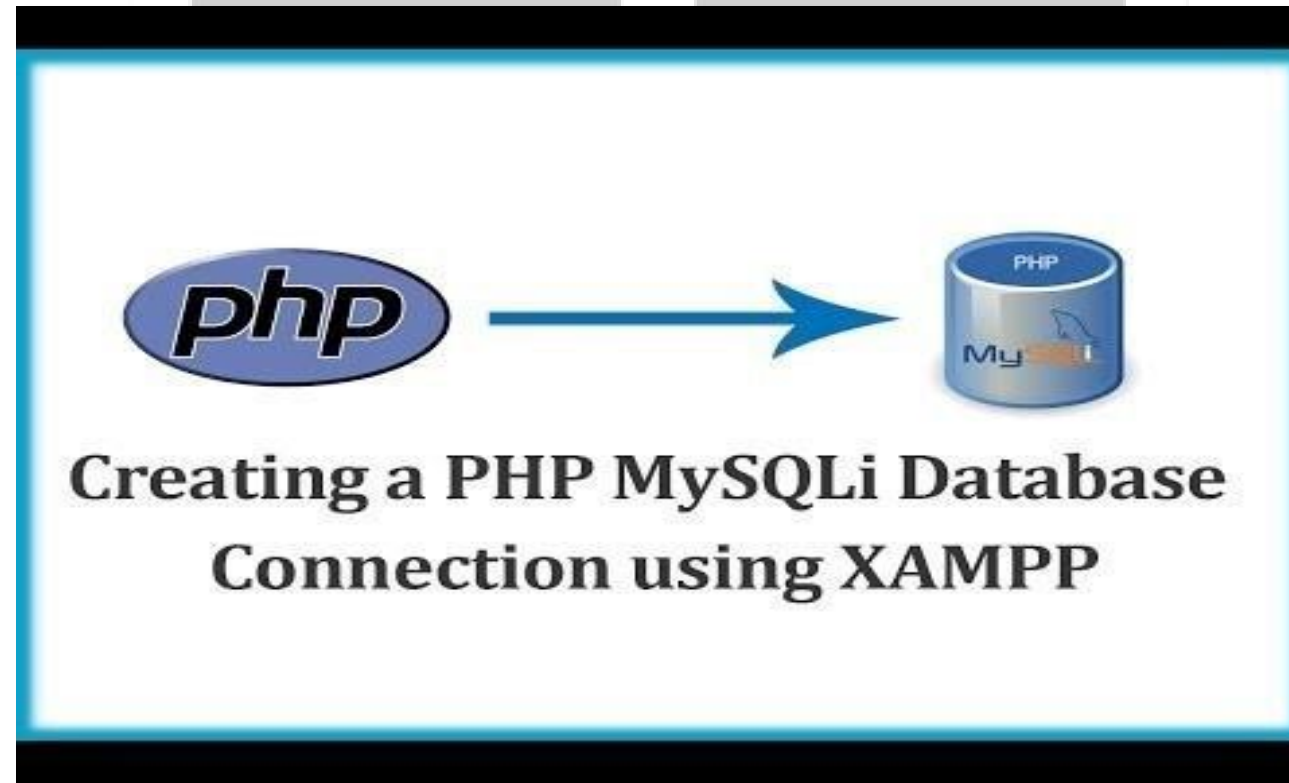
```
$pass = 'kumar'; //database password $conn
```

```
= mysqli_connect($host, $user, $pass);
```

```
mysqli_close($conn)
```

?>

## Mysqli\_connect Video Reference



## Mysqli\_query()

- This function is used to execute SQL queries
- `bool mysqli_query(resource $link, string $query)`



## Mysqli\_query() Create Database Example

```
<?php
    $host = 'localhost:3306';
    $user = 'shiva';
    $pass = 'kumar';

    $conn = mysqli_connect($host, $user, $pass);

    $sql = 'CREATE DATABASE test';
    if(mysqli_query($conn, $sql)){ echo
    "Database created successfully";
    }else{ echo "Error while create
    database");
    }
```

?>

## Mysqli\_query() Create Table Example

```
<?php
```

```
    $host = 'localhost:3306';
```

```
    $user = 'shiva';
```

```
    $pass = 'kumar';
```

```
    $dbname = 'test';
```

```
    $conn = mysqli_connect($host, $user, $pass,$dbname);
```

```
    $sql = 'CREAT TABLE emp(ename varchar(40),salary  
numeric(5,0))'; if(mysqli_query($conn, $sql)){ echo "Table created  
successfully";
```

```
    }else{ echo "Error while creating  
table"); }
```

?>

## Mysqli\_query() Insert Example

```
<?php
```

```
$host = 'localhost:3306';
```

```
$user = 'shiva';
```

```
$pass = 'kumar';
```

```
$dbname = 'test';
```

```
$conn = mysqli_connect($host, $user, $pass,$dbname);
```

```
$sql = 'INSERT INTO emp VALUES ("shiva", 9000)';
```

```
if(mysqli_query($conn, $sql)){ echo  
    "Record inserted successfully";
```

```
}else{ echo "Error while inserting  
    record");
```

```
}  
?>
```

## Mysqli\_query() Insert Example-2

```
<?php  
    $host = 'localhost:3306';  
    $user = 'shiva';  
    $pass = 'kumar';  
    $dbname = 'test';  
  
    $conn = mysqli_connect($host, $user, $pass,$dbname);  
  
    $sql = 'INSERT INTO emp(ename) VALUES ("kumar")';  
    if(mysqli_query($conn, $sql)){ echo  
        "Record inserted successfully";
```

```
        }else{ echo "Error while inserting  
                record");  
    }  
?>
```

## **Mysqli\_query() Delete Example**

```
<?php  
$host = 'localhost:3306';  
$user = 'shiva';  
$pass = 'kumar';  
$dbname = 'test';  
  
$conn = mysqli_connect($host, $user, $pass,$dbname);
```

```
$sql = DELETE FROM emp WHERE  
salary=1000'; if(mysqli_query($conn, $sql)){ echo  
"Record deleted successfully";  
}else{ echo "Error while deleting  
record");  
}
```

?>

## Mysqli\_query() Delete Example-2

```
<?php
    $host = 'localhost:3306';
    $user = 'shiva';
    $pass = 'kumar';
    $dbname = 'test';

    $conn = mysqli_connect($host, $user, $pass,$dbname);

    $sql = DELETE FROM emp WHERE salary < 50000 or ename="shiva";
    if(mysqli_query($conn, $sql)){ echo "Record deleted successfully";
    }else{ echo "Error while deleting
            record");
    }
?>
```

## Mysqli\_query() Update Example

```
<?php
    $host = 'localhost:3306';
    $user = 'shiva';
    $pass = 'kumar';
    $dbname = 'test';

    $conn = mysqli_connect($host, $user, $pass,$dbname);

    $sql = 'UPDATE emp SET salary=25400 WHERE
    ename="shiva"; if(mysqli_query($conn, $sql)){ echo "Record
    updated successfully";

    }

?>
```



```
}else{ echo "Error while updating  
record");
```

## Mysqli\_query() Update Example-2

```
<?php  
$host = 'localhost:3306';  
$user = 'shiva';  
$pass = 'kumar';  
$dbname = 'test';  
  
$conn = mysqli_connect($host, $user, $pass,$dbname);  
  
$sql = 'UPDATE emp SET salary=35400 WHERE salary < 30000 and salary >  
25000'; if(mysqli_query($conn, $sql)){ echo "Record updated successfully";  
  
}  
?>
```

# HTML and PHP

```
}else{ echo "Error while updating  
record");
```

```
?>  
}
```

## Mysqli\_query() Select Example

```
<?php
    $host = 'localhost:3306'; $user = 'shiva'; $pass = 'kumar'; $dbname = 'test';

    $conn = mysqli_connect($host, $user, $pass,$dbname);

    $sql = 'SELECT * FROM emp';
    $retval=mysqli_query($conn, $sql);

    if(mysqli_num_rows($retval) > 0){ while($row
    = mysqli_fetch_assoc($retval)){ echo "EMP
    NAME :{$row['ename']} <br> “.
        "EMP SALARY : {$row['salary']} <br> ".
        "-----<br>";
    }
    }else{ echo "0
    results";
    }
```

?>

## Mysqli\_query() Select Example-2

<?php

```
$host = 'localhost:3306'; $user = 'shiva'; $pass = 'kumar'; $dbname = 'test';
```

```
$conn = mysqli_connect($host, $user, $pass,$dbname);
```

```
$sql = 'SELECT ename FROM emp where salary > 30000';
```

```
$retval=mysqli_query($conn, $sql);
```

```
if(mysqli_num_rows($retval) > 0){ while($row  
= mysqli_fetch_assoc($retval)){  
    echo "EMP NAME :{$row['ename']} <br> “.  
        "-----<br>";
```

```
}
```

```
}else{ echo "0  
results";
```

```
}
```

?>  
**Mysqli\_query Video Reference**



## Displaying queries in tables

To populate a new database table with data you will first need an HTML page which will collect that data from the user. The following HTML code that and passes the information to a PHP script:

```
<form action="insert.php" method="post">  
Value1: <input type="text" name="field1-name" />  
Value2: <input type="text" name="field2-name" />  
Value3: <input type="text" name="field3-name" />  
Value4: <input type="text" name="field4-name" />  
Value5: <input type="text" name="field5-name" />  
<input type="Submit" /></form>
```

The above HTML code will show the user 5 text fields, in which the user can input data and a Submit button. Upon clicking the Submit button the data submitted by the user will be passed to a script named insert.php.

## Displaying queries in tables

The insert.php script can have a syntax similar to the following:

```
<?php
$username = "your_username";
$password = "your_password";
$database = "your_database";
$field1-name=$_POST['Value1'];
$field2-name=$_POST['Value2'];
$field3-name=$_POST['Value3'];
$field4-name=$_POST['Value4'];
$field5-name=$_POST['Value5'];
$mysqli = new mysqli("localhost", $username, $password, $database);
@mysqli_select_db($database) or die( "Unable to select database");
$query = "INSERT INTO tablename VALUES(',$field1-name',$field2-name',$field3-name',$field4-
name',$field5-name)";
```

```
$mysqli->query($query);  
$mysqli->close();  
?>
```

## Displaying queries in tables

After the user submits the information, the **insert.php** script will save it in the database table. Then you may want to output that information, so that the user can see it on the page.

```
<?php  
$username="username";  
$password="password";  
$database="your_database";  
$mysqli = new mysqli("localhost", $username, $password, $database);  
@mysql_select_db($database) or die( "Unable to select database");  
$query2="SELECT * FROM tablename";  
$result=$mysqli->query($query2);  
$num=$mysqli->mysqli_num_rows($result);
```



```
$mysqli->close();  
echo "<b>  
<center>Database Output</center>  
</b>  
<br>  
<br>";
```

## Displaying queries in tables

```
$i=0; while ($i <  
$num) {  
$field1-name=mysql_result($result,$i,"field1-name");  
$field2-name=mysql_result($result,$i,"field2-name");  
    $field3-name=mysql_result($result,$i,"field3-name"); $field4-  
        name=mysql_result($result,$i,"field4-name"); $field5-  
            name=mysql_result($result,$i,"field5-name");  
echo "<b>  
$field1-name $field2-name2</b>
```

```
<br>  
$field3-name<br>  
$field4-name<br>  
$field5-name<hr>  
<br>";  
$i++;  
}  
?>
```

# Assignment

## Assignment

1. Create a simple HTML form as shown and PHP script to display the data user enters back to the browser.

Name:

Age:

Choose a Color:

Red

▼

Choose the month you were born in:

January

▼

Submit Query

2. Create a PHP script to upload JPG files of size not more than 5MB.

## E-book Links

Module /Unit	Keyword/Topic	E-Book Name	Chapter	Page Number	URL	Comments
--------------	---------------	-------------	---------	-------------	-----	----------

# HTML and PHP

4	PHP Introduction and Installation	PHP6/MySQL Programming for the Absolute Beginner	. Chapter 1	Page No: 3-12	<a href="http://www.it-docs.net/ddata/391.pdf">http://www.it-docs.net/ddata/391.pdf</a>	Gives an overview about PHP and explains PHP installation.
4	PHP Basics	PHP6/MySQL Programming for the Absolute Beginner	. Chapter 2 ,3,4	Page No: 21-93 Page No. 102-25	<a href="http://www.it-docs.net/ddata/391.pdf">http://www.it-docs.net/ddata/391.pdf</a>	Explains the basics - input, variables, loops, decisions and arrays.
4	PHP Files	PHP6/MySQL Programming for the Absolute Beginner	. Chapter 6	Page No: 201 - 210	<a href="http://www.it-docs.net/ddata/391.pdf">http://www.it-docs.net/ddata/391.pdf</a>	Explains how to open, read, write and close a file using example.

## Web Reference Links

Sno	Chapter	Topic	URL	Notes
1	HTML and PHP	PHP Basics	<a href="https://www.w3schools.com/php/php_syntax.asp">https://www.w3schools.com/php/php_syntax.asp</a>	A tutorial link for PHP. Covers the basics of PHP.
			<a href="https://www.tutorialspoint.com/php/php_syntax_overview.htm">https://www.tutorialspoint.com/php/php_syntax_overview.htm</a>	A tutorial link for PHP. Covers the basics of PHP.
2	HTML and PHP	PHP Forms	<a href="https://www.w3schools.com/php/php_forms.asp">https://www.w3schools.com/php/php_forms.asp</a>	The links explains how form data is handled in PHP using examples.
3	HTML and PHP	PHP File Handling	<a href="https://www.tutorialspoint.com/php/php_files.htm">https://www.tutorialspoint.com/php/php_files.htm</a>	Explains how file handling is done in PHP.
4	HTML and PHP	Displaying queries in tables	<a href="https://www.siteground.com/tutorials/php-mysql/display-table-data/">https://www.siteground.com/tutorials/php-mysql/display-table-data/</a>	Explains inserting data into tables using MySQL queries

## Video Links

Sno	Topic	URL	Notes
1	PHP Basics	<a href="https://www.youtube.com/watch?v=vC4zncpdma0&amp;list=PLd1aagDQek4svljguIQvBXeXvQoKJHEdS">https://www.youtube.com/watch?v=vC4zncpdma0&amp;list=PLd1aagDQek4svljguIQvBXeXvQoKJHEdS</a>	A tutorial video link for PHP. It covers the installation of PHP and PHP basics in a series of short videos .
2	PHP Forms	<a href="https://www.youtube.com/watch?v=doobakPif3s">https://www.youtube.com/watch?v=doobakPif3s</a>	The video explains how form data is handled in PHP using examples.
3	PHP File Handling	<a href="https://www.youtube.com/watch?v=lvbTtrudQYY">https://www.youtube.com/watch?v=lvbTtrudQYY</a>	Explains how file handling is done in PHP.