# MEHRAN UNIVERSITY OF ENGINEERING & TECHNOLOGY

## DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING

## WEB ENGINEERING

**BATCH:** 18-CS

**SUBMITTED To:** Engr. MAHAVEER RATHI

**SUBMITTED By:** ________________________

# Table of Contents with Rubrics

| | |
|---|---|
| Total Marks | |
| Percentage | |
| Marks Obtained /15 | |

| | | | | | |
|---|---|---|---|---|---|
| Department of:<br>Computer Systems Engineering | | Subject | | | |
| | | WEB ENGINEERING | | | |
| Mehran University of Engineering<br>&Technology<br>Jamshoro | | Year | 3rd | Semester | 6th |
| | | Batch | 17CS | Duration | 03<br>Hours |

# PRACTICAL # 1

# TO PRACTICE THE BASIC TAGS OF HTML

**Webpage:**
A webpage is a simple document which requires browser to be displayed. All web pages have a unique address to be accessed on web.

**Website:**
A website is the collection of multiple web pages that are linked with each other and have a unique Domain name.

**Static webpage:**
The content of static pages can only be change by the developer by editing in the source code. Static web pages are quickly developed. Developers create static pages with HTML.

**Dynamic webpages:**
Dynamic webpages are created in script languages like PHP, JavaScript etc. Dynamic Web pages are capable of producing different content for different visitors from the same source code file. The website can display different content based on what operating system or browser the visitor is using.

**Web designing:**
Web design is the customer-facing part of the website. It is process of creating the visual impacts of site that how the customers interact with it, how to put together the principles of design and create a site that customers want to navigate around in.

**Web development:**
Web development is the data processing and the interaction of Webpages. We can manage the design by web development.

**Programming Language:**
A programming language is used to transform data. It does so by creating CPU instructions that rewrite the input data into the desired output. Programming languages usually needs a compiler to compile. E.g. C, C++, Java etc.

**Scripting language:**
Scripting language don’t need separate compiler we can compile it with browsers. A scripting language is a programming language which is interpreted, rather than compiled. E.g. Java script, Python etc.

## HTML:

- HTML is not a programming language; it is a **markup language.**
- A markup language is a set of **markup tags.**
- HTML uses **markup tags** to describe web pages.
- HTML markup tags are usually called HTML tags.
- HTML tags are keywords surrounded by **angle brackets** like <html>.
- <tagname>content</tagname>

## <Head> Tag:

- HTML tags normally **come in pairs** like <b> and </b>.
- The first tag in a pair is the **start tag;** the second tag is the **end tag.**
- Start and end tags are also called **opening tags** and **closing tags**.
- The <head> element is a container for metadata (data about data).
- HTML metadata is data about the HTML document. Metadata is not displayed.
- Metadata typically define document title, styles, links, scripts, and other Meta information.
- The following tags describe metadata: <title>, <style>, <meta>, <link>, <script>, and <base>.

**<! DOCTYPE html>:**

- <! DOCTYPE>is used to make sure that the website renders correctly in different browser. To be professional developer it is best to use <! DOCTYPE>

**Is it necessary to use <html>, <head> and <body> tags?**

- Omittingthe html,head,andbodytagsis certainly allowedbytheHTML specs. The underlyingreasonis that browsers havealways soughtto be consistent withexistingwebpages,andthe very early versions of HTML didn't definethose elements. WhenHTML2.0first did,itwas doneinaway thatthetagswouldbe inferredwhenmissing
- The html, head, and body tags are certainly allowed by the HTML specs.

## HTML Headings:

- HTML headings are defined with the <h1> to <h6> tags.
- <h1>This is a heading 1</h1>
- <h2>This is a heading 2</h2>
- ……………………………………..
- <h6>This is a heading 6</h6>

**Note:**H1 is largest heading & H6 is smallest heading type.

## HTML Paragraphs:

- HTML paragraphs are defined with the <p> tag.

  <p>This is a paragraph.</p>
  <p>This is another paragraph.</p>
- HTML tags are not case sensitive: <P> means the same as <p>. Many web sites use uppercase HTML tags.
- You may use lowercase tags because the World Wide Web Consortium (W3C) recommends lowercase in HTML 4, and demands lowercase tags in XHTML.

## HTML Links

HTML links are defined with the <a> Anchor tag.
Example: <a href="http://www.google.com">This is a link</a>

**Ordinary link:**
<ahref="http://www.google.com/">Google</a>

**Image-link:**
<a href="http://www.ebay.com/"><img src="a.jpg" alt="Alternate Text" /></a>

**Email / Mailto link:**
<a href="mailto:abc@xyz.com">Email Link</a>

- A named anchor:
- <a name="tips">Tips Section</a>
- <a href="#tips">Jump to the Tips Section</a>

## HTML Line Breaks

- Use the <br /> tag if you want a line break (a new line) without starting a new paragraph:
- **<br> or <br />**
- In XHTML, XML, and future versions of HTML, HTML elements with no end tag (closing tag) are not allowed.
- Even if <br> works in all browsers, writing <br /> instead is more **future proof**.

## HTML Formatting Tags

- HTML uses tags like <b> and <i> for formatting output, like **bold** or *italic* text.
- These HTML tags are called formatting tags.

| Tag | Description |
|---|---|
| **<b>** | Defines bold text |
| **<big>** | Defines big text |
| **<em>** | Defines emphasized text |
| **<i>** | Defines italic text |
| **<small>** | Defines small text |
| **<strong>** | Defines strong text |
| **<sub>** | Defines subscripted text |
| **<sup>** | Defines superscripted text |
| **<ins>** | Defines inserted text |
| **<del>** | Defines deleted text |

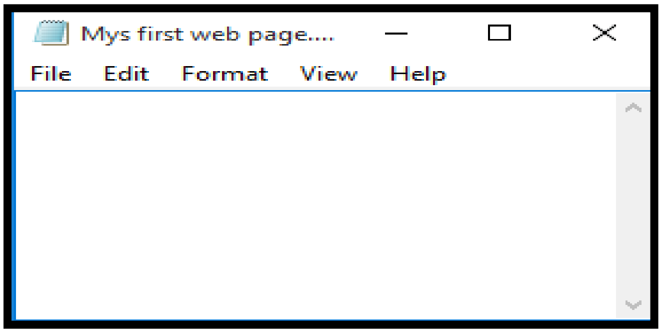
## HTML Attributes

- Attributes provide additional information about HTML elements.
- HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in the start tag.
- Attributes come in name/value pairs like: **name="value"**
- Attribute values should always be enclosed in quotes.
- Double style quotes are the most common, but single style quotes are also allowed.
- **Size Attributes**<imgsrc="w3schools.jpg" width="104" height="142">
- Attribute values should always be enclosed in quotes.
- Double style quotes are the most common, but single style quotes are also allowed.

# Making First Web Page

**Open Notepad & Type**

```
<!doctype html>
<html>
<head>my first page
<title>my first page</title>
</head>
<body>
<h1>this is heading 1</h1>
</body>
</html>
```



And save this file as a html file.

## LAB TAKS

**Task #1:** What is Html? Discuss what is difference between html tags and html element?

**Task # 2**: Make a webpage which will show your bio data (e.g: Name, fathers name, class, roll no, address, contact no, email address etc.). Use headings etc.

**Task # 3**: Apply 3 web links on your web page. Also place at least 3 images.

**Task # 4**: Make a web page which will show page header, body & footer. Size of header should be same as footer. Apply some font attributes in body section.

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ______________________________

Subject Teacher: ______________________________

Date: ______________________________

| | | | | | |
|---|---|---|---|---|---|
| Department of:<br>Computer Systems Engineering | | Subject | | | |
| | | WEB ENGINEERING | | | |
| Mehran University of Engineering<br>&Technology<br>Jamshoro | | Year | 3rd | Semester | 6th |
| | | Batch | 17CS | Duration | 03 Hours |

# PRACTICAL # 2

# TO MAKE USE OF VARIOUS TAGS RELATED WITH TABLES

### TABLES

- Tables are defined with the <table> tag.
- A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). td stands for "table data," and holds the content of a data cell. A <td> tag can contain text, links, images, lists, forms, other tables, etc.
- If you do not specify a border attribute, the table will be displayed without borders. Sometimes this can be useful, but most of the time, we want the borders to show.
- To display a table with borders, specify the border attribute:
- Header information in a table are defined with the <th> tag.
- All major browsers will display the text in the <th> element as bold and centered.

### Table related Tags

| Tag | Description |
|---|---|
| <table> | Defines a table |
| <th> | Defines a table header |
| <tr> | Defines a table row |
| <td> | Defines a table cell |
| <caption> | Defines a table caption |
| <thead> | Groups the header content in a table |
| <tbody> | Groups the body content in a table |

### HTML Lists

- The most common HTML lists are ordered and unordered lists:

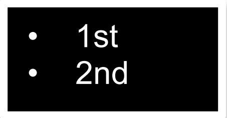
| An ordered list: | An unordered list: |
|---|---|
| 1. The first list item<br>2. The second list item<br>3. The third list item | • List item<br>• List item<br>• List item |

### Unordered Lists

- An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.
- The list items are marked with bullets (typically small black circles).

**Example:**

```
<ul>
   <li>1st </li>
   <li>2nd</li>
```



- 1st
- 2nd

```
</ul>
```

**Ordered ListsExample:**

```
<ol>
   <li>Mehran</li>
   <li>Sindh</li>
</ol>
```

1. Mehran
2. Sindh

**Definition Lists**

- A definition list is a list of items, with a description of each item.
- The <dl> tag defines a definition list.
- The <dl> tag is used in conjunction with <dt> (defines the item in the list) and <dd> (describes the item in the list):

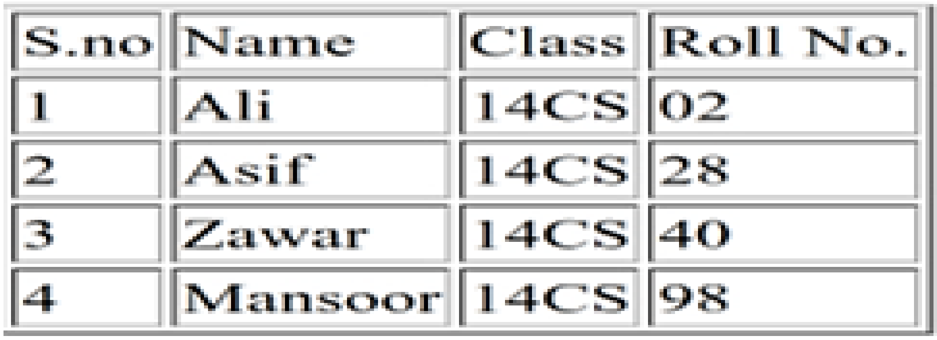
**Example:**

```
<dl>
    <dt>Cold Drinks</dt>
    <dd>- String / Coke</dd>
    <dt>Juices</dt>
    <dd>- Nestle /  Sheezan</dd>
</dl>
```

Cold Drink
  - String / Coke
Juices
  - Nestle /  Sheezan

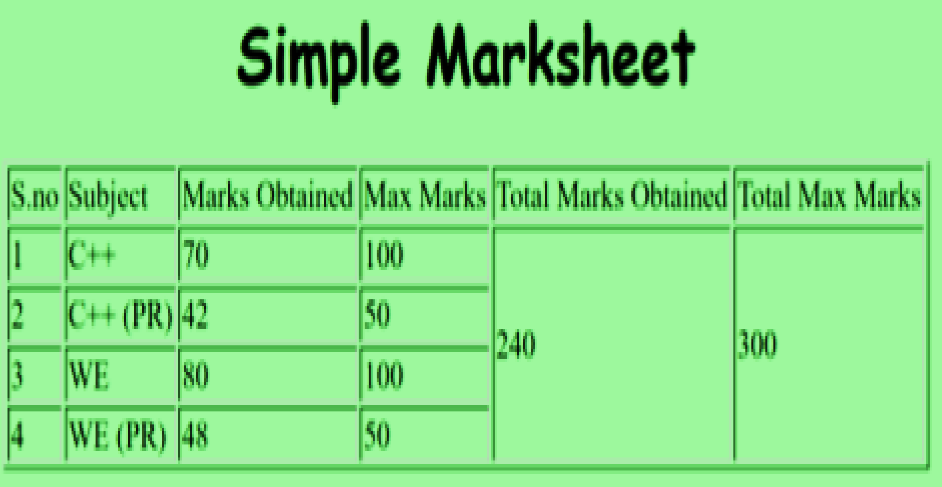
## LAB TAKS

**Task # 1**: Make a 4x4 table, as sample given below;



| S.no | Name | Class | Roll No. |
|---|---|---|---|
| 1 | Ali | 14CS | 02 |
| 2 | Asif | 14CS | 28 |
| 3 | Zawar | 14CS | 40 |
| 4 | Mansoor | 14CS | 98 |

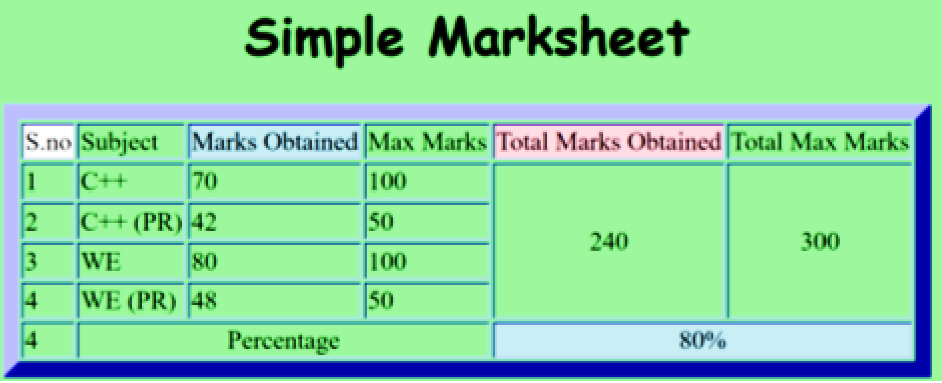
**Task # 2**: Make a mark sheet pattern, as sample given below;



## Simple Marksheet

| S.no | Subject | Marks Obtained | Max Marks | Total Marks Obtained | Total Max Marks |
|---|---|---|---|---|---|
| 1 | C++ | 70 | 100 | 240 | 300 |
| 2 | C++ (PR) | 42 | 50 | | |
| 3 | WE | 80 | 100 | | |
| 4 | WE (PR) | 48 | 50 | | |

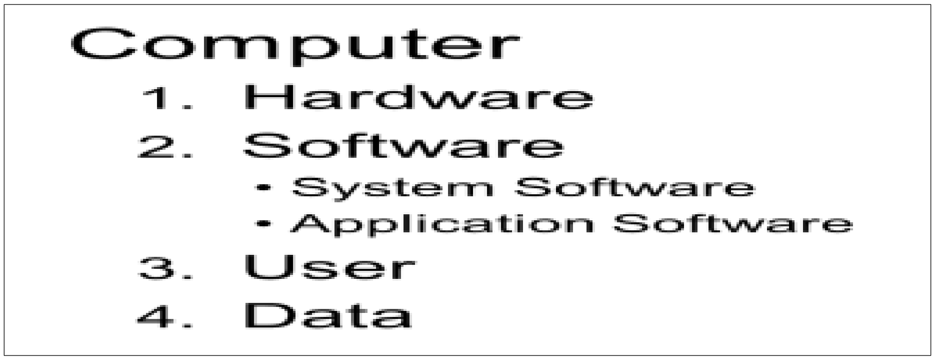
**Task # 3**: Highlight the few cells of table, as sample given below;



## Simple Marksheet

| S.no | Subject | Marks Obtained | Max Marks | Total Marks Obtained | Total Max Marks |
|---|---|---|---|---|---|
| 1 | C++ | 70 | 100 | 240 | 300 |
| 2 | C++ (PR) | 42 | 50 | | |
| 3 | WE | 80 | 100 | | |
| 4 | WE (PR) | 48 | 50 | | |
| 4 | Percentage | | | 80% | |

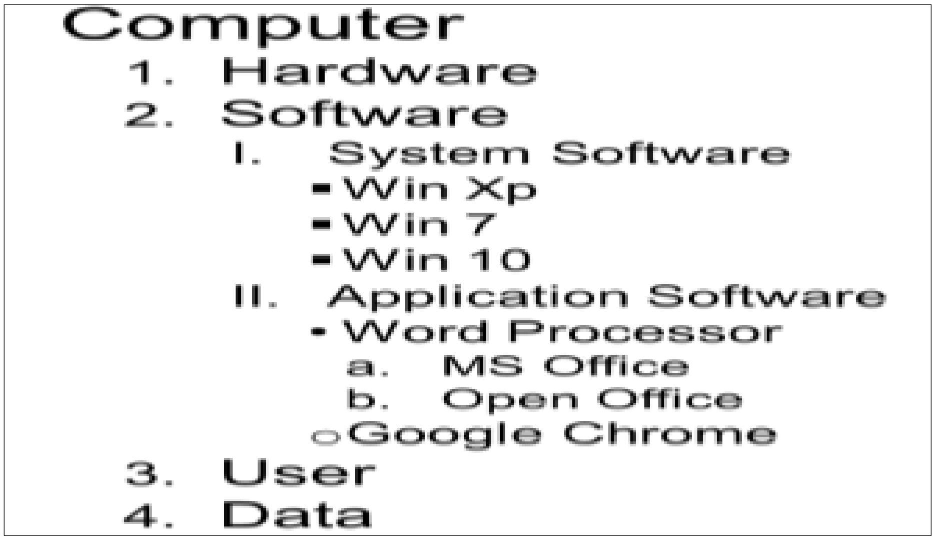
**Task # 4**: Apply various types of lists, sample given below;



Computer

1. Hardware
2. Software
   - System Software
   - Application Software
3. User
4. Data

**Task # 5**: Apply various types of lists, sample given below;



Computer

1. Hardware
2. Software
   - I. System Software
     - ▪ Win Xp
     - ▪ Win 7
     - ▪ Win 10
   - II. Application Software
     - • Word Processor
       - a. MS Office
       - b. Open Office
     - ○ Google Chrome
3. User
4. Data

**Task # 6**: Create current class timetable using HTML Table.

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ____________________________

Subject Teacher: ____________________________

Date: ____________________________

| | | | | | |
|---|---|---|---|---|---|
| Department of:<br>Computer Systems Engineering | | Subject | | | |
| | | WEB ENGINEERING | | | |
| Mehran University of Engineering<br>&Technology<br>Jamshoro | | Year | 3rd | Semester | 6th |
| | | Batch | 17CS | Duration | 03 Hours |

# PRACTICAL # 3

# TO DEMONSTRATE THE EFFECT OF FRAMES & IFRAMES

### HTML Frames

In HTML Frames are used to divide the page into multiple HTML pages. So, using frames we can load multiple different pages on the single HTML page. To divide the HTML page, we use two tags that are, each HTML document is called a frame, and each frame is independent of the others.

### Working with Frames

1) <FRAMESET></FRAMESET>
2) <FRAME></FRAME>

### <FRAMESET></FRAMESET>

***Type:*** Paired Tag
***Purpose:*** It divides the page into rows and columns of the defined sizes.

### <FRAME></FRAME>

***Type:*** Paired Tag
***Purpose:*** It is used to load different pages to the different section of the HTML page, like loading the different pages to the sections created by the frameset tag.

### Attributes of FRAMESET Tag

**ROWS:** It is used to divide the page into multiple rows according to the list of sizes assigned to it.
**COLS** (It is used to divide the page into multiple columns according to the list of sizes assigned to it).
e.g. <FRAMESET ROWS = “30%, *”> . . . </FRAMESET>
<FRAMESET COLS = “20%, 30%, 50%”> . . . </FRAMESET>

### Attributes of FRAME Tag

**SRC** (used to indicate the URL for the page to be loaded).
**MARGINHEIGHT** (Specifies the amount of space left at the top and bottom of the frame).
**MARGINWIDTH** (Specifies the amount of space at the left and right of the frame).
**NAME** (specifies the unique name or the pointer for the frame).
**NORESIZE** (Disables the frame’s resizing capabilities).
**SCROLLING** (Controls the appearance of the horizontal and vertical scrollbars of the frame).

### Example:

=====================================================================

***header.html***

```
<html>
        <body bgcolor = "lightblue">
        <h1><center><font face = "calibri" size = 6> 14 cs</font></center></h1>
        </body>
</html>
```

=========================================================================

***links.html***

```
<html>
<body bgcolor = "orange">
<br>    <br>
        <a href = "intro.html" target = "main" >
                <font face = "calibri" size = 4><b> introduction </b> </font>
        </a>
<br><br>
        <a href = "subjects.html" target = "main" >
                <font face = "calibri" size = 4><b> subjects </b></font>
        </a>
        <br>            <br>
        <a href = "photogallary.html" target = "main" >
                <font face = "calibri" size = 4><b> photo gallary</b></font>
        </a>
</body>
</html>
```

=========================================================================

***intro.htm***

```
<html>
<body bgcolor = "amber">
        <br>    <br>    <h1><center> introduction </center></h1>
</body>
</html>
```

=========================================================================

***subjects.html***

```
<html>
<body bgcolor = "amber">
        <br>    <br>
        <h1><center> subjects </center></h1>
</body>
</html>
```

=========================================================================

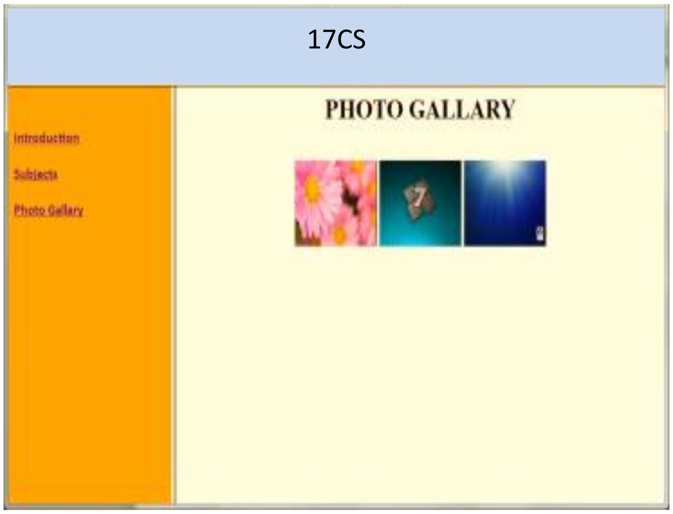
***photogallary.html***

```
<html>
        <body>
        <h1><center> photo gallary</center></h1>
        <br>    <center>
                <imgsrc = "image1.jpg" height = 100 width = 120>
                <imgsrc = "image2.jpg" height = 100 width = 120>
                <imgsrc = "image3.jpg" height = 100 width = 120>
        </center>
        </body>
</html>
```

=====================================OUTPUT============================

***frameset.html***

```
<html>
<frameset rows = "15%, *">
  <frame src = "header.html" name = "header"
  noresize>
  <frameset cols = "25%,*">
  <frame src = "links.html" name = "links"
  noresize>
  <frame src = "photogallary.html" name =
  "main" scrolling = auto>
  </frameset>
        </frameset>
</html>
```



17CS

Introduction

Subjects

Photo Gallery

PHOTO GALLARY

# LAB TASKS

**Task # 1:** Make a webpage in which 2 rows and 2 columns will be shown. In all frames you have to attach different HTML pages, which you have made earlier.

**Task # 2:** Make a webpage in which 2 rows and 2 columns will be shown. In all frames you have to provide different website.

**Task # 3:** Make a webpage, as shown below. Replace contents of these frames with your own choice.

<table>
<tr><td colspan="2"><b>Header</b></td></tr>
<tr><td><b>Menu 1<br>25</b></td><td rowspan="2"><b>Contents</b></td></tr>
<tr><td><b>Menu 1<br>25%</b></td></tr>
</table>

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ______________________________

Subject Teacher: ______________________________

Date: ______________________________

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| Department of:<br>Computer Systems Engineering |  | Subject |  |  |  |
|  |  | WEB ENGINEERING |  |  |  |
| Mehran University of Engineering &Technology Jamshoro |  | Year | 3rd | Semester | 6th |
|  |  | Batch | 17CS | Duration | 03 Hours |

# PRACTICAL # 4

# TO MANAGE THE DIV TAG & ITS ATTRIBUTES

The <DIV> tag is for Division, as in dividing or subdividing a Page into many Sections or functional blocks, for example: Header, Footer, Navigation pane: Main Content pane: Right Pane etc.

This HTML <DIV> tag is a container for other HTML items or elements. This HTML-Div Tag is used to layout HTML elements on a page and can be used to group HTML Elements together. Ultimately, we can say that <DIV> tag is container unit that encapsulates other page elements and divides the HTML document into sections.

You can use the <DIV> Tag to layout your page, The other way to layout your HTML web page is table. I would like to suggest you use Div tag for page payout.

Web designers use <DIV> elements to combine together HTML elements and implement CSS styles to many elements at once. For instance, by covering a set of paragraph elements into a <div> element, the designer can take advantage of CSS styles.

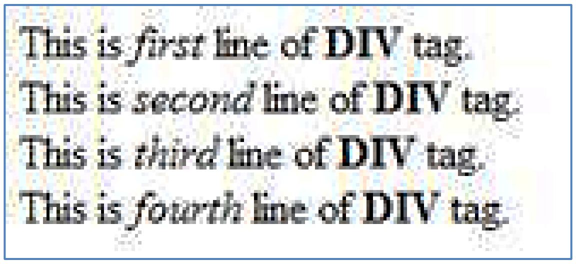
Designer can apply a design to all paragraphs at once by applying a design style to the <DIV> tag instead of coding the same style for each paragraph element.

Within a <DIV>, you can you can nest and align A) Paragraphs B) Headings C) Lists D) Definitions E) Pre-formatted text F) Images G) Tables etc. You can use the ALIGN attribute for Horizontal Text Alignment.

**Using the HTML Div Tag:**
The following section demonstrates, how to use div tag to layout your page.

**Listing 1:** Simple script to display the content of div tag.

```
<html>
<body>
<div> This is first line.  </div>
<div> This is second line. </div>
<div> This is third line. </div>
<div> This is fourth line. </div>
</body>
</html>
```



This is *first* line of DIV tag.
This is *second* line of DIV tag.
This is *third* line of DIV tag.
This is *fourth* line of DIV tag.

**Figure 1:** Above figure is out of script to display the content of div tag.

**HTML Div Tag - Alignment and Style**
First of all we will add border in each Div tag. We can do this by using in-line CSS. In-line CSS is CSS inside HTML which is used for adding style to page

- This is done by adding style=" border: 1px solid red;" to each the div's tag. That will add a red border to one html div.

- It should be kept in mind while designing website that as per requirement add the borders to HTML div tags. You should also take care: what each div contains, where it starts and ends.
- HTML div's alignment can be changed by using the align attribute. To align a div using the align attribute E.g.

**Listing 2:** HTML- Div tag with align property.

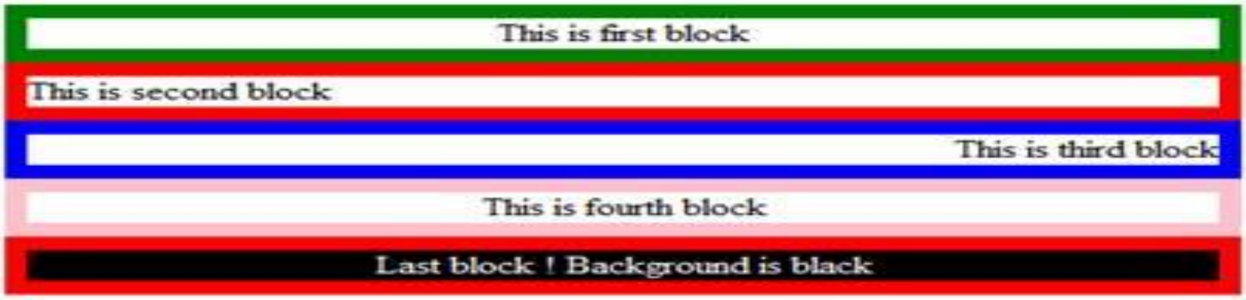
```
<div align="left">"contents"</div>
```

- HTML Div tag can also be used to set layout width and Height, E.g. <div width:100px; Height:100px;">

Following Script will clear the concept of use of align attribute and adding style

**Listing 3:** Script of HTML Div Tag Layout using the Align Attribute and adding Style

```
<html>
<body>
        <div style="width:500px;">
                <div align="center" width="400px;"style="border:9px solid green;">This is first
                block</div>
                <div align="left" width="400px;"style="border:9px solid red;">This is second
                block</div>
                <div align="right" width="400px;"style="border:9px solid blue;">This is third
                block</div>
                <div align="center" width="400px;"style="border:9px solid pink;">This is fourth
                block</div>
                <div align="center" bgcolor="green" width="900px;align="center"
                style="border:9px solid red;background:black;color:white;">Last block !
                Background is black</div>
        </div>
</body>
</html>
```



This is first block

This is second block

This is third block

This is fourth block

Last block ! Background is black

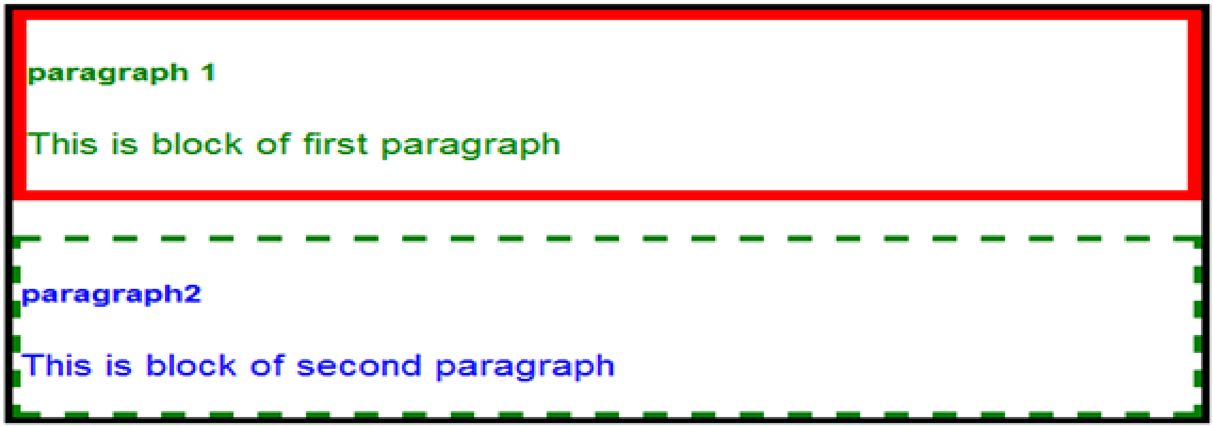
**Figure 2:** The figure shows output of applying alignment and style property on Div tag

*Note: If you will place<Div> elements or items inside of other <Div>elements, it will allow these elements to be further subdivided. Following Script will clear the concept of subdivision.*

**Listing 4:** HTML script for <DIV> inside </DIV>

```
<html>
<body>
        <div style="width:600px;">
                <div id="UgDiv" name="UgDiv" title="Div Element" style="font-family: Arial; font-
                size: 15pt; border: 4px solid black;">
                <div id="subDiv1" name="subDiv1" title="Subdivision Div Element" style="color:
                green; border: 7px solid red;">
                        <h5>paragraph 1</h5>
```

```
            <p>This is block of first paragraph</p>
        </div>
        <br /><div id="subDiv2" name="subDiv2" title="Subdivision Div Element"
        style="color: Blue;border: 4px dashed green;">
        <h5>paragraph2</h5>
        <p>This is block of second paragraph</p>
    </div>
</body>
</html>
```



paragraph 1

This is block of first paragraph

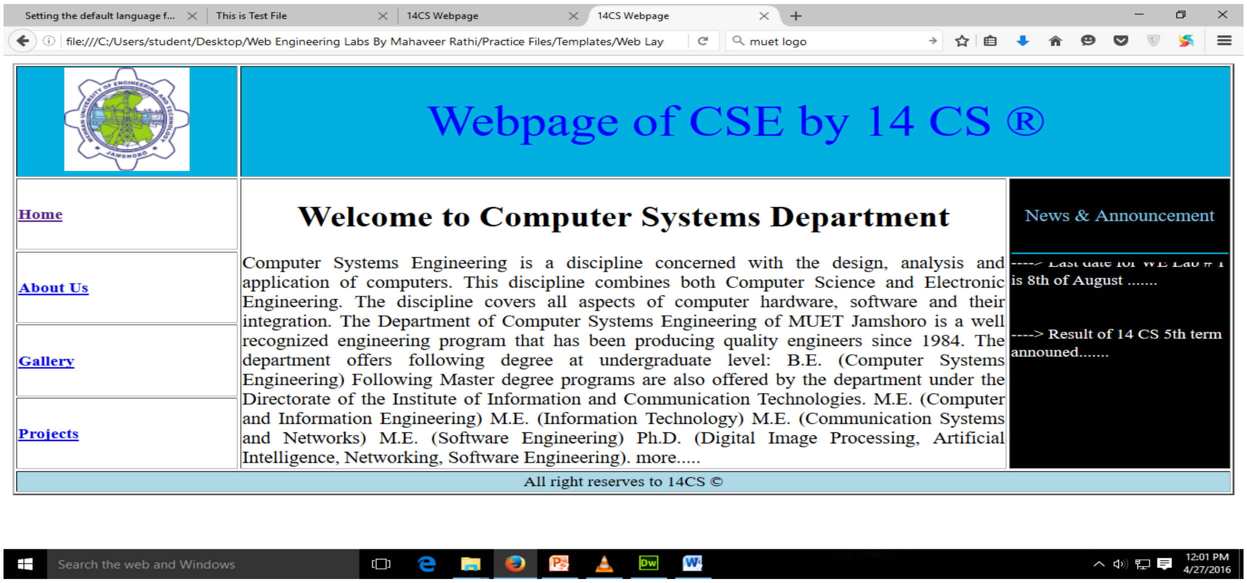
paragraph2

This is block of second paragraph

**Figure 3:** Output of subdivision script of div tag.

# LAB TASKS

**Task # 1:** Make a webpage which look likes as shown below;


Setting the default language f...
This is Test File
14CS Webpage
14CS Webpage
file:///C:/Users/student/Desktop/Web Engineering Labs By Mahaveer Rathi/Practice Files/Templates/Web Lay
muet logo
Search the web and Windows
12:01 PM
4/27/2016

Webpage of CSE by 14 CS ®

Home

About Us

Gallery

Projects

Welcome to Computer Systems Department

Computer Systems Engineering is a discipline concerned with the design, analysis and application of computers. This discipline combines both Computer Science and Electronic Engineering. The discipline covers all aspects of computer hardware, software and their integration. The Department of Computer Systems Engineering of MUET Jamshoro is a well recognized engineering program that has been producing quality engineers since 1984. The department offers following degree at undergraduate level: B.E. (Computer Systems Engineering) Following Master degree programs are also offered by the department under the Directorate of the Institute of Information and Communication Technologies. M.E. (Computer and Information Engineering) M.E. (Information Technology) M.E. (Communication Systems and Networks) M.E. (Software Engineering) Ph.D. (Digital Image Processing, Artificial Intelligence, Networking, Software Engineering). more.....

News & Announcement

----> Last date for WE Lab # 1 is 8th of August .......

----> Result of 14 CS 5th term announed.......

All right reserves to 14CS ©

**Task # 2:**

1. Create several divs of various widths, heights, and colors. Use margin to push them around the page.
2. Make some of the <div>'s into links (anchors). Then, make the background color of the div change on mouse over.
3. Try to position some of the <div>'s relatively in a vertical line down the middle of the page.
4. Create a class that you can use with <span> tags to make certain text twice the default size.

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ______________________________

Subject Teacher: ______________________________

Date: ______________________________

| | | | | | |
|---|---|---|---|---|---|
| Department of:<br>Computer Systems Engineering | | Subject | | | |
| | | WEB ENGINEERING | | | |
| Mehran University of Engineering &Technology<br>Jamshoro | | Year | 3rd | Semester | 6th |
| | | Batch | 17CS | Duration | 03 Hours |

# PRACTICAL # 5

## TO MANIPULATE THE VARIOUS FORMS ELEMENTS

### HTML Forms

HTML forms are used to pass data to a server. A form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, textarea, fieldset, legend, and label elements. The <form> tag is used to create an HTML form:

```
<form>
input elements
</form>
```

### HTML Forms - The Input Element

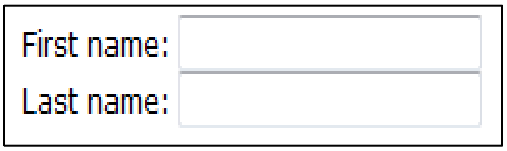
The most important form element is the input element. The input element is used to select user information. An input element can vary in many ways, depending on the type attribute. An input element can be of type text field, checkbox, password, radio button, submit button, and more.
The most used input types are described below.

### Text Fields

<input type="text" /> defines a one-line input field that a user can enter text into:

```
<form>
    First name: <input type="text" name="firstname" /> <br />
    Last name: <input type="text" name="lastname" />
</form>
```

How the HTML code above looks in a browser:

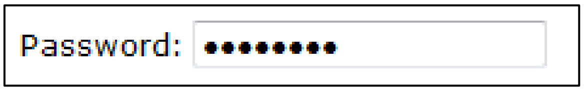


**Note:** The form itself is not visible. Also note that the default width of a text field is 20 characters.

### Password Field

<input type="password" /> defines a password field:

```
<form>
Password: <input type="password" name="pwd" />
</form>
```



How the HTML code above looks in a browser:

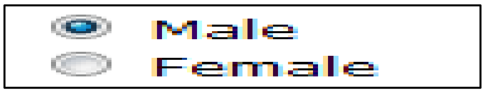
**Note:** The characters in a password field are masked (shown as asterisks or circles).

### Radio Buttons

<input type="radio" /> defines a radio button. Radio buttons let a user select ONLY ONE of a limited number of choices:

```
<form>
<input type="radio" name="sex" value="male" /> Male<br />
<input type="radio" name="sex" value="female" /> Female
</form>
```

How the HTML code above looks in a browser:



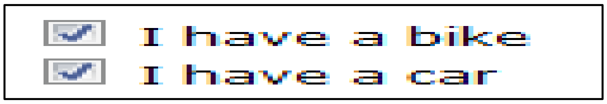
● Male
○ Female

### Checkboxes

<input type="checkbox" /> defines a checkbox. Checkboxes let a user select ZERO or MORE options of a limited number of choices.

```
        <form>
                <input type="checkbox" name="vehicle" value="Bike" /> I have a bike<br />
                <input type="checkbox" name="vehicle" value="Car" /> I have a car
        </form>
```

How the HTML code above looks in a browser:



☑ I have a bike
☑ I have a car

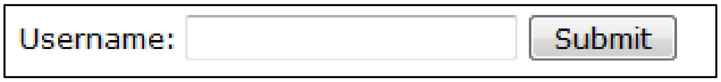
### Submit Button

<input type="submit" /> defines a submit button.

A submit button is used to send form data to a server. The data is sent to the page specified in the form's action attribute. The file defined in the action attribute usually does something with the received input:

```
        <form name="input" action="html_form_action.asp" method="get">
                Username: <input type="text" name="user" />
                <input type="submit" value="Submit" />
        </form>
```

How the HTML code above looks in a browser:



Username:

If you type some characters in the text field above, and click the "Submit" button, the browser will send your input to a page called "html_form_action.asp". The page will show you the received input.

### HTML <form> method Attribute

The method attribute specifies how to send form-data (the form-data is sent to the page specified in the action attribute).

The form-data can be sent as URL variables (with method="get") or as HTTP post (with method="post").

**Notes on the "get" method:**

- This method appends the form-data to the URL in name/value pairs
- This method is useful for form submissions where a user wants to bookmark the result
- There is a limit to how much data you can place in a URL (varies between browsers), therefore, you cannot be sure that all of the form-data will be correctly transferred
- Never use the "get" method to pass sensitive information! (password or other sensitive information will be visible in the browser's address bar)

**Notes on the "post" method:**

- This method sends the form-data as an HTTP post transaction
- Form submissions with the "post" method cannot be bookmarked
- The "post" method is more robust and secure than "get", and "post" does not have size limitations

**Syntax:** <form method="get|post">

### HTML <form> Action Attribute

The required action attribute specifies where to send the form-data when a form is submitted.

**Syntax:** <form action="*URL*">

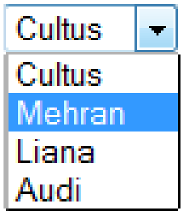
**Attribute Values**

| **Value** | **Description** |
|---|---|
| *URL* | Where to send the form data.<br><br>Possible values:<br>• An absolute URL - points to another web site (like action="http://www.example.com/example.htm")<br>• A relative URL - points to a file within a web site (like action="example.htm") |

### Select List

It is used to create drop down list of items that the user can select. <select></select> defines the drop down list and <option></option> defines the drop down list items.

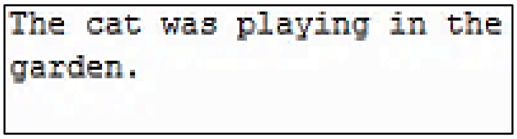
```
<select name="cars">
        <option value="cultus">Cultus</option>
        <option value="mehran">Mehran</option>
        <option value="liana">Liana</option>
        <option value="audi">Audi</option>
</select>
```



### Text Area

It is used to send multiline text to the server. <textarea></textarea> defines the text area.

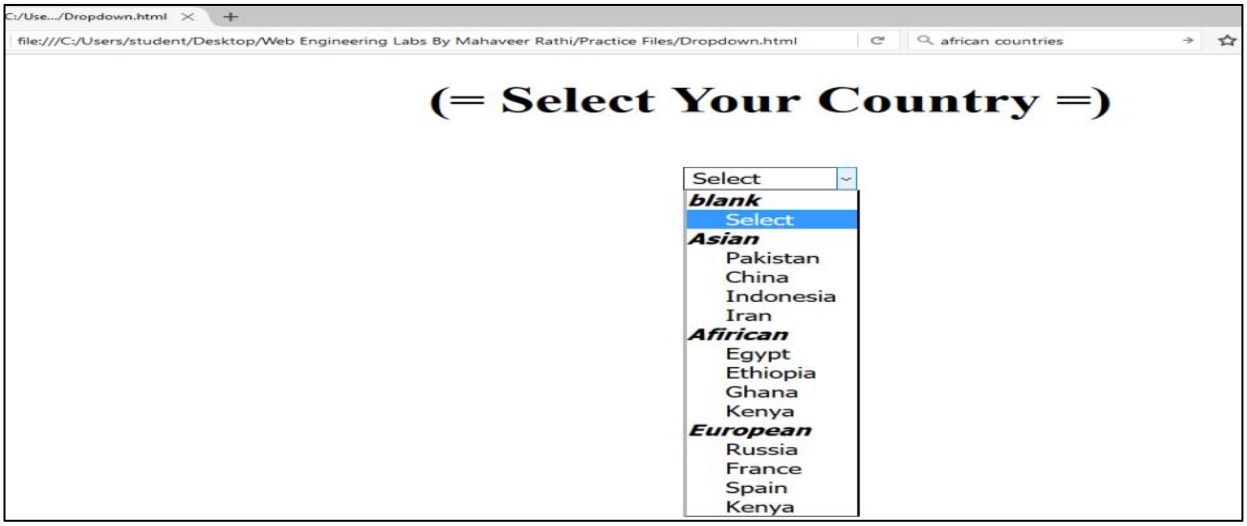
```
<textarea rows="10" cols="30">
        The cat was playing in the garden.
</textarea>
```



The cat was playing in the garden.

# LAB TASKS

- **Task # 1:** Make dropdown, as shown below;

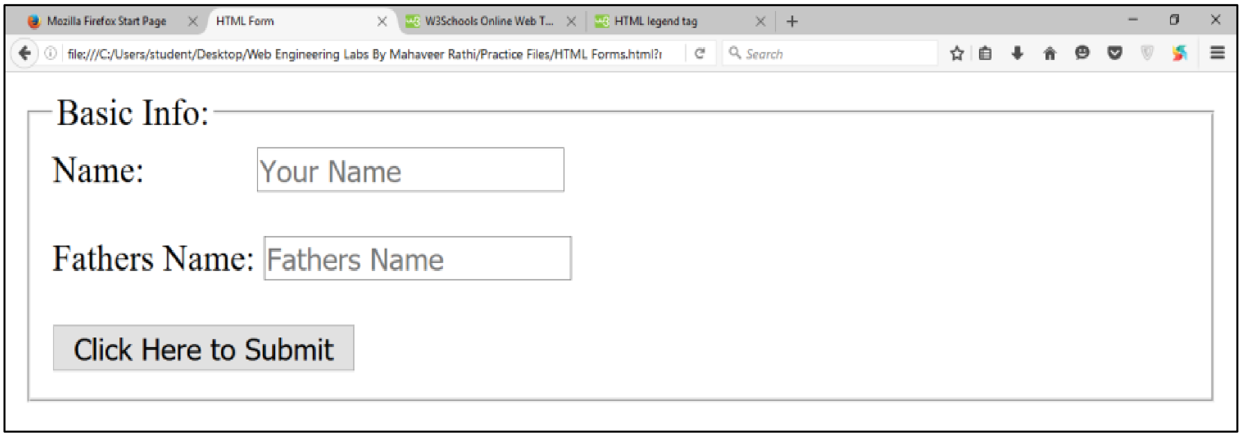

C:/Use.../Dropdown.html
file:///C:/Users/student/Desktop/Web Engineering Labs By Mahaveer Rathi/Practice Files/Dropdown.html
african countries

(= Select Your Country =)

Select

***blank***
Select
***Asian***
Pakistan
China
Indonesia
Iran
***African***
Egypt
Ethiopia
Ghana
Kenya
***European***
Russia
France
Spain
Kenya

**Task # 2:** Make HTML form, as shown below;


Mozilla Firefox Start Page
HTML Form
W3Schools Online Web T...
HTML legend tag
file:///C:/Users/student/Desktop/Web Engineering Labs By Mahaveer Rathi/Practice Files/HTML Forms.html?r
Search

Basic Info:

Name: Your Name

Fathers Name: Fathers Name

Click Here to Submit

**Task #3:**
Create a HTML form in which at least 5 different form elements must be used.

**Task #4:**
What is difference between get and post method describe briefly? Discuss with the help of example. Can we send the form data to another html page without get and post methods? If yes, then how? If No, then Why?

**Task #5:**
What is difference b/w <input type=" button" > and <input type=" submit">? Discuss with the help of example.

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ____________________________

Subject Teacher: ____________________________

Date: ____________________________

| | | | | | |
|---|---|---|---|---|---|
| Department of:<br>Computer Systems Engineering | | Subject | | | |
| | | WEB ENGINEERING | | | |
| Mehran University of Engineering<br>&Technology<br>Jamshoro | | Year | 3rd | Semester | 6th |
| | | Batch | 17CS | Duration | 03<br>Hours |

# PRACTICAL # 6

# TO EXECUTE THE JAVASCRIPT

### JavaScript

JavaScript is a scripting language used to enable programmatic access to objects within other applications. It is primarily used in the form of client-side JavaScript for the development of dynamic websites. JavaScript was influenced by many languages and was designed to look like Java, but to be easier for non-programmers to work with.

In order to write the JavaScript code in HTML we use the <SCRIPT></SCRIPT> and Set the LANGUAGE attribute to “Javascript” as,

```
        <SCRIPT LANGUAGE = “Javascript”> . . . </SCRIPT>
Or
        <SCRIPT TYPE = “text/javascript”> . . . </SCRIPT>
```

### JavaScript Programming Constructs

The JavaScript Programming constructs are simply the control structures that are used to implement the conditional or iterative logic into the program to complete the program. These structures may include the condition checking statements, iterative / loop structure or other types of JS statements.

**If Statement:** It checks the condition and executes the “if” body if and only if the condition is true else the body of “if” is skipped hence not executed.

***Syntax:***

```
        if (Condition)
        {
                //JS Code
        }
```

**If – else Statement:** It checks the condition and executes the “if” body, if the condition is true else the “else” body is executed.

***Syntax:***

```
        if (Condition)
        {
                //JS IF Code
        }
        else
        {
```

```
        //JS else Code
}
```

**Immediate If Statement:** It checks the condition and returns value1 if and only if the condition is true else the value2 is returned.

***Syntax:***

```
Var result = (Condition) ?value1 : value2
```

**5.2.4 Switch Case Statement:** It is used when the bunch of conditions depend up on single variable/expression. It evaluates the expression or takes the value of the variable provided in the parenthesis and compares it with the cases and if the value matches with any case than the corresponding case body is executed else the default block is executed.

***Syntax:***

```
Switch (expression/value/variable)
{
Case value1:
        //JS Code
break;

Case value N:
        //JS Code
break;

default:
        //JS Code
}
```

**For Loop:** It is used to execute the particular block of code for finite known number of times.

***Syntax:***

```
for (initialization expression; test expression; increment expression)
{
        //JS Code
}
```

**While Loop**: It is used to execute the particular block of code for unknown number of times or until the condition becomes false.

***Syntax:***

```
while (Condition)
{
        //JS Code
}
```

**Do While Loop:** It is used to execute the particular block of code for unknown number of times. It is like while loop but do while loop is executed at least once regardless of the condition also here the loop condition is checked at last.

***Syntax:***

```
do
{
        //JS Code
} while (Condition);
```

# LAB TASKS

**Task # 1:** Make a webpage, which will ask user whether inputted no is even/odd & generate alert box accordingly;

**Task # 2:** Write a JavaScript program to convert temperatures to and from Celsius, Fahrenheit.

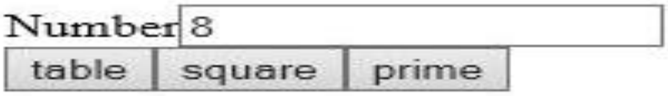
**Task #3:** Create a program in JavaScript which will ask a number from user and display weather inputted number is prime or not.

```
e.g         Enter Number:  5
            It is a prime (display msg)
            Enter Number:6
            It is not prime(display msg)
```

**Task #4:** Create a program in JavaScript, which print the following pattern by using nested loops.

```
        *
      * *
    * * *
  * * * *
* * * * *
```

**Task #5:** Create a program in JavaScript that ask a number from user. And generate three buttons. If user clicks on table button then it prints the table of inputted number, if user enter square button than it displays the square of inputted number, if user clicks on prime button then it shows weather the inputted number is prime or not.



Number 8

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ______________________________

Subject Teacher: ______________________________

Date: ______________________________

| | | | | | |
|---|---|---|---|---|---|
| Department of:<br>Computer Systems Engineering | | Subject | | | |
| | | WEB ENGINEERING | | | |
| Mehran University of Engineering<br>&Technology<br>Jamshoro | | Year | 3rd | Semester | 6th |
| | | Batch | 17CS | Duration | 03 Hours |

# PRACTICAL # 7

# TO PERFORM THE CLIENT-SIDE VALIDATION & DATE OBJECT IN JS

### Form Validation

Ideally, users will fill the web form with necessary information and finish their job successfully. However, people often make mistakes. This is where web form validation comes into play. The goal of web form validation is to ensure that the user provided necessary and properly formatted information needed to successfully complete an operation.

### Validation Methods

User's input can be validated on the server and on the client (web browser). Thus, we have server-side and client-side validation. We'll discuss pros and cons of each.

### Server-Side Validation

In the server-side validation, information is being sent to the server and validated using one of server-side languages. If the validation fails, the response is then sent back to the client, page that contains the web form is refreshed and a feedback is shown. This method is secure because it will work even if JavaScript is turned off in the browser and it can't be easily bypassed by malicious users. On the other hand, users will have to fill in the information without getting a response until they submit the form. This results in a slow response from the server.

### Client-Side Validation

Server-side validation is enough to have a successful and secure form validation. For better user experience, however, you might consider using client-side validation. This type of validation is done on the client using script languages such as JavaScript. By using script languages user's input can be validated as they type. This means a more responsive, visually rich validation.

With client-side validation, form never gets submitted if validation fails. Validation is being handled in JavaScript methods that you create (or within frameworks/plugins) and users get immediate feedback if validation fails.

Main drawback of client-side validation is that it relies on JavaScript. If users turn JavaScript off, they can easily bypass the validation. This is why validation should always be implemented on both the client and server. By combining server-side and client-side methods we can get the best of the two: fast response, more secure validation and better user experience.

### Date object

The Date object is used to work with dates and times. The Date object is useful when you want to display a date or use a timestamp in some sort of calculation. In JavaScript, you can either make a Date object by supplying the date of your choice, or you can let JavaScript create a Date object based on your visitor's system clock. It is usually best to let JavaScript simply use the system clock.

Date objects are created with new Date(). There are four ways of instantiating a date:

var d = new Date();

var d = new Date(*milliseconds*);

var d = new Date(*dateString*);

var d = new Date(*year*, *month*, *day*, *hours*, *minutes*, *seconds*, *milliseconds*);

Most parameters above are optional. Not specifying, causes 0 to be passed in.

Once a Date object is created, a number of methods allow you to operate on it. Most methods allow you to get and set the year, month, day, hour, minute, second, and milliseconds of the object, using either local time or UTC (universal, or GMT) time.

All dates are calculated in milliseconds from 01 January, 1970 00:00:00 Universal Time (UTC) with a day containing 86,400,000 milliseconds.

Some examples of initiating a date:

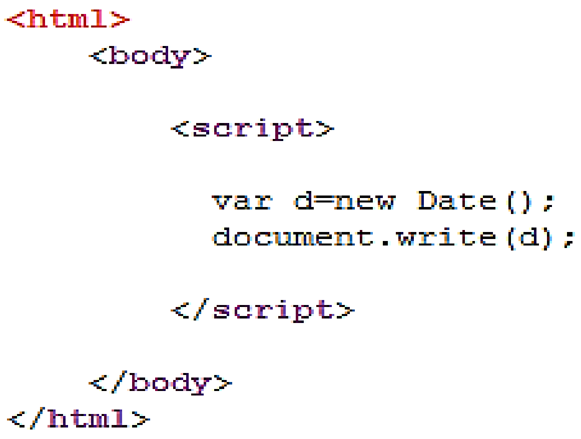
var today = new Date()

var d1 = new Date("October 13, 1975 11:13:00")

var d2 = new Date(79,5,24)

var d3 = new Date(79,5,24,11,33,0)

### Displaying Today's Date



```
<html>
    <body>

        <script>

          var d=new Date();
          document.write(d);

        </script>

    </body>
</html>
```

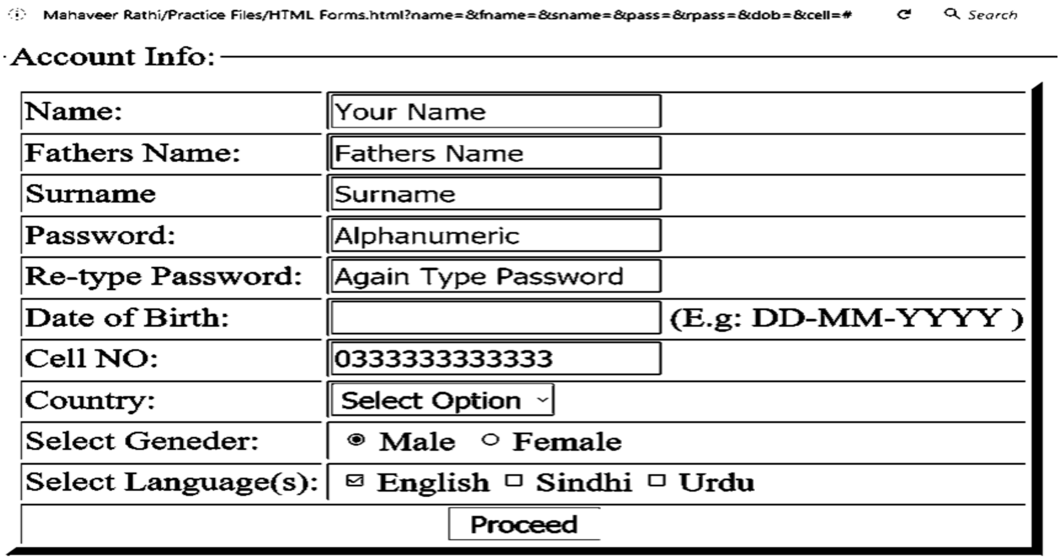
### Date Object Methods

| Method | Description |
|---|---|
| getDate() | Returns the day of the month (from 1-31) |
| getDay() | Returns the day of the week (from 0-6) |
| getFullYear() | Returns the year (four digits) |
| getHours() | Returns the hour (from 0-23) |
| getMilliseconds() | Returns the milliseconds (from 0-999) |
| getMinutes() | Returns the minutes (from 0-59) |
| getMonth() | Returns the month (from 0-11) |

| | |
|---|---|
| getSeconds() | Returns the seconds (from 0-59) |
| getTime() | Returns the number of milliseconds since midnight Jan 1, 1970 |
| getTimezoneOffset() | Returns the time difference between UTC time & local time, in minutes |
| getYear() | Deprecated. Use the getFullYear() method instead |
| setDate() | Sets the day of the month of a date object |
| setFullYear() | Sets the year (four digits) of a date object |
| setHours() | Sets the hour of a date object |
| setMilliseconds() | Sets the milliseconds of a date object |
| setMinutes() | Set the minutes of a date object |
| setMonth() | Sets the month of a date object |
| setSeconds() | Sets the seconds of a date object |
| setTime() | Sets a date and time by adding or subtracting a specified number of milliseconds to/from midnight January 1, 1970 |
| setYear() | Deprecated. Use the setFullYear() method instead |
| toString() | Converts a Date object to a string |

# LAB TASKS

- **Task # 1:** Make a webpage, as shown below;



Mahaveer Rathi/Practice Files/HTML Forms.html?name=&fname=&sname=&pass=&rpass=&dob=&cell=#

Account Info:

| | | |
|---|---|---|
| Name: | Your Name | |
| Fathers Name: | Fathers Name | |
| Surname | Surname | |
| Password: | Alphanumeric | |
| Re-type Password: | Again Type Password | |
| Date of Birth: | | (E.g: DD-MM-YYYY ) |
| Cell NO: | 03333333333333 | |
| Country: | Select Option | |
| Select Geneder: | ● Male ○ Female | |
| Select Language(s): | ☑ English □ Sindhi □ Urdu | |
| | Proceed | |

**Task 2:** Create a web page which checks the NIC validation using JavaScript. If format of NIC is valid then form can be submitted.

**Task 3:** Create a form in html which contains name, caste, rollno, department and submit button and check for all the fields must contains text. If a user enters any special key or number, then it shows the error message of "Invalid data".

**Task # 4:** Make a webpage, which will ask your date of birth and display your age in years, months & days.

**Task # 5:** Write a JavaScript program to display the current day and time in the following format.

"Today is : Saturday."
"Current Time : 6 PM : 1 : 20"

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ______________________________

Subject Teacher: ______________________________

Date: ______________________________

# PRACTICAL # 8

# TO DEVELOP DHTML BASED WEBPAGES

## DHTML Events

By using JavaScript, we have the ability to create dynamic web pages. Events are actions that can be detected by JavaScript. Every element on a web page has certain events which can trigger a JavaScript. For example, we can use the onClick event of a button element to indicate that a function will run when a user clicks on the button. We define the events in the HTML tags.

**Examples of events:**

- A mouse click
- A web page or an image loading
- Mousing over a hot spot on the web page
- Selecting an input field in an HTML form
- Submitting an HTML form
- A keystroke

Events are normally used in combination with functions, and the function will not be executed before the event occurs!

### onLoad and onUnload

The onLoad and onUnload events are triggered when the user enters or leaves the page.
The onLoad event is often used to check the visitor's browser type and browser version, and load the proper version of the web page based on the information.

### onFocus, onBlur and onChange

The onFocus, onBlur and onChange events are often used in combination with validation of form fields.
Below is an example of how to use the onChange event. The checkEmail() function will be called whenever the user changes the content of the field:

```
<input type="text" size="30" id="email" onChange="checkEmail()">
```

### OnSubmit

The onSubmit event is used to validate ALL form fields before submitting it.

Below is an example of how to use the onSubmit event. The checkForm() function will be called when the user clicks the submit button in the form. If the field values are not accepted, the submit should be cancelled. The function checkForm() returns either true or false. If it returns true the form will be submitted, otherwise the submit will be cancelled:

```
<form method="post" action="abc.htm" onSubmit="return checkForm()">
```

### onMouseOver and onMouseOut

onMouseOver and onMouseOut are often used to create "animated" buttons.

Below is an example of an onMouseOver event. An alert box appears when an onMouseOver event is detected:

<a href="http://www.muet.edu.pk" onMouseOver=" alert ('An onMouseOver event'); return false">Mouse Over Me! </a>

## Form Events

| Event | Description |
|---|---|
| onblur | The event occurs when an element loses focus |
| onchange | The event occurs when the content of a form element, the selection, or the checked state have changed (for <input>, <keygen>, <select>, and <textarea>) |
| onfocus | The event occurs when an element gets focus |
| onfocusin | The event occurs when an element is about to get focus |
| onfocusout | The event occurs when an element is about to lose focus |
| oninput | The event occurs when an element gets user input |
| oninvalid | The event occurs when an element is invalid |
| onreset | The event occurs when a form is reset |
| onsearch | The event occurs when the user writes something in a search field (for <input="search">) |
| onselect | The event occurs after the user selects some text (for <input> and <textarea>) |
| onsubmit | The event occurs when a form is submitted |

## Drag Events

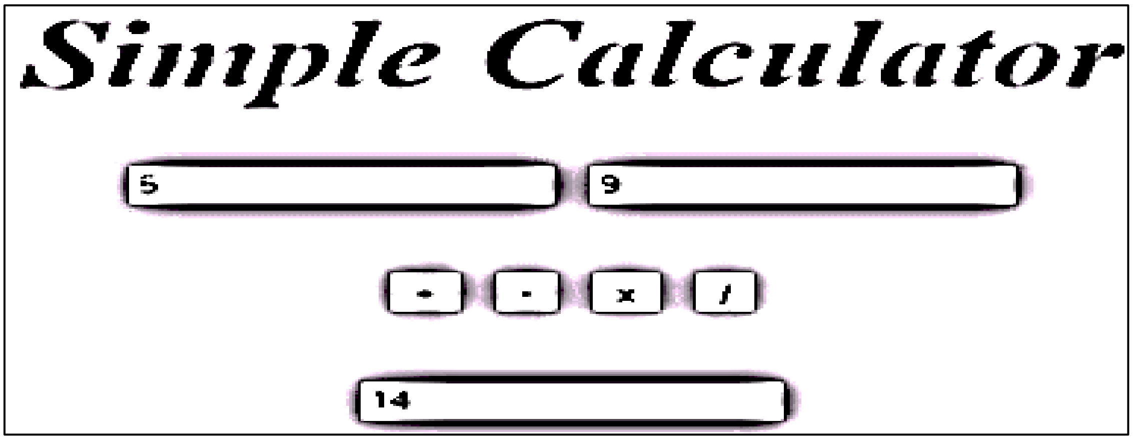
| Event | Description |
|---|---|
| ondrag | The event occurs when an element is being dragged |
| ondragend | The event occurs when the user has finished dragging an element |
| ondragenter | The event occurs when the dragged element enters the drop target |
| ondragleave | The event occurs when the dragged element leaves the drop target |
| ondragover | The event occurs when the dragged element is over the drop target |
| ondragstart | The event occurs when the user starts to drag an element |
| ondrop | The event occurs when the dragged element is dropped on the drop target |

# LAB TASKS

**Task # 1:** Make a webpage, generate alert box when user setFocus on an input box.

**Task # 2:** Make a webpage having a single button. If, user clicks 1st time on button then background color of the page should be changed as sky blue, when user press 2nd time on button then background color of the page color should be changed to orange color/any other color of your choice.

**Task # 3:** Create a calculator in JavaScript by using the functions including with some styles (e.g.; Background color, font size, borders etc.)



Simple Calculator

5

9

14

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ______________________________

Subject Teacher: ______________________________

Date: ______________________________

| | | | | | |
|---|---|---|---|---|---|
| Department of:<br>Computer Systems Engineering | | Subject | | | |
| | | WEB ENGINEERING | | | |
| Mehran University of Engineering &Technology<br>Jamshoro | | Year | 3rd | Semester | 6th |
| | | Batch | 17CS | Duration | 03 Hours |

# PRACTICAL # 9

## TO PRACTICE THE CASCADING STYLE SHEETS (CSS) EFFECTS ON HTML PAGES

### Introduction

Cascading Style Sheets (CSS) is a language that allows you to define how you want your html document to look. This concerns features such as typeface, background, link colors, margins, and placement of objects on a page.

### Why to use CSS?

1. Style sheets make managing the look of multiple web pages MUCH easier by separating CONTENT from DISPLAY information.

2. They make web pages faster to download, very important when you are trying to reach people with older computers and modems.

3. The W3C is slowly phasing out old ways of coding HTML in its standards, for example the use of the <font> tag. Professional standards are moving toward style sheets.

### Where to use CSS?

CSS style rules can be coded in three places:

1. Inline - right in the HTML tag
2. Internal Style sheet - at the beginning of an html document between the <head></head> tags, and enclosed by the <style type="text/css"></style> tags.
3. External Style Sheet - in a separate document that is used as a reference for multiple html pages, and there is a kind of "link" (example below) in the html pages that tells the browser where to look for the instructions:

```
<link rel="Stylesheet" media="screen" type="text/css" href="eportfolio.css" />
```

### Cascading Styles Priority Orders

1. Inline style (inside an HTML element)
2. Internal style sheet (inside the <head> tag)
3. External style sheet
4. Browser default

So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style declared inside the <head> tag, in an external style sheet, or in a browser (a default value).

**Syntax:** The CSS syntax is made up of three parts: a selector, a property and a value:

```
selector {property: value}
```

**Note:** If you wish to specify more than one property, you must separate each property with a semicolon. The example below shows how to define a center aligned paragraph, with a red text color:

```
P
{
text-align: center;
color: black;
font-family: arial
}
```

**Grouping:** You can group selectors. Separate each selector with a comma. In the example below we have grouped all the header elements. All header elements will be displayed in green text color:

```
h1, h2, h3, h4, h5, h6
{         color: green;
}
```

**The class Selector**

With the class selector you can define different styles for the same type of HTML element.

Say that you would like to have two types of paragraphs in your document: one right-aligned paragraph, and one center-aligned paragraph. Here is how you can do it with styles:

```
p.right {text-align: right}

p.center {text-align: center}
```

You have to use the class attribute in your HTML document:

```
<p class="right">
This paragraph will be right-aligned.
</p>
<p class="center">
This paragraph will be center-aligned.
</p>
```

**Note:** Only one class attribute can be specified per HTML element! The example below is wrong:

```
<p class="right" class="center">
This is a paragraph.
</p>
```

You can also omit the tag name in the selector to define a style that will be used by all HTML elements that have a certain class. In the example below, all HTML elements with class="center" will be center-aligned:

```
.center {text-align: center}
```

In the code below both the h1 element and the p element have class="center". This means that both elements will follow the rules in the ".center" selector:

```
<h1 class="center">
This heading will be center-aligned
</h1>
<p class="center">
This paragraph will also be center-aligned.
</p>
```

Do **NOT** start a class name with a number! It will not work in Mozilla/Firefox.

### The id Selector

You can also define styles for HTML elements with the id selector. The id selector is defined as a #.

The style rule below will match the element that has an id attribute with a value of "green":

```
#green {color: green}
```

### CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date. A comment will be ignored by browsers. A CSS comment begins with "/*", and ends with "*/", like this:

```
/* This is a comment */
p
{
text-align: center;
/* This is another comment */
color: black;
font-family: arial
}
```

**<u>Below listed are the charts of various CSS properties:</u>**

## <u>CSS Text</u>

| **Property: Value (Description)** |
|---|
| Color: *Color* (Sets the color of a text) |
| Direction: Ltr, rtl (Sets the text direction) |
| letter-spacing Normal, l*ength*<br>*(*Increase or decrease the space between characters) |
| text-align: Left, right, center, justify (Aligns the text in an element) |
| text-decoration: None, underline, overline, line-through, blink<br>(Adds decoration to text) |
| text-indent: *Length, %* (Indents the first line of text in an element) |

| |
|---|
| text-shadow : None, *color*, *length* |
| text-transform None, capitalize, uppercase, lowercase<br>(Controls the letters in an element) |
| unicode-bidi : Normal/ embed/ bidi-override |
| white-space: Normal, pre, nowrap<br>(Sets how white space inside an element is handled) |
| word-spacing Normal, *length*<br>*(*Increase or decrease the space between words) |

**CSS Font**

| **Property: Value** |
|---|
| Font*: font-style/ font-variant/ font-weight*<br>*font-size/line-height/ font-family/* caption<br>icon/ menu/ message-box/ small-caption/status-bar |
| font-family*: family-name / generic-family* |
| font-size: xx-small/x-small/small/medium/large<br>x-large/xx-large/smaller/larger/*length/%* |
| font-size-adjust: none / *number* |
| font-stretch: normal /wider/ narrower/ ultra-condensed/ extra-condensed /condensed<br>/ semi-condensed/ semi-expanded/ expanded<br>extra-expanded/ ultra-expanded |
| font-style: normal/italic/oblique |
| font-variant: normal/small-caps |
| font-weight: normal/bold/bolder/lighter/100/900 |

**CSS Border**

| **Property: Value** |
|---|
| Border*: border-width/border-style/border-color* |
| border-bottom: *border-bottom-width/border-style/border-color* |
| border-bottom-color: *border-color* |
| border-bottom-style: *border-style* |
| border-bottom-width: thin/medium/thick/*length* |
| border-color: color |
| border-left: *border-left-width/border-style/border-color* |
| border-left-color: *border-color* |
| border-left-style: *border-style* |
| border-left-width: thin/medium/thick/*length* |
| border-right: *border-right-width/ border-style/ border-color* |
| border-right-color: *border-color* |
| border-right-style: *border-style* |
| border-right-width: thin/medium/thick/*length* |

| |
|---|
| border-style: none/hidden/dotted/dashed/solid/double /groove /ridge /inset /outset |
| border-top: *border-top-width/border-style/border-color* |
| border-top-color: *border-color* |
| border-top-style: *border-style* |
| border-top-width: thin/medium/thick/*length* |
| border-width: thin/medium/thick/*length* |

### CSS Dimension

| **Property : Value** |
|---|
| Height : auto/ *length / %* |
| line-height: normal/*number/ length / %* |
| max-height :none / *length / %* |
| max-width: none/ *length / %* |
| min-height: *length / %* |
| min-width: *length / %* |
| Width: Auto / *length / %* |

**Example:**

```
<html>
<head> <title>John’s Homepage</title>
<link rel="stylesheet" type="text/css" href="base.css">
<style type="text/css">
        body {
                text-align: left;           font-size: 1em;
                }
        a {
                color: #6666FF;             font-size: 1.2em;
                }
        #nav {
                border: 1px dashed blue;
                }
        #nav a {
                font-size: 1.2em;           color: #990000;
                }
        #content {
                text-align: right;          border: 1px solid green;
                }
        .contentp {
                color: #E62B86;             background: #EFB3C6;
                }
        p.important {
                color: #AE00FF;             background: #C4B87F;
                }
</style></head>
<body>
<div id="nav">
```

```
    <a href="index.html">Home</a>       <a href="courses.html">Courses</a>
    <a href="personal.html">Personal</a>
</div>
<div id="content">
    <p style="color: red; background-color: yellow;">This is my brief description for my
webpage.<br />
    Contact: <a href="Mahaveer.rathi@gmail.com">Contact</a><br />
    <a href="http://www.muet.edu.pk ">Go to the Home Page</a>
    </p>
    <p>This is my new test p.</p>
    <p class="contentp">This is my first test paragraph</p>
    <p class="contentp">This is my second test paragraph</p>
    <p class="contentp">This is my third test paragraph</p>
    <p class="important"> This is my important paragraph!</p>
    <div>
    <imgsrc="palmtree.jpg" style="border: solid; border-color: green;" alt="a picture of a
palm tree"/>
    <imgsrc="palmtree.jpg" alt="a picture of a palm tree"/>
    </div>
</div>
</body>
</html>
```

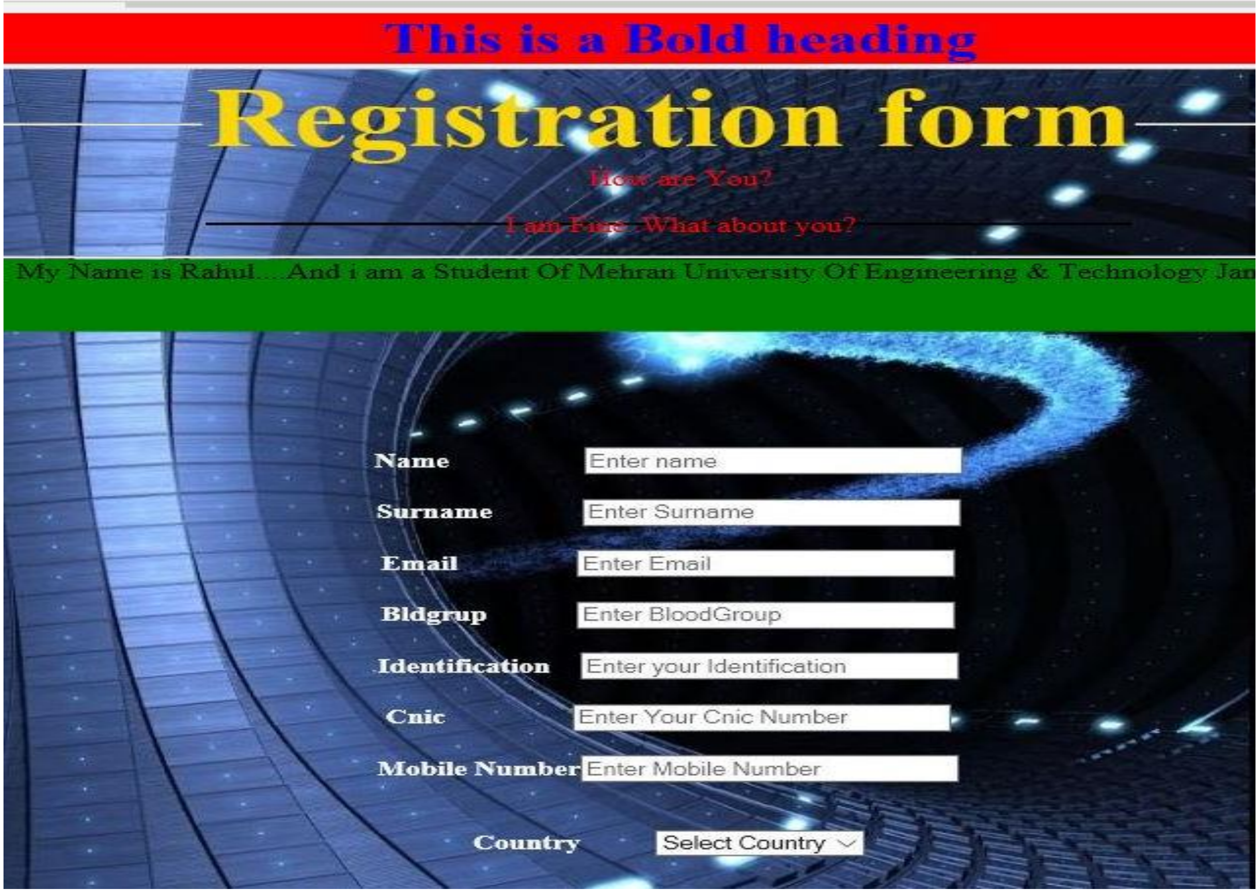
# LAB TASKS

**Task# 1:** Type a paragraph in the HTML page with <h3> style

**Task# 2:**

.Create h3 style (with no leading dot), existing sheet
• Font= Arial, Helvetica, sans serif
• Font size= 24 px
• Font color red, weight bold or bolder

**Task #3:** Create html form use different input fields apply External CSS



This is a Bold heading

Registration form

How are You?

I am Fine ..What about you?

My Name is Rahul....And i am a Student Of Mehran University Of Engineering & Technology Jan

| | |
|---|---|
| Name | Enter name |
| Surname | Enter Surname |
| Email | Enter Email |
| Bldgrup | Enter BloodGroup |
| Identification | Enter your Identification |
| Cnic | Enter Your Cnic Number |
| Mobile Number | Enter Mobile Number |
| Country | Select Country |

**Task# 4:** Make a Zebra Crossing Table using internal CSS as shown below

| Id | Name | Subject |
|---|---|---|
| 1001 | Asghar | Physics |
| 1002 | Susheel | Economics |
| 1003 | Shahzad | Chemistry |
| 1004 | Rameez | Zoology |
| 1004 | Paras | Math |
| 1004 | Sonu | Botany |

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ____________________________

Subject Teacher: ____________________________

Date: ____________________________

|  |  |  |  |  |
|---|---|---|---|---|
| Department of:<br>Computer Systems Engineering | Subject |  |  |  |
|  | WEB ENGINEERING |  |  |  |
| Mehran University of Engineering &Technology<br>Jamshoro | Year | 3rd | Semester | 6th |
|  | Batch | 17CS | Duration | 03 Hours |

# PRACTICAL # 10

# TO EXECUTE THE SERVER-SIDE LANGUAGE (PHP)

## PHP

PHP is a powerful tool for making dynamic and interactive Web pages. PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

PHP is a general-purpose server-side scripting language originally designed for Web development to produce dynamic Web pages. It is one of the first developed server-side scripting languages to be embedded into an HTML source document rather than calling an external file to process data. The code is interpreted by a Web server with a PHP processor module which generates the resulting Web page. It also has evolved to include a command-line interface capability and can be used in standalone graphical applications. PHP can be deployed on most Web servers and also as a standalone shell on almost every operating system and platform free of charge. A competitor to Microsoft's Active Server Pages (ASP) server-side script engine and similar languages, PHP is installed on more than 20 million Web sites and 1 million Web servers.

PHP was originally created by Rasmus Lerdorf in 1995. The main implementation of PHP is now produced by The PHP Group and serves as the formal reference to the PHP language. PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.

While PHP originally stood for Personal Home Page, it is now said to stand for PHP: Hypertext Preprocessor, a recursive acronym.

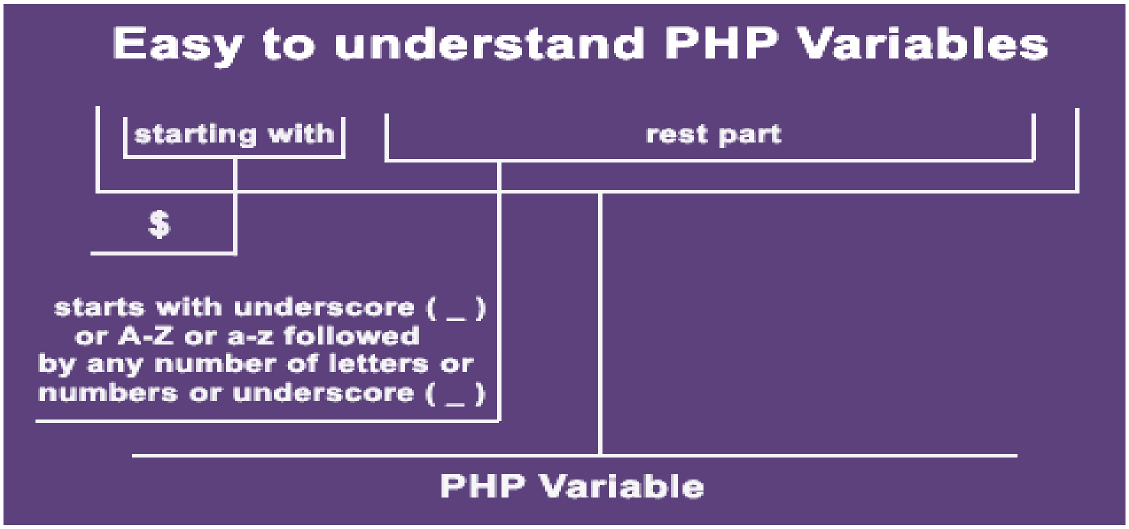
## PHP Variables

Variable is a symbol or name that stands for a value. Variables are used for storing values such as numeric values, characters, character strings, or memory addresses so that they can be used in any part of the program. PHP variables are used to hold values or expressions.

A variable can have a short name, like x, or a more descriptive name, like carName.

Rules for PHP variable names:

- Variables in PHP starts with a $ sign, followed by the name of the variable
- The variable name must begin with a letter or the underscore character
- A variable name can only contain alpha-numeric characters & underscores (A-z, 0-9, & _)
- A variable name should not contain spaces
- Variable names are case sensitive (y and Y are two different variables)


Easy to understand PHP Variables
starting with
rest part
$
starts with underscore ( _ ) or A-Z or a-z followed by any number of letters or numbers or underscore ( _ )
PHP Variable

**Valid and invalid PHP variables**

```
<?php
        $abc = 'Welcome';  //valid
        $Abc = 'W3resource.com'; //valid
        $9xyz = 'Hello world';  //invalid; starts with a number
        $_xyz = 'Hello world';  //valid; starts with an underscore
        $_9xyz = 'Hello world';  //valid
        $aäa = 'Hello world';  //valid; 'ä' is (Extended) ASCII 228.
?>
```

**PHP variable name is case-sensitive**

Consider the following example :

```
<?php
        $abc = 'Welcome';
        $ABC = 'Good Bye';
        echo "Value of abc : $abc";
        echo "Value of ABC : $ABC";
?>
```

*Output of the above example*

```
Value of abc : Welcome
Value of ABC : Good Bye
```

In the above example the different capitalization schemes make for different variables.

**PHP is a loosely type language**

In language such as C, C++ and Java the programmer must declare the name and type of the variable before use it. In PHP the type of the variable does not need to be declared before use it, because types are associated with values rather than variables. As a result a variable can change the type of its value as much as we want.

As previously mentioned you don't need to declare variables or their type before using them in PHP. In the following example none of the variables are declared before they are used, fact is $height is floating number and $width is an integer.

```
<?php
        $height = 3.5;
        $width = 4;
        $area=$height*$width;
        echo "Area of the rectangle is : $area";
?>
```

**PHP Variable Scope**
The scope of a variable is the portion of the script in which the variable can be referenced.
PHP has four different variable scopes:

- local
- global
- static
- parameter

**Local Scope**

A variable declared within a PHP function is local and can only be accessed within that function. (the variable has local scope):

```
<?php
$a = 5; // global scope
function myTest()
{
echo $a; // local scope
}
myTest();
?>
```

The script above will not produce any output because the echo statement refers to the local scope variable $a, which has not been assigned a value within this scope.

You can have local variables with the same name in different functions, because local variables are only recognized by the function in which they are declared.

Local variables are deleted as soon as the function is completed.

**Global Scope**

Global scope refers to any variable that is defined outside of any function.

Global variables can be accessed from any part of the script that is not inside a function.

To access a global variable from within a function, use the global keyword:

```
<?php
$a = 5;
$b = 10;
function myTest()
{
global $a, $b;
$b = $a + $b;
}
```

```
myTest();
echo $b;
?>
```

The script above will output 15.

PHP also stores all global variables in an array called $GLOBALS[index]. Its index is the name of the variable. This array is also accessible from within functions and can be used to update global variables directly.

The example above can be rewritten as this:

```
<?php
$a = 5;
$b = 10;
function myTest()
{
$GLOBALS['b'] = $GLOBALS['a'] + $GLOBALS['b'];
}
myTest();
echo $b;
?>
```

**Static Scope**
When a function is completed, all of its variables are normally deleted. However, sometimes you want a local variable to not be deleted.

To do this, use the static keyword when you first declare the variable:

```
static $rememberMe;
```

Then, each time the function is called, that variable will still have the information it contained from the last time the function was called.

Note: The variable is still local to the function.

**Parameters**
A parameter is a local variable whose value is passed to the function by the calling code.

Parameters are declared in a parameter list as part of the function declaration:

```
functionmyTest($para1,$para2,...)
{
// function code
}
```

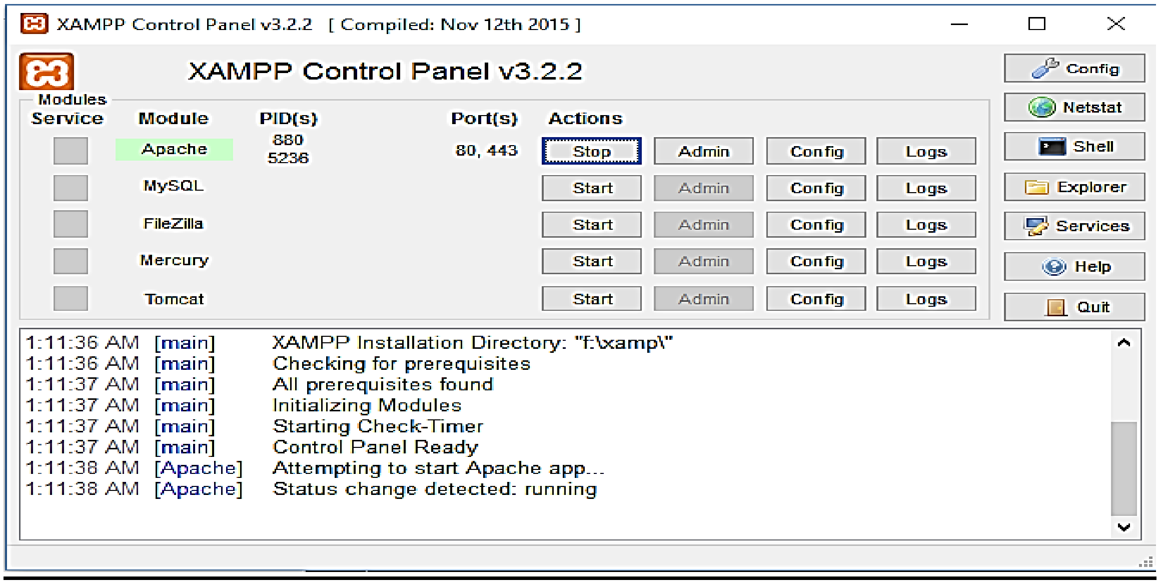
Parameters are also called arguments. We will discuss them in more detail when we talk about functions.

**Example:**

```
<?php
	echo "<h1>This is a Header </h1>";
	$date = date('D/M/Y');
	echo "<p>The Date today is $date";
?>
```

## Running PHP File:

- First Run XAMP and Start Apache Server

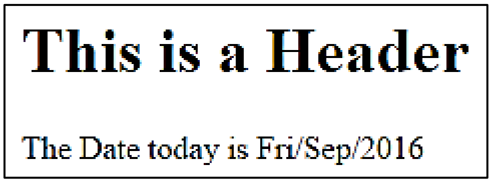

XAMPP Control Panel v3.2.2 [ Compiled: Nov 12th 2015 ]
XAMPP Control Panel v3.2.2
Config
Netstat
Shell
Explorer
Services
Help
Quit
Modules

| Service | Module | PID(s) | Port(s) | Actions | | | |
|---|---|---|---|---|---|---|---|
| □ | Apache | 880<br>5236 | 80, 443 | Stop | Admin | Config | Logs |
| □ | MySQL | | | Start | Admin | Config | Logs |
| □ | FileZilla | | | Start | Admin | Config | Logs |
| □ | Mercury | | | Start | Admin | Config | Logs |
| □ | Tomcat | | | Start | Admin | Config | Logs |

```
1:11:36 AM  [main]     XAMPP Installation Directory: "f:\xamp\"
1:11:36 AM  [main]     Checking for prerequisites
1:11:37 AM  [main]     All prerequisites found
1:11:37 AM  [main]     Initializing Modules
1:11:37 AM  [main]     Starting Check-Timer
1:11:37 AM  [main]     Control Panel Ready
1:11:38 AM  [Apache]   Attempting to start Apache app...
1:11:38 AM  [Apache]   Status change detected: running
```

- Open Browser and type localhost

## RESULT



# This is a Header

The Date today is Fri/Sep/2016

# LAB TASKS

**Task # 1:** Make A Simple Mark sheet Program using PHP Give Subject Name and Marks using Variables and echo it in HTML Table

**Task # 2:** Make validation form, in which student's data (Name, Roll #, Class & Section) mandatory to be filled using PHP.

**Task # 3:** Create Simple 3 Step Form using PHP. In First file take inputs using form, in second step show option for modify or submit form in third step show a successful submission and redirect it to first file.

**Task # 4:** Write a program to calculate and print the factorial of a number using for loop.

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ____________________________

Subject Teacher: ____________________________

Date: ____________________________

| Department of:<br>Computer Systems Engineering | | Subject | | | |
|---|---|---|---|---|---|
| | | WEB ENGINEERING | | | |
| Mehran University of Engineering<br>&Technology<br>Jamshoro | | Year | 3rd | Semester | 6th |
| | | Batch | 17CS | Duration | 03<br>Hours |

# PRACTICAL # 11

# TO MANAGE SERVER-SIDE CONDITIONAL CONTROL STRUCTURES

### Conditional Control Structures

Conditional statements are used to perform different actions based on different conditions. Very often when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this.

In PHP we have the following conditional statements:

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

### if Statement

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

**Syntax**

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

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

### if...else Statement

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

**Syntax**

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

### if...elseif .... else Statement

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

**Syntax**

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

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

**Syntax**

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

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

**PHP Loops**
Loops execute a block of code a specified number of times, or while a specified condition is true. Often when you write code, you want the same block of code to run over and over again in a row. Instead of adding several almost equal lines in a script we can use loops to perform a task like this.

In PHP, we have the following looping statements:

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

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

**Syntax**

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

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

**Syntax**

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

**The for Loop**

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

**Syntax**

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

**Parameters:**

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

***Note***: The init and increment parameters above can be empty or have multiple expressions (separated by commas).

# LAB TASKS

**Task # 1:** Make a PHP page which will generate a table of inputted number.

**Task # 2:** Make a PHP page which will show whether the inputted number is even / odd.

**Task # 3:** Create Simple server side validation form, which will ask username & password. If inputted username should be “Your Roll No” & and password should be “15CS” then it will show you greeting else generate an error.

**Task # 4:** Write a PHP script that creates the following table using for loops. Add cellpadding="3px" and cellspacing="0px" to the table tag.

| | | | | |
|---|---|---|---|---|
| 1 * 1 = 1 | 1 * 2 = 2 | 1 * 3 = 3 | 1 * 4 = 4 | 1 * 5 = 5 |
| 2 * 1 = 2 | 2 * 2 = 4 | 2 * 3 = 6 | 2 * 4 = 8 | 2 * 5 = 10 |
| 3 * 1 = 3 | 3 * 2 = 6 | 3 * 3 = 9 | 3 * 4 = 12 | 3 * 5 = 15 |
| 4 * 1 = 4 | 4 * 2 = 8 | 4 * 3 = 12 | 4 * 4 = 16 | 4 * 5 = 20 |
| 5 * 1 = 5 | 5 * 2 = 10 | 5 * 3 = 15 | 5 * 4 = 20 | 5 * 5 = 25 |
| 6 * 1 = 6 | 6 * 2 = 12 | 6 * 3 = 18 | 6 * 4 = 24 | 6 * 5 = 30 |

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ____________________________

Subject Teacher: ____________________________

Date: ____________________________

| | | | | | |
|---|---|---|---|---|---|
| Department of:<br>Computer Systems Engineering | | Subject | | | |
| | | WEB ENGINEERING | | | |
| Mehran University of Engineering<br>&Technology<br>Jamshoro | | Year | 3rd | Semester | 6th |
| | | Batch | 17CS | Duration | 03 Hours |

# PRACTICAL # 12

# TO DESIGN THE DATABASES WITH THE HELP OF PHP

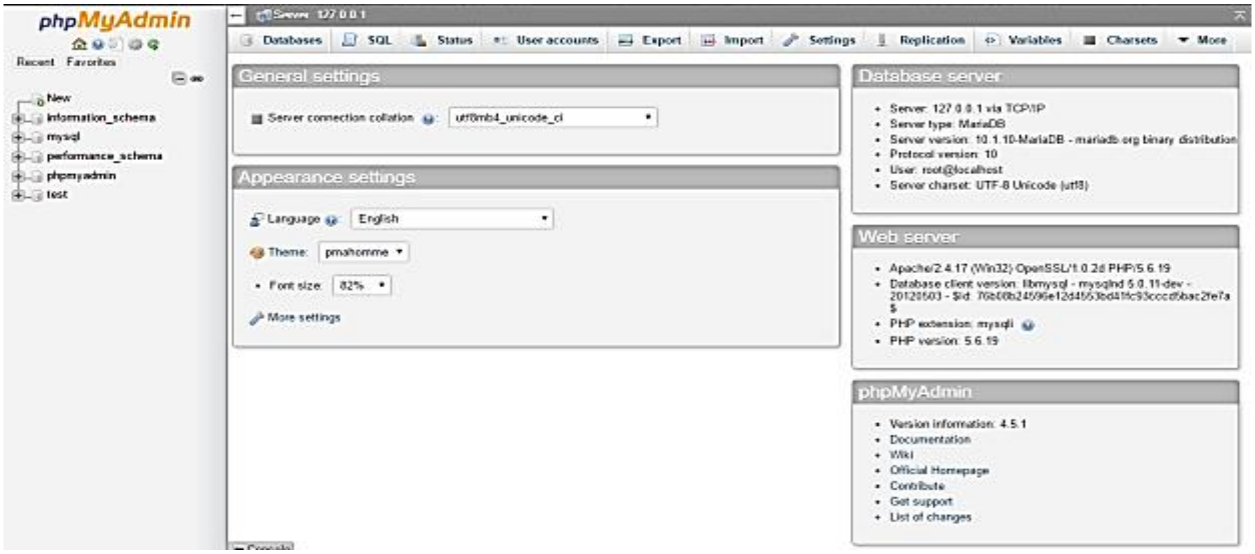
## Database in PHP

PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database.

## Creating Database

- STEP1:
  Open Web Browser and navigate to following URL

  https://localhost/phpmyadmin



Server connection collation: utf8mb4_unicode_ci

Language: English

Theme: pmahomme

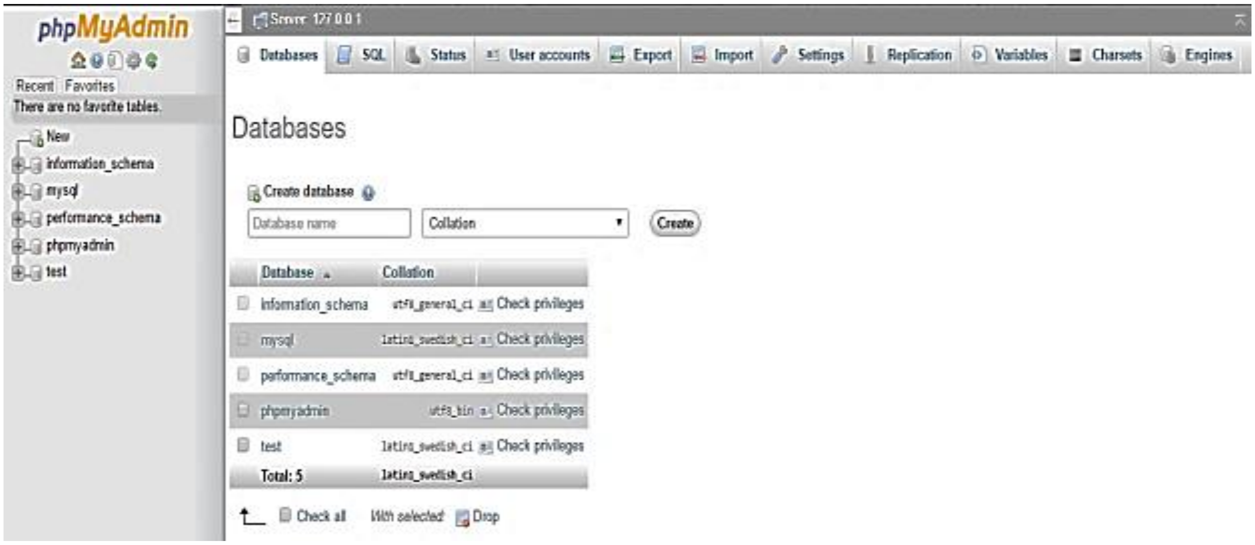
Font size: 82%

- Server: 127.0.0.1 via TCP/IP
- Server type: MariaDB
- Server version: 10.1.10-MariaDB - mariadb.org binary distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

- Apache/2.4.17 (Win32) OpenSSL/1.0.2d PHP/5.6.19
- Database client version: libmysql - mysqlnd 5.0.11-dev - 20120503 - $Id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a $
- PHP extension: mysqli
- PHP version: 5.6.19

- Version information: 4.5.1

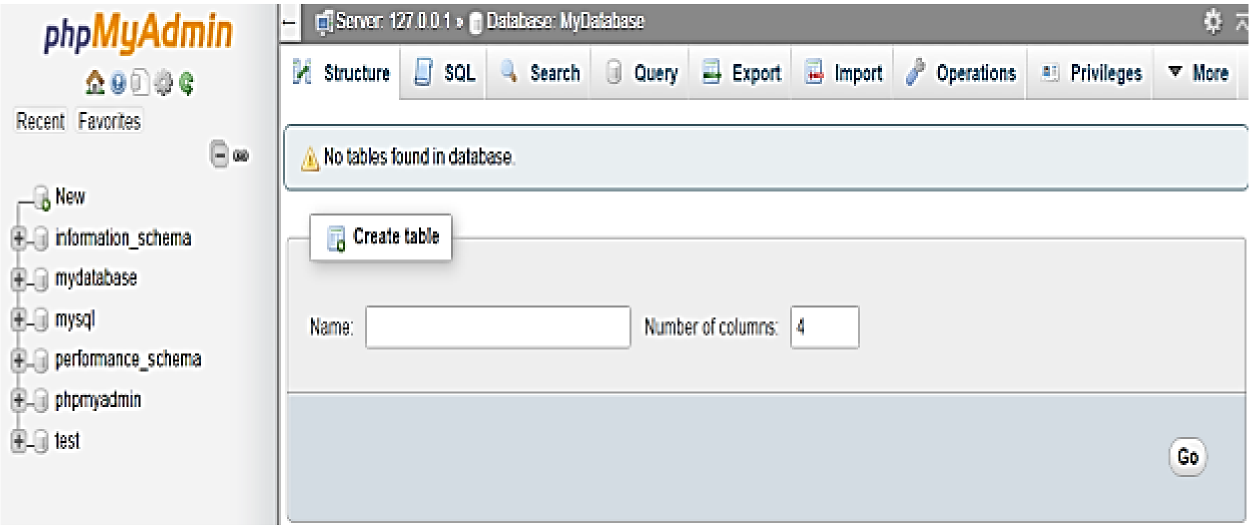
Step2: Click On +New Button and enter Database name and click Create



Databases

| Database | Collation | |
|---|---|---|
| information_schema | utf8_general_ci | Check privileges |
| mysql | latin1_swedish_ci | Check privileges |
| performance_schema | utf8_general_ci | Check privileges |
| phpmyadmin | utf8_bin | Check privileges |
| test | latin1_swedish_ci | Check privileges |
| Total: 5 | latin1_swedish_ci | |

Step3: Create Table in Database by giving Table name,Rows and clicking Go Button



No tables found in database.

Name:

Number of columns: 4

Step4: Create Columns in Table by giving column and data type of column and save

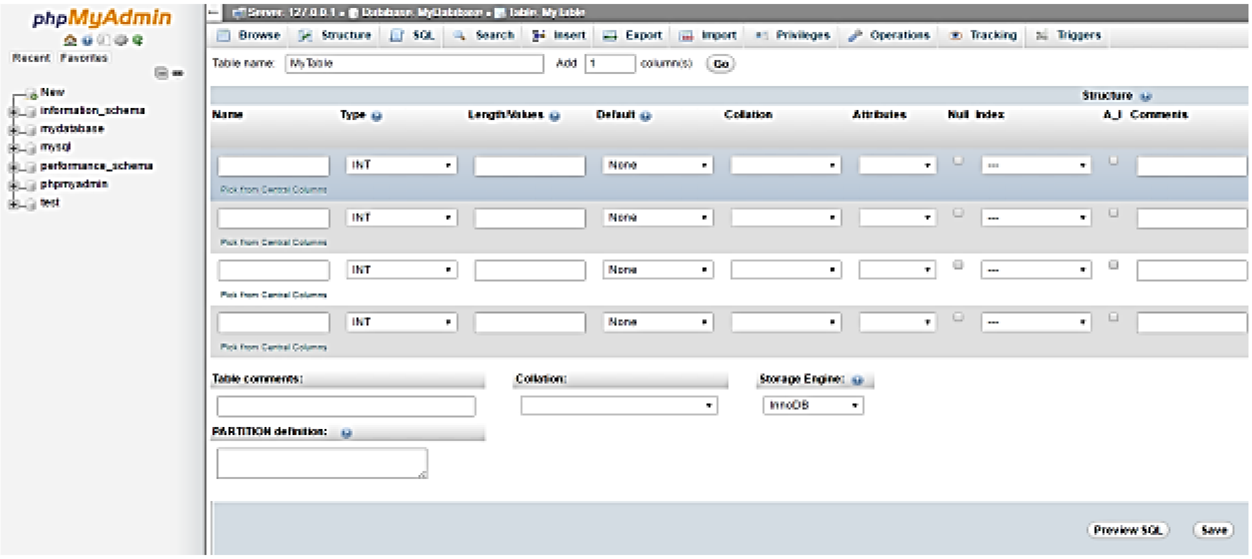


Table name: MyTable Add 1 column(s)

| Name | Type | Length/Values | Default | Collation | Attributes | Null | Index | A_I | Comments |
|---|---|---|---|---|---|---|---|---|---|
| | INT | | None | | | □ | --- | □ | |
| | INT | | None | | | □ | --- | □ | |
| | INT | | None | | | □ | --- | □ | |
| | INT | | None | | | □ | --- | □ | |

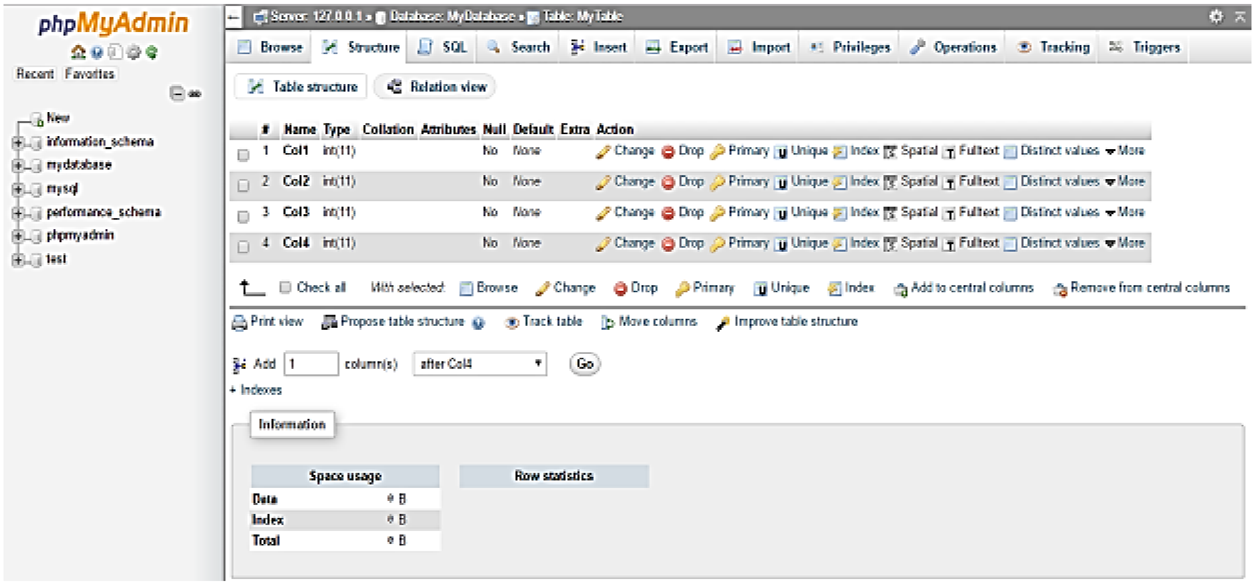
Table comments:

Collation:

Storage Engine: InnoDB

PARTITION definition:

Final View of Table is as shown below



|  | # | Name | Type | Collation | Attributes | Null | Default | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | Col1 | int(11) |  |  | No | None |  | Change Drop Primary Unique Index Spatial Fulltext Distinct values More |
| □ | 2 | Col2 | int(11) |  |  | No | None |  | Change Drop Primary Unique Index Spatial Fulltext Distinct values More |
| □ | 3 | Col3 | int(11) |  |  | No | None |  | Change Drop Primary Unique Index Spatial Fulltext Distinct values More |
| □ | 4 | Col4 | int(11) |  |  | No | None |  | Change Drop Primary Unique Index Spatial Fulltext Distinct values More |

| Space usage |  |
|---|---|
| Data | 0 B |
| Index | 0 B |
| Total | 0 B |

### Connecting to Database

- PHP provides **mysql_connect**function to open a database connection. This function takes five parameters and returns a MySQL link identifier on success , or FALSE on failure

Code for creating database connection in PHP is given below;

```
<?php
$host = "localhost";
$user = "root";
$pass = "";
$db = "mydatabase";
$con = mysqli_connect($host,$user,$pass,$db)
?>
```

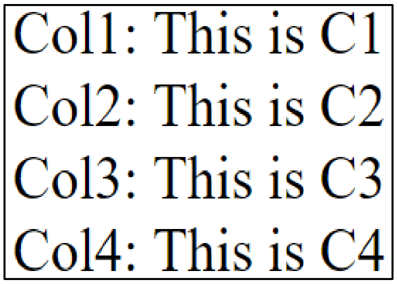
### Running SQL Query in PHP:

- Running SQL Query in PHP using mysqli_query(connection,query)

```
<?php
$query = "SELECT * FROM My Table";
$res = mysqli_query($con,$query);
?>
```

### Example: Fetching data from Database.

```
<?php
$host = "localhost";
$user = "root";
$pass = "";
$db = "mydatabase";
$con = mysqli_connect($host,$user,$pass,$db);
$query = "SELECT * FROM mytable";
$res = mysqli_query($con,$query);
$output =  mysqli_fetch_row($res);
echo "Col1: $output[0] </br> Col2: $output[1] </br>
          Col3: $output[2]</br>Col4: $output[3]";
?>
```



```
Col1: This is C1
Col2: This is C2
Col3: This is C3
Col4: This is C4
```

# LAB TASKS

**Task # 1:** Create a database for registration form, which takes Name, Roll Number, Section via PHP file and store form data in Database using PHP.

**Task # 2:** Create a webpage which shows name, marks of subjects and grade from already stored database.

**Task # 3:** Create a webpage which will take Student Roll Number in a search box and display the student data accordingly: Name, Roll Number and Section from database.

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ____________________________

Subject Teacher: ____________________________

Date: ____________________________

| | | | | | |
|---|---|---|---|---|---|
| Department of:<br>Computer Systems Engineering | | Subject | | | |
| | | WEB ENGINEERING | | | |
| Mehran University of Engineering<br>&Technology<br>Jamshoro | | Year | 3rd | Semester | 6th |
| | | Batch | 17CS | Duration | 03 Hours |

# PRACTICAL # 13

# TO DEVELOP THE LIVE WEBSITE (WEB HOSTING)

### Web hosting:

- The total costs of setting up your own in-house Web commerce site are expensive.
- Hosting services allow businesses to start electronic commerce inexpensively.
- Web hosting services provide all the services that an ISP does
  Your Web or commerce site resides on the same server as several other sites.
- It is inexpensive, requires very little of an online store's time to maintain
- Security concerns arise from unrelated online businesses sharing the same server.

### Usage limitations:

- Most hosting plans limit how much data you can transfer to and from their servers in a given month. Once you reach 500MB or 1GB of data transfer, most companies charge you by the megabyte for any additional traffic

### What should you look for in a host?

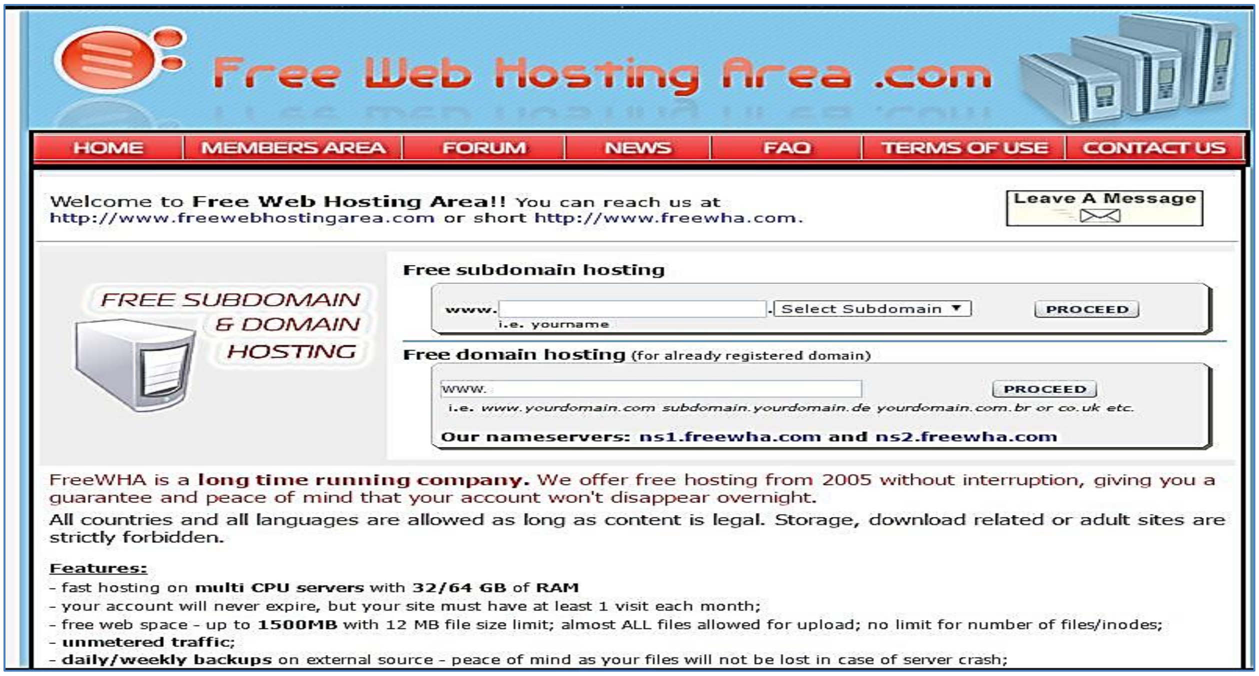
- Script and extension support
- SQL databases
- CGI scripts
- FrontPage 2000 extensions
- PHP
- Active Server Pages.

### Publishing Your First Website

Go to any Free Domain Hosting Site, e.g;

- 1) www.freewebhostingarea.com
- 2) www.000webhosts.com

### FreeWebHostingArea.com Home Page



Welcome to **Free Web Hosting Area!!** You can reach us at http://www.freewebhostingarea.com or short http://www.freewha.com.

**Free subdomain hosting**

www.________ . Select Subdomain ▼

i.e. yourname

**Free domain hosting** (for already registered domain)

www.

i.e. *www.yourdomain.com subdomain.yourdomain.de yourdomain.com.br or co.uk etc.*

**Our nameservers: ns1.freewha.com and ns2.freewha.com**

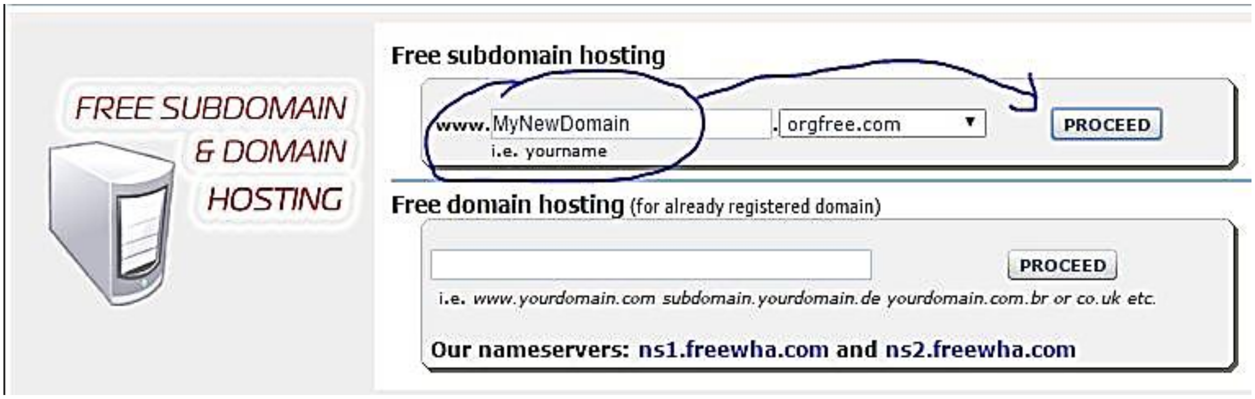
FreeWHA is a **long time running company.** We offer free hosting from 2005 without interruption, giving you a guarantee and peace of mind that your account won't disappear overnight.

All countries and all languages are allowed as long as content is legal. Storage, download related or adult sites are strictly forbidden.

**Features:**

- fast hosting on **multi CPU servers** with **32/64 GB** of **RAM**
- your account will never expire, but your site must have at least 1 visit each month;
- free web space - up to **1500MB** with 12 MB file size limit; almost ALL files allowed for upload; no limit for number of files/inodes;
- **unmetered traffic;**
- **daily/weekly backups** on external source - peace of mind as your files will not be lost in case of server crash;

- **Enter Your Website Name and Choose Domain**



**Free subdomain hosting**

www.MyNewDomain . orgfree.com ▼

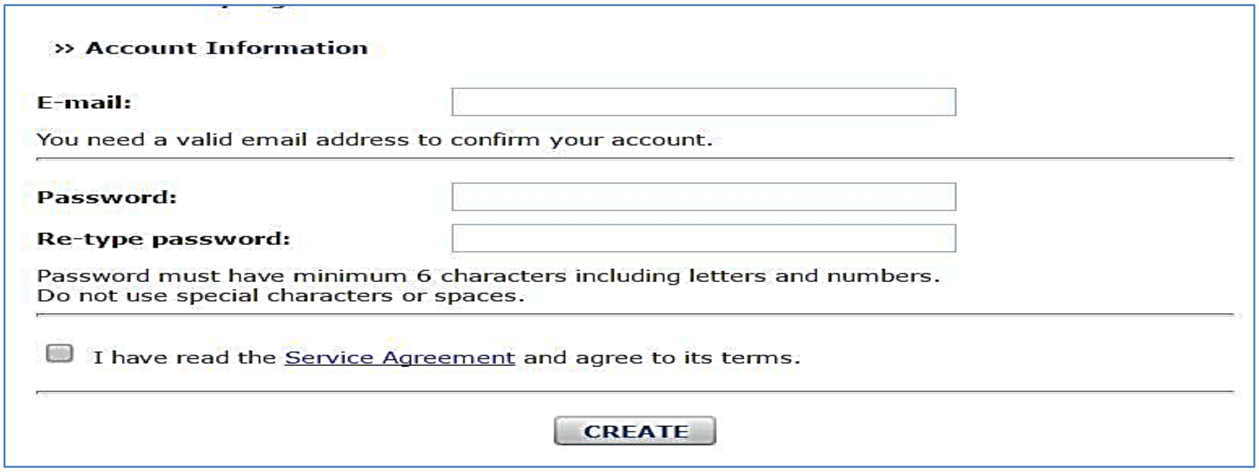
i.e. yourname

**Free domain hosting** (for already registered domain)

i.e. *www.yourdomain.com subdomain.yourdomain.de yourdomain.com.br or co.uk etc.*

**Our nameservers: ns1.freewha.com and ns2.freewha.com**

- **Create Your Account**



**>> Account Information**

**E-mail:**

You need a valid email address to confirm your account.

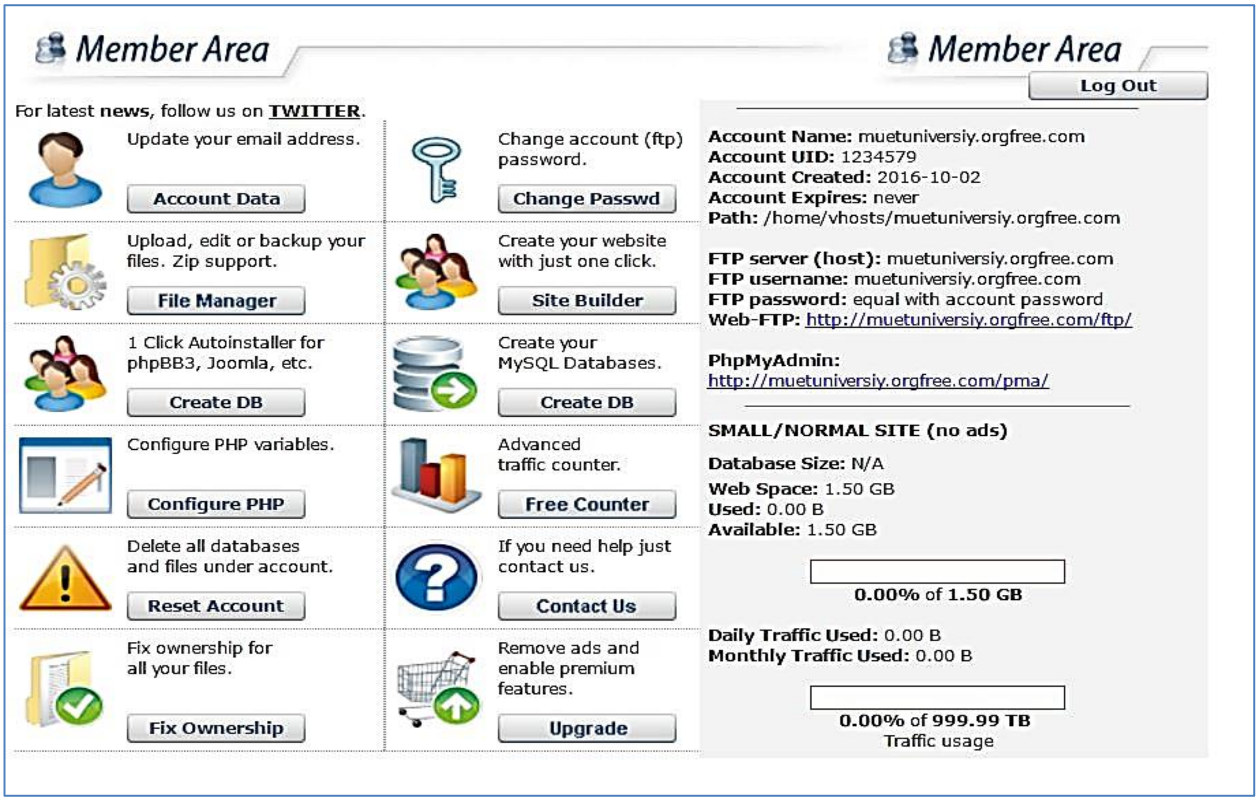
**Password:**

**Re-type password:**

Password must have minimum 6 characters including letters and numbers.
Do not use special characters or spaces.

□ I have read the Service Agreement and agree to its terms.

- **Go to Members Area & Login to your account**



For latest **news**, follow us on **TWITTER**.

Update your email address.

Change account (ftp) password.

Upload, edit or backup your files. Zip support.

Create your website with just one click.

1 Click Autoinstaller for phpBB3, Joomla, etc.

Create your MySQL Databases.

Configure PHP variables.

Advanced traffic counter.

Delete all databases and files under account.

If you need help just contact us.

Fix ownership for all your files.

Remove ads and enable premium features.

**Account Name:** muetuniversiy.orgfree.com
**Account UID:** 1234579
**Account Created:** 2016-10-02
**Account Expires:** never
**Path:** /home/vhosts/muetuniversiy.orgfree.com

**FTP server (host):** muetuniversiy.orgfree.com
**FTP username:** muetuniversiy.orgfree.com
**FTP password:** equal with account password
**Web-FTP:** http://muetuniversiy.orgfree.com/ftp/

**PhpMyAdmin:**
http://muetuniversiy.orgfree.com/pma/

**SMALL/NORMAL SITE (no ads)**

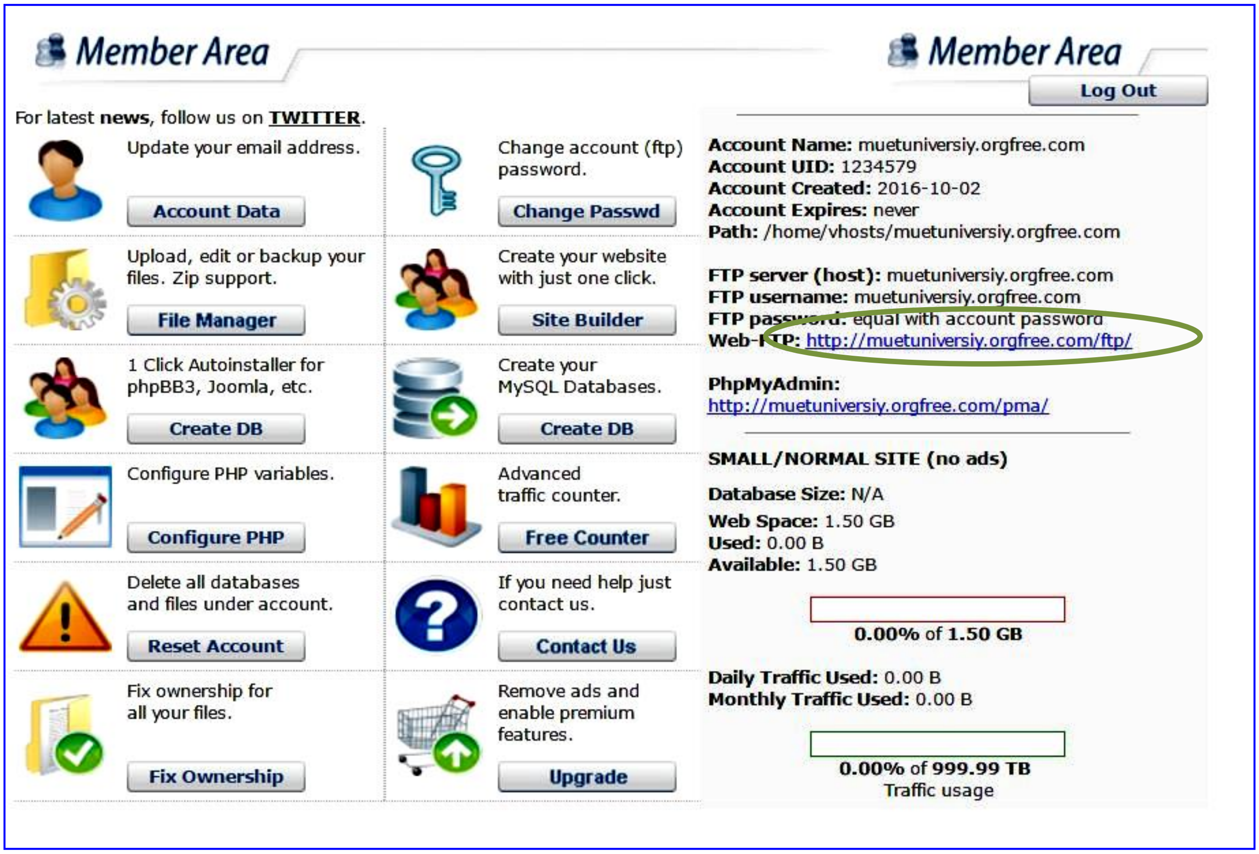
**Database Size:** N/A
**Web Space:** 1.50 GB
**Used:** 0.00 B
**Available:** 1.50 GB

**0.00%** of **1.50 GB**

**Daily Traffic Used:** 0.00 B
**Monthly Traffic Used:** 0.00 B

**0.00%** of **999.99 TB**
Traffic usage

- **Now publish your site on Web FTP Server**



For latest **news**, follow us on **TWITTER**.

Update your email address.

Change account (ftp) password.

Upload, edit or backup your files. Zip support.

Create your website with just one click.

1 Click Autoinstaller for phpBB3, Joomla, etc.

Create your MySQL Databases.

Configure PHP variables.

Advanced traffic counter.

Delete all databases and files under account.

If you need help just contact us.

Fix ownership for all your files.

Remove ads and enable premium features.

**Account Name:** muetuniversiy.orgfree.com
**Account UID:** 1234579
**Account Created:** 2016-10-02
**Account Expires:** never
**Path:** /home/vhosts/muetuniversiy.orgfree.com

**FTP server (host):** muetuniversiy.orgfree.com
**FTP username:** muetuniversiy.orgfree.com
**FTP password:** equal with account password
**Web-FTP:** http://muetuniversiy.orgfree.com/ftp/

**PhpMyAdmin:**
http://muetuniversiy.orgfree.com/pma/

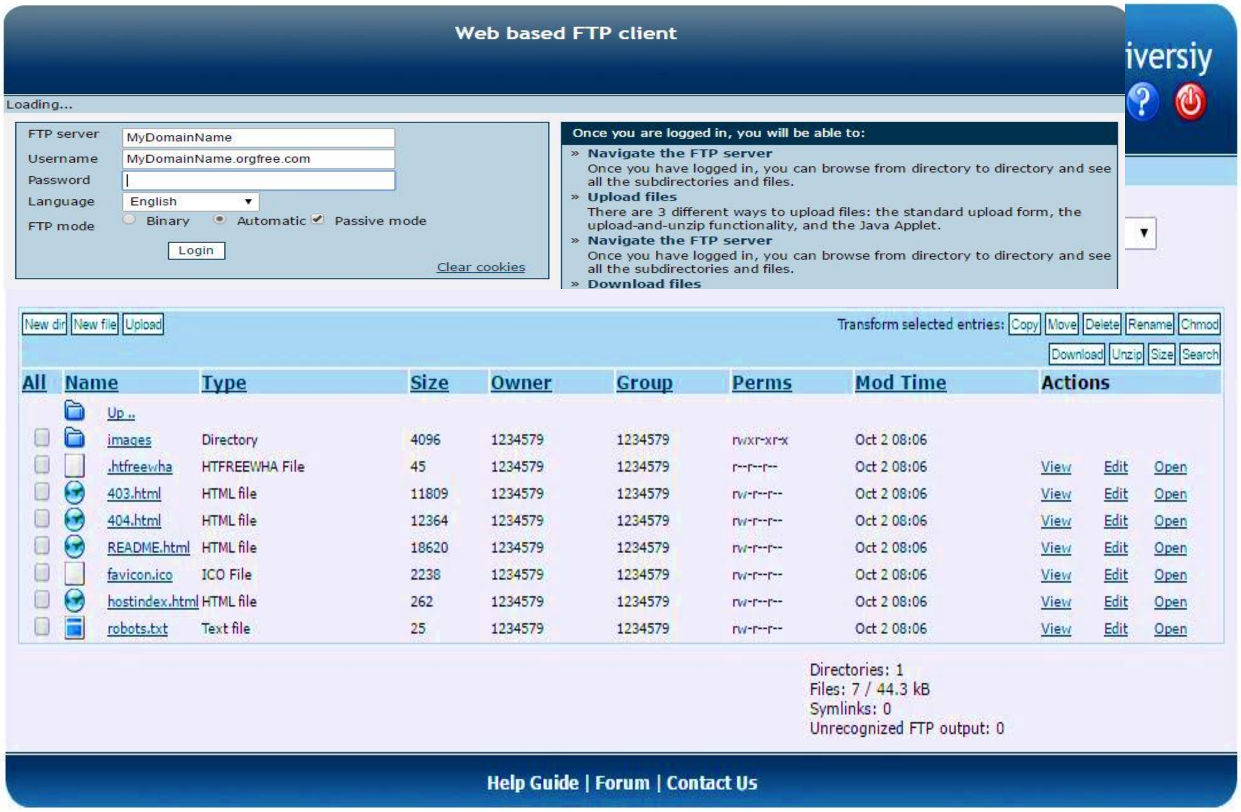
**SMALL/NORMAL SITE (no ads)**

**Database Size:** N/A
**Web Space:** 1.50 GB
**Used:** 0.00 B
**Available:** 1.50 GB

**0.00%** of **1.50 GB**

**Daily Traffic Used:** 0.00 B
**Monthly Traffic Used:** 0.00 B

**0.00%** of **999.99 TB**
Traffic usage



FTP server: MyDomainName
Username: MyDomainName.orgfree.com
Password:
Language: English
FTP mode: ○ Binary ● Automatic ☑ Passive mode

Once you are logged in, you will be able to:

» **Navigate the FTP server**
Once you have logged in, you can browse from directory to directory and see all the subdirectories and files.

» **Upload files**
There are 3 different ways to upload files: the standard upload form, the upload-and-unzip functionality, and the Java Applet.

» **Navigate the FTP server**
Once you have logged in, you can browse from directory to directory and see all the subdirectories and files.

» **Download files**

Transform selected entries:

| All | Name | Type | Size | Owner | Group | Perms | Mod Time | Actions | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Up .. | | | | | | | | | |
| □ | images | Directory | 4096 | 1234579 | 1234579 | rwxr-xr-x | Oct 2 08:06 | | | |
| □ | .htfreewha | HTFREEWHA File | 45 | 1234579 | 1234579 | r--r--r-- | Oct 2 08:06 | View | Edit | Open |
| □ | 403.html | HTML file | 11809 | 1234579 | 1234579 | rw-r--r-- | Oct 2 08:06 | View | Edit | Open |
| □ | 404.html | HTML file | 12364 | 1234579 | 1234579 | rw-r--r-- | Oct 2 08:06 | View | Edit | Open |
| □ | README.html | HTML file | 18620 | 1234579 | 1234579 | rw-r--r-- | Oct 2 08:06 | View | Edit | Open |
| □ | favicon.ico | ICO File | 2238 | 1234579 | 1234579 | rw-r--r-- | Oct 2 08:06 | View | Edit | Open |
| □ | hostindex.html | HTML file | 262 | 1234579 | 1234579 | rw-r--r-- | Oct 2 08:06 | View | Edit | Open |
| □ | robots.txt | Text file | 25 | 1234579 | 1234579 | rw-r--r-- | Oct 2 08:06 | View | Edit | Open |

Directories: 1
Files: 7 / 44.3 kB
Symlinks: 0
Unrecognized FTP output: 0

## LAB TASK

**Task # 1:** Create your account on any free web server & place at least 3pages website on it. Also place snapshots of your member's area control panel & your website. Write link of your website as well.

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ______________________________

Subject Teacher: ______________________________

Date: ______________________________

# PRACTICAL # 14

# TO PROCEED WITH THE VARIOUS TOOLS OF CMS – JOOMLA

## CMS

A content management system (CMS) is a computer application that supports the creation and modification of digital content. It is often used to support multiple users working in a collaborative environment.

CMS features vary widely. Most CMSs include Web-based publishing, format management, history editing and version control, indexing, search, and retrieval. By their nature, content management systems support the separation of content and presentation.

A web content management system (WCM or WCMS) is a CMS designed to support the management of the content of Web pages. Most popular CMSs are also WCMSs. Web content includes text and embedded graphics, photos, video, audio, maps, and program code (e.g., for applications) that displays content or interacts with the user.

Content Management System (CMS) typically has two major components:

- A content management application (CMA) is the front-end user interface that allows a user, even with limited expertise, to add, modify, and remove content from a website without the intervention of a webmaster.
- A content delivery application (CDA) compiles that information and updates the website.

## Features:

- SEO-friendly URLs
- Integrated and online help
- Modularity and extensibility
- User and group functionality
- Templating support for changing designs
- Install and upgrade wizards
- Integrated audit logs
- Compliance with various accessibility frameworks and standards, such as WAI-ARIA

## Advantages:

- Reduced need to code from scratch
- The ability to create a unified look and feel
- Version control

- Edit permission management

## Joomla!

Joomla! is one of the world's most popular software packages used to build, organize, manage and publish content for websites, blogs, Intranets and mobile applications. Owing to its scalable MVC architecture its also a great base to build web applications.

## Features for User

### Multilingual

Joomla! is the most popular and widely supported open source multilingual CMS platform in the world, offering over 65 languages.

### Well Supported

Worldwide, enthusiastic community is filled with individuals, and teams of world class developers and business consultants who actively help at no cost in the forums.

There are thousands of professional Joomla! service providers throughout the world who can help build, maintain and market Joomla! projects.

### Easy Updates

One of the big challenges with any software is keeping it up to date. Fortunately, Joomla! has a "One Click Version Update" feature to make this process super easy for users of any skill level.

### Integrated Help System

Joomla! has an in-app contextual help system to help every level of user to operate their Joomla. Most pages have a help button in the top right, helping you fully understand all options on that page.

### Media Manager

The Media Manager is the tool for easily uploading, organizing and managing your media files and folders. You can even handle more types of files, thanks to the configurable MIME settings. The Media Manager is integrated into the Article Editor so you can access images and all other media files for easy usage and enhancement of your written content.

### Banner Management

With the banner manager you have the possibility to easily add advertising and monetize your website. The tool allows you to create clients and campaigns, to add as many banners as you need, even custom codes, to set impression numbers, track the clicks and much more...

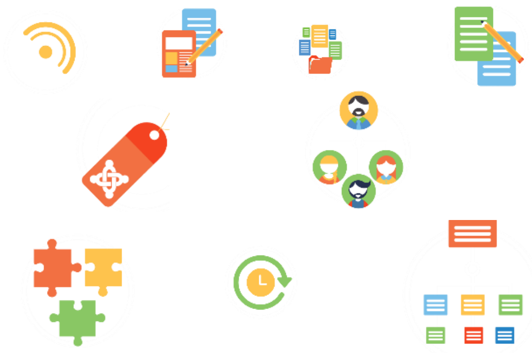
### Contact Management

The contacts component allows you to add several contacts, departments and categories, and extend the basic contact information with miscellaneous information and an image. Easily set up a contact form for each contact you create and allow access to the public or just to some registered users, or create a listing of these contacts.

### Search better, Search Smarter

With the built in search and smart search, your website visitors will be able to quickly and easily find the appropriate information on your site. And even more, thanks to the statistics you can analyze your visitors needs and streamline your content even better to serve them. You have the ability to use the included smart indexing, advanced search options, auto suggest searches - making Joomla! search the best in class right out of the box.

- Content Management
- Nested categorization
- Tagging
- Frontend Editing
- Content Versioning
- Syndication and Newsfeed Management
- Menu Manager
- Powerful Extensibility
- Extensive ACL for all your access control needs



## Features for Designers

### Design Uncoupled

Joomla! was one of the pioneers in open source CMS's by adopting a MVC design strategy. MVC means that views are strictly separate from the business logic. This is a huge advantage since you own the views or can override them to achieve superior custom designs. Joomla! not only gives you the design freedom that you have always craved but can help you make your sites stand out in the crowd!

### Responsive with Bootstrap

Joomla! is Mobile Ready and allows you to build more than just websites but online applications that can respond to virtually any device. Joomla! core templates are built with Bootstrap making it responsive out of the box. Which means you have a toolset to work with which makes creating templates even easier!

### Do More with Less

Spend less time coding and reduce the tedious tasks associated with building interfaces in Joomla! 3. Joomla! now features LESS CSS and jQuery which means you can write less code to achieve greater results. In addition the Icomoon font icon library provides a wealth of retina-optimized icons. The Joomla! User Interface (JUI) library gives you a standardized backend & frontend interface.

### Override Away!

With a highly advanced override system, designers get an awesome amount of power over how pages & elements of pages are presented without touching any of the core code! Practically any HTML generated by Joomla! can be customized to your project.

### Beautiful Fonts for that extra edge

Designers know the power of fonts for expressing ideas and design strategies. With Joomla! you do not need to get constrained by standard Web fonts. The Joomla! core itself opens a whole new world of expression because it allows you the freedom to use Google fonts to make that new design dream come true !

### Template Management

Templates in Joomla! are more than a framework for managing your designs but a powerful tool suite for making your site look exactly the way you want. You have complete control of your presentation since you can either use a single template for the entire site or a separate template for each site section or menu item. The level of visual control goes a step further with powerful template overrides, allowing you to customize each part of your pages.

## Features for Developers

### User Management

Joomla! has a registration system that allows users to configure personal options. Out of the box, there are nine user groups with various types of permissions on what users are allowed to access, edit, publish and administer.

### System Features

Speedy page loads are possible with page caching, granular-level module caching, and GZIP page compression.

### Microdata library implementation

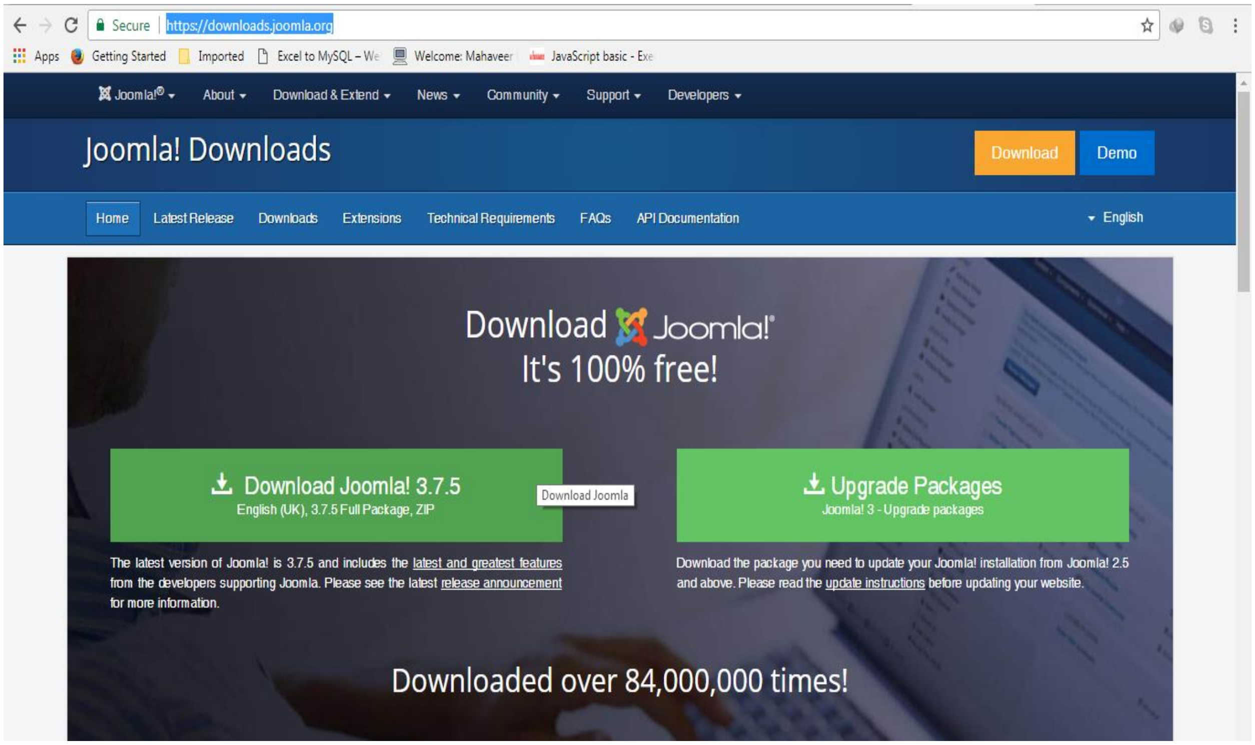
Developers will now be able to incorporate microdata more easily into their extensions and sites. From automating the Author tag in articles, to generating detailed markup for directories of information, the microdata library will significantly enhance how you can optimise SEO with Joomla.

### PHP 7 Support

As of Joomla! 3.5.0, Joomla! supports the most recent release of Zend's PHP: PHP 7, which provides a significant performance boost to your website. Joomla! users can enjoy the benefit of that performance improvement. Joomla! still provides legacy PHP support going back to PHP version 5.3.10 for web servers that have not yet upgraded.

## Installing Joomla

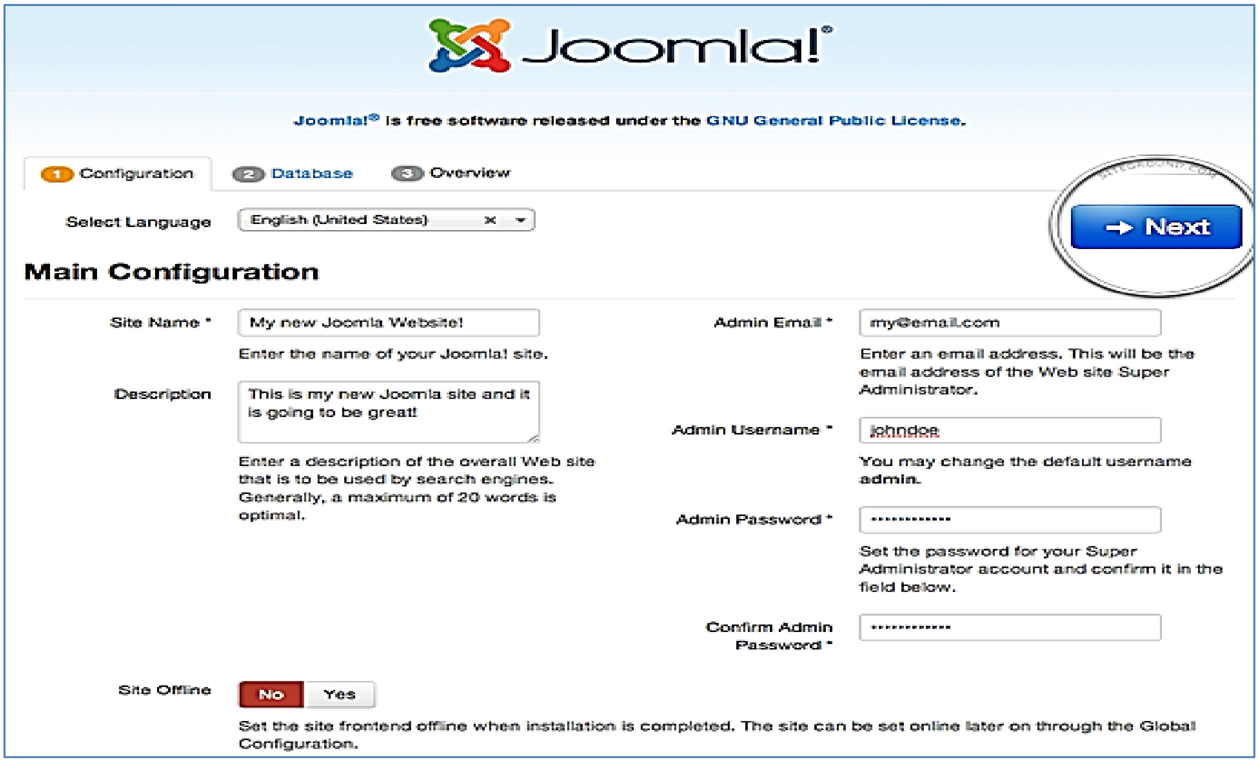
Download Joomla from: https://downloads.joomla.org/



The latest version of Joomla! is 3.7.5 and includes the latest and greatest features from the developers supporting Joomla. Please see the latest release announcement for more information.

Download the package you need to update your Joomla! installation from Joomla! 2.5 and above. Please read the update instructions before updating your website.

Downloaded over 84,000,000 times!



Joomla!® is free software released under the GNU General Public License.

Select Language: English (United States)

## Main Configuration

Site Name *: My new Joomla Website!

Enter the name of your Joomla! site.

Description: This is my new Joomla site and it is going to be great!

Enter a description of the overall Web site that is to be used by search engines. Generally, a maximum of 20 words is optimal.

Admin Email *: my@email.com

Enter an email address. This will be the email address of the Web site Super Administrator.

Admin Username *: johndoe

You may change the default username admin.

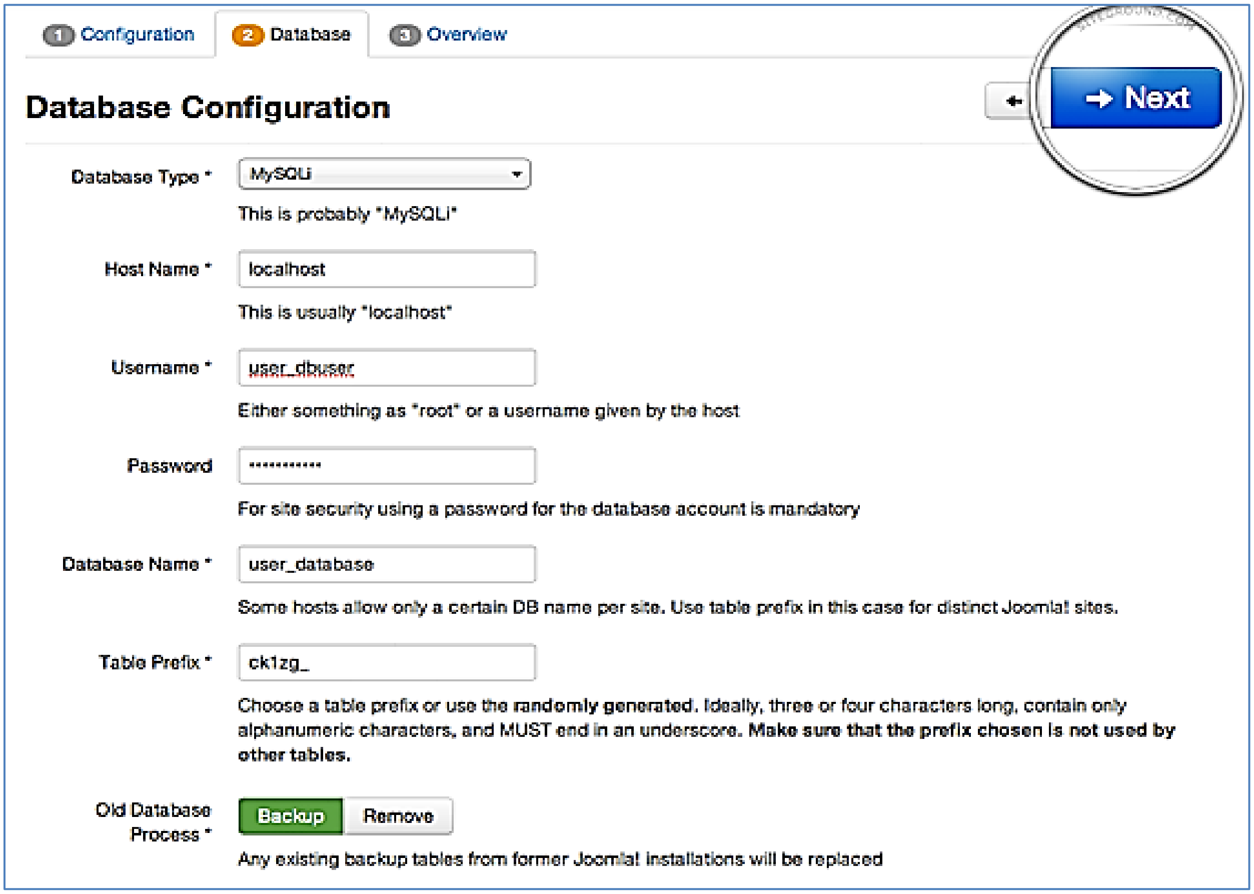
Admin Password *: ••••••••••••

Set the password for your Super Administrator account and confirm it in the field below.

Confirm Admin Password *: ••••••••••••

Site Offline: [✓ No] Yes

Set the site frontend offline when installation is completed. The site can be set online later on through the Global Configuration.



## Database Configuration

Database Type *: MySQLi

This is probably "MySQLi"

Host Name *: localhost

This is usually "localhost"

Username *: user_dbuser

Either something as "root" or a username given by the host

Password: ••••••••••••

For site security using a password for the database account is mandatory

Database Name *: user_database

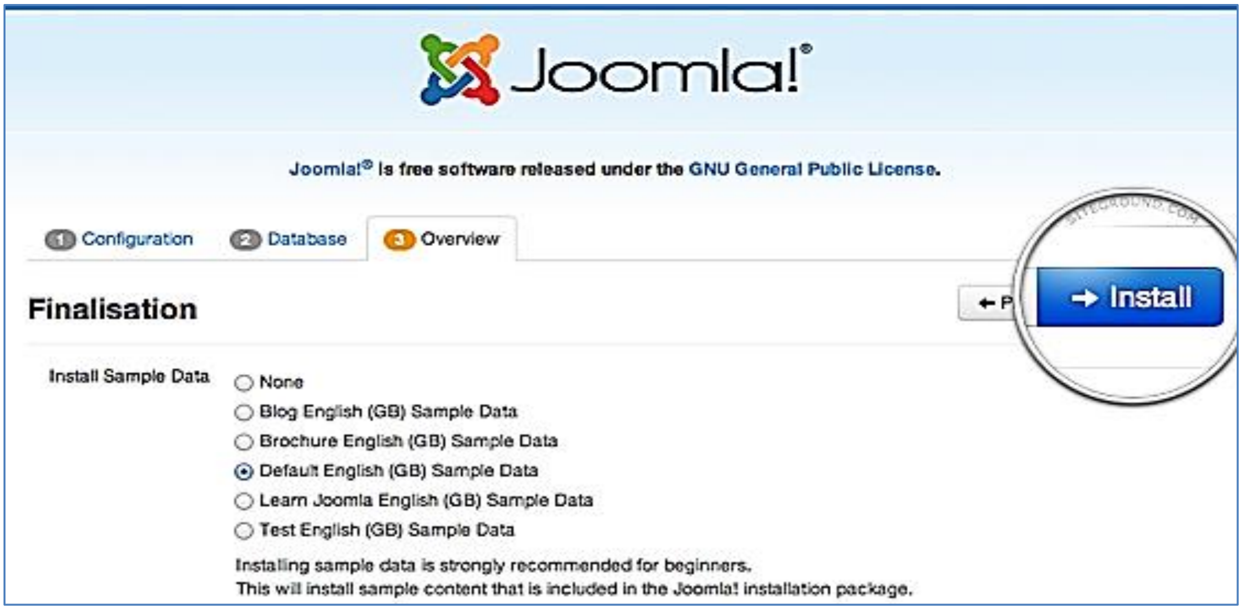
Some hosts allow only a certain DB name per site. Use table prefix in this case for distinct Joomla! sites.

Table Prefix *: ck1zg_

Choose a table prefix or use the randomly generated. Ideally, three or four characters long, contain only alphanumeric characters, and MUST end in an underscore. Make sure that the prefix chosen is not used by other tables.

Old Database Process *: [✓ Backup] Remove

Any existing backup tables from former Joomla! installations will be replaced



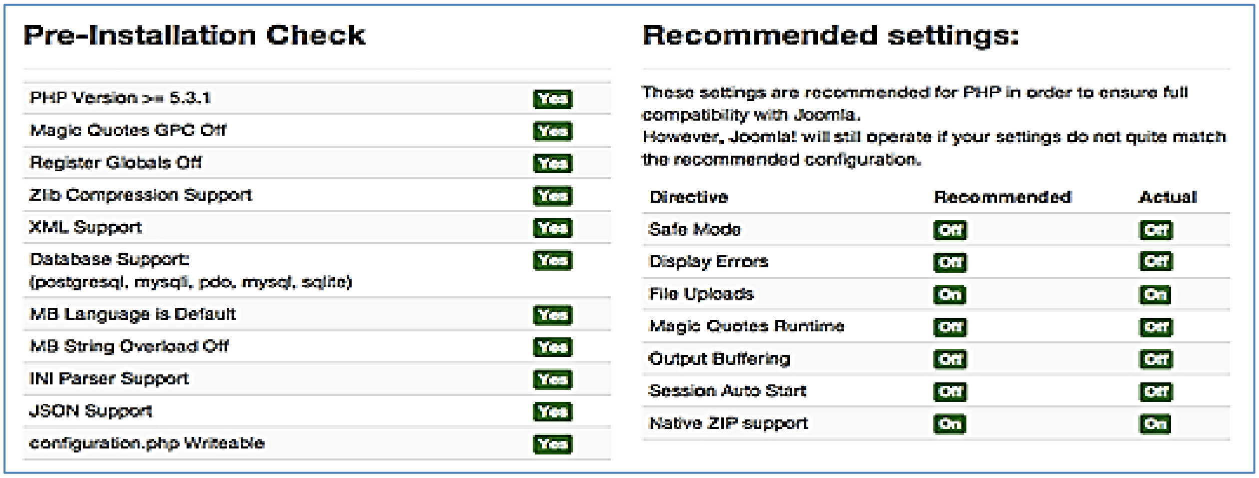
Joomla!® is free software released under the GNU General Public License.

### Finalisation

Install Sample Data

○ None
○ Blog English (GB) Sample Data
○ Brochure English (GB) Sample Data
● Default English (GB) Sample Data
○ Learn Joomla English (GB) Sample Data
○ Test English (GB) Sample Data

Installing sample data is strongly recommended for beginners.
This will install sample content that is included in the Joomla! installation package.



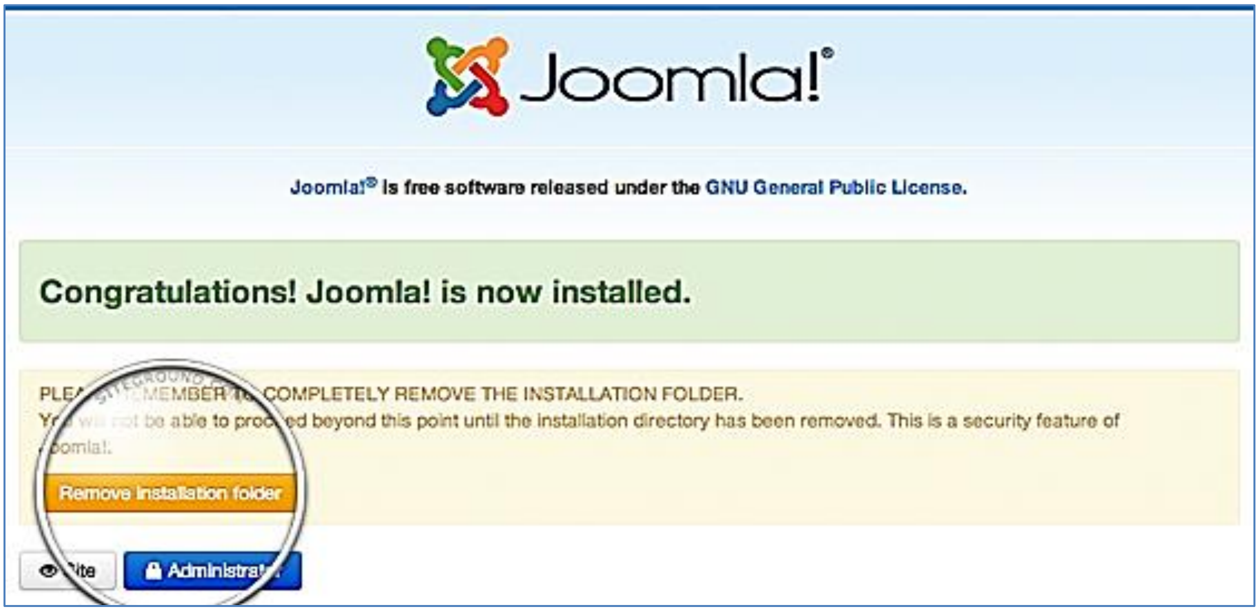
### Pre-Installation Check

| | |
|---|---|
| PHP Version >= 5.3.1 | Yes |
| Magic Quotes GPC Off | Yes |
| Register Globals Off | Yes |
| Zlib Compression Support | Yes |
| XML Support | Yes |
| Database Support:<br>(postgresql, mysqli, pdo, mysql, sqlite) | Yes |
| MB Language is Default | Yes |
| MB String Overload Off | Yes |
| INI Parser Support | Yes |
| JSON Support | Yes |
| configuration.php Writeable | Yes |

### Recommended settings:

These settings are recommended for PHP in order to ensure full compatibility with Joomla.
However, Joomla! will still operate if your settings do not quite match the recommended configuration.

| Directive | Recommended | Actual |
|---|---|---|
| Safe Mode | Off | Off |
| Display Errors | Off | Off |
| File Uploads | On | On |
| Magic Quotes Runtime | Off | Off |
| Output Buffering | Off | Off |
| Session Auto Start | Off | Off |
| Native ZIP support | On | On |



Joomla!® is free software released under the GNU General Public License.

**Congratulations! Joomla! is now installed.**

PLEASE REMEMBER TO COMPLETELY REMOVE THE INSTALLATION FOLDER.
You will not be able to proceed beyond this point until the installation directory has been removed. This is a security feature of Joomla!.

Now open relative link of Joomla folder on your XAMPP server.

E.g., http://localhost/Joomla/

# LAB TASK

**Task # 1:** Install Joomla. Add at least 2 menus and install a new template and deploy it on your website and place snaps of it.

| Rubrics | Marks Obtained | |
|---|---|---|
| | 0 | 1 |
| Completeness & Accuracy | | |
| Timelines | | |

Student ID: ____________________________

Subject Teacher: ____________________________

Date: ____________________________