Guru Nanak College, Budhlada

Semester – 1
[2023 – 2024]

WEB DESIGNING PROJECT FILE

Submitted By:

Name: **Mitesh Kumar**
Class: **B.Voc SD – 1**
Roll No.: **11116**

Submitted To:

Assistant Professor **Bhawna Bansal**

Table of Contents

Introduction to HTML

- HTML stands for Hyper Text Markup Language.
- HTML is the standard markup language for creating Web pages.
- HTML describes the structure of a Web page.
- HTML consists of a series of elements.
- HTML elements tell the browser how to display the content.

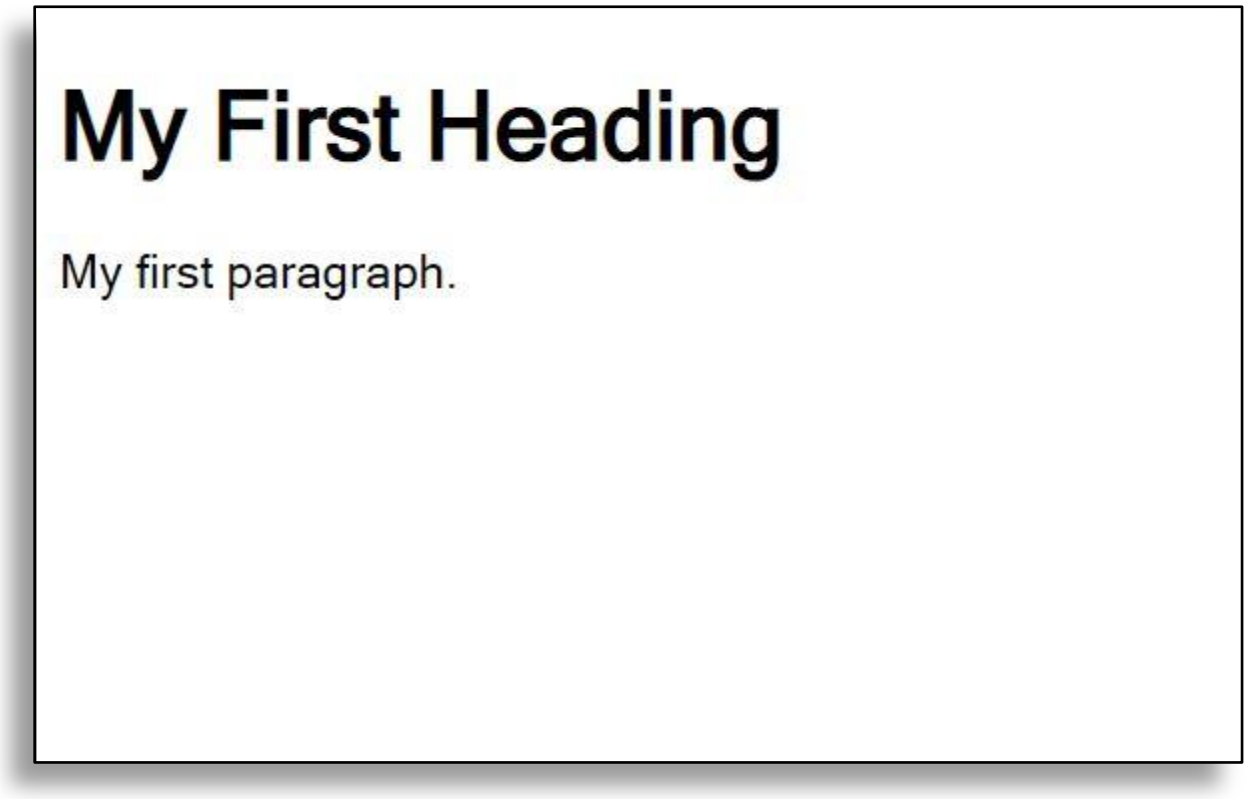
Following is the basic structure of a HTML document:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

To display output of this document as a webpage on a browser, we will have to save the document as a (**.html**) file.

In this way we can create a basic webpage.

The output of the example given in introduction section given will be:



By default, browser leaves some empty spaces at top and left sides of webpage, so in output we are seeing blank space at top and left sides that shouldn't be there. These things can be fixed using concept of styling in CSS (Cascading Style Sheet).

Structure of HTML Page

- A webpage structure is defined by using **head tag** and **body tag** in the **root html tag** in an html document.
- These three tags are must for defining a basic structure of an html page.
- Without any of these the basic html page structure remains incomplete.
- In head tag we also add a title tag in which we insert text to be displayed as the name of a new webpage.

Analyzing Above Given Example

- In HTML we use multiple elements (tags) to form web-pages.
- Each element is written with the following *syntax* :

  <tag-name> Content goes here… </tag-name>
- Each tag name has its own meaning and function on a webpage.

In above example, we used some statements and tags. Let's understand them one by one:

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document.
- The <html> element is the root element of an HTML page.
- The <head> element contains meta-information about the HTML page.

- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab).
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading.
- The <p> element defines a paragraph.

Output of the given code will be like:

In output, FIRST font tag had no attributes means no designing. So the output display is just simple text.

In SECOND font tag, due to use of attributes of font tag, text size has been changed, its color too and its font-family too.

Font Tag & Its Attributes

- Font tag is used to display text on webpage without any formatting.
- Font tag doesn't print any new line at beginning or at end of the text like paragraph <p> tag does.

See the following example:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<font>Hello World</font>
<font face='cambria' size='7' color='purple'>
    Hello World
</font>

</body>
</html>
```

- Here in example we are using two font tags.
- One contains simple "Hello World" text.
- Second one contains exactly the same text. But this time we have used attributes in font tag. These all three are font tag attributes i.e. FACE, SIZE and COLOR.
- FACE attribute defines font-family of the text.
- SIZE attribute defines font-size of the text. We can pass values from 1 to 7. By default, value of SIZE attribute is set to 3.
- COLOR attribute defines the color of the text. By default it is black.

Output of the example will be as follows:

Anchor tags are always underlined by default, if we don't change the styling of anchor tags.

In output, clicking first link will redirect us to college website.

Clicking on second link will redirect us to given file location.

Anchor Tag

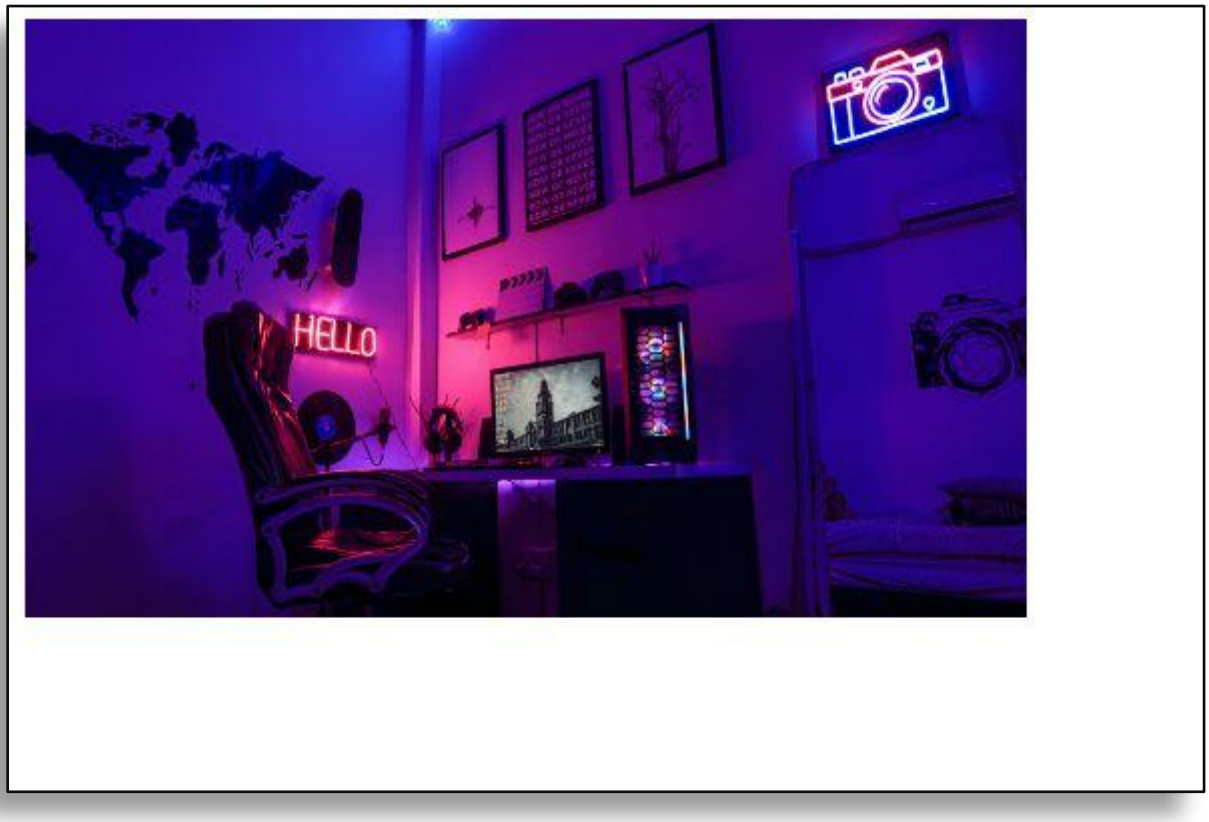
- Anchor tag is used to make a connection between two or more files.
- We can use anchor <a> tag with a reference attribute to pass the location of the linked page.
- Linked page can be a local file or can be online URL.
- When we click on text of anchor tag, we will be redirected to given address.

See the following example:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<a href='https://www.gncbudhlada.org'>
    Go to GNC Budhlada Website
</a>
<a href='otherHtmlPage.html'>
    Go to other html page
</a>

</body>
</html>
```

- Here in example we are using two anchor tags.
- One contains address of GNC college with text “Go to GNC Budhlada Website”.
- Second one contains address of other html local file with text “Go to other html page”.

Output of the example will be:



In output, given address image has been displayed which is occupying width of 500px and height of 300px.

If by chance, we had passed wrong address of image, the text in alt attribute would be displayed with a image icon instead of image.

Like this:

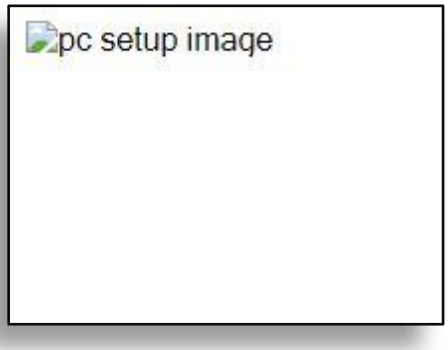


Image Tag

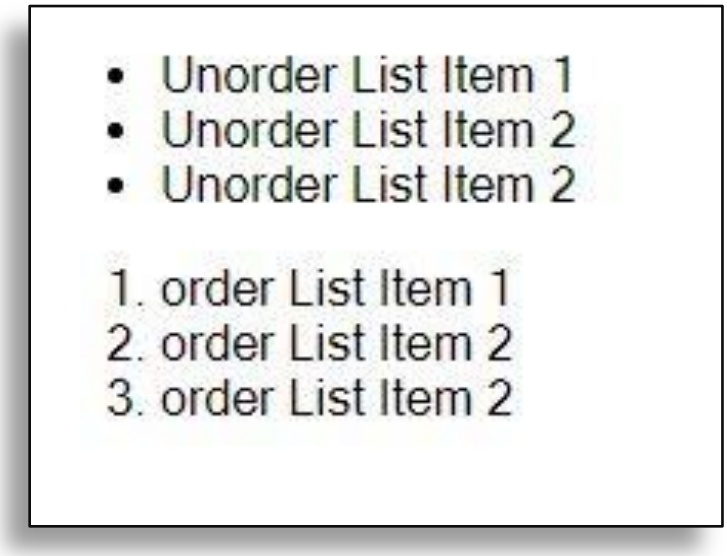
- We include images in web pages to make them attractive.
- We can include images in web pages using image <img> tag of html.

See the following example:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<img src='pic.jpg' alt='pc setup image' width='500'
height='300'>

</body>
</html>
```

- In example, we are using an image tag <img> in which we are giving location of image in source (src) attribute with name and extension of the image. In this attribute, we can pass address of local image or an image from internet.
- In alt attribute, we can pass image description if by some reason browser fails to fetch image from given address.
- In width and height attributes, we can pass numerical values to set width and height of image. It can effect real aspect ratio of the image.
- If we pass one of these two, other will automatically get set with respect to original aspect ratio of the image.

Output of the code will be:



- In output, two lists are created.
- Upper is unordered list with disc symbols. If we pass values like square or circle in type attr of UL, respective symbols will be displayed.
- Same is with ordered list. We can see 1 2 3 numbering to represent order of items. We can pass values like “a”, “A”, “I” etc. to change order style.

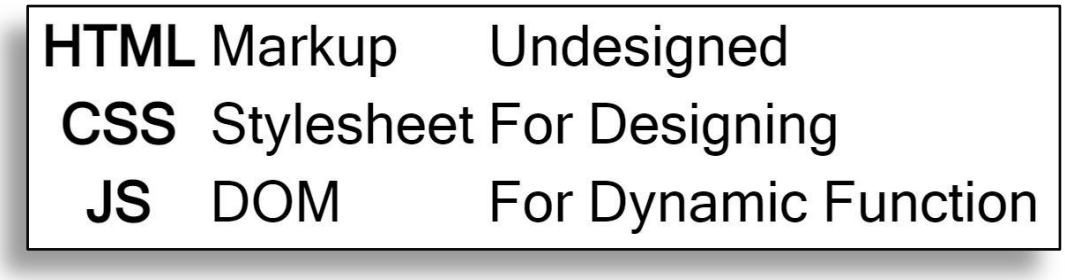
Lists in HTML

- Lists in html are used to demonstrate some set of data in lists form.
- In html there are 2 types of lists: UNORDERED and ORDERED.
- UNORDERED lists are demonstrated by <UL> tag while ORDERED lists are demonstrated by <OL> tag.
- In each type of list we use <LI> list item tag to insert items into list.

See the following example:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<ul type=''>
    <li>Unorder List Item 1</li>
    <li>Unorder List Item 2</li>
    <li>Unorder List Item 2</li>
</ul>

<ol type=''>
    <li>order List Item 1</li>
    <li>order List Item 2</li>
    <li>order List Item 2</li>
</ol>

</body>
</html>
```

Output of the code will be:



HTML	Markup	Undesigned
CSS	Stylesheet	For Designing
JS	DOM	For Dynamic Function

- In output, there are 3 rows as in code.
- Each <th> tag is representing text in heading form and <td> tags are representing text in normal form.
- By default, in html table looks like the given output with no border or designing.
- But we can make design change using CSS.

Tables in HTML

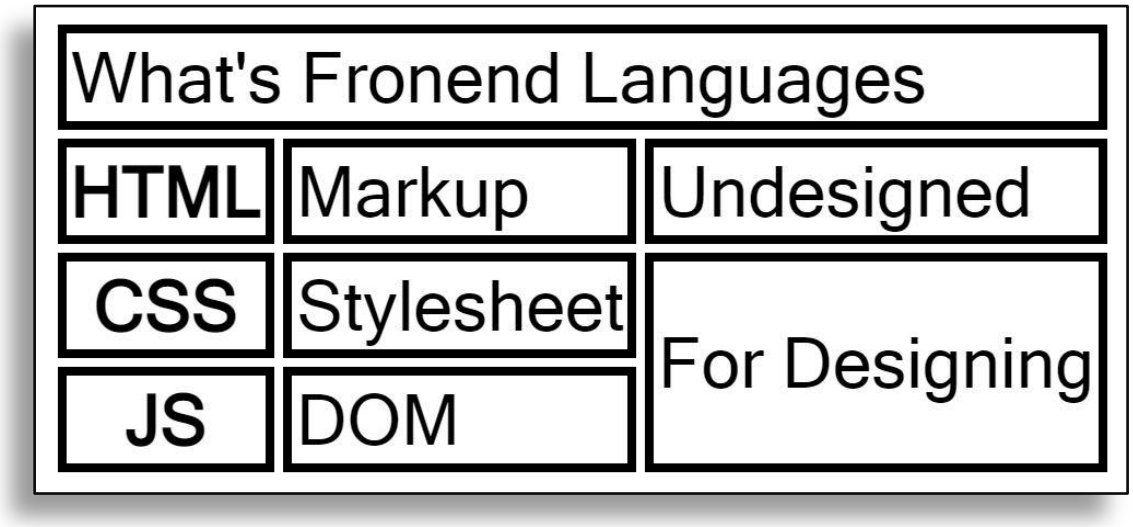
- ❖ We use tables in web pages for the same use as of lists.
- ❖ Tables help to represent some set of data in rows and columns form.

See the following example:

```
<table>
    <tr>
        <th>HTML</th>
        <td>Markup</td>
        <td>Undesigned</td>
    </tr>
    <tr>
        <th>CSS</th>
        <td>Stylesheet</td>
        <td>For Designing</td>
    </tr>
    <tr>
        <th>JS</th>
        <td>DOM</td>
        <td>For Dynamic Function</td>
    </tr>
</table>
```

- ❖ In example, we are creating a table using <table> tag.
- ❖ In table tag, there is <tr> tag that represent table row.
- ❖ We took 3 <tr> tags. In each are inserting a <th> table heading and 2 <td> table data tags.

Output will be:



- In code above, first row tr contained only one td of which colspan was set to 3. So there's no need to create more td tags in first row tr. It's occupying space of 3 cells now.
- In third row, last td was set to rowspan 2 means it will get included in after row of table. So it completed the need to create third td tag in lastmost row.
- It is occupying its own space now and of JS row's third item also.

Row span and Col span Attributes

- In html tables, to extend occupied space of a cell, we can use row span and col span attributes in that cell.
- We can pass a numeric value in these attributes to say to browser that cell must occupy space of n rows or m columns or of both nm rows and columns.

See the following example:

```
<table>
    <tr>
        <td>What's Frontend Languages</td>
    </tr>
    <tr>
        <th>HTML</th>
        <td>Markup</td>
        <td>Undesigned</td>
    </tr>
    <tr>
        <th>CSS</th>
        <td>Stylesheet</td>
        <td rowspan='2'>For Designing</td>
    </tr>
    <tr>
        <th>JS</th>
        <td>DOM</td>
    </tr>
</table>
```

Output of given code will be:

Tag	Meaning	Example
b	Bold Text	**I am bold**
strong	Important Text	**I am strong**
i	Italic Text	*I am italic*
em	Emphasized Text	*I am emphasized*
mark	Marked Text	I am marked
del	Deleted Text	~~I am deleted with strikeout~~
ins	Inserted Text	I am inserted with underline
sub	Subscript Text	Normal Text$_{\text{I am subscript}}$
sup	Superscript Text	Normal Text$^{\text{I am superscript}}$

So this are all the tags meant for text formatting.

Further designing (if wished) can be applied in these tags using CSS.

Text Formatting Tags

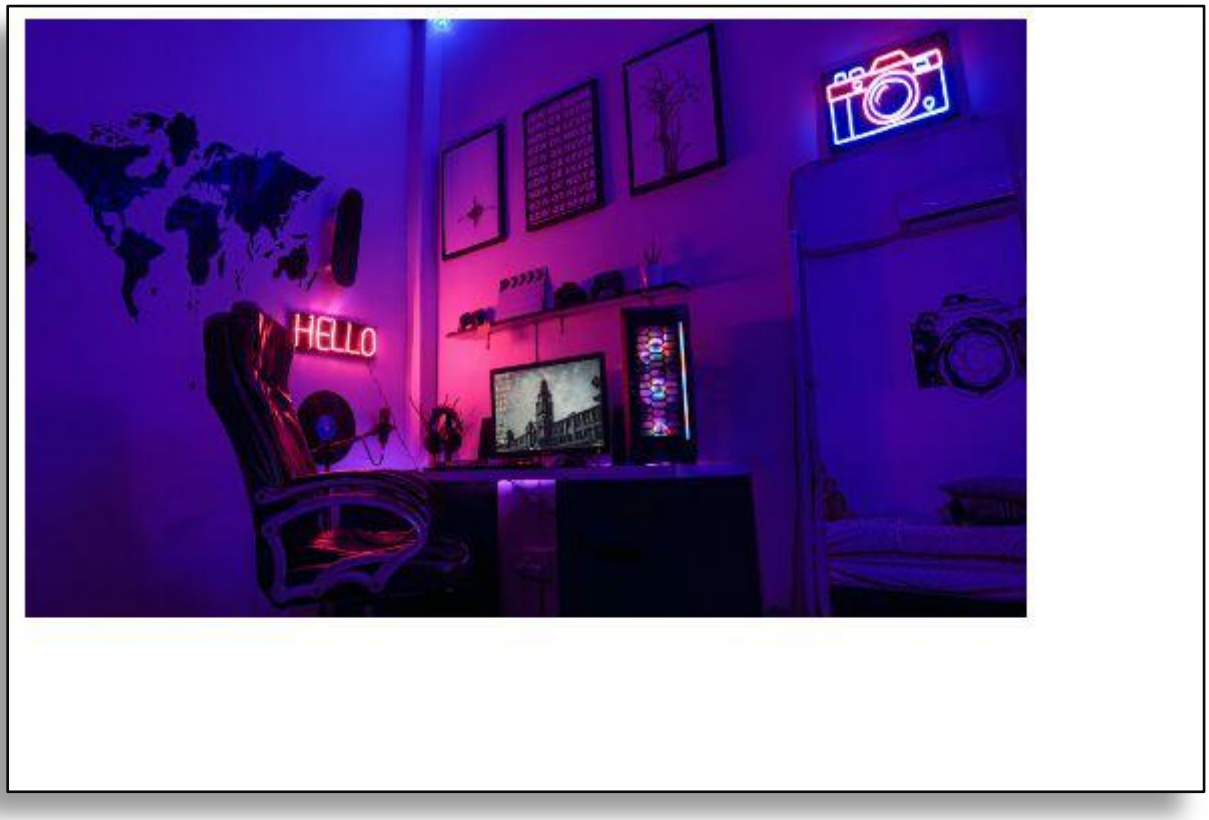
- Html contains several tags to format text of web pages.
- These can be used to display text in special way, so that it can grab attention of user.

See the following example:

```
<b>I am bold</b>
<strong>I am strong</strong>
<i>I am italic</i>
<em>I am emphasized</em>
<mark>I am marked</mark>
<del>I am deleted with strikeout</del>
<ins>I am inserted with underline</ins>
<sub>I am subscript</sub>
<sup>I am superscript</sup>
```

- Here we are using all tags meant for text formatting.
- None of these prints new line character at beginning or end of text.
- Output of these texts is given on next page in a table format with meaning of each tag.

Output will be:



- Output this time will be same as previous one in visual way.
- But if we hover our cursor to given coordinates area it will become pointer means the area is clickable.
- And on clicking the region we will be redirected to given address.

Image Mapping

- We use images in web pages to make web pages attractive and understandable.
- But what if we want to convert some limited regions of image into clickable links such that we get redirected to some other address on clicking defined area of image.
- This can be achieved using html concept of image mapping.

See the example below:

```
<body>

    <img src='pic.jpg' alt='pc setup image'
    width='500' height='300' usemap='#map'>

    <map name='map'>
        <!-- coords x1 y1 x2 y2 -->
        <area shape='rect' coords='0 0 300 100'
        href='https://www.instagram.com'>
    </map>

</body>
```

- In example, we are adding attribute USEMAP in <IMG> tag with a value assigned using HASHTAG #.
- Then we are using map tag with name value same as that of usemap attribute without hashtag this time.
- In map we can use one or more area tags. This time we are using only one.
- We set area shape and coordinates with HREF attribute..

Output of the example will be as follows:

Button Tag

- The <button> tag defines a clickable button.
- Always specify type attribute for a button to tell browser what type of button it is.
- Important types of buttons are 'button', 'reset', 'submit'.

See the example:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<button type='button'>Button</button>
<button type='reset'>Reset</button>
<button type='submit'>Submit</button>

</body>
</html>
```

- In example, we created three buttons.
- One with type button. We can add any event in JavaScript.
- Second with type reset. To reset any data on click.
- Third with type submit. To be used in forms to submit filled data.
- On hovering on any of these with mouse, mouse cursor will change to default to pointer.

Output will be:

- In code, we set input types to text for name, email for email and number for number to stop data from get submitted if we filled data with wrong format.
- Name can contain any thing, so we used text for it.
- Email must include @ and . (dot), so if we press submit without filling any of these, our action will be stopped and we will be notified that we missed something.
- Contact number must include only numeric digits, so we used number input type.

Registration Page

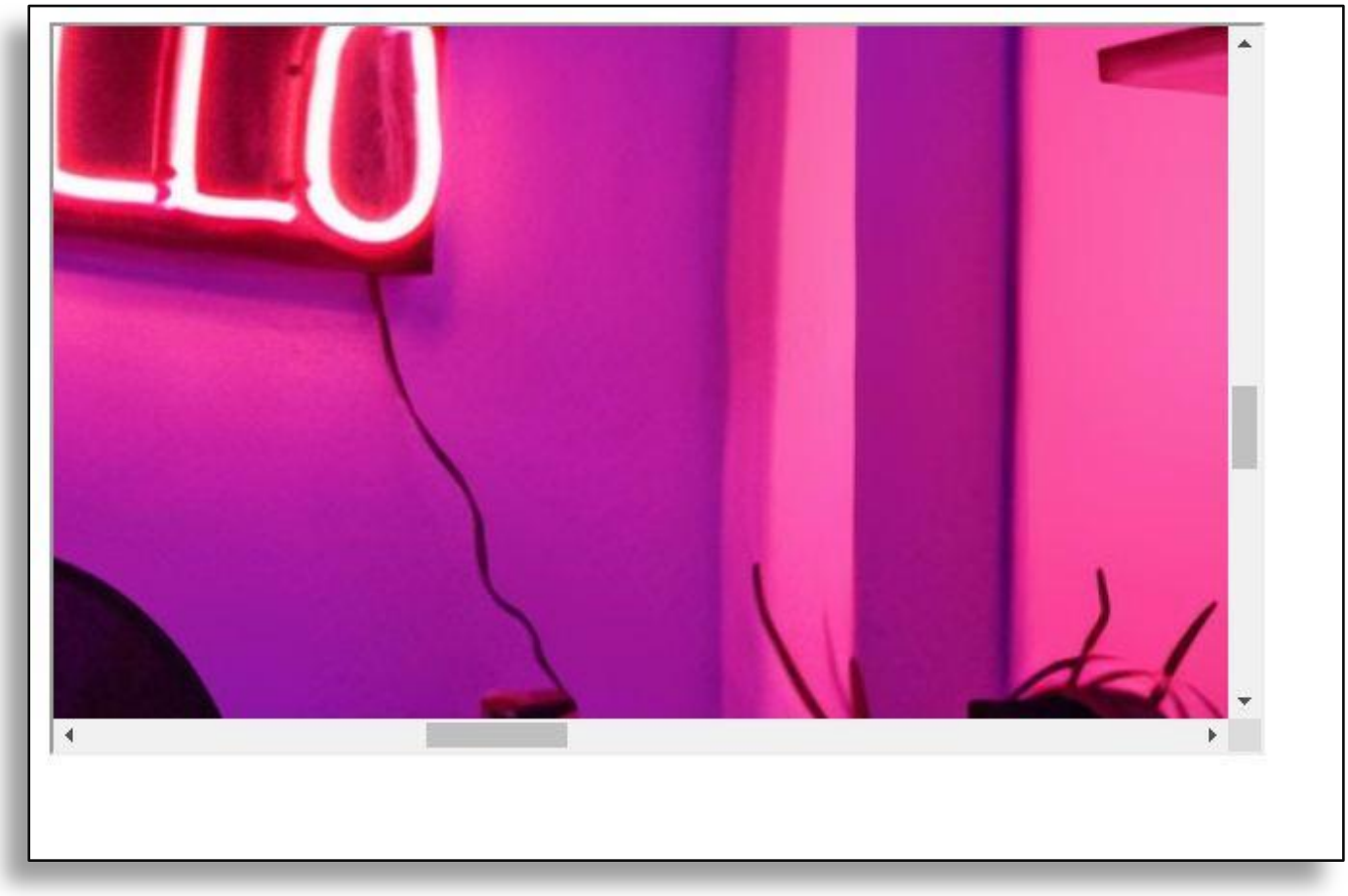
- Now we are going to understand FORM tag.
- It is used to create any type of forms in html web pages.
- Let's understand through example of registration page.

See the example:

```
<form action='#'>
    <label for='name'>Name</label>
    <br>
    <input type='text' id='name'>
    <br>
    <label for='email'>Email</label>
    <br>
    <input type='email' id='email'>
    <br>
    <label for='contact'>Contact Number</label>
    <br>
    <input type='number' id='contact'>
    <br><br>
    <button type='submit'>Submit</button>
</form>
```

We have created form under FORM tag. Inside it we have used label tags, input tags and one button with type submit to submit the data.

Output of above code:



We can see that when width was 500 and height 300, image given didn't compressed with frame's settings, but frame created a new window for content inside it and added scrollbars itself for overflowing content.

Frame Tag

- Frame tag in html is used to define one particular window or frame.
- We can set its width and height etc through its attributes.
- If material inserted in frame tag is overflowing the frame tag width and height then horizontal and vertical scrollbars will appear automatically.
- Although frame is not supported in browsers nowadays, we are using IFRAME tag instead of FRAME tag.

See the example below:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<iframe src='pic.jpg' width='500' height='300'>

</body>
</html>
```

- In example we used iframe tag in body with source attribute passing location of file to be displayed in it.
- We are also setting width and height of frame using attributes.

Output of webpage using INLINE CSS will look like this:

And it's the same output we expected !!

P tag is orange in background and white in text.

Introduction of CSS

- CSS means CASCADING STYLE SHEET.
- These sheets are linked with html file to apply rules written in this file to html tags on webpage.
- It's main purpose is designing of basic page of html.
- There are three types of CSS: inline, internal, and external.

Inline CSS

Inline CSS is CSS used alongwith html tag using style attribute.

See following example:

```
<body>

<p style="
backgroud-color: orange;
color: white;
">
Hello World
</p>

</body>
```

- In code, we used style attribute with paragraph tag.
- In style attribute, properties are written inside pair of inverted commas.
- We set background-color of p tag to orange and its text-color to white.

Output of code with INTERNAL CSS will look like this:

And it's the same output we expected !!

- First p is orange in background and white in text.
- Second is green in background.
- H4 is green in background and black in text.
- Third p is brown in background and white in text.

Internal CSS

Internal CSS is CSS written in html file inside head section inside <style> tag.

We can apply internal css using tag name, or we can assign IDs and CLASSES to html tags, and then use that IDs and CLASS names to target and design html document.

See following example:

```
<head>
    <title>Page Title</title>
    <style>
        p{
            background-color: orange;
            color: white;
        }
        .green{
            background-color: green;
        }
        #brown{
            background-color: brown;
        }
    </style>
</head>
<body>
    <p>Hello World</p>
    <p class='green'>Hello World</p>
    <h4 class='green'>Heading 4</h4>
    <p id='brown'>Brown</p>
</body>
```

- We are using all three methods to write rules of CSS.
- We can assign same CLASS name to number of elements.
- But we can assign ID name unique to each element. We cannot reuse same ID name for any element.

- In output, first p tag background should be orange and text white, second p tag background green and text white, h4 background green and text auto means black, and last p tag background should be brown and text white.

Now lets see what's the external CSS is?

External CSS:

External CSS is nothing different from internal css if we talk of how can we design elements of html.

Just a little difference is that it is written outside of html file in separate (.css) file. It is linked with html file using <link> tag in html head section.

See the following example:

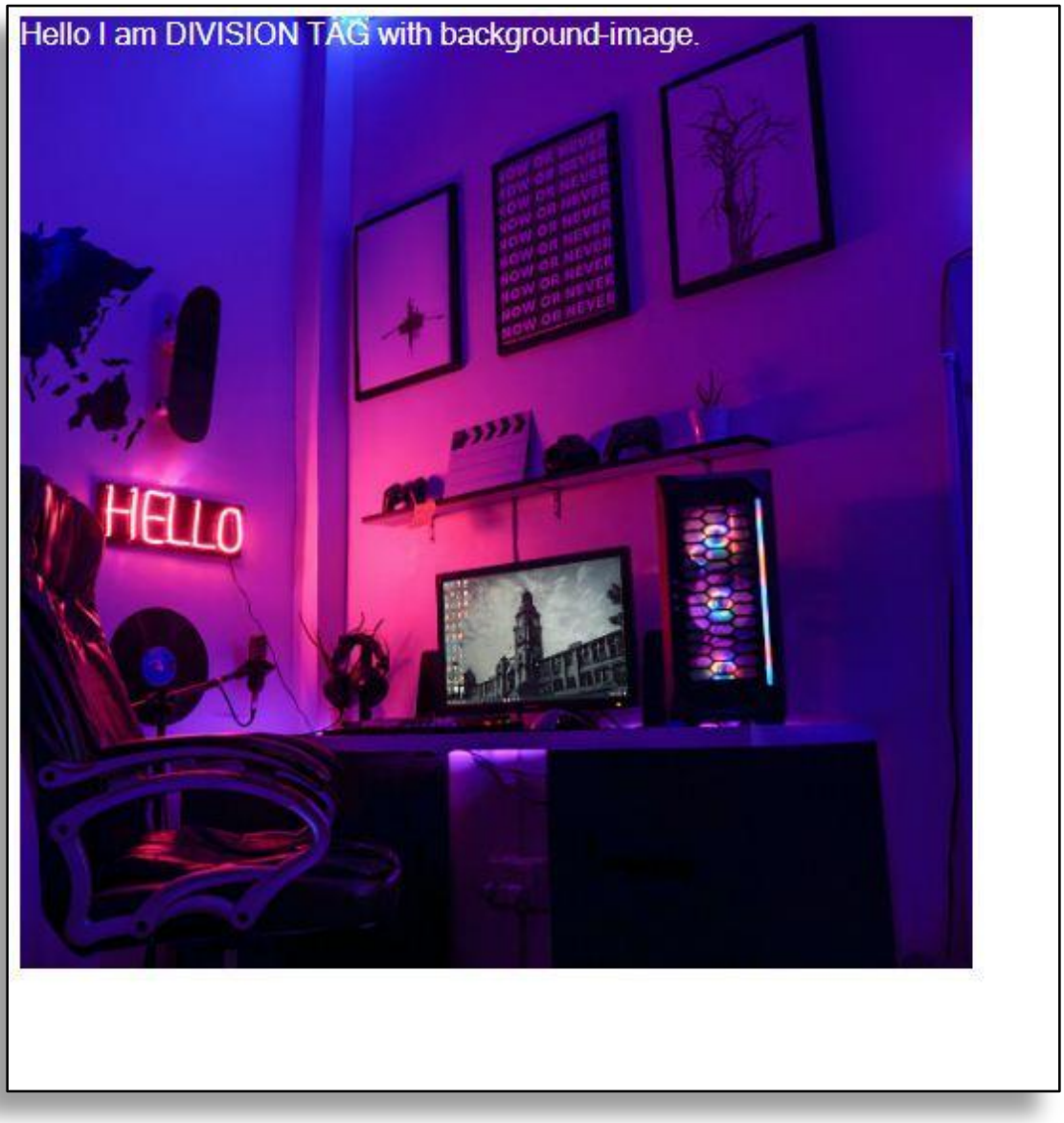
```
<!DOCTYPE html>
<html>

<head>
    <title>Page Title</title>
    <link rel='stylesheet' href='style.css'>
</head>
<body>
    <p>Hello World</p>
    <p class='green'>Hello World</p>
    <h4 class='green'>Heading 4</h4>
    <p id='brown'>Brown</p>
</body>

</html>
```

Output of code with EXTERNAL CSS is same as previous one:

```
p{
    background-color: orange;
    color: white;
}
.green{
    background-color: green;
}
#brown{
    background-color: brown;
}
```

- In above example, we used previous html elements and previous css rules. But this time we separated html and css files.
- We created a style.css file in same folder as of index.html file.
- We included link tag in html head section with relation stylesheet attribute and href reference attribute to tell location of the CSS file to the browser.
- If we see its output it should be the same as the previous one because we didn't made any changes, we just made an external file for designing and linked to the html file.

Output of the example will be:



Now if we use any text or content in div it will be displayed on applied background image. We can make edits in CSS for text color if it is not visible on darker background-images.

So this was all about background images in CSS.

Background Image in CSS

- We talked a lot about how to apply now let's see how to apply an image as a background of any html element using CSS.
- It is way to easy.

See the following example now:

```
<!DOCTYPE html>
<html>

<head>
    <title>Page Title</title>
    <link rel='stylesheet' href='style.css'>
</head>
<body>
    <div>

    </div>
</body>

</html>
```

```
div{
    width: 500px;
    height: 500px;
    background-image: url('pic.jpg');
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
}
```

- In example, we took a empty tag.
- In CSS, we set its width and height to 500 and 500 pixels both.
- Then we used background-image property with value url(), inside url location of image.
- Then we used background-size property to set image's size. We can use values like contain, cover, 100% 100% in it. Currently we used cover.
- Then we used background-position property to tell browser that image should be at start at center or at end. We used center in example.
- Then we used background-repeat property to tell browser that background can repeat itself in x-repeat y-repeat or both or no-repeat. We used no-repeat currently.