|  |  |
| --- | --- |
| **Ex.No:1** |  |
| **Introduction to HTML, Browsers and HTML, Editor’s Offline and Online, Tags, Attribute and Elements, DOCTYPE Element** | |

**a) Explain the procedure to create a HTML program using different editors.**

There are basically two types of editors available for developing HTML Pages:

1. Text Editors
2. WYSIWYG Editors

**Text Editors:**

These are text-based editors where the developers can write their codes and compile them. The code appears in the same manner we write it, thus it requires basic knowledge of HTML. Some of these editors also provide features of making a project, managing all the files related to the web, etc. Examples of HTML Text editors include-Notepad++, VSCode,Sublime Text.

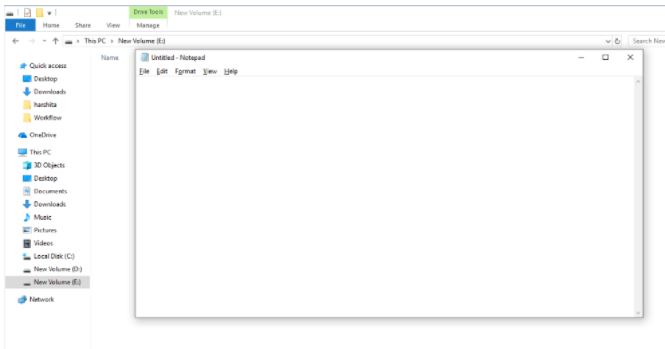
**WYSIWYG Editors:**

’What you see is what you get’ is its full form. WYSIWYG are editors that provide the preview of the output of the source code i.e. as it would appear on a browser. There is a drag and drop feature available in most of them that eases the handling. It does not require any hardcore knowledge of HTML, thus enabling non-technicals to easily develop websites. Examples include-Adobe Dreamweaver, Amaya, BlueGriffon, etc.

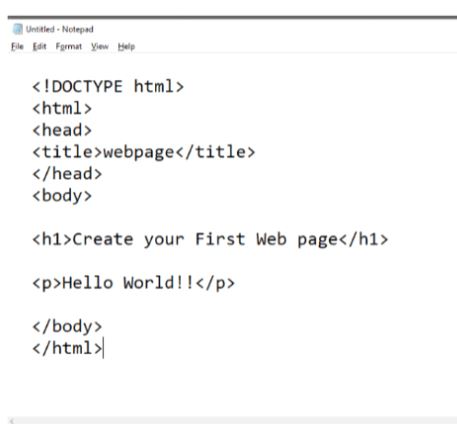
**Developing HTML Page with Notepad**

Notepad is a simple text editor and suitable for beginners to learn HTML. It is available in all versions of Windows, from where you easily access it.

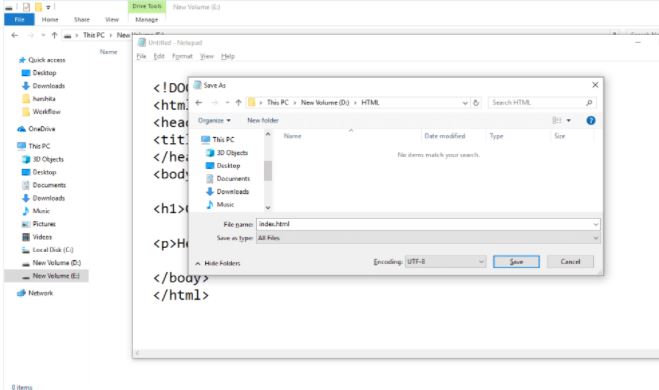
**Step 1: Open Notepad (Windows)**



**Step 2: Write code in HTML**

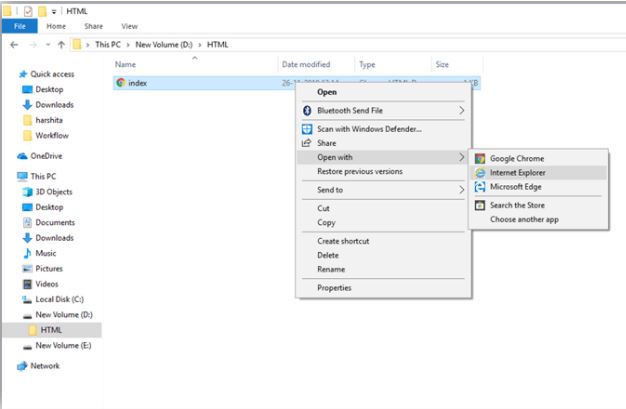
****

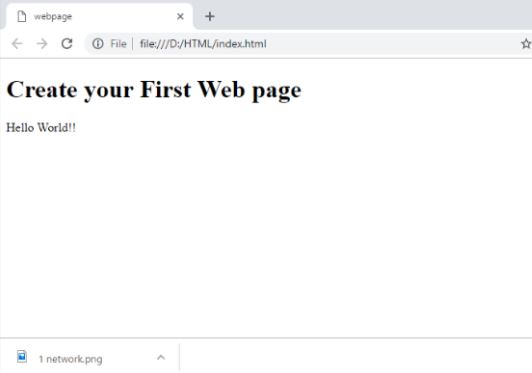
**Step 3: Save the HTML file with .htm or .html extension.**



**Step 4: Open the HTML page in your web browser.**

To run the HTML page, you need to open the file location, where you have saved the file and then either double-click on file or click on open with option



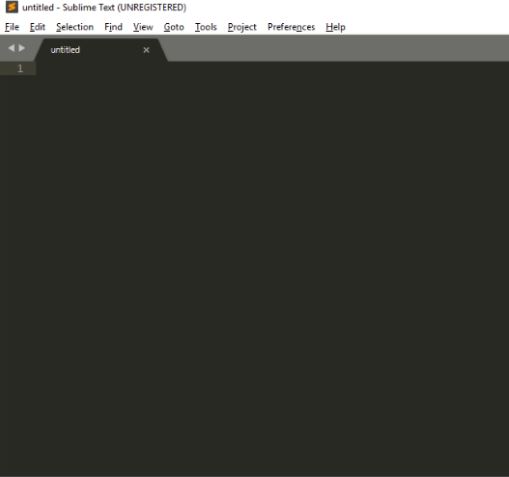


**HTML code with Sublime Text-editor.(Recommended after learning basics of HTML)**

When you will learn the basics of HTML, then you can use some professional text editors, which will help you to write an efficient and fast code. So to use Sublime Text editors, first it needs to download and install from internet.

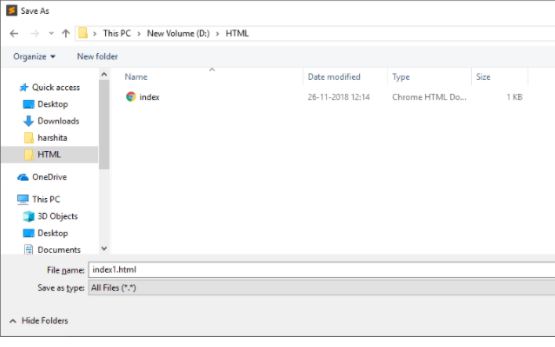
**Step 1: Open Sublime Text editor:**

To open Sublime Text editor go to Start screen ⤏ type Sublime Text⤏ Open it. To open a new page press CTRL+N.

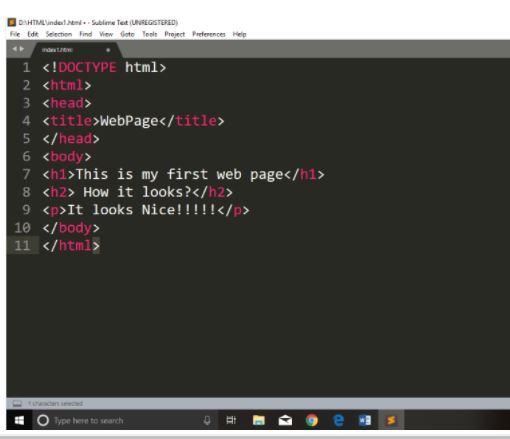


**Step 2: Save the page before writing any code.**

To save your page in Sublime Text press Ctrl+S or go to File option ⤏ save, to save a file use extension .htm or .html. We recommend to save the file first then write the code because after saving the page sublime text editor will give you suggestions to write code.

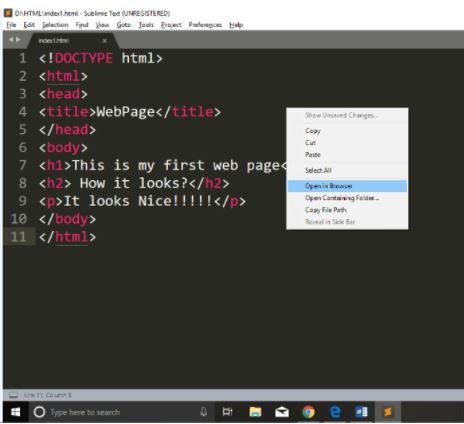


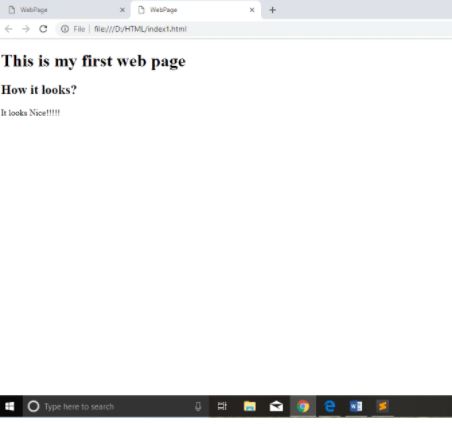
**Step 3: Write the code in Sublime Text editor**

****

**Step 4: Open the HTML page in your Browser**

To execute or open this page in Web browser just **right click** by mouse on sublime text page and click on **Open in Browser**.

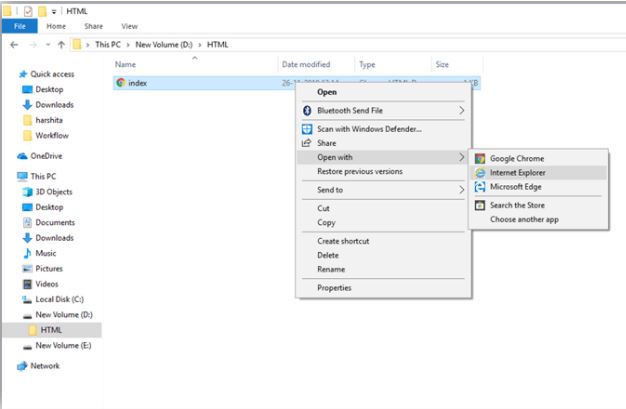
****

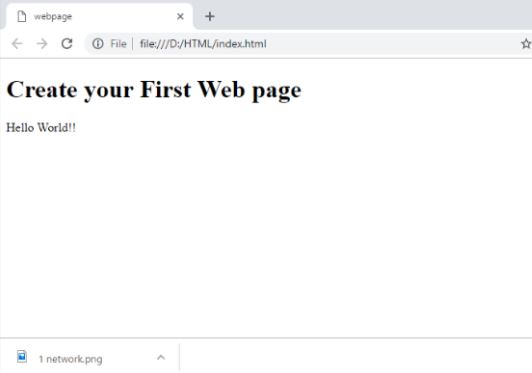
****

**b) Explain the procedure to run a HTML program using Browser**

**Step 4: Open the HTML page in your web browser.**

To run the HTML page, you need to open the file location, where you have saved the file and then either double-click on file or click on open with option





**c) List some of the offline and online editors to create and run HTML pages.**

Atom.

Notepad ++

Sublime Text.

Visual Studio Code.

Adobe Dreamweaver CC.

Froala.

CoffeeCup.

**d) Explain with syntax, the use of DOCTYPE in HTML page.**

The HTML document type declaration, also known as DOCTYPE , is the first line of code required in every HTML or XHTML document. The DOCTYPE declaration is an instruction to the web browser about what version of HTML the page is written in. This ensures that the web page is parsed the same way by different web browsers.

**e) Write a HTML program to experience the HTML Page Structure.**

<!DOCTYPE……..>

<html>

<head>

<title> Title of the page </title>

</head>

<body>

This is my First Page

</body>

</html>

|  |  |
| --- | --- |
| **Ex.No:2** |  |
| **Comments, Headings, Paragraphs.Formatting Text** | |

**a) Write a html program that makes use of basic tags like <html>,<head>,<title>,<body>,<p>,<hr>,<br><h1> to <h6>,<!-- --> tags and their attributes.**

**Description:**

**Block Formatting Tags**

We can use following tags to format blocks of text within HTML document. Some of the most frequently used Block-formatting tags are :

**Body tag (<body>)**

* Body tag contains some attributes such as bgcolor, background etc.
* *bgcolor*is used for setting the background color of a webpage which takes background color name or hexadecimal number such as #000000 to #FFFFFF
* background attribute used for setting mage as a background for webpage and it will take the path of the image which you can place as the background image in the browser.
* **Syntax:**
* <body bgcolor=”name/#rrggbb” background=”image name”> . . . </body>

**Paragraph tag (<p>)**

* HTML documents are divided into paragraphs.
* Paragraphs are defined with the <p> tag.
* Most text is part of a paragraph of information. Each paragraph is aligned to the left, right or center of the page by using an attribute called as align.
* **Syntax:**
* <p [align=”left” | “right” | “center”] > . . . </p>

**Heading tag(<Hn>)**

* HTML is having six levels of heading that are commonly used.
* The largest heading tag is <h1>.
* The different levels of heading tags are <h1> ,<h2>, <h3>, <h4>, <h5><h6>.
* Each heading tag has an attribute called as align which can be set to left, center, or right. By default all headings align left.
* ***Syntax:***
* <h1 [align=”left” | “right” | “center”]> . . . </h1>
* <h2 [align=”left” | “right” | “center”]> . . . </h2> ……….
* <h6 [align=”left” | “right” | “center”]> . . . </h6>

**<hr> tag**

* This tag places a horizontal line across the screen.
* These lines are used to break up the page.
* This tag does not require an end tag.
* This tag also contains attributes which determines how the rule will be displayed.
* It can be aligned but by default is centered on the screen.
* The *size* attribute specifies the thickness of the rule in pixels.
* *noshade*draws the rule as a single thick line rather than giving it‟s default 3D appearance.
* ***Syntax:***
* <hr align=”left” | “right” | “center” size=”n” [noshade]/>.

**Comments**

* Comments can be inserted into the HTML code to make it more readable and understandable.
* Comments are ignored by the browser and are not displayed.
* HTML comment begins with “<!--“ and ends with “-->”.
* There should not be a space between angular bracket and exclamation mark. Each comment can contain as many lines of text as you like. If comment is having more lines, then each line must start and end with -- and must not contain -- within its body.
* <! -- this is a single line comment line - ->
* <! -- this is a multiline comment --
* -- spawned over --
* -- three line -->

**Line break tag**

* This tag is used to the break the line and start from the next line.
* It is an empty tag.
* **<br/>**

**Program:**

<html>

<head>

<title>Block Formatting Tags</title>

</head>

<body>

<h1 align="centre">surampalem</h1>

<h2 align="left">Geography</h2>

<p>The city is situated between the eastern ghats and the Bay of Bengal.The city coordinates lies between 17.704N and 83.2977E</p>

<hr align="left" size="10" [nonshade]/>

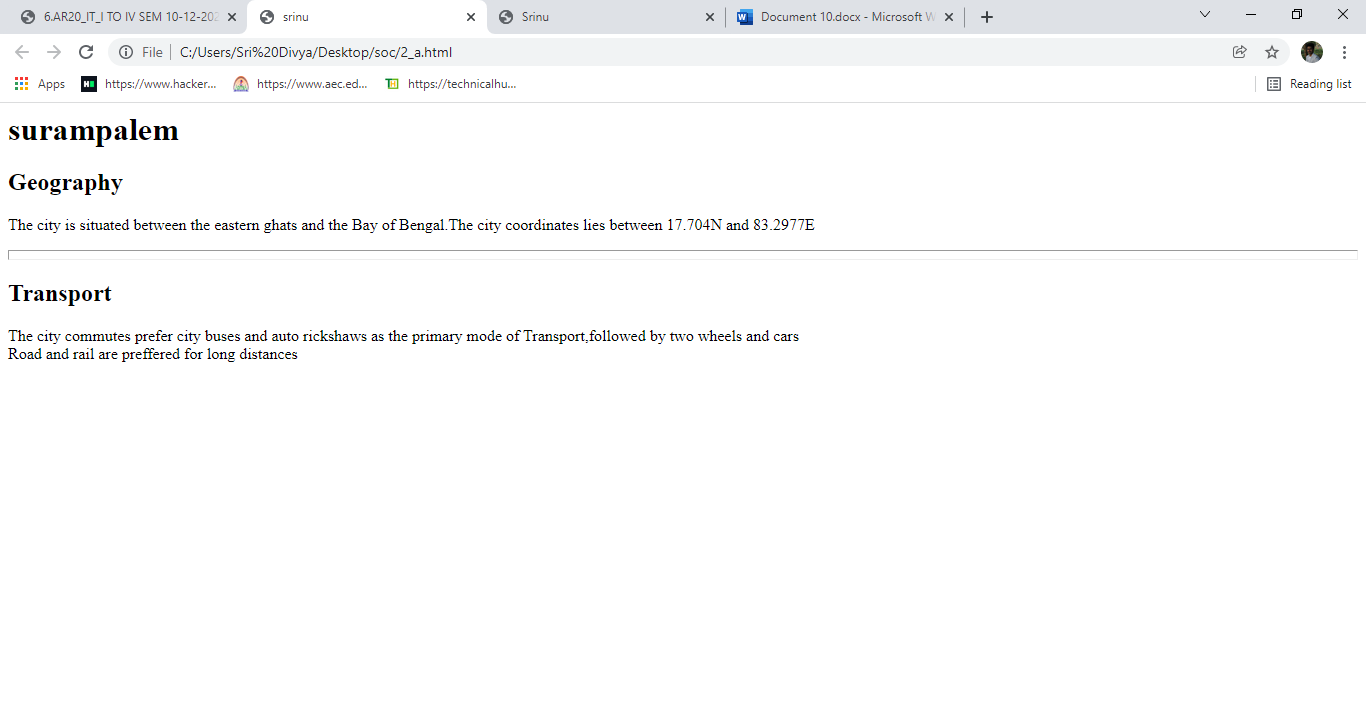
<h2 align="left">Transport</h2>

<p>The city commutes prefer city buses and auto rickshaws as the primary mode of Transport,followed by two wheels and cars <br>Road and rail are preffered for long distances</p>

</body>

</html>

**Output:**



**b) Write a HTML program, that makes use of text formatting tags like<b>,<i>,<u>,<strong>,<sub>,<sup>,<tt>,<pre>**

**Description:**

**Text Formatting Tags**

We can use character/text formatting tags to format a text block that is as small as a single character or as large as an entire document. Some of the most frequently used Character-formatting tags are :

**1. Boldface tag**

* This tag is used for implement bold effect on the text
* <b> ……. </b>

**2. Italic tag**

* This implements italic effects on the text.
* <i>…….</i>

**3. Underline tag**

* This is used to specify that the selected text be displayed with underline.
* <u>. . . </u>

**4. strong tag**

* This tag is used to always emphasized the text
* <strong>. . . </strong>

**5. sub and sup tag**

* These tags are used for subscript and superscript effects on the text.
* <sub>. . .</sub>
* <sup>. . .</sup>

**6.tt tag**

* This tag is used to give typewriting effect on the text
* <tt>. . .</tt>

**7.Pre-formatted text tag**

* It Considers spaces, new lines etc. and as it is prints the information.

<PRE>. . .</PRE>

**Program:**

<html>

<head>

<title>Text Formatting Tags</title>

</head>

<body>

<b>Body tag</b>

<big>body tag contains some attributes such as bgcolor,bg etc.</big>

<i>bg color</i>is used for setting the backgroud color of a webpage which tasks bgcolor name or hexadecimal numbers such as <strong>#000000 to #FFFFFF</strong>

<i>backgroud</i>attribute used for setting page as a bag for webpage and it will take the path of the image which you can place as the bg image in the browser

<tt>syntax body body<sub>bg color </sub=name<sup>#rrggbb</sup>backgroud="img name">

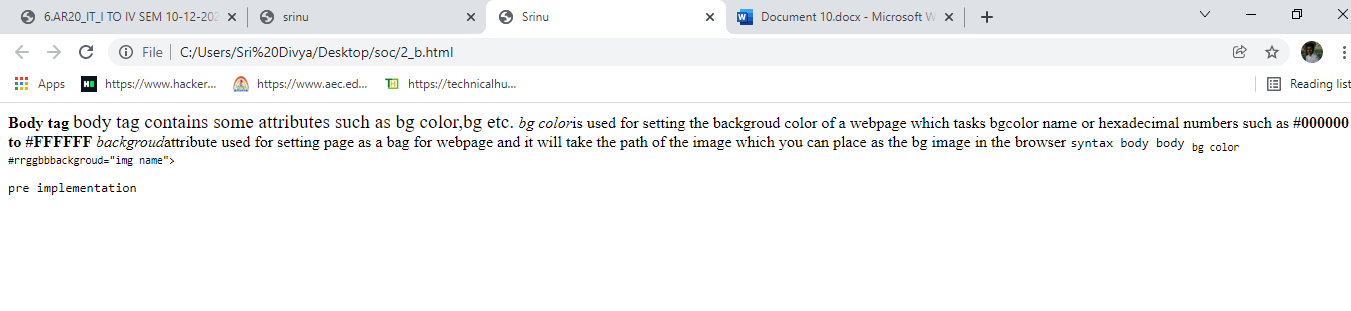
</tt>

<pre>pre implementation</pre>

</body>

</html>

**Output:**



**c) Write a HTML program, to explain the working of <font> tag and its attributes.**

**font tag**

* This sets font type, size, color and relative values for a particular text.
* Absolute font sizes are can be set from 1 to 7.
* Relative font sizes are set by using +/- 1 to 7.
* The color of the text is set by *color* attribute. This takes hex value which represents the amounts of red, green and blue in a chosen color.
* ***Syntax:***
* < font face=”font name ” size=”[+|-]n” color=”#rrggbb”>
* **face:** The style of the letter can be specified like arial, times new roman etc…
* **size:** The size can be specified .
* **color:** The color of the text to be displayed is specified through the color attribute.

**Program:**

<html>

<head>

<title>Srinu</title>

</head>

<body>

<h1 align="centre">surampalem</h1>

<h2 align="left">Geography</h2>

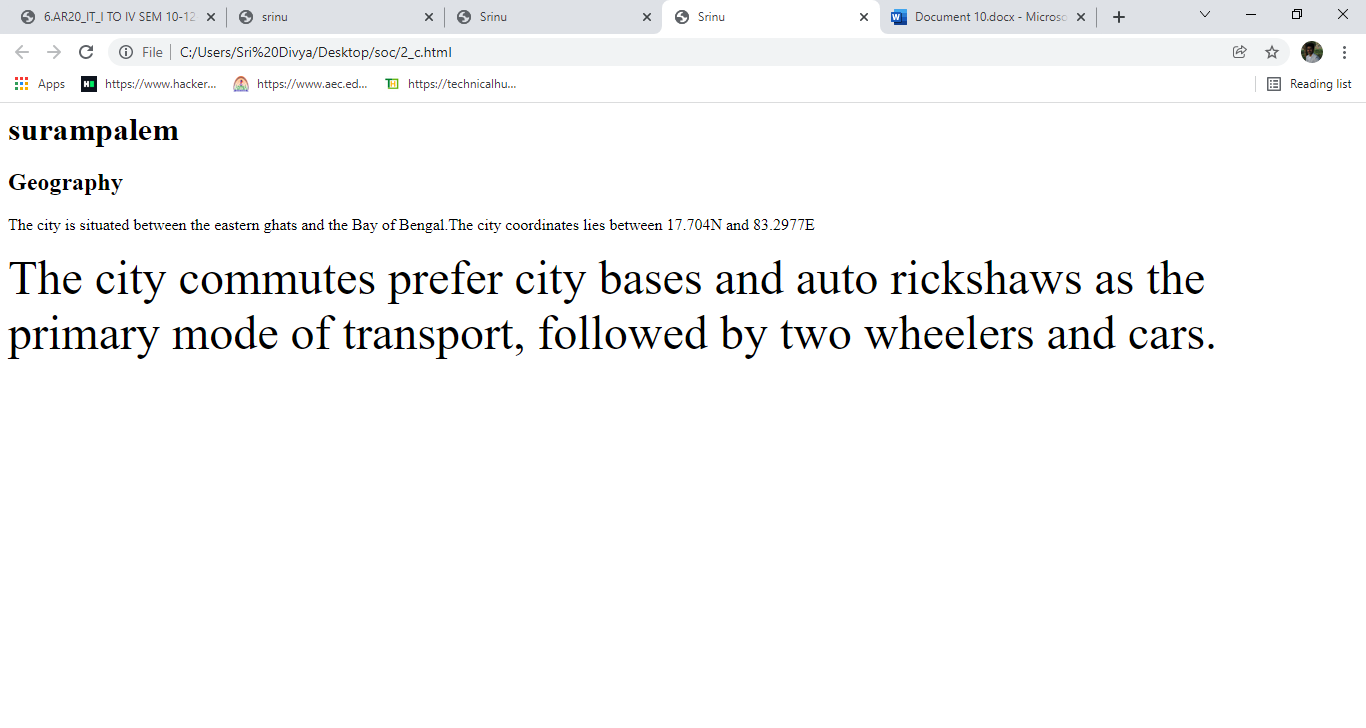
<p>The city is situated between the eastern ghats and the Bay of Bengal.The city coordinates lies between 17.704N and 83.2977E</p>

<p><font face="time new roman" size="7" color="black">The city commutes prefer city bases and auto rickshaws as the primary mode of transport, followed by two wheelers and cars.</p>

</body>

</html>

**Output:**



|  |  |
| --- | --- |
| **Ex.No:3** |  |
| Lists and Linksa)Write a HTML program, to explain the working of lists. Note: It should have an ordered list, unordered list, nested lists and ordered list in an unordered list and definition lists.Description: List is a collection of items**.**   * Types of lists are * Ordered lists * Unordered lists * Nested lists * Directory lists * Definition lists   **Ordered Lists:**   * Ordered lists are also called as Numbered lists or Sequenced lists. * In Ordered list, the list of items has an order that is signified by numbers, hence called as a numbered list. * An ordered list should start with <OL> element, which is immediately followed by a <LI> and end of ordered list is specified by </OL> element. * Different ordered list types like numeral list; alphabet list etc can be specified with TYPE tag. * Optional parameter with <OL> tag is START attribute, which indicates the starting number or alphabet.   **Syntax:**   * <OL [type={“1” or “I”or “A” or “a” or “i”}] START=n> * <LI>item1 </LI> * <LI>item2 </LI> * : * : * <LI>item n </LI> * </OL>   **Different Ordered list types**   * Type=”1” (default) e.g.1,2,3,4….. * Type=”A” Capital letters e.g.A,B,C… * Type=”a” Small letters e.g. a,b,c…… * Type=”I” Large roman letters e.g. I, II, III,…   **Unordered Lists:**   * Unordered lists are also called as UnNumberedlists . * The unordered list items are used to present a list of items, which are typically separated by white space/ or marked by bullets. * An unordered list should start with <UL> element, which is immediately followed by a <LI> and end of ordered list is specified by </UL> element. * TYPE attribute can also be added to <UL> tag that it indicates that the displayed bullet along with the list of items circle, square or disk. * By default it is disc.   **Syntax:**   * <UL [type={square or disc or circle}] > * <LI>item1 </LI> * <LI>item2 </LI> * : * : * <LI>item n </LI> * </UL>   **Nested Lists:**   * Both ordered list and unordered list are used. * We can put an ordered list in an unordered list and viceversa.   **Directory Lists:**   * A Directory list element is used to present a list of items containing up to 20 characters each. * A directory list must start with <DIR> element, which s immediately followed by <LI> element.   **Definition Lists:**   * A definition list is a list of Definition terms. * To create definition list it must start with <DL> and immediately followed by <DT> tag * <DD> tag is used to give description of the definition  Program: **Ordered List:** <html><head><title>Departments of Engineering</title></head><body><p>Ordered List</p><ol type="a" start="1"><li>CE</li><li>EEE</li><li>Mech</li><li>ECE</li><li>CSE</li><li>IT</li><li>Min.</li><li>PT</li><li>Ag.E</li><li>H&BS</li></ol></body></html>Output: Capture  **Unordered List:**  <html>  <head>  <title>Unordered List</title>  </head>  <body>  <h1 align="center">Aditya Engineering College</h1>  <h2 align="left">The courses offered by AEC</h2>  <ul type=circle>  <li>CE</li>  <li>EEE</li>  <li>Mech</li>  <li>ECE</li>  <li>CSE</li>  <li>IT</li>  <li>Min.</li>  <li>PT</li>  <li>Ag.E</li>  </ul>  </body>  </html>  **Output:**  Capture  **Nested List:**  <html>  <head>  <title>nested list</title>  </head>  <body>  <ol type="A">  <li>CSE</li>  <ul type="square">  <li>SPM</li>  <li>CD</li>  </ul>  <li>IT</li>  <ul type="disc">  <li>LP</li>  <li>CG</li>  </ul>  </ol>  </body>  </html>  **Output:**  Capture  **Definition List:**  <html>  <head>  <title>Definition List</title>  </head>  <body>  <dl>  <dt>HTML:</dt>  <dd>The The abbrevation is Hyper Text Mark Up Language and is used for creation of Web pages</dd><br>  <dd>Used for performing validation at the client side</dd>  </dl>  </body>  </html>  **Output:**  Capture  **b)Write a HTML program, to explain the working of hyperlinks using tag and href, target Attributes.**  **Note: Use text to link →https://www.aec.edu.in/**  **Use image to link →https://www.aec.edu.in/?p=Gallery**  **Description:**  **Hyperlinks**   * It is a block formatting tag * The real power and flexibility of HTML is in Hyperlinks. * Hyperlinks are created with anchor tag**(<a>)** * We can use hyperlink to * 1. create links with HTML pages * 2. link different HTML pages * 3. access services at other Internet sites   **Anchor tag:**   * The anchor tag is created by <a> . . .</a> tags. * The tag has three sections: the address of the referenced document, a piece of text displayed as link, and the closing tag.   ***Syntax:***   * <a href=”address”> Text </a> * *href*means Hypertext references that can be used for giving the path of a file which you want to link. * The “Text” between the <a> . . . </a> tags acted as a hyperlink. This text is called *“hypertext”.* When we click on this text, the linked page or file will be displayed.   We can also replace this text with image. In that case, the image will act as a hyperlink.   * The target attribute specifies where to open the linked element.   **Syntax**   * <a target="\_blank|\_self|\_parent|\_top|*framename*"> * Attribute Values  |  |  | | --- | --- | | Value | Description | | \_blank | Opens the linked document in a new window or tab | | \_self | Opens the linked document in the same frame as it was clicked  (this is default) | | \_parent | Opens the linked document in the parent frame | | \_top | Opens the linked document in the full body of the window | | *Framename* | Opens the linked document in the named iframe |  Program: **Using Text as a Link**  <html>  <head>  <title>Linking to Aditya</title>  </head>  <body>  <a href="https://www.aec.edu.in">Click the link to redirect to Aditya website</a>  </body>  </html> Output: | |

**Using Image as a Link**

FirsrCreate a working directory and save an image for gallery with name gallery.jpg.

<html>

<head>

<title>Image Link Demo</title>

</head>

<body>

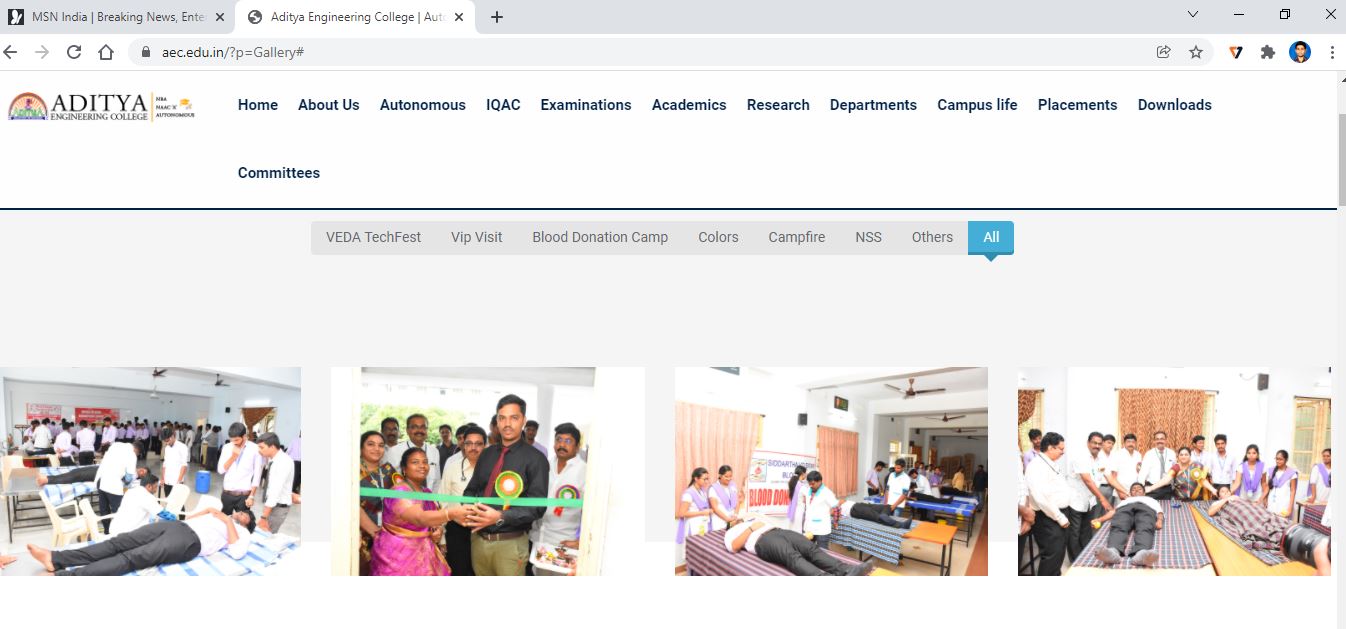
<p>To redirect to Aditya gallery click the image below:</p><br>

<a href="https://www.aec.edu.in/?p=Gallery"><img src="Gallery.jpg" height="200" width="200" border="10"></a>

</body>

</html>

**Output:**



|  |  |
| --- | --- |
| **Ex.No:4** |  |
| **HTML Images** | |

**a)Create a HTML document that has your image and your friend’s image with a specific height and width. Also when clicked on the images it should navigate to their respective profiles.**

**Description:**

**Images**

* Images are second aspect of pleasant Web experience.
* We can add images to an HTML page to either improve its appearance or present important information.
* To add images to an HTML page, we can use several image formats. These formats include „*gif‟, „jpg‟ and „png‟.*
* Images can be added in two different ways:
* 1. By using *„background‟* attribute of <body> tag
* 2. By using <img> tag

**The Image tag**

* We can use image tag in an HTML page to add images along with text.
* An image added using the image tag occupies space within the HTML page.
* To add image to HTML page, use *<img>*tag.
* ***Syntax:***
* <imgsrc=”URL” height=”n” width=”n” align = “left” | “right” | “top”|”middle” alt = ” string ” border=n >

**Attributes:**

* ***src :***
* The important parameter is SRC, because it specifies the location of the source file
* If the file is included in the current directory, no need to specify the path otherwise given the entire path.
* We can also use uniform resource locator(URL).
* b) ***align :*** used to specify the vertical alignment of an image
* c) ***height*** : used to specify the vertical area that an image will occupy in HTML page
* d) ***Width*** : used to specify the horizontal area that an image will occupy in HTML page
* e) ***alt :*** used to specify the text when browser unable to display the image or image not available.
* f) ***border:*** Specifies the border width along the image.

P**rogram:**

**Friend1.html**

<html>

<head>

<title>hello world 1</title>

</head>

<body>

<p>NAME:freind</p>

<p>ROLL NO:1235</p>

<p>BRANCH:IT</p>

[<p>EMAIL:\*\*\*\*\*\*\*\*@gmail.com</p](mailto:%3cp%3eEMAIL:********@gmail.com%3c/p)>

<p>PHONE NO:916\*\*\*9165</p>

</body>

</html>

**Friend2.html**

<html>

<head>

<title>hello world 2</title>

</head>

<body>

<p>NAME:friend 2</p>

<p>ROLL NO:1221</p>

[<p>EMAil:\*\*\*\*\*\*\*@gmail.com</p](mailto:%3cp%3eEMAil:*******@gmail.com%3c/p)>

<p>984\*\*\*\*\*10</p>

</body>

</html>

**Main.html**

<html>

<head>

<title>hello world</title>

</head>

<body>

<a href="Friend1.html">

<imgsrc="billgates.jpg" alt="It's gates">

</a>

<a href="Friend2.html">

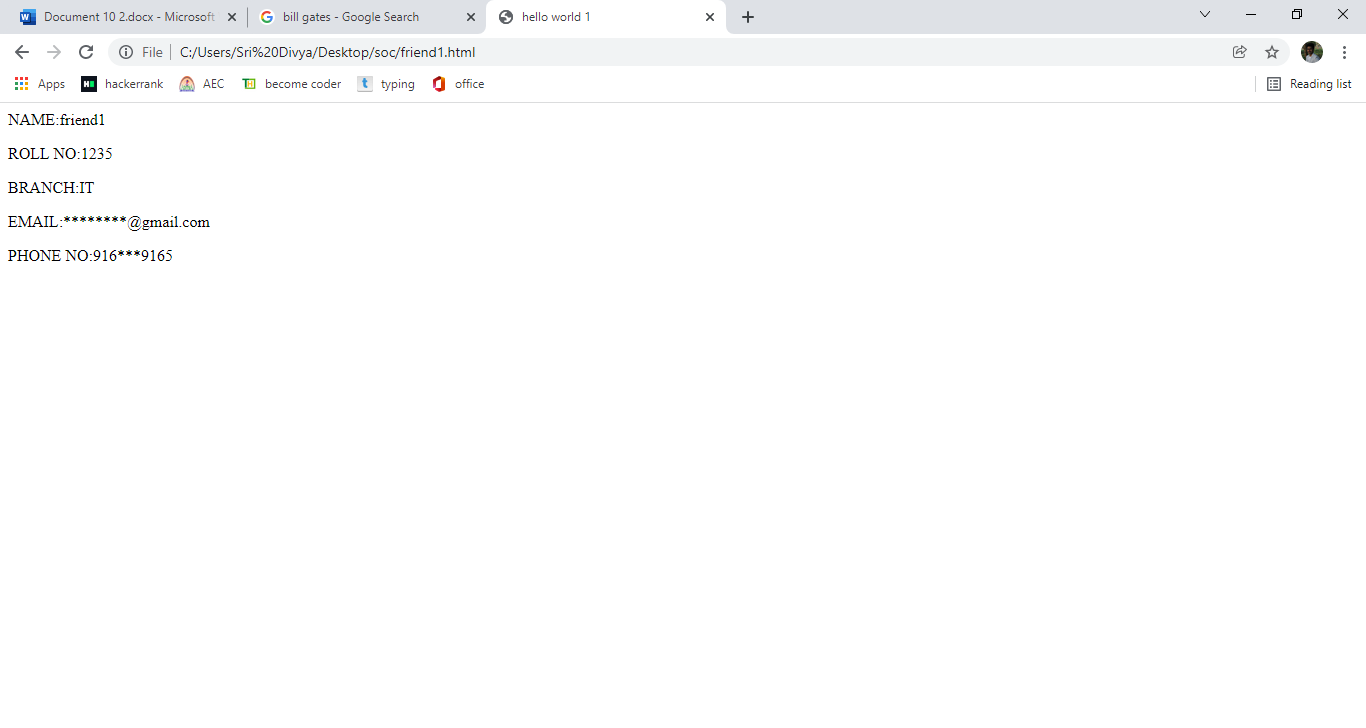
<imgsrc="satyanadella.jpg" alt="It'ssatyanadella">

</a>

</body>

</html>

**Output:**



**b) Write a HTML program, in such a way that, rather than placing large images on a page, the preferred technique is to use thumbnails by setting the height and width parameters to something like to 100\*100 pixels. Each thumbnail image is also a link to a full sized version of the image. Create an image gallery using this technique.**

**Program:**

n=input("Enter n value ")

<html>

<head>

<title>Image Demo</title>

</head>

<body>

<a href="my\_image.jpg"><imgsrc="my\_image.jpg" height="100" width="100" alt="Some scenarie" border="10"></a>

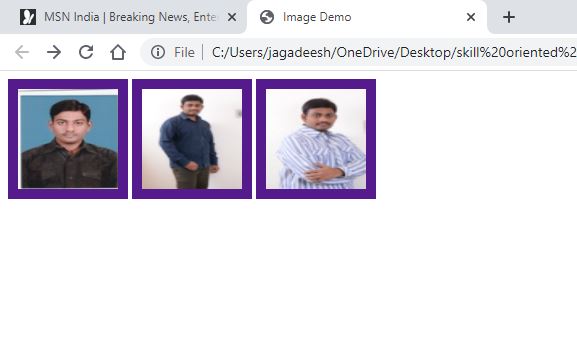
<a href="my\_image2.jpg"><imgsrc="my\_image2.jpg" height="100" width="100" alt="Some scenarie" border="10"></a>

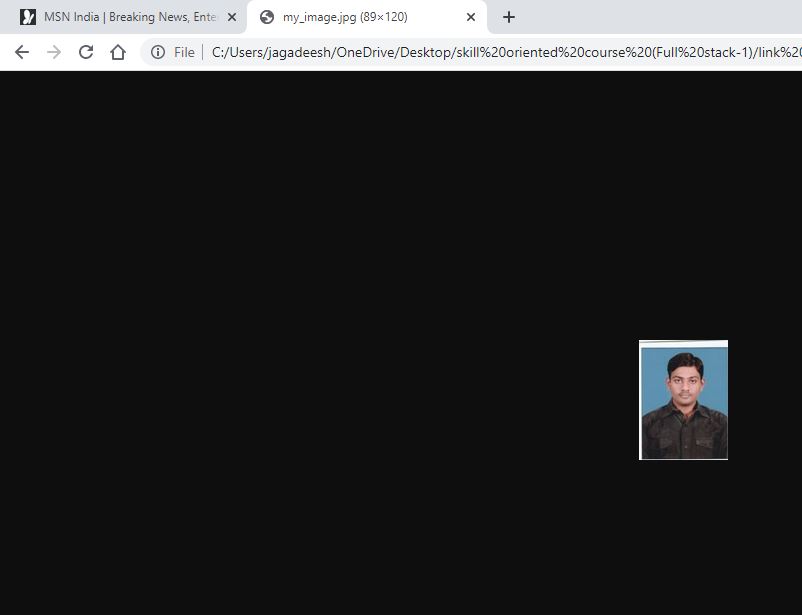
<a href="my\_image3.jpg"><imgsrc="my\_image3.jpg" height="100" width="100" alt="Some scenarie" border="10"></a>

</body>

</html>

**Output:**



****

|  |  |
| --- | --- |
| **Ex.No:5** |  |
| **HTML Tables** | |

**a)Write a HTML program, to explain the working of tables. (use tags :<table>,<tr>,<th>,<td> and attributes border,rowspan,columnspan)**

**Description:**

**Tables**

* Tables are defined with the **<table>** tag.
* A table is divided into rows with the **<tr>** tag
* 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.
* Header information in a table are defined with the **<th>** tag.
* All major browsers display the text in the <th> element as bold and centered

We can use various elements to specify the details of a table. Many table elements also take attributes, which allows you to further specify the look of the table.

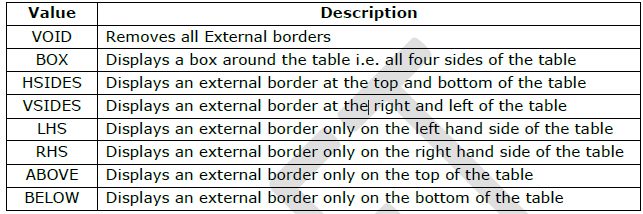
* TABLE
* Table row
* Table data
* Table Heading

**The TABLE Element**

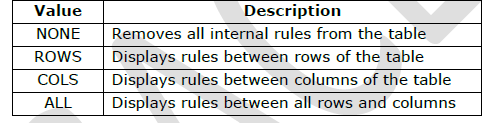
* The TABLE element is the container element for table and uses the <TABLE>...</TABLE> tags to enclose all the other table tags.
* If the <TABLE> tag is omitted or not closed, the browser ignores all the other tags that you specify for the table.
* Everything that we write between these two tags will be within a table.
* The attributes of the table will control in formatting of the table.

***The <TABLE> tag has the following attributes:***

* 1. ***align***: used to specify the alignment of a table in a HTML page.
* ***2.border*:** used to specify the thickness of the table border in pixels.
* ***3.bgcolor***: used to specify the background color for the table.
* 4. ***frame***: used to specify the which side of the outer border is visible in the browser.
* We must specify the border attribute before you specify the FRAME attribute.



***rules***: used to specify the borders between cells. You must specify the border attribute before you specify the FRAME attribute.



**Cell Attributes**

* ***cellspacing***: used to specify the spacing between cells in pixels.
* ***cellpadding***: used to specify the spacing between cell content and cell wall in pixels.
* ***height***: used to specify the height of table in pixels or %.
* ***width***: used to specify the width of table in pixels or %.
* ***Syntax:***
* ***<****TABLE align=”left | right | center” border=”n” bgcolor=”#rrggbb” cellspacing=”n”cellpadding=”n” frame=”value” rules=”value” height=”n | %” width=”n | %”>*
* *</TABLE>*

**Table Row:**

* Table row element is used to create rows in a table.
* The Table row element uses the <TR> tag to create a row.
* The <TR> tag has the following attributes:
* ***align***: used to specify the horizontal alignment of the contents for cells of a row.
* ***valign:*** used to specify the vertical alignment of the cell content for all cells of the row.
* ***bgcolor:*** used to specify the background color of the row.
* ***Syntax***:
* <TR *align=”left | right | center”*
* *valign=”top | bottom |middle”*
* *bgcolor=”#rrggbb”* > ….. </TR>

**Table Data Element:**

* The <TD> tag has the following attributes:
* ***colspan:*** used to specify the number of columns the cell can span.
* ***rowspan*:** used to specify the number of rows the cell can span
* **align:** used to specify the horizontal alignment of the data within a cell.
* ***valign***: used to specify the vertical alignment of data within the cell.
* ***bgcolor***: used to specify the background color of the cell.
* **Syntax:**
* <TD *align=”left | right | center”*
* *valign=”top | bottom |middle”*
* *bgcolor=”#rrggbb”*
* *colspan=”n”*
* *rowspan=”n”*> ….. </TD>

**Program:**

<html>

<head>

<title>Table Creation</title>

</head>

<body>

<table align="center" border="3" bgcolor="pink" frames="void" rules="all" cellpadding="8" cellspacing="8" height="10px" width="5pts">

<caption align="center">Demo Table</caption><br>

<tr align="center" valign="middle" bgcolor="green">

<th>Col1</th>

<th>Col2</th>

<th>Col3</th>

</tr>

<tr>

<td align="center" valign="top">1</td>

<td align="center" valign="bottom">2</td>

<td align="right" valign="middle">3</td>

</tr>

<tr>

<td bgcolor="orange">4</td>

<td bgcolor="orange">5</td>

<td bgcolor="orange">6</td>

</tr>

<tr>

<td colspan="2">7</td>

<!--<td>8</td>-->

<td>9</td>

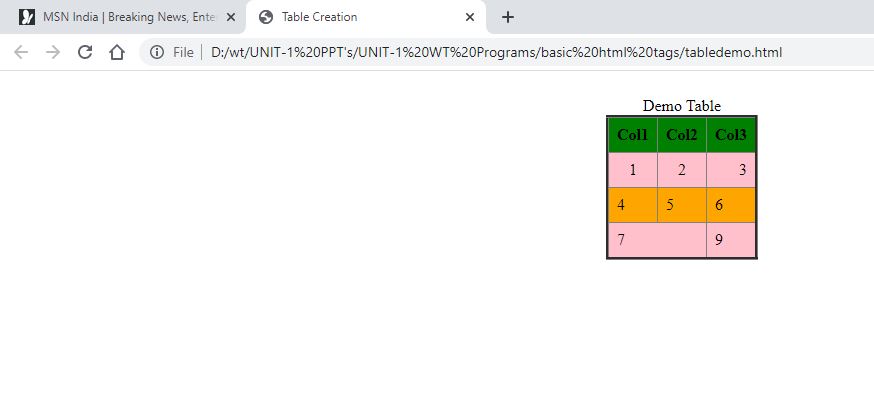
</tr>

</table>

</body>

</html>

**Output:**



**b) Write a HTML program, to explain the working of tables by preparing a timetable. (Note: Use <caption> tag to set the caption to the table & also use cell spacing, cell padding, border,rowspan, colspan etc.).**

**Program:**

<html>

<head>

<title>IT Time Table</title>

</head>

<body>

<h1 align = "center"><b>ADITYA ENGINEERING COLLEGE</b></h1>

<h3 align = "center">An Autonomous Institution</h3>

<h4 align = "center">Approved by AICTE, Permanently Affiliated to JNTUK & Accredited by NBA, NAAC with 'A' Grade<br>Recognized by UGC under the section 2(f) and 12(B) of the UGC act 1956<br> Aditya Nagar ADB Road, Surampalem</h4>

<h2 align = "center">DEPARTMENT OF INFORMATION TECHNOLOGY</h2>

<h3 align = "center"><u>B.Tech III SEM TIME TABLE</u></h3>

<table align = "center" border="2" cellspacing="0" cellpadding="7">

<tr>

<throwspan="2">Day</th>

<th>1</th>

<th>2</th>

<th>3</th>

<th></th>

<th>4</th>

<th>5</th>

<th>6</th>

<th>7</th>

</tr>

<tr>

<th>9:30-10:20</th>

<th>10:20-11:10</th>

<th>11:10-12:00</th>

<th>12:00-12:50</th>

<th>12:50-1:50</th>

<th>1:50-2:40</th>

<th>2:40-3:30</th>

<th>3:30-4:20</th>

</tr>

<tr align="center">

<th>MON</th>

<td>SE<br>(BRSS)</td>

<td>DM<br>(DVLP)</td>

<td>OOP<br>(MRB)</td>

<td>LUNCH</td>

<td colspan="4">USP LAB</br>(SN/SS)</td>

</tr>

<tr align ="center">

<th>TUE</th>

<td>ADS<BR>(AW)</td>

<td colspan="3">OOP LAB<br>(SS/GS)</td>

<td rowspan="4"><b>LUNCH</b></td>

<td>SE<br>(BRSS)</td>

<td>OOP<br>(MRB)</td>

<td>OS <br> (DC)</td>

</tr>

<tr align ="center">

<th>WED</th>

<td> ADS<br> (AW)</td>

<td> OS<br> (DC)</td>

<td> SE<br> (BRSS)</td>

<td> DM<br> (DVLP)</td>

<td colspan ="4">OS LAB<br> (DC/SN)</td>

</tr>

<tr align="center">

<th> THU</th>

<td> DM<br> (DVLP)</td>

<td> ADS<br> (AW)</td>

<td> SE<br> (BRSS)</td>

<td> OOP<br> (MRB)</td>

<td> OS<br> (DC)</td>

<td> DM<br> (DCLP)</td>

<td> ADS<br> (AW)</td>

</tr>

<tr align="center">

<th> FRI</th>

<td> SE<br> (BRSS)</td>

<td> OS <br> (DC)</td>

<td> OS<br> (DC)</td>

<td> OOP<br> (MRB)</td>

<td>ADS<br> (AW)</td>

<td> ADS<br> (AW)</td>

<td> DM<br> (DVLP)</td>

</tr>

<tr align="center">

<th>MON</th>

<td>DM<br>(DVLP)</td>

<td>OS<br>(DC)</td>

<td>OOP<br>(MRB)</td>

<td>LUNCH</td>

<td colspan="4">SKILL LAB<br>(SJ/GS)</td>

</tr>

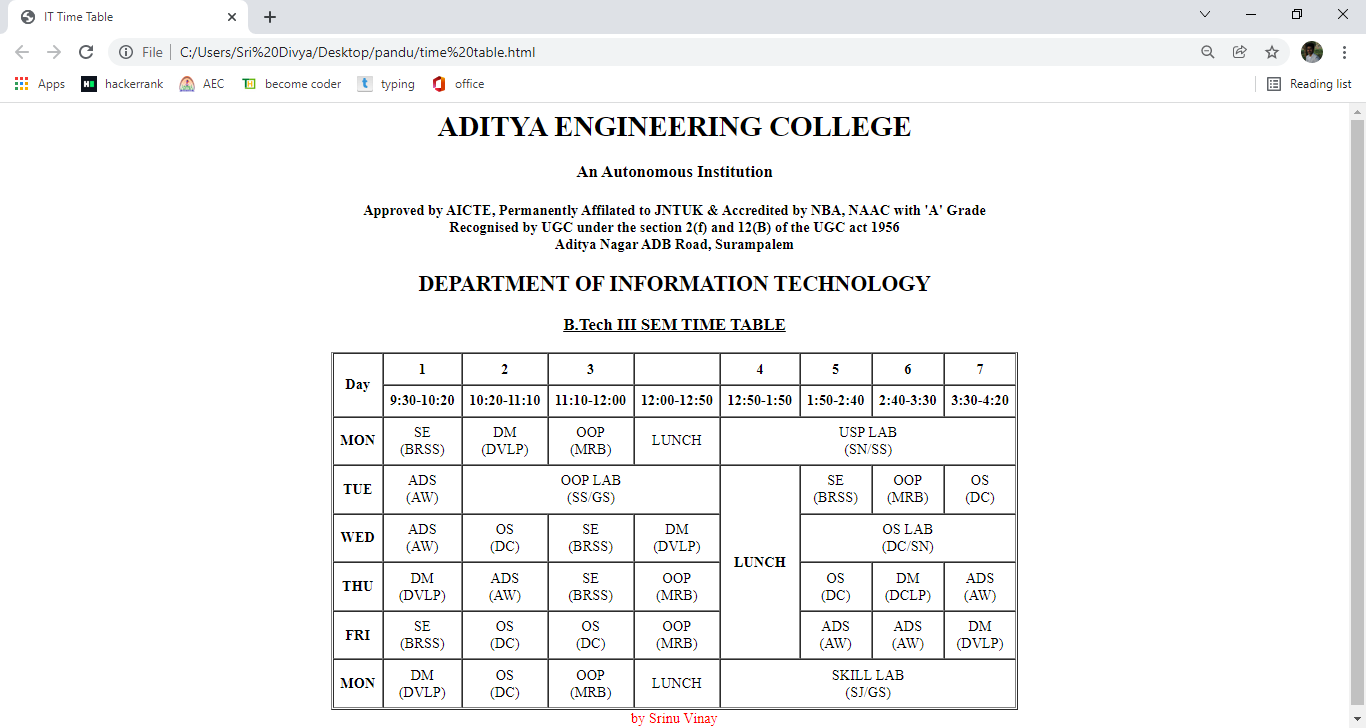
<caption align ="bottom" style="color:red">by Srinu Vinay</a></caption>

</table>

</body>

</html>

**Output:**



|  |  |
| --- | --- |
| **Ex.No:6** |  |
| **Frames and Forms** | |

**a)Write a HTML program, to explain the working of frames, such that page is to be divided into 3 parts on either direction. (Note: first frame → image, second frame → paragraph, third frame → hyperlink. And also make sure of using “no frame” attribute such that frames to be fixed).**

**Description:**

**Frames:**

* Frames are not supported in HTML5.
* A frame provides a facility to display one page at a time.
* Collection of these frames into a set provides a facility to display more than one page at a time on browser and is referred as a **frameset.**
* **A Frameset** is a collection of frames.
* **A web page** containing frame elements is called a framed page.
* **A framed page** begins with **<frameset>**& ends with **</frameset>** tag.
* Each individual frame is identified through **<frame>** tag.

**Syntax:**

* <frameset rows/cols="45%,55%">
* <frame name="framename" src=“filename" noresize scrolling="yes">
* <frame name=" framename " src=" filename " noresize>
* </frameset>

**Program:**

**Image.html**

<html>

<head>

<title>Clg logo</title>

</head>

<body>

<imgsrc="adityalogo.jfif" alt="adityaclg logo">

</body>

</html>

**Para.html**

<html>

<head>

<title>Aditya details</title>

</head>

<body>

<h1>Aditya Engineering College was established in the academic year 2001-02 under the aegis of Aditya Academy, Kakinada with the approval of AICTE and Affiliated to JNTU with an intake of 180 in three UG Courses in Engineering &Technology.The College is situated in an eco-friendly area of 180 acres with thick greenery at Surampalem, Gandepalli Mandal, East Godavari District, Andhra Pradesh. The College is 15 KM away from SamalkotRailway Station on Howrah-Chennai Railway line in South Central Railway. The College is 35 Km away from Kakinada and Rajahmundry on ADB Road. The College has four academic Buildings with a total carpet area of 44,524 Sq. Mts. apart from two boys hostels and one girls hostel buildings. The particulars of academic buildings and the departments / offices accommodated are as follows.

</h1>

</body>

</html>

**Link.html**

<html>

<head>

<title>Clg link</title>

</head>

<body>

<a href="https://aec.edu.in">click on me</a>

</body>

</html>

**Main.html**

<html>

<head>

<title>frameset</title>

</head>

<frameset rows="50%,50%" border="50">

<frame name="image" src="frameimg.html" noresize scrolling="yes">

<frameset cols="50%,50%" border="50" frameborder="50">

<frame name="paragraph" src="framep.html" noresize scrolling="yes">

<frame name="link" src="framelink.html" scrolling="yes">

</frameset>

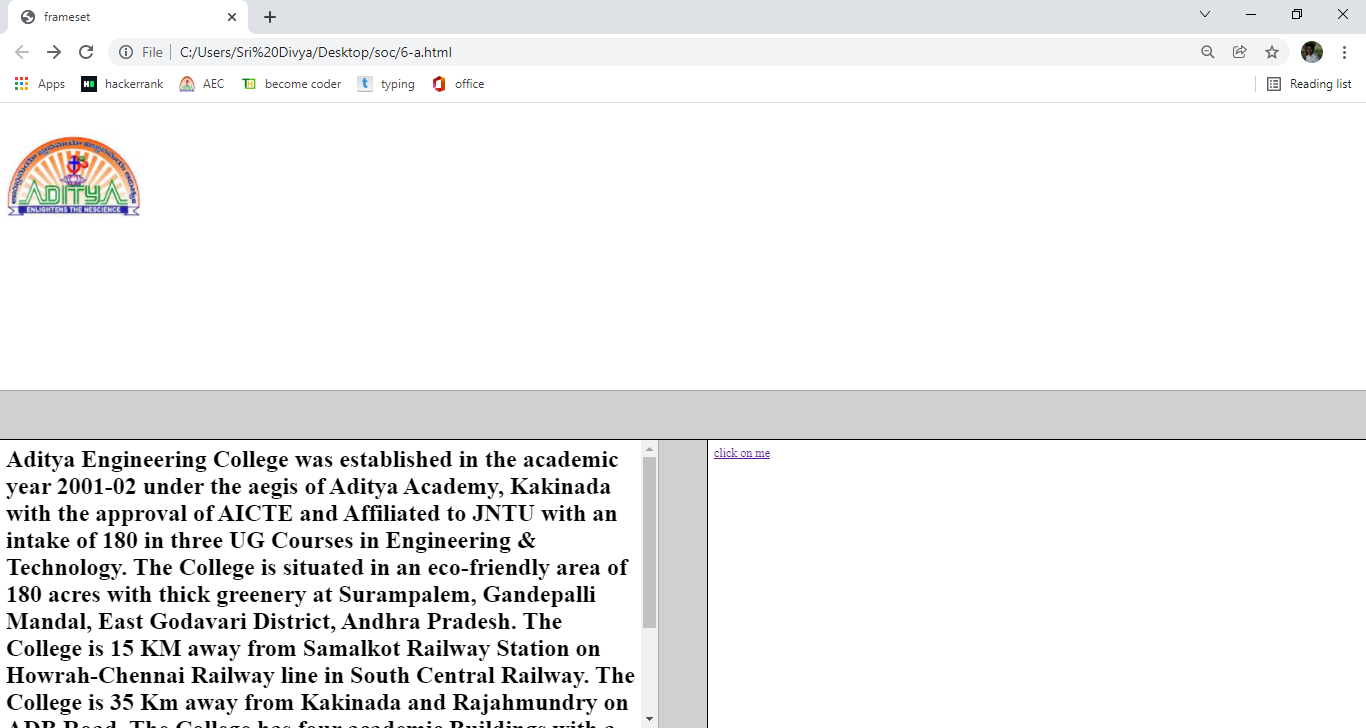
</frameset>

<body>

</body>

</html>

**Output:**



**b)Write a HTML program, to explain the working of forms by designing Registration form. (Note: Include text field, password field, number field, date of birth field, checkboxes, radio buttons, list boxes using<select>&<option> tags,<textarea>, and two buttons ie: submit and reset. Use tables to provide a better view).**

**Description:**

**Forms**

* HTML provides several user interactive elements such as **label, text fields, buttons, radio button, check box, combo box, list box, text area and password fields**.
* The **<form>** tag is used to create forms.
* The important attributes of **<form>** tag are name, **METHOD** and **ACTION.**
* **Method:** This parameter specifies how the data entered in the form can be sent to destination.
* There are 2 methods of sending information GET & POST

**Labels**

* HTML is not provides any separate tag for label.
* The text which is preceded by the interfacing element is taken as a label.

**Text field**

* Text is typically required to place one line string.
* Text fields can be created by using <input>tag.
* Syntax:<input type=”text” size=n name=”text name”>

**Buttons**

There are three types of buttons

* + **Button**
* Syntax:<input type=button value=”click me”>
  + **Submit**
* Syntax:<input type=submit value=”submit”>
  + **Reset**
* Syntax: <input type=reset value=”clear”>

**Radio buttons**

* + To display set of option to user, where he can select one among them, radio buttons are used.
  + ***Syntax***:<input type=”radio” name=”text name” value=”text” checked>
  + All the name attributes of group of radio buttons must be same and all the value attributes are different.
  + Value indicates the value that is stored while selecting particular button, which returns to name of group.
  + Checked parameter indicates the initial selection that we give as default.

**Checkboxes**

* These are similar to radio buttons as we can select more than one option among the options.
* Syntax: <input type=”checkbox” name=”text name” value=”text” checked>

**Combo box**

* Combo box displays one at a time and by clicking on combo arrow displays other options available.
* Syntax: <select name=“selectname” size=n multiple>
* <option value=“text” label=“labelname” selected>
* <option>
* </select>

**TextArea**

* Textarea is typically required to place multi line string.
* Syntax:<textarea name=”text name” cols=n rows=n></teaxtarea>

**Password field**

* When user typed the information in a text field it is displayed as it is.
* If we want to display it in encoded form, use the password field.
* Syntax: <input type=”password” name=”pwd” size=n >

**Program:**

<html>

<head>

<title>Formdemo</title>

</head>

<body>

<table align="center">

<caption="Formdemo">

<form name="f1" method="get">

<tr>

<td>Username: </td>

<td><input type="text" name="uname" size="10" maxlength="15"></td>

</tr>

<tr>

<td>Password:</td>

<td><input type="password" name="pwd" size="10" maxsize="15"></td>

</tr>

<tr>

<td>Address:</td>

<td><textarea name="ta" rows="5" cols="10"></textarea></td>

</tr>

<tr>

<td>Languages known:</td>

<td><input type="checkbox" name="c1" value="Te" checked>Telugu</br>

<input type="checkbox" name="c2" value="Ta">Tamil</br>

<input type="checkbox" name="c3" value="Hi">Hindi</br>

<input type="checkbox" name="c4" value="En">English</br>

</td>

</tr>

<tr>

<td>Programming Proficiency:</td>

<td>

<input type="radio" name="r1" value="C">C</br>

<input type="radio" name="r1" value="Java">Java</br>

<input type="radio" name="r1" value="Python" checked>Python</br>

<input type="radio" name="r1" value="PHP">PHP</br>

</td>

</tr>

<tr>

<td>Willing to work at:</td>

<td>

<select name="s1" multiple>

<option value="Mu">Mumbai</option>

<option value="ch" label="Chennai">Chennai</option>

<option value="Hy">Hyderbad</option>

<option value="viz" selected>Vizag</option>

</select>

</td>

</tr>

<tr>

<td colspan="2"><input type="submit" value="submit" >

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

</td>

</tr>

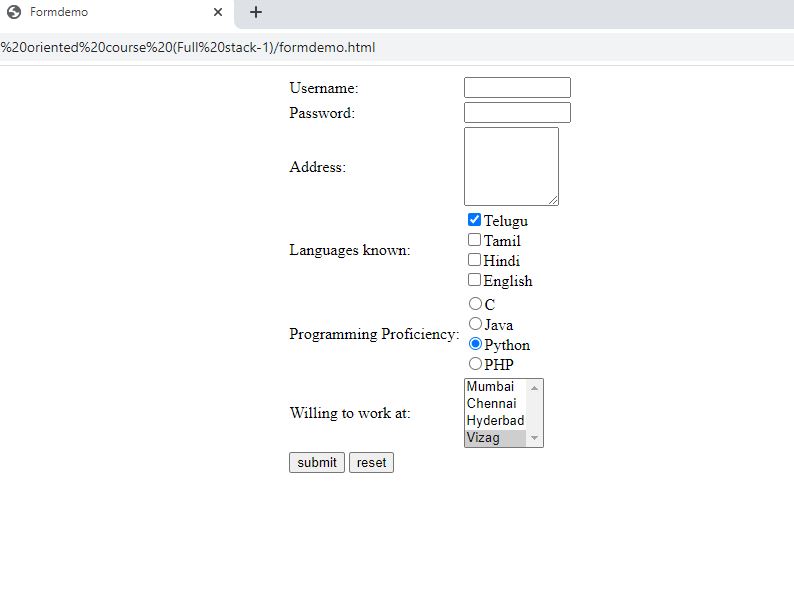
</form>

</table>

</body>

</html>

**Output:**



|  |  |
| --- | --- |
| **Ex.No:7** |  |
| **HTML 5** | |

**a)Write a HTML program, that makes use of <article> ,<aside> ,<figure> ,<figcaption> ,<footer> ,<header> ,<main> ,<nav> ,<section> ,<div> ,<span> tags**

**Description:**

**Article Tag:**

* The <article> tag specifies independent, self-contained content.
* An article should make sense on its own and it should be possible to distribute it independently from the rest of the site.
* Potential sources for the <article> element:

1. Forum post

2. Blog post

3. News story

4. Comment

* Syntax:
* <article>….</article>

**Aside Tag:**

* The <aside> tag defines some content aside from the content it is placed in.
* The aside content should be related to the surrounding content.
* Syntax:
* <aside>…</aside>

**Figure and Figcaption Tags:**

* The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
* While the content of the <figure> element is related to the main flow, its position is independent of the main flow, and if removed it should not affect the flow of the document.
* The <figcaption> tag defines a caption for a <figure> element.
* The <figcaption> element can be placed as the first or last child of the <figure> element.
* Syntax:
* <figure>
* <imgsrc=“” alt=“”>
* <figcaption>….</figcaption>
* </figure>

**Footer Tag:**

* The <footer> tag defines a footer for a document or section.
* A <footer> element should contain information about its containing element.
* A <footer> element typically contains:
* authorship information
* copyright information
* contact information
* sitemap
* back to top links
* related documents
* You can have several <footer> elements in one document.
* Contact information inside a <footer> element should go inside an <address> tag.

**Syntax:**

* <footer>
* <address>….</address>
* </footer>

**Header Tag:**

* The <header> element represents a container for introductory content or a set of navigational links.
* A <header> element typically contains:
* one or more heading elements (<h1> - <h6>)
* logo or icon
* authorship information
* You can have several <header> elements in one document.
* A <header> tag cannot be placed within a <footer>, <address>, <article> or another <header> element.
* Syntax:
* <header>….</header>

**Main Tag:**

* The <main> tag specifies the main content of a document.
* The content inside the <main> element should be unique to the document. It should not contain any content that is repeated across documents such as sidebars, navigation links, copyright information, site logos, and search forms.
* There must not be more than one <main> element in a document. The <main> element must NOT be a descendant of an <article>, <aside>, <footer>, <header>, or <nav> element.

**Syntax:**

* <main>….</main>

**Nav Tag:**

* The <nav> tag defines a set of navigation links.
* Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of **navigation links**.

**Syntax**:

* <nav>
* <a href””>…</a>
* <a href””>…</a>
* <a href””>…</a>
* </nav>

**Section Tag:**

* The <section> tag defines sections in a document, such as chapters, headers, footers, or any other sections of the document.

**Syntax:**

* <section>…</section>

**Div Tag:**

* The <div> tag defines a division or a section in an HTML document.
* The <div> tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.
* The <div> tag is easily styled by using the class or id attribute.
* Any sort of content can be put inside the <div> tag!

**Syntax:**

* <div>…. </div>

**Span Tag:**

* The <span> tag is an inline container used to mark up a part of a text, or a part of a document.
* The <span> tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.
* The <span> tag is much like the [<div>](https://www.w3schools.com/tags/tag_div.asp) element, but <div> is a block-level element and <span> is an inline element.

**Syntax:**

* <span>….</span>

**Program:**

<html>

<head>

<title>HTML5 Tags Demo</title>

</head>

<body>

<header>

<imgsrc="aboutaec.jpg">

</header>

<main>

<nav>

<a href="#">HOME</a>&nbsp;&nbsp;&nbsp;

<a href="#">ABOUT US</a>&nbsp;&nbsp;&nbsp;

<a href="#">GOVERNING BODY</a>&nbsp;&nbsp;&nbsp;

<a href="#">AUTONOMOUS</a>&nbsp;&nbsp;&nbsp;

<a href="#">ACADEMICS</a>&nbsp;&nbsp;&nbsp;

<a href="#">CONTACT US</a>&nbsp;&nbsp;&nbsp;

</nav>

<aside>

<article>

<p>Aditya Engineering College located at surampalem is a well versed

college in East Godavari District.

It is located in a vast area of 200 acres. <br>

Aditya a centre for excellence is striving for the growth of the students

who are the drtiving force of the institution.<br></p>

</article>

</aside>

<figure>

<imgsrc="mickey.png">

<figcaption>A wonderful creation of waltdisney</figcapion>

</figure>

<section>

IT department established in the year 2001 is striving hard for the well

being of the students. Many leaders and achievers were produced from this

department.

</section>

<div>

Mr. M Rajababu the leader of the department is a hard working person who

mingles with the faculty and also understands the problems of the students

and try to solve those problems as quickly as possible.

</diV>

<p>Aditya is spending an overall amount of 25million <span

style="color:red;">$</span> every year</p>

</main>

<footer>

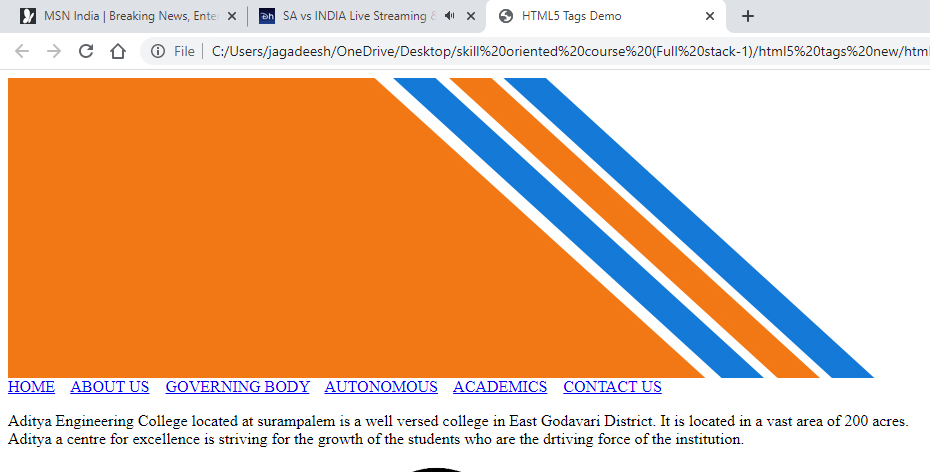
<p align="center">Copyright @Aditya Engineering Coolege, Surampalem</p>

</footer>

</body>

</html>

**Output:**



**b)Write a HTML program, to embed audio and video into HTML web page.**

**Description:**

**Audio Tag:**

* The <audio> tag defines sound, such as music or other audio streams.
* Currently, there are 3 supported file formats for the <audio> element: MP3, Wav, and Ogg
* <audio controls>  
     <source src="" type="audio/mpeg/wav/ogg">  
     
  </audio>

**Video Tag:**

* The <video> tag specifies video, such as a movie clip or other video streams.
* Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg
* Syntax:
* <video height=“pixels” width=“pixels” src=“url” preload=“auto/metadata /none” poster=“url” muted autoplay controls loop>…</video>

**Program:**

<html>

<head>

<title>MultiMedia Content</title>

</head>

<body>

<audio controls>

<source src="master the blaster.mp3" type="audio/mpeg">

</audio><br>

<video width="auto" height="auto" poster="saaho.jpg" src="Saaho Bang.mp4"

controls>

<track src=subtitles\_en.vtt" kind="subtitles" srclang="en" label="English"

default>

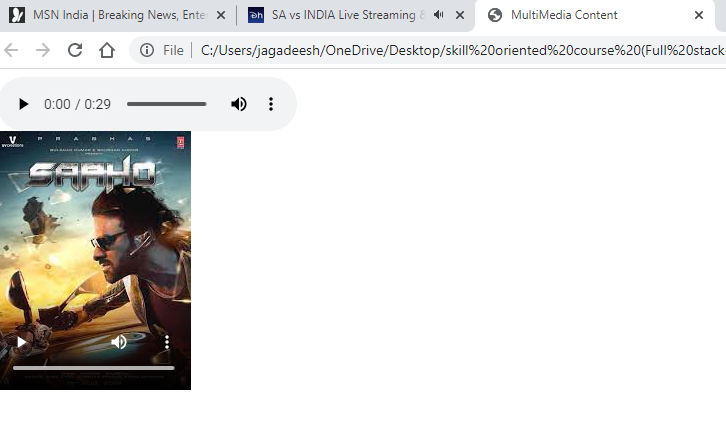
<track kind="description" label="Trailer of Saaho">

</video>

</body>

</html>

**Output:**



|  |  |
| --- | --- |
| **Exp No:8** |  |
| **Cascading Style Sheets, types of CSS, Selector forms** | |

**Write a program to apply different types (or levels of styles or style specification formats) - inline, internal, external styles to HTML elements. (identify selector, property and value).**

**Description:**

**Introduction to CSS:**

* **CSS** stands for **C**ascading **S**tyle **S**heets
* Styles define **how to display** HTML elements
* Styles were added to HTML 4.0 **to solve a problem**
* **External Style Sheets** can save a lot of work
* External Style Sheets are stored in **CSS files**

**Types of CSS:**

* There are three ways of inserting a style sheet:

1. Inline styles
2. Internal style sheet
3. External style sheet

**Inline CSS:**

* **Inline sheets**  can be used to format **only one tag** at a time
* The **inline** cascading style sheet is a kind of style sheet which the styles can be applied to **html tags** only.
* Using inline sheets, we can apply uniform style on tags for the whole document.
* **Disadvantage:** Inline sheet is not much suitable for web page designing because the actual contents of web page are mixed with the presentation.

***Syntax:***

* <Tag style="property : value " >

**Internal CSS:**

* **Advantage** of Internal style sheet comparing with inline sheets, at a time **several** tags can be formatted with **internal** sheets, where as in **inline sheets only one tag** at a time can be formatted.
* **Disadvantage** : when we want to apply style to more than one document at a time then internal sheet of no use.

**Syntax:**

* <head>
* <style type=“text/css”>
* Tagname{
* Tagproperties;
* }
* </style>
* </head>

**External CSS:**

* When we want to apply style to more than one document at a time then external sheets are used.
* Total style elements are defined in a separate document and this document is added to required web page.
* By using this, we can use this style sheets in different web pages. So we can achieve **reusability** by using external sheets.
* The document where all the style formats are placed , should have extension .**css**
* This page can be called in the web page by using **LINK** tag.

***Syntax:***

* <link rel=”stylesheet” type=”text/css’’ href=”sample.css”>
* **rel:** Specifies relationship between documents.
* **type:** indicates which type we are including.
* **href**: indicates style sheet document address.

**Program:**

**Inline CSS:**

<!DOCTYPE html>

<html>

<body>

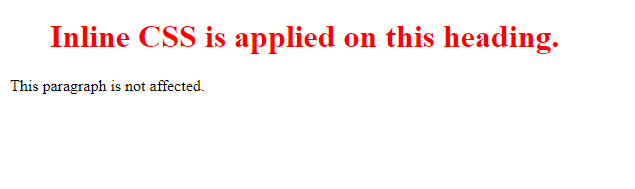
<h1 style="color:red;margin-left:40px;">Inline CSS is applied on this heading.</h1>

<p>This paragraph is not affected.</p>

</body>

</html>

**Output:**



**Internal CSS:**

<html>

<head>

<style>

.nav{

background-color:yellow;

list-syle-type:none;

text-align:center;

margin:0;

padding:0;

}

.navli{

display:inline-block;

font-size:20px;

padding:20px;

}

</style>

</head>

<body>

<ul class="nav">

<li><a href="#home">Home</a></li>

<li><a href="#about us">About Us</a></li>

<li><a href="#achievements">Achievements</a></li>

<li><a href="#clients">Clients</a></li>

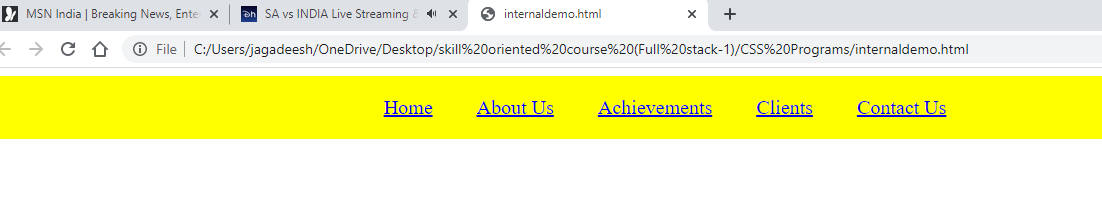
<li><a href="#contact us">Contact Us</a></li>

</ul>

</body>

</html>

**Output:**



**External CSS:**

**Style.css**

.nav{

background-color:yellow;

list-syle-type:none;

text-align:center;

margin:0;

padding:0;

}

.navli{

display:inline-block;

font-size:20px;

padding:20px;

}

**External.html**

<html>

<head>

<link rel=”stylesheet” type=”text/css” href=”style.css”>

</head>

<body>

<ul class="nav">

<li><a href="#home">Home</a></li>

<li><a href="#about us">About Us</a></li>

<li><a href="#achievements">Achievements</a></li>

<li><a href="#clients">Clients</a></li>

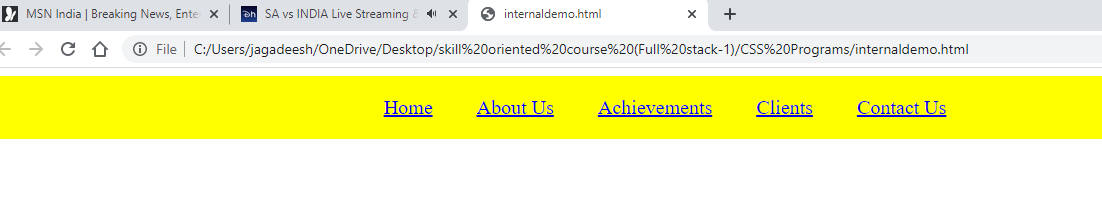
<li><a href="#contact us">Contact Us</a></li>

</ul>

</body>

</html>

**Output:**



|  |  |
| --- | --- |
| **Ex.No: 9** |  |
| **Selectors in CSS** | |

**Write a program to apply different types of selector forms**

1. **Simple selector (element, id, class, group, universal)**
2. **Combinator selector (descendant, child, adjacentsibling, general sibling)**
3. **Pseudo-class selector**
4. **Pseudo-element selector**
5. **Attribute selector**

**Description:**

Selectors are used to apply special effects.

Types of selectors are:

1. Simple selector

2. Class selector

3. Generic Selector

4. Universal selector

5. Id Selector

**Simple Selector:**

* The simple selector form is a single element to which the property and value is applied.
* Syntax:
* <head>
* <title>Simle selectors</title>
* <style type="text/css">
* tagname{
* properrties;
* }
* </style>
* </head>

**Class Selector:**

* Using class selector we can apply different styles to same element.
* Syntax:<head>
* <style type=“text/css”>
* Tagname.classname{
* Properties;
* }
* </style>
* </head>
* <tagname class=“classname”>….</tagname>

**Generic Selector:**

* The class can be defined in the generalized form.
* So that the particular class can be applied to any tag.
* Syntax: <head>
* <style type=“text/css”>
* .classname{
* Properties;
* }
* </style>
* </head>
* <tagname class=“classname”>….</tagname>

**Universal Selector:**

* This selector can be applied to all the elements in the document.
* This selector is denoted by \* symbol.
* Syntax:<head>
* <title>Universal selectors</title>
* <style type="text/css">
* \*
* {
* properties;
* }

**Id Slector:**

* The id selector is used to specify a style for a single, unique element.
* The id selector uses the id attribute of the HTML element, and is defined with a "#".
* Do **not** start an ID name with a number
* **Syntax:**
* #para1  
   {  
  text-align:center;  
  color:red;  
   }
* <tagname id=“idname”> ……</tagname>

**Attribute Slector:**

* The [attribute] selector is used to select elements with a specified attribute.
* The [attribute="value"] selector is used to select elements with a specified attribute and value.
* The [attribute~="value"] selector is used to select elements with an attribute value containing a specified word.
* The [attribute|="value"] selector is used to select elements with the specified attribute starting with the specified value.
* The [attribute^="value"] selector is used to select elements whose attribute value begins with a specified value.
* The [attribute$="value"] selector is used to select elements whose attribute value ends with a specified value.
* The [attribute\*="value"] selector is used to select elements whose attribute value contains a specified value.

**Program:**

**Simple Slector Demo:**

<html>

<head>

<style type="text/css">

p{

font-style:Tahoma;

font-size:40px;

border:5px double #ccc;

}

</style>

<body>

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

<p>All theparagraph's are displayed with the mentioned styles</p><br>

</body>

</html>

**Output:**



**Class Selctor Demo:**

<html>

<head>

<style type="text/css">

p{

font-style:Tahoma;

font-size:40px;

border:5px double #ccc;

}

p.redpara{

color:none;

background-color:red;

border:5px solid green;

}

</style>

<body>

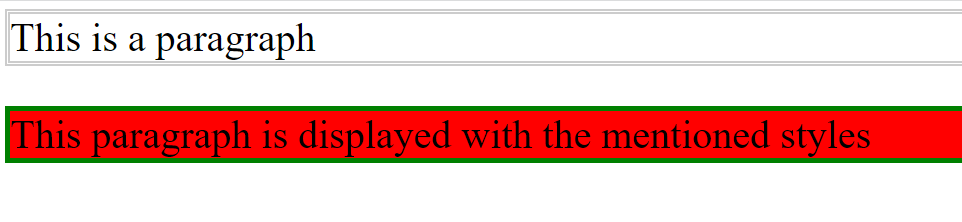
<p>This is a paragraph</p<br>

<p class="redpara">This paragraph is displayed with the mentioned styles</p><br>

</body>

</html>

**Output:**



**Id Selector Demo:**

<html>

<head>

<style type="text/css">

#red{

color:none;

background-color:red;

border:5px solid green;

}

</style>

<body>

<p id="red">This paragraph is displayed with the mentioned styles</p><br>

<div id="red">This division is displayed wit mentioned styles</div><br>

<section id="red">This section is dispalyed with the mentioned styles</section><br>

</body>

</html>

**Output:**



**Generic Slector Demo:**

<html>

<head>

<style type="text/css">

.red{

color:none;

background-color:red;

border:5px solid green;

}

</style>

<body>

<p class="red">This paragraph is displayed with the mentioned styles</p><br>

<div class="red">This division is displayed wit mentioned styles</div><br>

<section class="red">This section is dispalyed with the mentioned styles</section><br>

</body>

</html>

**Output:**



**Universal Slector Demo:**

<html>

<head>

<style type="text/css">

\*{

color:none;

background-color:red;

border:5px solid green;

}

</style>

<body>

<p >This paragraph is displayed with the mentioned styles</p><br>

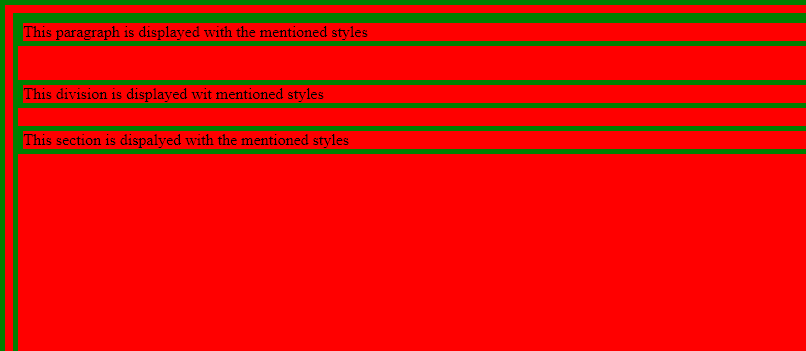
<div>This division is displayed wit mentioned styles</div><br>

<section>This section is dispalyed with the mentioned styles</section><br>

</body>

</html>

**Output:**



**Attribute Selector Demo:**

<!DOCTYPE html>

<html>

<head>

<style>

input[type=text] {

width: 150px;

display: block;

margin-bottom: 10px;

background-color: yellow;

}

input[type=button] {

width: 120px;

margin-left: 35px;

display: block;

}

</style>

</head>

<body>

<form name="input" action="" method="get">

Firstname:<input type="text" name="Name" value="Peter" size="20">

Lastname:<input type="text" name="Name" value="Griffin" size="20">

<input type="button" value="Example Button">

</form>

</body>

</html>

**Output:**



|  |  |
| --- | --- |
| **Ex.No:10** |  |
| **Color, Background and CSS Box Model** | |

**a)Write a program to demonstrate the various ways you can reference a color in CSS.**

**Description:**

**Color Properties in CSS:**

* Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.
* Background-color:colorname/rgb/hex;(applied for background)
* Color:colorname/rgb/hex;(applied for text)
* Border:width in pixels type colorname/rgb/hex (applied for borders)

**Program:**

<!DOCTYPE html>

<html>

<head>

<style>

#p1 {background-color:rgba(255,0,0,0.3);}

#p2 {background-color:rgba(0,255,0,0.3);}

#p3 {background-color:rgba(0,0,255,0.3);}

#p4 {background-color:hsl(120,60%,70%);}

#p5 {background-color:hsl(290,100%,50%);}

#p6 {background-color:hsl(290,60%,70%);}

#p7 {background-color:hsla(120,60%,70%,0.3);}

#p8 {background-color:hsla(290,100%,50%,0.3);}

#p9 {background-color:hsla(290,60%,70%,0.3);}

div {

color: blue;

border: 10px solid currentcolor;

padding: 15px;

}

</style>

</head>

<body>

<p id="p1">Red</p>

<p id="p2">Green</p>

<p id="p3">Blue</p>

<p id="p4">Grey</p>

<p id="p5">Yellow</p>

<p id="p6">Cerise</p>

<p id="p7">Pastel green</p>

<p id="p8">Violet</p>

<p id="p9">Pastel violet</p>

<div>

This div element has a blue text color and a blue border.

</div>

</body>

</html>

**Output:**



**b)Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down.**

**Program:**

!DOCTYPE html>

<html>

<head>

<style>

img {

-webkit-box-reflect: right 20px;

}

</style>

</head>

<body>

<h1>CSS Image Reflection</h1>

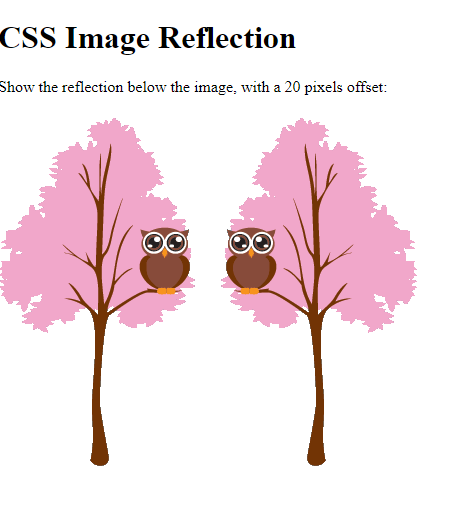
<p>Show the reflection below the image, with a 20 pixels offset:</p>

<imgsrc="./images/img\_tree.gif">

</body>

</html>

**Output:**



|  |  |
| --- | --- |
| **Ex.No:11** |  |
| **CSS Box Model, CSS Font and Image Proerpties** | |

**a)Write a program, to explain the importance of CSS Box model using**

**i. Content**

**ii. Border**

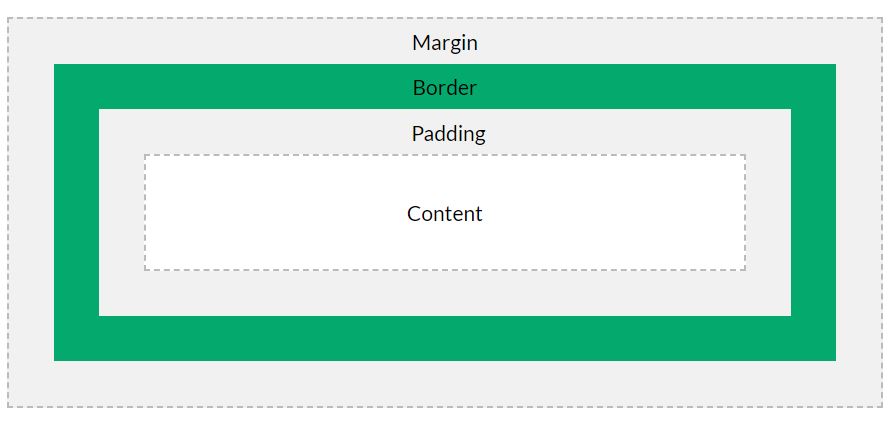
**iii. Margin**

**iv. padding**

**Description:**

**Box Model:**

* In CSS, the term "box model" is used when talking about design and layout.
* The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.
* The image below illustrates the box model:



* **Content** - The content of the box, where text and images appear
* **Padding** - Clears an area around the content. The padding is transparent
* **Border** - A border that goes around the padding and content
* **Margin** - Clears an area outside the border. The margin is transparent
* The box model allows us to add a border around elements, and to define space between elements.
* div {  
    width: 300px;  
    border: 15px solid green;  
    padding: 50px;  
    margin: 20px;  
  }

**Program:**

<!DOCTYPE html>

<html>

<head>

<style>

div {

background-color: lightgrey;

width: 300px;

border: 15px solid green;

padding: 50px;

margin: 20px;

}

</style>

</head>

<body>

<h2>Demonstrating the Box Model</h2>

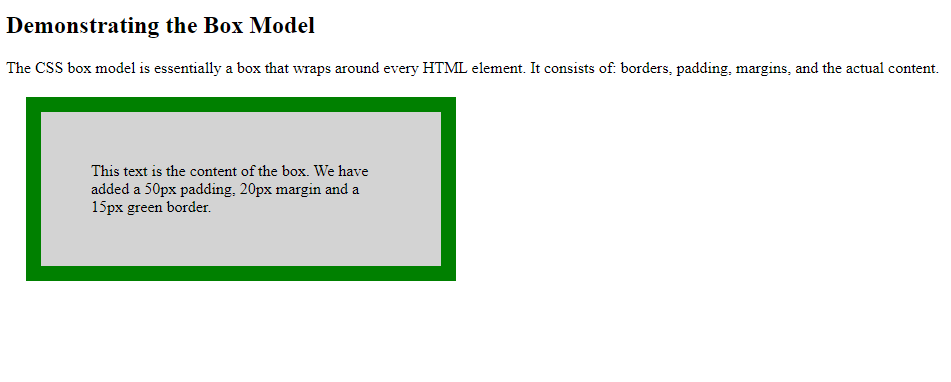
<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>

<div>This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. </div>

</body>

</html>

**Output:**



**b)Write a program using the following terms related to CSS font and text:**

**i. font-size**

**ii. font-weight**

**iii. font-style**

**iv. text-decoration**

**v. text-transformation**

**vi. text-alignment**

**Description:**

**Font Properties in CSS:**

* Using a font that is easy to read is important.
* The font adds value to your text. It is also important to choose the correct color and text size for the font.
* In CSS there are five generic font families:
* **Serif** fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
* **Sans-serif** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
* **Monospace** fonts - here all the letters have the same fixed width. They create a mechanical look.
* **Cursive** fonts imitate human handwriting.
* **Fantasy** fonts are decorative/playful fonts.
* All the different font names belong to one of the generic font families
* In CSS, we use the font-family property to specify the font of a text.
* The font-family property should hold several font names as a "fallback" system, to ensure maximum compatibility between browsers/operating systems.
* Start with the font you want, and end with a generic family (to let the browser pick a similar font in the generic family, if no other fonts are available).
* The font names should be separated with comma.
* font-family: *family-name*|*generic-family*|initial|inherit;
* Font-family:Georgia|PalatinoLinotype|BookAntiqua|Times New Roman|Arial|Helvetica|ArialBlack|Impact|Lucida Sans Unicode|Tahoma|Verdana|CourierNew|LucidaConsole|initial
* In a small-caps font, all lowercase letters are converted to uppercase letters.
* However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.
* The font-variant property specifies whether or not a text should be displayed in a small-caps font.
* font-variant: normal|small-caps|initial|inherit;
* The font-variant-caps property controls the usage of alternate glyphs for capital letters.
* font-variant-caps: normal|small-caps|all-small-caps|petite-caps|all-petite-caps|unicase|titling-caps|initial|inherit|unset;
* The font-size property sets the size of a font.
* font-size:medium|xx-small|x-small|small|large|x-large|xx-large|smaller|larger|*length*|%|initial|inherit;
* The font-style property specifies the font style for a text.
* font-style: normal|italic|oblique|initial|inherit;
* The font-weight property sets how thick or thin characters in text should be displayed.
* font-weight: normal|bold|bolder|lighter|*number*|initial|inherit;

**Text Properties in CSS:**

* CSS has a lot of properties for formatting text
* The color property is used to set the color of the text. The color is specified by:
* a color name - like "red"
* a HEX value - like "#ff0000"
* an RGB value - like "rgb(255,0,0)"
* Look at CSS Color Values for a complete list of possible color values.
* The default text color for a page is defined in the body selector.
* Text-color/color:colorname/rgb/hex/hsl|initial|inherit;
* The text-align property is used to set the horizontal alignment of a text.
* A text can be left or right aligned, centered, or justified.
* When the text-align property is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers)
* Text-align: left|right|center|justify|initial|inherit;
* The text-decoration property is used to set or remove decorations from text.
* The value text-decoration: none; is often used to remove underlines from links
* The other text-decoration values are used to decorate text
* Text-decoration: *text-decoration-line* *text-decoration-color* *text-decoration-style*|initial|inherit;
* The text-decoration-line property sets the kind of text decoration to use (like underline, overline, line-through).
* Text-decoration-line:none|underline|overline|line-through|initial|inherit;
* The text-decoration-style property sets the style of the text decoration (like solid, wavy, dotted, dashed, double).

text-decoration-style: solid|double|dotted|dashed|wavy|initial|inherit;

* The text-transform property is used to specify uppercase and lowercase letters in a text.
* It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word
* Text-transform:none|capitalize|uppercase|lowercase|initial|inherit;
* The text-indent property is used to specify the indentation of the first line of a text
* Text-indent:px|pts|cm|em|initial|inherit;
* The direction property specifies the text direction/writing direction within a block-level element.
* direction: ltr|rtl|initial|inherit;
* The text-overflow property specifies how overflowed content that is not displayed should be signaled to the user. It can be clipped, display an ellipsis (...), or display a custom string.
* Both of the following properties are required for text-overflow:
* text-overflow: clip|ellipsis|*string*|initial|inherit;
* The letter-spacing property is used to specify the space between the characters in a text.
* Letter spacing can be given either a positive or –ve value.
* Letter-spacing:+-normal|px|pts|cm|em|initial|inherit;
* The line-height property is used to specify the space between lines as floating point value
* Line-height:normal|value|length|initial|inherit;
* The word-spacing property is used to specify the space between the words in a text
* word-spacing: normal|px|pts|cm|em|inherit;
* The white-space property specifies how white-space inside an element is handled.
* White-space: normal|nowrap|pre|pre-line|pre-wrap|initial|inherit;
* The unicode-bidi property is used together with the direction property to set or return whether the text should be overridden to support multiple languages in the same document.
* unicode-bidi: normal|embed|bidi-override|initial|inherit;
* The vertical-align property sets the vertical alignment of an element.
* vertical-align: baseline|*length*|sub|super|top|text-top|middle|bottom|text-bottom|initial|inherit;
* The text-shadow property adds shadow to text.
* In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px)
* text-shadow: *h-shadow v-shadow blur-radius color*|none|initial|inherit;

**Program:**

<html>

<head>

<style>

.font{

font-family:Times;

font-style:oblique;

font-weight:900;

font-size:65%;

border:2px solid;

}

.text{

text-align:center;

text-decoration:overline;

text-transform:capitalize;

border:2px solid;

}

</style>

</head>

<body>

<section class="font">

This text is displayed with applied font properties.

</section>

<div class="text">

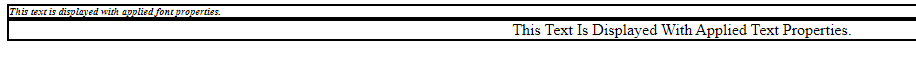
This text is displayed with applied text properties.

</div>

</body>

</html>

**Output:**



|  |  |
| --- | --- |
| **Ex.No:12** |  |
| **Animations**  **a)Write a CSS program, to apply 2D transformations in a web page.** | |

**Description:**

**2D Transforms:**

The CSS 2D transforms are used to re-change the structure of the element as translate, rotate, scale and skew etc.

Following is a list of 2D transforms methods:

* **translate(x,y):** It is used to transform the element along X-axis and Y-axis.
* **translateX(n):** It is used to transform the element along X-axis.
* **translateY(n):** It is used to transform the element along Y-axis.
* **rotate():** It is used to rotate the element on the basis of an angle.
* **scale(x,y):** It is used to change the width and height of an element.
* **scaleX(n):** It is used to change the width of an element.
* **scaleY(n):** It is used to change the height of an element.
* **skewX():** It specifies the skew transforms along with X-axis.
* **skewY():**It specifies the skew transforms along with Y-axis.
* **matrix():** It specifies matrix transforms.

**Program1**

<!DOCTYPE html>

<html>

<head>

<style>

div {

width: 300px;

height: 100px;

background-color: yellow;

border: 1px solid black;

}

div#myDiv {

transform: scaleX(1);

transform: scaleY(1);

transform: skewX(1);

transform: skewY(1);

transform: rotate(-20deg);

}

</style>

</head>

<body>

<div>

This a normal div element.

</div>

<div id="myDiv">

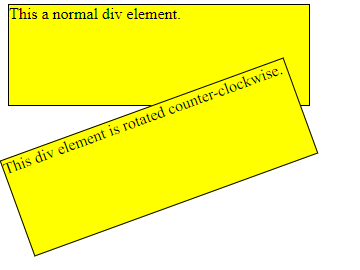
This div element is rotated counter-clockwise.

</div>

</body>

</html>

**Output:**



**b) Write a CSS program, to apply 3D transformations in a web page.**

**Descrription:**

**3D Tranforms:**

The CSS 3D transforms facilitates you to move an element to X-axis, Y-axis and Z-axis. Following is a list of 3D transforms methods:

|  |  |
| --- | --- |
| **Function** | **Description** |
| matrix3D(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n) | It specifies a 3D transformation, using a 4x4 matrix of 16 values. |
| translate3D(x,y,z) | It specifies a 3D translation. |
| translateX(x) | It specifies 3D translation, using only the value for the X-axis. |
| translateY(y) | It specifies 3D translation, using only the value for the Y-axis. |
| translateZ(z) | It specifies 3D translation, using only the value for the Z-axis. |
| scale3D(x,y,z) | It specifies 3D scale transformation |
| scaleX(x) | It specifies 3D scale transformation by giving a value for the X-axis. |
| scaley(y) | It specifies 3D scale transformation by giving a value for the Y-axis. |
| scaleZ(z) | It specifies 3D scale transformation by giving a value for the Z-axis. |
| rotate3D(X,Y,Z,angle) | It specifies 3D rotation along with X-axis, Y-axis and Z-axis. |
| rotateX(angle) | It specifies 3D rotation along with X-axis. |
| rotateY(angle) | It specifies 3D rotation along with Y-axis. |
| rotateZ(angle) | It specifies 3D rotation along with Z-axis. |
| perspective(n) | It specifies a perspective view for a 3D transformed element. |

**Program2**

<!DOCTYPE html>

<html>

<head>

<style>

div {

width: 300px;

height: 100px;

background-color: yellow;

border: 1px solid black;

}

div#myDiv {

transform: scaleX(1);

transform: scaleY(1);

transform: scaleZ(1);

transform: rotateZ(-20deg);

}

</style>

</head>

<body>

<div>

This a normal div element.

</div>

<div id="myDiv">

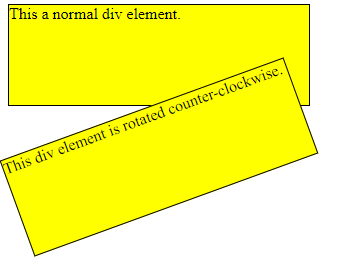
This div element is rotated counter-clockwise.

</div>

</body>

</html>

**Output:**



**c) Write a CSS program, to apply Animations in a web page.**

**Description:**

**Animations:**

An animation makes an element change gradually from one style to another. You can add as many as properties you want to add. You can also specify the changes in percentage.0% specify the start of the animation and 100% specify its completion.

## *How CSS animation works*

When the animation is created in the [@keyframe rule](https://www.javatpoint.com/css-keyframes-rule), it must be bound with selector; otherwise the animation will have no effect.

The animation could be bound to the selector by specifying at least these two properties:

* The name of the animation
* The duration of the animation

## *CSS animation properties*

|  |  |
| --- | --- |
| **Property** | **Description** |
| @keyframes | It is used to specify the animation. |
| animation | This is a shorthand property, used for setting all the properties, except the animation-play-state and the animation-fill- mode property. |
| animation-delay | It specifies when the animation will start. |
| animation-direction | It specifies if or not the animation should play in reserve on alternate cycle. |
| animation-duration | It specifies the time duration taken by the animation to complete one cycle. |
| animation-fill-mode | it specifies the static style of the element. (when the animation is not playing) |
| animation-iteration-count | It specifies the number of times the animation should be played. |
| animation-play-state | It specifies if the animation is running or paused. |
| animation-name | It specifies the name of @keyframes animation. |
| animation-timing-function | It specifies the speed curve of the animation. |

**Program3**

<!DOCTYPE html>

<html>

<head>

<style>

div {

width: 100px;

height: 100px;

background-color: red;

animation-name: example;

animation-duration: 4s;

}

@keyframes example {

0% {background-color: red;}

25% {background-color: yellow;}

50% {background-color: blue;}

100% {background-color: green;}

}

</style>

</head>

<body>

<h1>CSS Animation</h1>

<div></div>

<p><b>Note:</b> When an animation is finished, it goes back to its original style.</p>

</body>

</html>

**Output**

