

Module – 2

HTML Tables and Forms:

- 2.1 Lists,
- 2.2 Tables,
- 2.3 Forms,
- 2.4 Frames
- 2.5 CSS: Introduction,
- 2.6 Levels of style sheets,
- 2.7 Style specification formats,
- 2.8 Selector forms,
- 2.9 Property value forms,
- 2.10 Font properties,
- 2.11 List properties,
- 2.12 Color,
- 2.13 Alignment of text,
- 2.14 The box model,
- 2.15 Background images,
- 2.16 The and <div> tags,
- 2.17 Conflict resolution.

2.1 LISTS

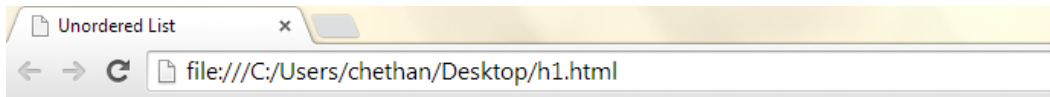
2.1.1 Unordered Lists:

The tag, which is a block tag, creates an unordered list. Each item in a list is specified with an tag (li is an acronym for *list item*). Any tags can appear in a list item, including nested lists. When displayed, each list item is implicitly preceded by a bullet.

```
<html>
<head>
    <title> Unordered List </title>
</head>
<body>
<h1> Some Common Single-Engine Aircraft </h1>
    <ul>
        <li> Cessna skyhawk</li>
        <li> Beechcraft Bonaza</li>
        <li> piper Cherokee</li>
```


</body>

</html>



Some Common Single-Engine Aircraft

- Cessna skyhawk
- Beechcraft Bonaza
- piper Cherokee

2.1.2 Ordered Lists:

- Ordered lists are lists in which the order of items is important. This orderedness of a list is shown in the display of the list by the implicit attachment of a sequential value to the beginning of each item. The default sequential values are Arabic numerals, beginning with 1. An ordered list is created within the block tag .
- The items are specified and displayed just as are those in unordered lists, except that the items in an ordered list are preceded by sequential values instead of bullets.

<html>

<head>

<title> ordered List </title>

</head>

<body>

<h3> Cessna 210 Engine Starting Instructions </h3>

 Set mixture to rich

 Set propeller to high RPM

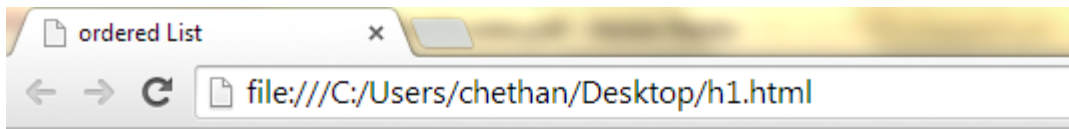
 Set ignition switch to "BOTH"

 Set auxiliary fuel pump switch to "LOW PRIME"

 When fuel pressure reaches 2 to 2.5 PSI, push starter button

</body>

</html>



Cessna 210 Engine Starting Instructions

1. Set mixture to rich
2. Set propeller to high RPM
3. Set ignition switch to "BOTH"
4. Set auxiliary fuel pump switch to "LOW PRIME"
5. When fuel pressure reaches 2 to 2.5 PSI, push starter button

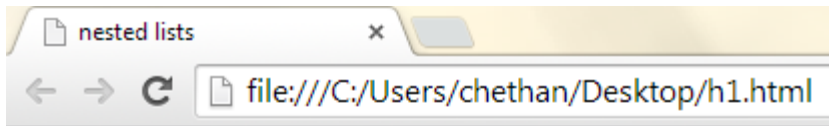
2.1.3 Nested Lists:

```
<html>
<head>
    <title> nested lists </title>
</head>
<ol>
    <li> Information Science </li>
    <ol>
        <li>OOMD</li>
        <li>Java & J2ee</li>
    </ol>
    <ul>
        <li>classes and methods</li>
        <li>exceptions</li>
        <li>applets</li>
        <li>servelets</li>
    </ul>
    <li>Computer Networks</li>
    <ul>
        <li>Part 1</li>
        <li>Part 2</li>
    </ul>
    <li>DBMS</li>
    <li>Operations Research</li>
</ol>
    <li> Computer Science</li>
<ol>
```

```

    <li>Compiler Design</li>
    <li>FLAT</li>
<ul>
    <li>NFA</li>
    <li>DFA</li>
    <li>CFG</li>
</ul>
    <li>Computer Graphics</li>
    <li>Artificial Intelligence</li>
</ol>
</ol>
</html>

```

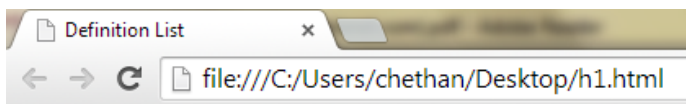


1. Information Science
 1. OOMD
 2. Java & J2ee
 - classes and methods
 - exceptions
 - applets
 - servelets
 3. Computer Networks
 - Part 1
 - Part 2
 4. DBMS
 5. Operations Research
2. Computer Science
 1. Compiler Design
 2. FLAT
 - NFA
 - DFA
 - CFG
 3. Computer Graphics
 4. Artificial Intelligence

2.1.4 Definition Lists:

- As the name implies, definition lists are used to specify lists of terms and their definitions, as in glossaries. A definition list is given as the content of a <dl> tag, which is a block tag.
- Each term to be defined in the definition list is given as the content of a <dt> tag. The definitions themselves are specified as the content of <dd> tags.
- The defined terms of a definition list are usually displayed in the left margin; the definitions are usually shown indented on the line or lines following the term.

```
<html>
<head>
  <title> Definition List </title>
</head>
<body>
  <h3> Single-Engine Cessna Airplanes </h3>
  <dl >
    <dt> 152 </dt>
      <dd> Two-place trainer </dd>
    <dt> 172 </dt>
      <dd> Smaller four-place airplane </dd>
    <dt> 182 </dt>
      <dd> Larger four-place airplane </dd>
    <dt> 210 </dt>
      <dd> Six-place airplane - high performance
    </dd>
  </dl>
</body>
</html>
```



Single-Engine Cessna Airplanes

152	Two-place trainer
172	Smaller four-place airplane
182	Larger four-place airplane
210	Six-place airplane - high performance

2.2 TABLES

A table is a matrix of cells. The cells in the top row often contain column labels, those in the leftmost column often contain row labels, and most of the rest of the cells contain the data of the table. The content of a cell can be almost any document element, including text, a heading, a horizontal rule, an image, and a nested table.

2.2.1 Basic Table Tags:

- A table is specified as the content of the block tag **<table>**.
- There are two kinds of lines in tables: the line around the outside of the whole table is called the *border*; the lines that separate the cells from each other are called *rules*.
- It can be obtained using **border** attribute. The possible values are “border” or any number.
- The table heading can be created using **<caption>** tag.
- The table row can be created using **<tr>** tag.
- The column can be created either by using **<th>** tag (stands for table header which is suitable for headings) or **<td>** tag (stands for table data which is suitable for other data).

```
<html>
<head>
  <title> Table with text and image </title>
</head>
<body>
<table border = "border">
<caption>VTU Memo </caption>
  <tr>
    <th> VTU </th>
    <th> Image </th>
  </tr>
  <tr>
    <td> Funny image </td>
    <td> <img src = "img(13).jpg" alt = "cant display"/></td>
  </tr>
  <tr>
    <td> True Story </td>
    <td> <img src = "img(19).jpg" alt = "cant display"/></td>
  </tr>
</table>
</body>
```

</html>

VTU Memo	
VTU	Image
Funny image	
True Story	

2.2.2 The rowspan and colspan Attributes:

Multiple-level labels can be specified with the rowspan and colspan attributes.

<html>

<head>

<title>row-span and column-span</title>

</head>

```

<body>
  <p> Illustration of Row span</p>
  <table border="border">
    <tr>
      <th rowspan="2"> ATME</th>
      <th>ISE</th>
    </tr>
    <tr>
      <th>CSE</th>
    </tr>
  </table>
  <p> Illustration of Column span</p>
  <table border="border">
    <tr>
      <th colspan="2"> ATME </th>
    </tr>
    <tr>
      <th>ISE</th>
      <th>CSE</th>
    </tr>
  </table>
</body>
</html>

```

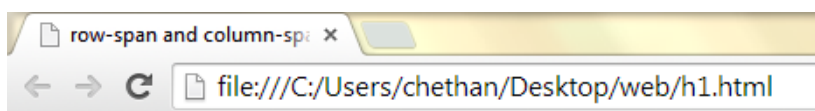


Illustration of Row span

ATME	ISE
	CSE

Illustration of Column span

ATME
ISE CSE

2.2.3 The align and valign Attributes:

- The placement of the content within a table cell can be specified with the align and valign attributes in the <tr>, <th>, and <td> tags.

- The align attribute has the possible values left, right, and center, with the obvious meanings for horizontal placement of the content within a cell.
- The default alignment for th cells is center; for td cells, it is left. The valign attribute of the <th> and <td> tags has the possible values top and bottom.
- The default vertical alignment for both headings and data is center.

```

<html>
<head>
<title> Align and valign </title>
</head>
<body>
<p>Table having entries with different alignments</p>
<table border="border">
  <tr align = "center">
    <th> </th>
    <th> Column Label </th>
    <th> Another One </th>
    <th> Still Another </th>
  </tr>
  <tr>
    <th> Align </th>
    <td align = "left"> Left</td>
    <td align = "center"> Center </td>
    <td align = "right"> right </td>
  </tr>
  <tr>
    <th> <br/>Valign<br/><br/><br/></th>
    <td> Deafult </td>
    <td valign = "top"> Top</td>
    <td valign = "bottom"> Bottom</td>
  </tr>
</table>
</body>
</html>

```

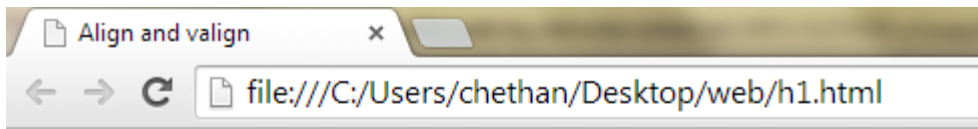


Table having entries with different alignments

	Column Label	Another One	Still Another
Align	Left	Center	right
Valign	Deafult	Top	Bottom

2.2.4 The cellpadding and cellspacing Attributes:

Cellspacing is the distance between cells.

Cellpadding is the distance between the edges of the cell to its content.

```
<html>
<head>
  <title> cell spacing and cell padding </title>
</head>
<body>
  <h3>Table with space = 10, pad = 50</h3>
  <table border = "7" cellspacing = "10" cellpadding = "50">
    <tr>
      <td> Kswamy</td>
      <td>Chethan </td>
    </tr>
  </table>
  <h3>Table with space = 50, pad = 10</h3>
  <table border = "7" cellspacing = "50" cellpadding = "10">
    <tr>
      <td> Divya </td>
      <td>Chethan </td>
    </tr>
  </table>
</body>
</html>
```

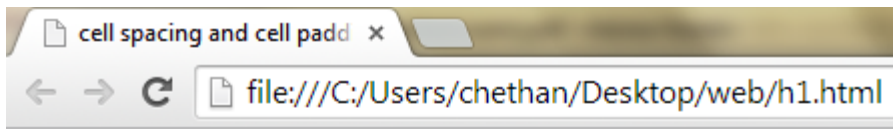


Table with space = 10, pad = 50

Kswamy	Chethan
--------	---------

Table with space = 50, pad = 10

Kswamy	Chethan
--------	---------

2.2.5 Table Sections:

- Tables naturally occur in two and sometimes three parts: header, body, and footer. (Not all tables have a natural footer.)
- These three parts can be respectively denoted in XHTML with the thead, tbody, and tfoot elements.
- The header includes the column labels, regardless of the number of levels in those labels.
- The body includes the data of the table, including the row labels.
- The footer, when it appears, sometimes has the column labels repeated after the body.
- In some tables, the footer contains totals for the columns of data above.
- A table can have multiple body sections, in which case the browser may delimit them with horizontal lines that are thicker than the rule lines within a body section.

2.3 FORMS

The most common way for a user to communicate information from a Web browser to the server is through a form. XHTML provides tags to generate the commonly used objects on a screen form. These objects are called *controls* or *widgets*. There are controls for single-line and multiple-line text collection, checkboxes, radio buttons, and menus, among others. All control tags are inline tags.

2.3.1 The <form> Tag:

All of the controls of a form appear in the content of a <form> tag. A block tag, <form>, can have several different attributes, only one of which, *action*, is required. The *action* attribute specifies the URL of the application on the Web server that is to be called when the user clicks the *Submit* button. Our examples of form elements will not have corresponding application programs, so the value of their *action* attributes will be the empty string ("").

2.3.2 The <input> Tag:

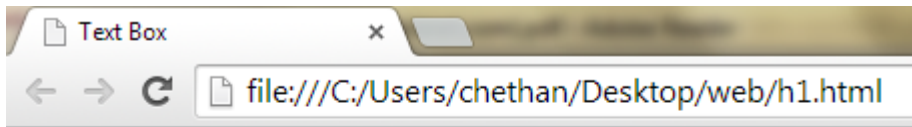
Many of the commonly used controls are specified with the inline tag <input>, including those for text, passwords, checkboxes, radio buttons, and the action buttons *Reset*, *Submit*, and *plain*.

- **Text Box**

- It is a type of input which takes the text.
- Any type of input can be created using <input>
- The *type* attribute indicates what type of input is needed for the text box, the value should be given as text.
- For any type of input, a name has to be provided which is done using *name* attribute.
- The size of the text can be controlled using *size* attribute.
- Every browser has a limit on the number of characters it can collect. If this limit is exceeded, the extra characters are chopped off. To prevent this chopping, *maxlength* attribute can be used. When *maxlength* is used, users can enter only those many characters that is given as a value to the attribute.

```
<html>
<head>
    <title>Text Box</title>
</head>
<body>
<form action = " ">
    <p> <label>Enter your Name:
        <input type = "text" name = "myname" size = "20" maxlength = "20" />
    </label> </p>
```

```
</form>
</body>
</html>
```

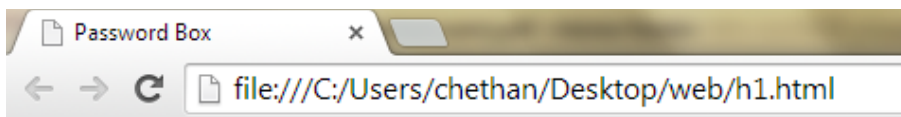


Enter your Name:

2.3.3 Password Box

- If the contents of a text box should not be displayed when they are entered by the user, a password control can be used.
- In this case, regardless of what characters are typed into the password control, only bullets or asterisks are displayed by the browser.

```
<html>
<head>
  <title>Password Box</title>
</head>
<body>
<form action = " ">
  <p> <label>Enter the email id:
    <input type = "text" name = "myname" size = "24" maxlength = "25" /> </label> </p>
  <p> <label>Enter the password:
    <input type = "password" name = "mypass" size = "20" maxlength = "20" />
    </label> </p>
</form>
</body>
</html>
```



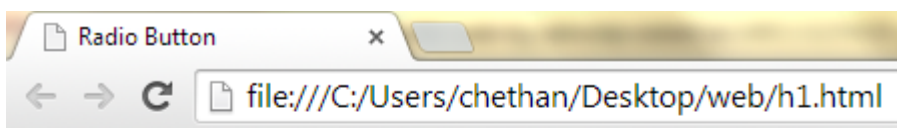
Enter the email id:

Enter the password:

2.3.4 Radio Button

- Radio buttons are special type of buttons which allows the user to select only individual option
- Radio buttons are created using the input tag with the *type* attribute having the value **radio**.
- When radio buttons are created, values must be provided with the help of *value* attribute.
- All the radio buttons which are created would have same name. This is because the radio buttons are group elements.
- If one of the radio buttons has to be selected as soon as the web page is loaded, checked attribute should be used. The value also would be checked.

```
<html>
<head>
  <title>Radio Button</title>
</head>
<body>
  <h3>Age Category ?</h3>
  <form action = " ">
    <p>
      <label><input type="radio" name="age" value="under20" checked = "checked"/>0-19 </label>
      <label><input type="radio" name="age" value="20-35"/>20-35</label>
      <label><input type="radio" name="age" value="36-50"/>36-50 </label>
      <label><input type="radio" name="age" value=" over50"/>over50</label>
    </p>
  </form>
</body>
</html>
```



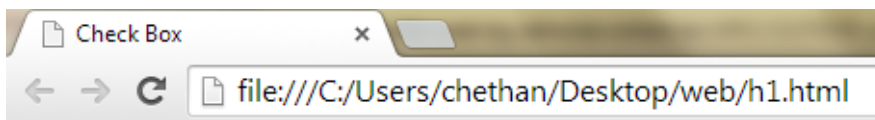
Age Category ?

☒ 0-19 ☐ 20-35 ☐ 36-50 ☐ over50

2.3.5 Check Box

- Check box is a type of input using which multiple options can be selected.
- Check box can also be created using the `<input>` tag with the *type* having the value “checkbox”.
- During the creation of check box, the value should be provided using the *value* attribute.
- All the checkbox which are created would have the same name because they are group elements.
- If one of the check box have to be selected as soon as the page is loaded, checked attribute should be used with the value checked.

```
<html>
<head>
    <title>Check Box</title>
</head>
<body>
    <h3>Grocery Checklist</h3>
    <form action = " ">
        <p>
            <label><input type="checkbox" name="groceries" value="milk" checked="checked"/>Milk</label>
            <label><input type="checkbox" name=" groceries" value="bread"/> Bread </label>
            <label><input type="checkbox" name=" groceries" value="eggs"/>Eggs</label>
        </p>
    </form>
</body>
</html>
```



Grocery Checklist

☒ Milk ☐ Bread ☐ Eggs

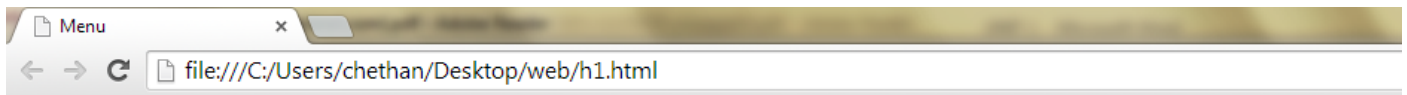
2.3.6 The `<select>` Tag:

- Menu items is another type of input that can be created on the page.
- To create the menu item, `<select>` tag is used.
- To insert the item in the menu, `<option>` tag is used.

```

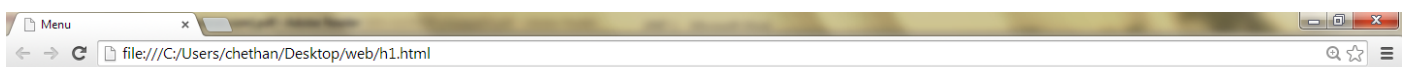
<html>
<head>
    <title> Menu </title>
</head>
<body>
<p> ATME Branches - Information Science, Computer Science, Electronics, Electrical, Mechanical </p>
<form action = "">
    <p> With size = 1 (the default)
        <select name = "branches">
            <option> Information Science </option>
            <option> Computer Science </option>
            <option> Electronics </option>
            <option> Electrical </option>
            <option> Mechanical </option>
        </select>
    </p>
</form>
</body>
</html>

```



ATME Branches - Information Science, Computer Science, Electronics, Electrical, Mechanical

With size = 1 (the default) Information Science ▾

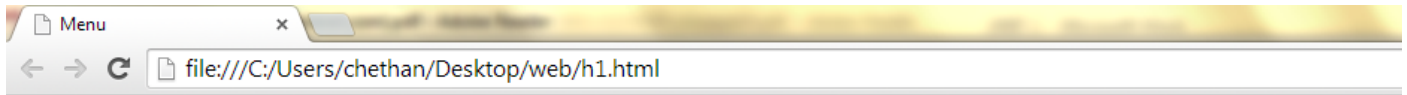


ATME Branches - Information Science, Computer Science, Electronics, Electrical, Mechanical

With size = 1 (the default) Computer Science ▾

- Information Science
- Computer Science
- Electronics
- Electrical
- Mechanical

If you give `<select name = "branches" size = "3">`, then you will get a scroll bar instead of drop down menu. It is as shown in the output given below:



ATME Branches - Information Science, Computer Science, Electronics, Electrical, Mechanical

With size = 1 (the default)

Information Science

Computer Science

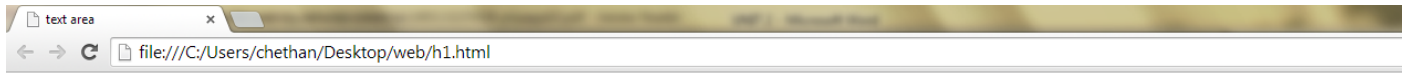
Electronics

Electrical

2.3.7 The <textarea> Tag:

- Text area is a type of input using which multiple statements can be entered.
- Text area is created using <textarea> tag.
- Text area should have the name.
- During the creation of text area, it should be mentioned how many sentences can be entered. This is done using *rows* attribute.
- Similarly, it should also be mentioned how many characters can be entered in a line. This is done using *cols* attribute.
- If the value given to rows is exceeded i.e. if users enter sentences more than specified, the *scroll bar* automatically appears.

```
<html>
<head>
  <title> text area </title>
</head>
<body>
<form action=" ">
  <h3> Enter your comments</h3>
  <p>
    <textarea name="feedback" rows="5" cols="100">
      (Be Brief and concise)
    </textarea>
  </p>
</form>
</body>
</html>
```



Enter your comments

(Be Brief and concise)

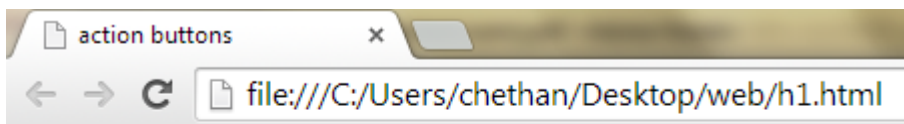
2.3.8 The Action Buttons:

The **Reset** button clears all of the controls in the form to their initial states. The **Submit** button has two actions: First, the form data is encoded and sent to the server; second, the server is requested to execute the server-resident program specified in the action attribute of the `<form>` tag.

The purpose of such a server-resident program is to process the form data and return some response to the user. Every form requires a **Submit** button.

The **Submit** and **Reset** buttons are created with the `<input>` tag.

```
<html>
<head>
    <title> action buttons </title>
</head>
<body>
<form action=" ">
    <p>
        <input type="SUBMIT" value="SUBMIT"/>
        <input type="RESET" value="RESET"/>
    </p>
</form>
</body>
</html>
```

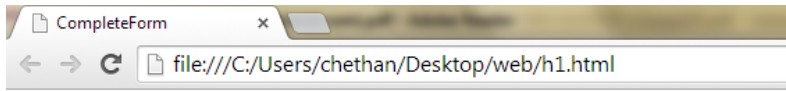


SUBMIT RESET

NOTE: A *plain* button has the type button. *Plain* buttons are used to choose an action.

2.3.9 Example of a Complete Form:

```
<html>
<head>
    <title> CompleteForm</title>
</head> <body>
<h1>Registration Form</h1>
<form action=" ">
<p> <label>Enter your email id:
    <input type = "text" name = "myname" size = "24" maxlength = "25" />
</label> </p>
<p> <label>Enter the password:
    <input type = "password" name = "mypass" size = "20" maxlength = "20" />
</label> </p>
<p>Sex</p>
<p>
    <label><input type="radio" name="act" value="one"/>Male</label>
    <label><input type="radio" name="act" value="two"/>Female</label>
</p>
<p>Which of the following Accounts do you have?</p>
<p>
    <label><input type="checkbox" name="act" value="one"/>Gmail</label>
    <label><input type="checkbox" name="act" value="two"/>Facebook</label>
    <label><input type="checkbox" name="act" value="three"/>Twitter</label>
    <label><input type="checkbox" name="act" value="four"/>Google+</label>
</p>
<p> Any Suggestions?</p>
    <p> <textarea name="feedback" rows="5" cols="100"> </textarea> </p>
    <p>Click on Submit if you want to register</p>
    <p> <input type="SUBMIT" value="SUBMIT"/>
    <input type="RESET" value="RESET"/>
</p>
</form>
</body>
</html>
```



Registration Form

Enter your email id:

Enter the password:

Sex

☒ Male ☐ Female

Which of the following Accounts do you have?

☒ Gmail ☒ Facebook ☐ Twitter ☐ Google+

Any Suggestions?

Click on Submit if you want to register

SUBMIT

RESET

2.4 FRAMES

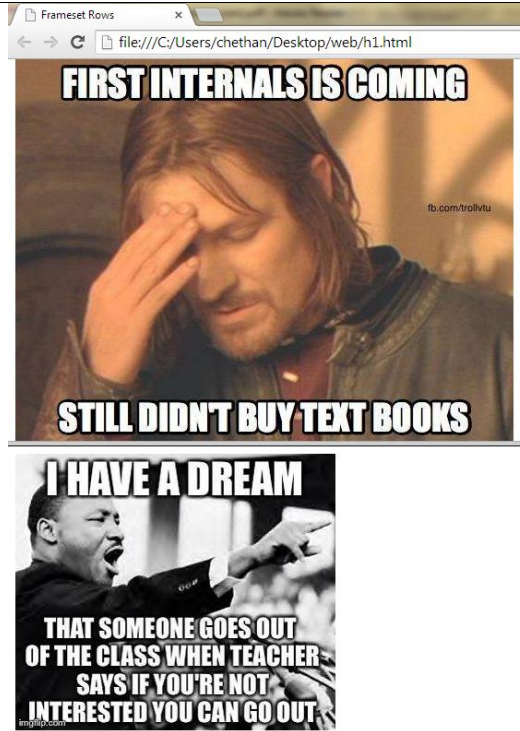
The browser window can be used to display more than one document at a time. The window can be divided into rectangular areas, each of which is a *frame*. Each frame is capable of displaying its own document.

2.4.1 Framesets:

- The number of frames and their layout in the browser window are specified with the `<frameset>` tag.
- A frameset element takes the place of the body element in a document. A document has either a body or a frameset but cannot have both.
- The `<frameset>` tag must have either a *rows* or a *cols* attribute. (or both)
- To create horizontal frames, *rows* attribute is used.
- To create vertical frames, *cols* attribute is used.
- The values for these attributes can be numbers, percentages and asterisks.
- Two or more values are separated by commas & given in quoted string.

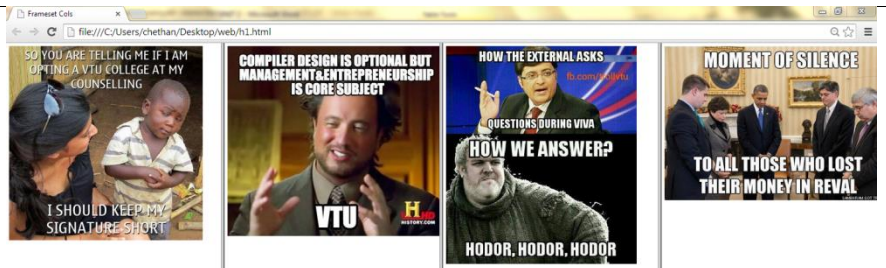
To Demonstrate Horizontal Frames using *rows* Attribute

```
<html>
<head>
<title>Frameset Rows</title>
</head>
<frameset rows = "*,*">
<frame src = "Framerow1.html"/>
<frame src = ""Framerow2.html"/>
</frameset>
</html>
```



To Demonstrate Vertical Frames using *cols* Attribute

```
<html>
<head>
<title>Frameset Cols</title>
</head>
<frameset cols
="25%,25%,25%,25%">
<frame src = "FrameCol1.html"/>
<frame src = "FrameCol2.html"/>
<frame src = "FrameCol3.html"/>
<frame src = "FrameCol4.html"/>
</frameset>
</html>
```



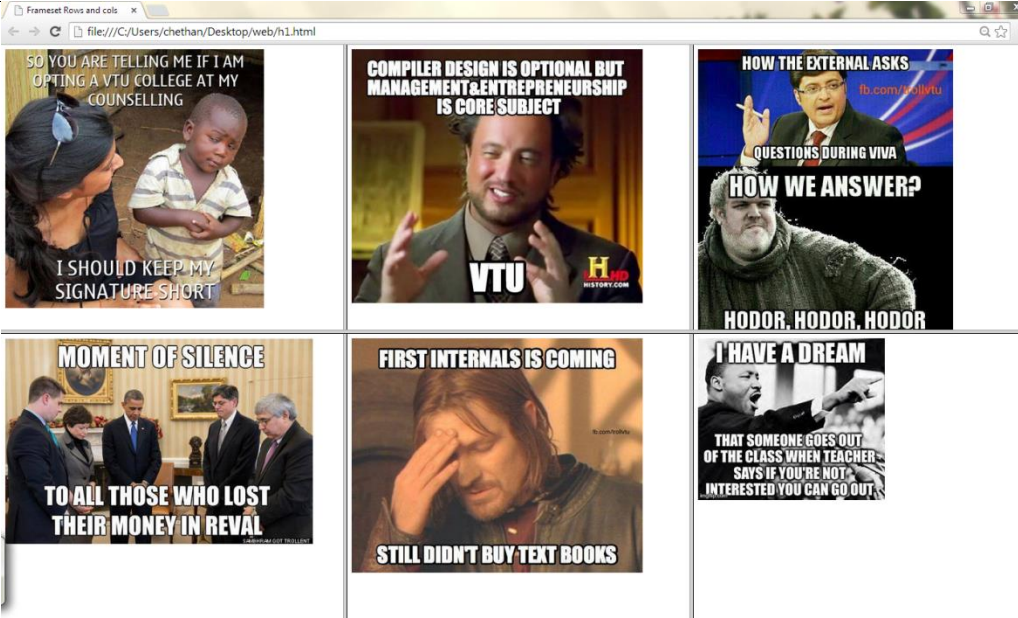
Note: Here, the programs FrameRow1.html, FrameRow2.html, FrameCol1.html, FrameCol2.html, FrameCol3.html, FrameCol4.html are programs to display images. They must be coded separately.

<pre> <html> <head> <title>frame row 1</title> </head> <body> </body> </html> </pre>	<pre> <html> <head> <title>frame col 1</title> </head> <body> </body> </html> </pre>	<pre> <html> <head> <title>frame col 3</title> </head> <body> </body> </html> </pre>
<pre> <html> <head> <title>frame row 2</title> </head> <body> </body> </html> </pre>	<pre> <html> <head> <title>frame col 3</title> </head> <body> </body> </html> </pre>	<pre> <html> <head> <title>frame col 4</title> </head> <body> </body> </html> </pre>

```

<html>
<head>
<title>Frameset Rows and cols</title>
</head>
<frameset rows = "50,50" cols =
"*,*,*">
<frame src = "FrameCol1.html"/>
<frame src = "FrameCol2.html"/>
<frame src = "FrameCol3.html"/>
<frame src = "FrameCol4.html"/>
<frame src = "FrameRow1.html"/>
<framesrc = "FrameRow2.html"/>
</frameset>
</html>

```



2.4.2 SYNTACTIC DIFFERENCES BETWEEN HTML AND XHTML

PARAMETERS	HTML	XHTML
Case Sensitivity	Tags and attributes names are case insensitive	Tags and attributes names must be in lowercase
Closing tags	Closing tags may be omitted	All elements must have closing tag
Quoted attribute values	Special characters are quoted. Numeric values are rarely quoted.	All attribute values must be quoted including numbers
Explicit attribute values	Some attribute values are implicit. For example: <table border>. A default value for border is assumed	All attribute values must be explicitly stated
id and name attributes	Both <i>id</i> and <i>name</i> attributes are encouraged	Use of <i>id</i> is encouraged and use of <i>name</i> is discouraged
Element nesting	Rules against improper nesting of elements (for example: a form element cannot contain another form element) are not enforced.	All nesting rules are strictly enforced

2.5 CSS: Introduction:

XHTML style sheets are called *cascading* style sheets because they can be defined at three different levels to specify the style of a document. Lower level style sheets can override higher level style sheets, so the style of the content of a tag is determined, in effect, through a cascade of style-sheet applications.

2.6 Levels of style sheets:

- The three levels of style sheets, in order from lowest level to highest level, are inline, document level, and external.
- **Inline style sheets** apply to the content of a single XHTML element.
- **Document-level style sheets** apply to the whole body of a document.
- **External style sheets** can apply to the bodies of any number of documents.
- Inline style sheets have precedence over document style sheets, which have precedence over external style sheets.
- Inline style specifications appear within the opening tag and apply only to the content of that tag.
- Document-level style specifications appear in the document head section and apply to the entire body of the document.
- External style sheets stored separately and are referenced in all documents that use them.
- External style sheets are written as text files with the MIME type text/css.
- They can be stored on any computer on the Web. The browser fetches external style sheets just as it fetches documents.

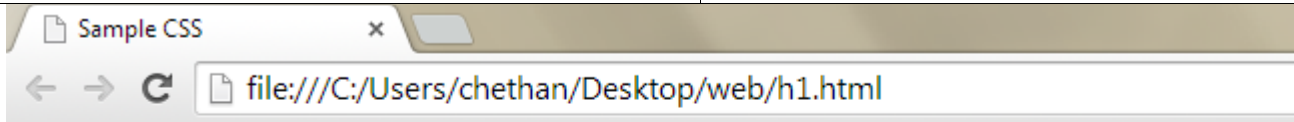
- The <link> tag is used to specify external style sheets. Within <link>, the rel attribute is used to specify the relationship of the linked-to document to the document in which the link appears. The href attribute of <link> is used to specify the URL of the style sheet document.

EXAMPLE WHICH USES EXTERNAL STYLE SHEET

```
<html>
<head>
<title>Sample CSS</title>
<link rel = "stylesheet" type = "text/css" href =
"Style1.css" />
</head>
<h1>Kendaganna swamy</h1>
</html>
```

Style1.css

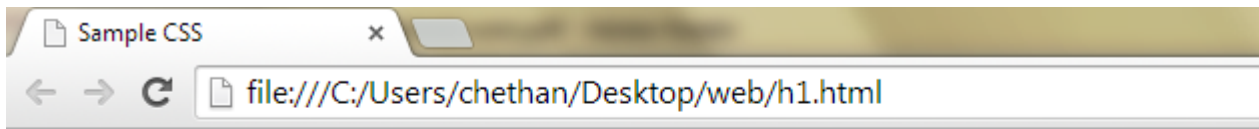
```
h1
{
    font-family: 'Lucida Handwriting';
    font-size: 50pt;
    color: Red;
}
```



Kendaganna swamy

EXAMPLE WHICH USES DOCUMENT LEVEL STYLE SHEET

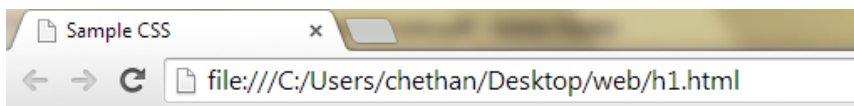
```
<html>
<head>
    <title>Sample CSS</title>
    <style type = "text/css">
h1
{
    font-family: 'Lucida Handwriting';
    font-size: 50pt;
    color: Red;
}
</style>
</head>
    <h1>Kendaganna swamy</h1>
</html>
```

Kendaganna swamy

EXAMPLE WHICH USES INLINE STYLE SHEET

```
<html>
<head>
<title>Sample CSS</title>
</head>
<h1 style="font-family: 'Lucida Handwriting';
font-size: 50pt;
color: Red;"> Chethan Kswamy</h1>
</html>
```



Chethan Kswamy

2.7 STYLE SPECIFICATION FORMATS

Inline Style Specification:

Style = "Property1 : Value1; Property2 : Value2; Property3 : Value3; Property_n : Value_n;"

Document Style Specification:

`<style type = "text/css"> Rule list </style>` Each style rule in a rule list has two parts: a selector, which indicates the tag or tags affected by the rule, and a list of property–value pairs. The list has the same form as the quoted list for inline style sheets, except that it is delimited by braces rather than double quotes. So, the form of a style rule is as follows:

Selector { *Property1 : Value1; Property2 : Value2; Property3 : Value3; Property_n : Value_n;* }

[For examples on all three levels of style sheets along with specifications, Please refer the previous examples] .

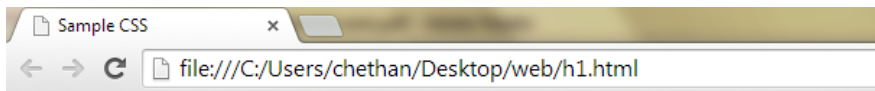
2.8.SELECTOR FORMS

Simple Selector Forms:

In case of simple selector, a tag is used. If the properties of the tag are changed, then it reflects at all the places when used in the program.

The selector can be any tag. If the new properties for a tag are not mentioned within the rule list, then the browser uses default behaviour of a tag.

```
<html>
<head>
    <title>Sample CSS</title>
    <style type = "text/css">
p { font-family: 'Lucida Handwriting'; font-size: 50pt; color: Red; }
</style>
</head>
    <body>
        <p>Kendaganna Swamy </p>
        <p>Sunil </p>
        <p>Siddiq shariff</p>
    </body>
</html>
```



Kendaganna Swamy

Sunil

Siddiq shariff

Class Selectors:

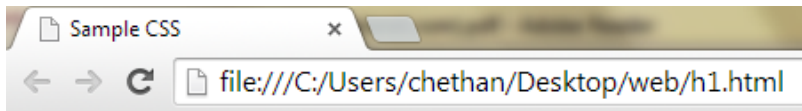
In class selector, it is possible to give different properties for different elements

```
<html>
<head>
    <title>Sample CSS</title>
    <style type = "text/css">
        p.one { font-family: 'Lucida Handwriting'; font-size: 25pt; color: Red; }
```

```

        p.two { font-family: 'Monotype Corsiva'; font-size: 50pt; color: green; }
</style>
</head>
    <body>
        <p class = "one">Kendaganna Swamy</p>
        <p class = "two">Sunil</p>
    </body>
</html>

```



Kendaganna Swamy

Sunil

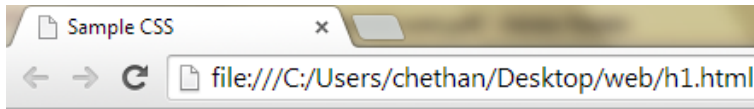
Generic Selectors:

In case of generic selector, when the class is created, it would not be associated to any particular tag. In other words, it is generic in nature.

```

<html>
<head>
    <title>Sample CSS</title>
    <style type = "text/css">
        .one { font-family: 'Monotype Corsiva'; color: green; }
    </style>
</head>
    <body>
        <p class = "one">KSwamy</p>
        <h1 class = "one">Sunil</h1>
        <h6 class = "one"> Siddiq </h6>
    </body>
</html>

```



KSwamy

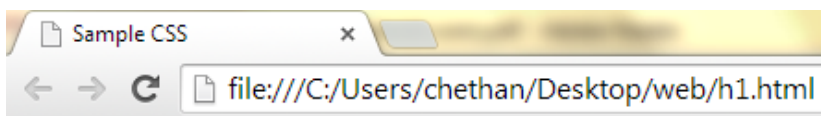
Sunil

Siddiq

id Selectors:

An id selector allows the application of a style to one specific element.

```
<html>
<head>
<title>Sample CSS</title>
<style type = "text/css">
    #one { font-family: 'lucida calligraphy'; color: purple; }
    #two { font-family: 'comic sans ms'; color: orange; }
</style>
</head>
<body>
    <p id = "two">Kswamy</p>
    <h1 id = "one">Sunil</h1>
</body>
</html>
```



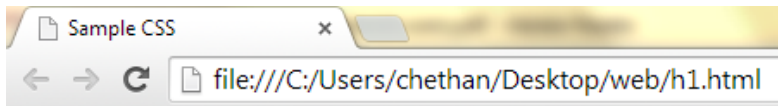
Kswamy

Sunil

Universal Selectors:

The universal selector, denoted by an asterisk (*), applies its style to all elements in a document.

```
<html>
<head>
<title>Sample CSS</title>
<style type = "text/css">
    * { font-family: 'lucida calligraphy'; color: purple; }
</style>
</head>
<body>
    <h1>Kswamy</h1>
    <h2>Sunil</h2>
    <h3>Siddiq</h3>
    <p>Gagana</p>
</body>
</html>
```



Kswamy

Sunil

Siddiq

Gagana

Pseudo Classes:

Pseudo class selectors are used if the properties are to be changed dynamically.

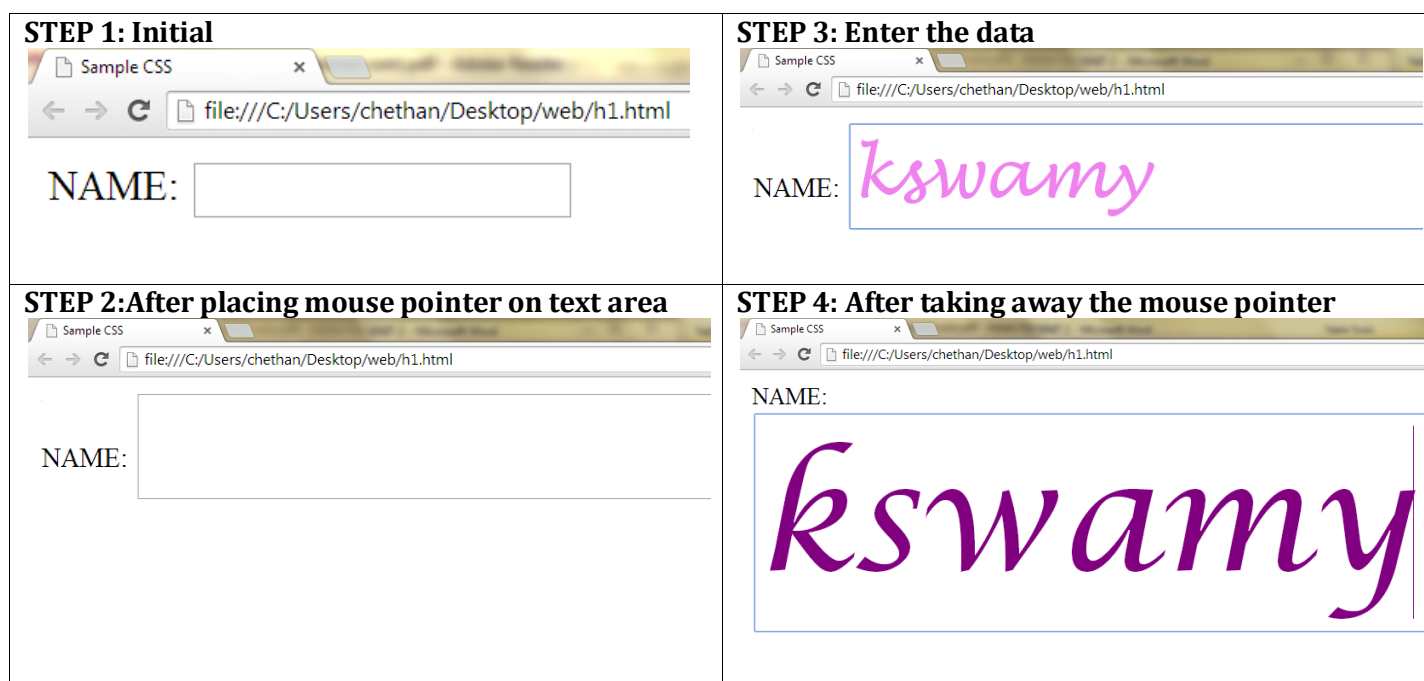
For example: when mouse movement happens, in other words, hover happens or focus happens.

```
<html>
<head>
    <title>Sample CSS</title>
<style type = "text/css">
```

```

input:focus { font-family: 'lucida calligraphy'; color: purple; font-size:100; }
input:hover { font-family: 'lucida handwriting'; color: violet; font-size:40; }
</style>
</head>
<body>
    <form action = " ">
        <p>
            <label> NAME: <input type = "text" />
            </label>
        </p>
    </form>
</body>
</html>

```



2.9 PROPERTY VALUE FORMS

CSS includes 60 different properties in seven categories: fonts, lists, alignment of text, margins, colours, backgrounds, and borders. Property values can appear in a variety of forms.

- Keyword property values are used when there are only a few possible values and they are predefined.
- A number value can be either an integer or a sequence of digits with a decimal point and can be preceded by a sign (+ or -).
- Length values are specified as number values that are followed immediately by a two-character abbreviation of a unit name. The possible unit names are px, for pixels; in, for inches; cm, for centimeters; mm, for millimeters; pt, for points.

- Percentage values are used to provide a measure that is relative to the previously used measure for a property value. Percentage values are numbers that are followed immediately by a percent sign (%). Percentage values can be signed. If preceded by a plus sign, the percentage is added to the previous value; if negative, the percentage is subtracted.
- There can be no space between url and the left parenthesis.
- Color property values can be specified as color names, as six-digit hexadecimal numbers, or in RGB form. RGB form is just the word rgb followed by a parenthesized list of three numbers that specify the levels of red, green, and blue, respectively. The RGB values can be given either as decimal numbers between 0 and 255 or as percentages. Hexadecimal numbers must be preceded with pound signs (#), as in #43AF00.

2.10 FONT PROPERTIES

Font Families:

The font-family property is used to specify a list of font names. The browser uses the first font in the list that it supports. For example, the property:

font-family: Arial, Helvetica, Futura

tells the browser to use Arial if it supports that font. If not, it will use Helvetica if it supports it. If the browser supports neither Arial nor Helvetica, it will use Futura if it can. If the browser does not support any of the specified fonts, it will use an alternative of its choosing. If a font name has more than one word, the whole name should be delimited by single quotes, as in the following example:

font-family: 'Times New Roman'

Font Sizes:

The font-size property does what its name implies. For example, the following property specification sets the font size for text to 10 points:

font-size: 10pt

Many relative font-size values are defined, including xx-small, x-small, small, medium, large, x-large, and xx-large. In addition, smaller or larger can be specified. Furthermore, the value can be a percentage relative to the current font size.

Font Variants:

The default value of the font-variant property is normal, which specifies the usual character font. This property can be set to small-caps to specify small capital characters. These characters are all uppercase, but the letters that are normally uppercase are somewhat larger than those that are normally lowercase.

Font Styles:

The font-style property is most commonly used to specify italic, as in

font-style: italic

Font Weights:

The font-weight property is used to specify the degree of boldness, as in

font-weight: bold

Besides bold, the values normal, bolder, and lighter can be specified. Specific numbers also can be given in multiples of 100 from 100 to 900, where 400 is the same as normal and 700 is the same as bold.

Font Shorthands:

If more than one font property must be specified, the values can be stated in a list as the value of the font property. The order in which the property values are given in a font value list is important. The order must be as follows: The font names must be last, the font size must be second to last, and the font style, font variant, and font weight, when they are included, can be in any order but must precede the font size and font names.

font: bold 14pt 'Times New Roman'

```
<html>
```

```
<head>
```

```
  <title>Font Properties</title>
```

```
<style type = "text/css">
```

```
p.one
```

```
{
    font-family: 'lucida calligraphy';
    font-weight:bold;
    font-size:75pt;
    color: purple;
}
```

```
h1.two
```

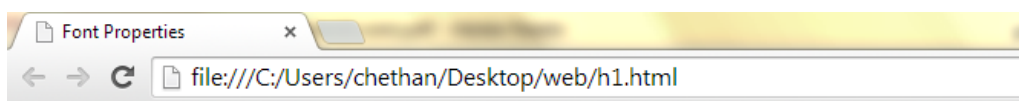
```
{
    font-family: 'cambria';
    color: violet;
    font-style:italics;
}
```



```

p.three
{
    font: small-caps italic bold 50pt 'times new roman'
}
</style>
</head>
<body>
    <p class = "one">Kswamy Chethan</p>
    <h1 class = "two">Sunil Kumar </h1>
    <p class = "three">Siddiq Shariff</p>
</body>
</html>

```



Kswamy Chethan

Sunil Kumar

SIDDIQ SHARIFF

Text Decoration:

The text-decoration property is used to specify some special features of text.

The available values are line-through, overline, underline, and none, which is the default.

```

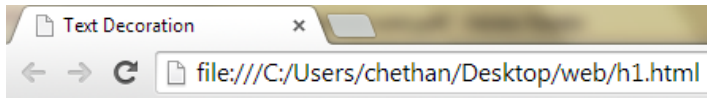
<html>
<head>
    <title>Text Decoration</title>
    <style type = "text/css">
h1.one {text-decoration: line-through;}
h1.two {text-decoration: overline;}
h1.three {text-decoration: underline;}
</style>
</head>

```

```

<body>
    <h1 class = "one">Kswamy Chethan </h1>
    <p>[This is line-through]</p><br/>
    <h1 class = "two"> Sunil Kumar </h1>
    <p>[This is overline]</p><br/>
    <h1 class = "three"> Siddiq Shariff </h1>
    <p>[This is underline]</p><br/>
</body>
</html>

```



~~Kswamy Chethan~~

[This is line-through]

Sunil Kumar

[This is overline]

Siddiq Shariff

[This is underline]

2.11 LIST PROPERTIES

Two presentation details of lists can be specified in XHTML documents: the shape of the bullets that precede the items in an unordered list and the sequencing values that precede the items in an ordered list. The list-style-type property is used to specify both of these. The **list-style-type** property of an unordered list can be set to disc, circle, square, or none.

```

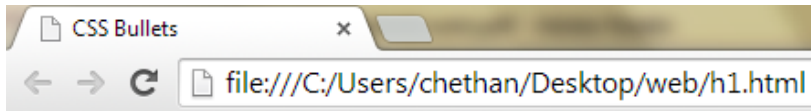
<html>
<head>
    <title>CSS Bullets</title>
    <style type = "text/css">
        li.one { list-style-type:disc}
        li.two{ list-style-type:square}
        li.three{ list-style-type:circle}
    </style>
</head>

```

```

<body>
<h3>Crazy Boy's </h3>
    <ul>
        <li class = "one"> Kendeganna Swamy</li>
        <li class = "two"> Sunil Kumar</li>
        <li class = "three"> Siddiq Shariff</li>
    </ul>
</body>
</html>

```



Crazy Boy's

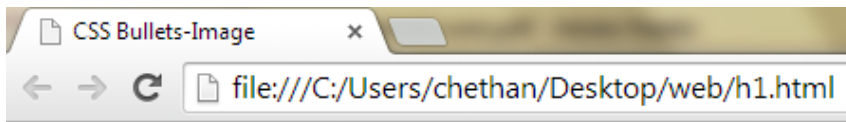
- Kendeganna Swamy
- Sunil Kumar
- Siddiq Shariff

Bullets in unordered lists are not limited to discs, squares, and circles. Any image can be used in a list item bullet. Such a bullet is specified with the `list-style-image` property, whose value is specified with the url form.

```

<html>
<head>
    <title>CSS Bullets-Image</title>
    <style type = "text/css">
li.image {list-style-image: url(bullet.png); font-size:25pt;}
    </style>
</head>
<body>
<h1>Crazy Boy's </h3>
    <ul>
        <li class = "image"> Kendeganna Swamy</li>
        <li class = "image "> Sunil Kumar</li>
        <li class = "image"> Siddiq Shariff</li>
    </ul>
</body>
</html>

```



Crazy Boy's



Kendeganna Swamy



Sunil Kumar



Siddiq Shariff

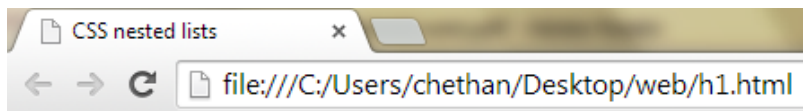
The following example illustrates the use of different sequence value types in nested lists:

```
<html>
<head>
<title> CSS nested lists </title>
<style type = "text/css">
    ol { list-style-type:upper-roman;}
    ol ol {list-style-type:upper-alpha;}
    ol ol ol {list-style-type:decimal;}
</style>
</head>
    <ol>
        <li> Information Science </li>
    </ol>
<li>OOMD</li>
<li>Java & J2ee</li>
    <ol>
        <li>classes and methods</li>
        <li>exceptions</li>
        <li>applets</li>
        <li>servelets</li>
    </ol>
<li>Computer Networks</li>
    <ol>
        <li>Part 1</li>
        <li>Part 2</li>
    </ol>
<li>DBMS</li>
```

```

<li>Operations Research</li>
</ol>
<li> Computer Science</li>
  <ol>
    <li>Compiler Design</li>
    <li>FLAT</li>
  </ol>
<li>NFA</li>
<li>DFA</li>
<li>CFG</li>
</ol>
  <li>Computer Graphics</li>
  <li>Artificial Intelligence</li>
</ol>
</ol>
</html>

```



- I. Information Science
 - A. OOMD
 - B. Java & J2ee
 - 1. classes and methods
 - 2. exceptions
 - 3. applets
 - 4. servelets
 - C. Computer Networks
 - 1. Part 1
 - 2. Part 2
 - D. DBMS
 - E. Operations Research
- II. Computer Science
 - A. Compiler Design
 - B. FLAT
 - 1. NFA
 - 2. DFA
 - 3. CFG
 - C. Computer Graphics
 - D. Artificial Intelligence

2.12. COLOR

Color Groups:

Three levels of collections of colours might be used by an XHTML document. The smallest useful set of colours includes only those that have standard names and are guaranteed to be correctly displayable by all browsers on all color monitors. This collection of 17 colours is called the *named colours*.

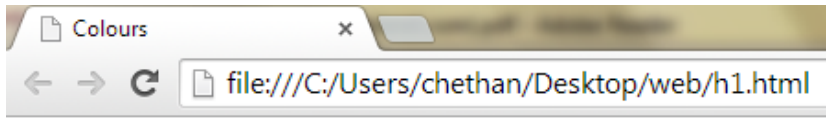
Name	Hexadecimal Code	Name	Hexadecimal Code
aqua	00FFFF	olive	808000
black	000000	orange	FFA500
blue	0000FF	purple	800080
fuchsia	FF00FF	red	FF0000
gray	808080	silver	C0C0C0
green	008000	teal	008080
lime	00FF00	white	FFFFFF
maroon	800000	yellow	FFFF00
navy	000080		

Larger set of colors, called the Web palette, consists of 216 colors. The colors of the Web palette can be viewed at http://www.web-source.net/216_color_chart.htm

Color Properties:

The color property is used to specify the foreground color of XHTML elements.

```
<html>
<head>
    <title>Colours</title>
<style type = "text/css">
    p.one { color: pink; }
    p.two { color: # 9900FF; }
    p.three { background-color:#99FF00;}
</style>
</head>
    <body>
        <p class = "one">Kendaganna Swamy</p>
        <p class = "two">Sunil Kumar </p>
        <p class = "three">Siddiq Shariff</p>
    </body>
</html>
```



Kendaganna Swamy

Sunil Kumar

Siddiq Shariff

2.13 ALIGNMENT OF TEXT

The text-indent property can be used to indent the first line of a paragraph. This property takes either a length or a percentage value. The text-align property, for which the possible keyword values are left, center, right, and justify, is used to arrange text horizontally.

The float property is used to specify that text should flow around some element, often an image or a table. The possible values for float are left, right, and none, which is the default.

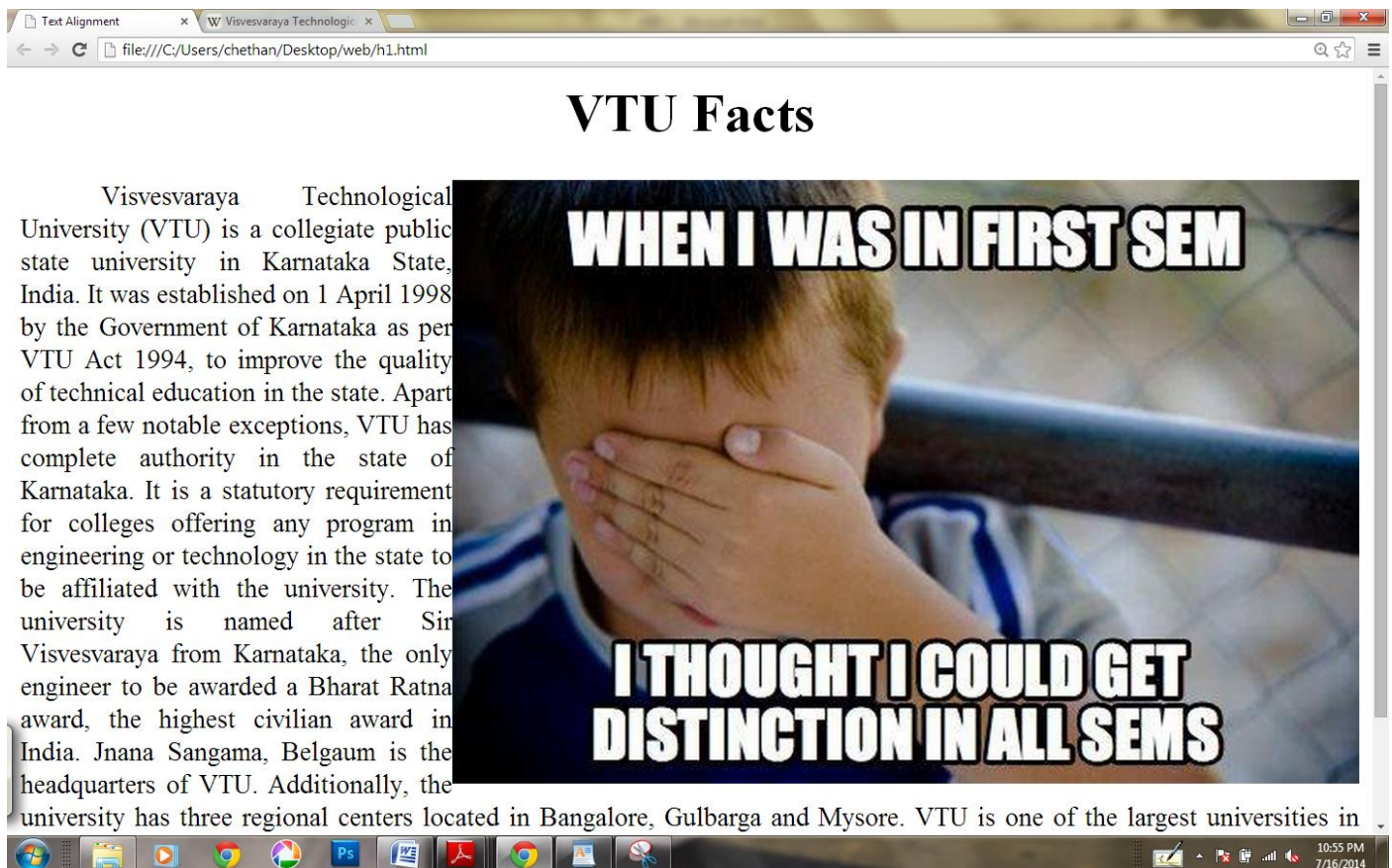
```
<html>
<head>
<title>Text Alignment</title>
<style type = "text/css">
h1.one {text-align: center}
p.two {text-indent: 0.5in; text-align: justify;}
img{float:right}
</style>
</head>
<body>
<h1 class = "one">VTU Facts</h1>
<p>
<img src = "img19.jpg" alt="error"/>
</p>
<p class = "two">Visvesvaraya Technological University (VTU) is a collegiate public state university in
Karnataka State, India. It was established on 1 April 1998 by the Government of Karnataka as per VTU Act
1994, to improve the quality of technical education in the state. Apart from a few notable exceptions, VTU
has complete authority in the state of Karnataka. It is a statutory requirement for colleges offering any
program in engineering or technology in the state to be affiliated with the university.
```

The university is named after Sir Visvesvaraya from Karnataka, the only engineer to be awarded a Bharat Ratna award, the highest civilian award in India. Jnana Sangama, Belgaum is the headquarters of VTU. Additionally, the university has three regional centers located in Bangalore, Gulbarga and Mysore.

VTU is one of the largest universities in India with 208 colleges affiliated to it with an intake capacity of over 67100 undergraduate students and 12666 postgraduate students. The university encompasses various technical & management fields which offers a total of 30 undergraduate and 71 postgraduate courses. The university has around 1800 PhD candidates.

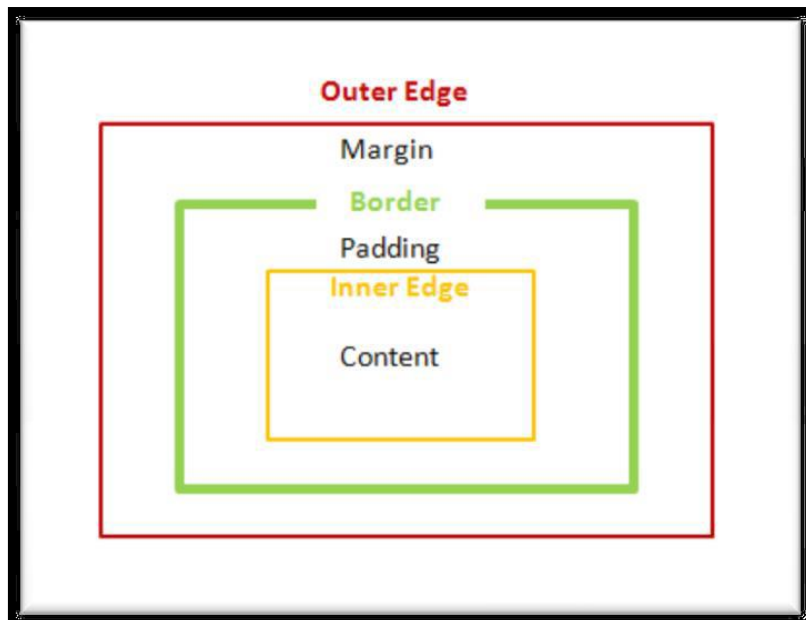
</body>

</html>



2.14 THE BOX MODEL

- On a given web page or a document, all the elements can have borders.
- The borders have various styles, color and width.
- The amount of space between the content of the element and its border is known as *padding*.
- The space between border and adjacent element is known as *margin*.



Borders:

Border-style

- It can be dotted, dashed, double
- Border-top-style
- Border-bottom-style
- Border-left-style
- Border-right-style

Border-width

- It can be thin, medium, thick or any length value
- Border-top-width
- Border-bottom-width
- Border-left-width
- Border-right-width

Border-color

- Border-top-color
- Border-bottom-color
- Border-left-color
- Border-right-color

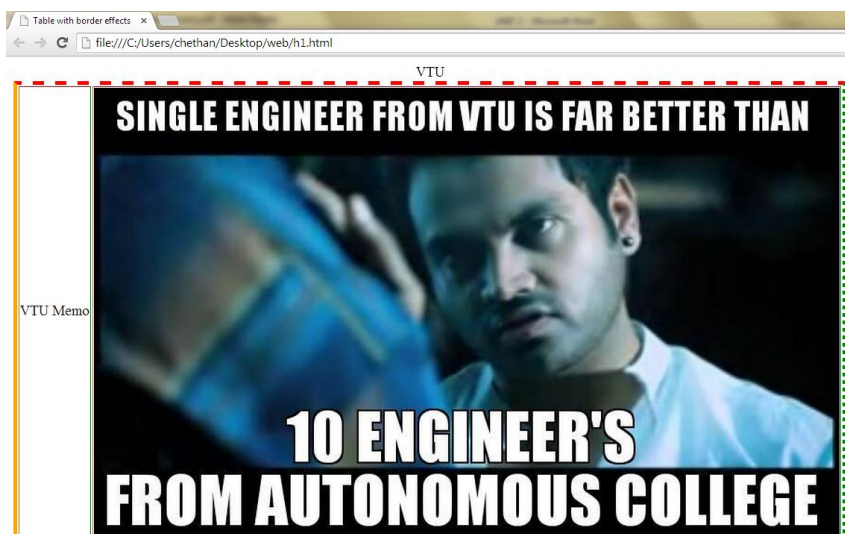
<html>

<head>

```

<title> Table with border effects </title>
<style type = "text/css">
table
{
    border-width:thick;
    border-top-color:red;
    border-left-color:orange;
    border-bottom-color:violet;
    border-right-color:green;
    border-top-style:dashed;
    border-bottom-style:double;
    border-right-style:dotted;
}
</style>
</head>
<body>
    <table border = "border">
<caption>VTU </caption>
    <tr>
        <td> VTU Memo </td>
        <td> <img src = "img9.jpg" alt = "cant display"/></td>
    </tr>
</table>
</body>
</html>

```



Margins and Padding:

The margin properties are named margin, which applies to all four sides of an element: margin-left, margin-right, margin-top, and margin-bottom.

The padding properties are named padding, which applies to all four sides: padding-left, padding-right, padding-top, and padding-bottom.

```
<html>
<head>
    <title> Margins and Padding </title>
<style type = "text/css">
p.one
{
    margin:0.1in;
    padding:0.5in;
    background-color:#FF33FF;
    border-style:dotted;
}

p.two
{
    margin:0.5in;
    padding:0.1in;
    background-color:#00FF33;
    border-style:dashed;
}

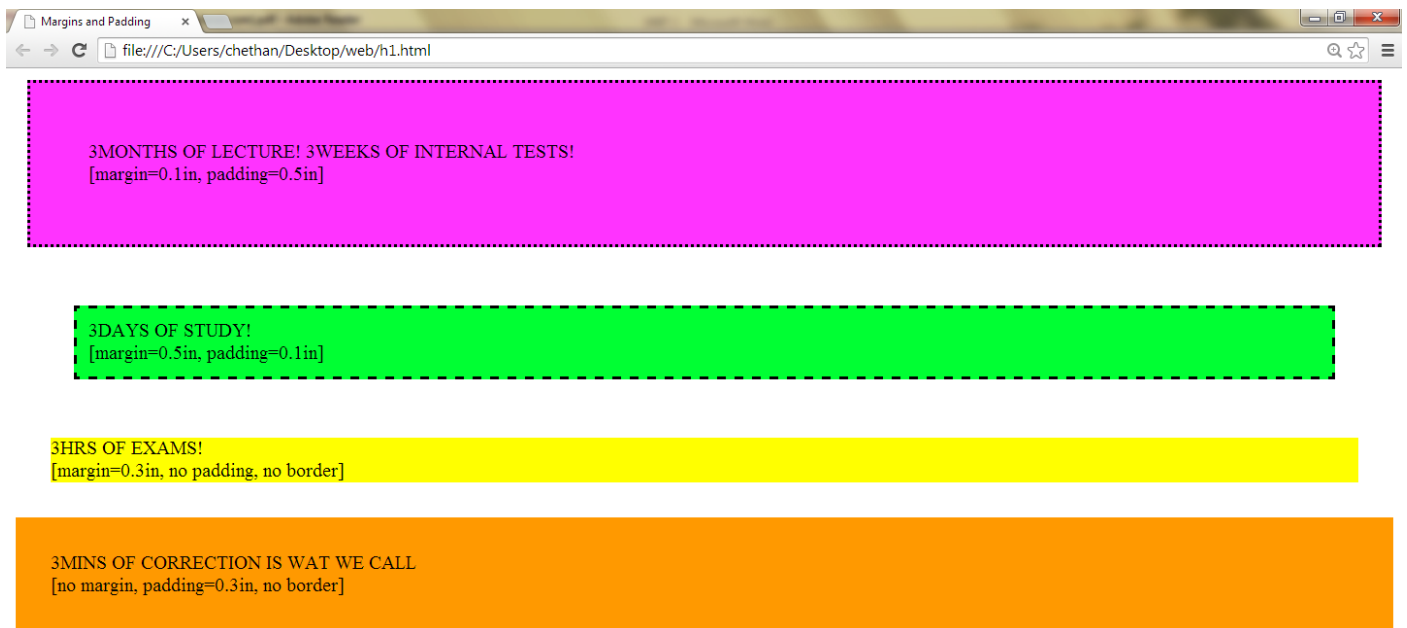
p.three
{
    margin:0.3in;
    background-color:#FFFF00;
}

p.four
{
    padding:0.3in;
    background-color:#FF9900;
}
</style>
</head>
<body>
```

```

    <p class = "one"> 3MONTHS OF LECTURE! 3WEEKS OF INTERNAL TESTS!<br/>
[margin=0.1in, padding=0.5in]</p>
    <p class = "two"> 3DAYS OF STUDY!<br/>
[margin=0.5in, padding=0.1in]</p>
    <p class = "three"> 3HRS OF EXAMS!<br/>
[margin=0.3in, no padding, no border]</p>
    <p class = "four"> 3MINS OF CORRECTION IS WAT WE CALL<br/>
[no margin, padding=0.3in, no border]</p>
</body>
</html>

```



2.15 BACKGROUND IMAGES

The background-image property is used to place an image in the background of an element.

```

<html>
<head>
<title>Background Image</title>
<style type = "text/css">
body { background-image:url(bk.jpg);}
p { text-align: justify; color:white;font-size:25pt;}
</style>
</head>
<body>
<p class = "two">Visvesvaraya Technological University (VTU) is a collegiate public state university in
Karnataka State, India. It was established on 1 April 1998 by the Government of Karnataka as per VTU Act
1994, to improve the quality of technical education in the state. Apart from a few notable exceptions, VTU

```

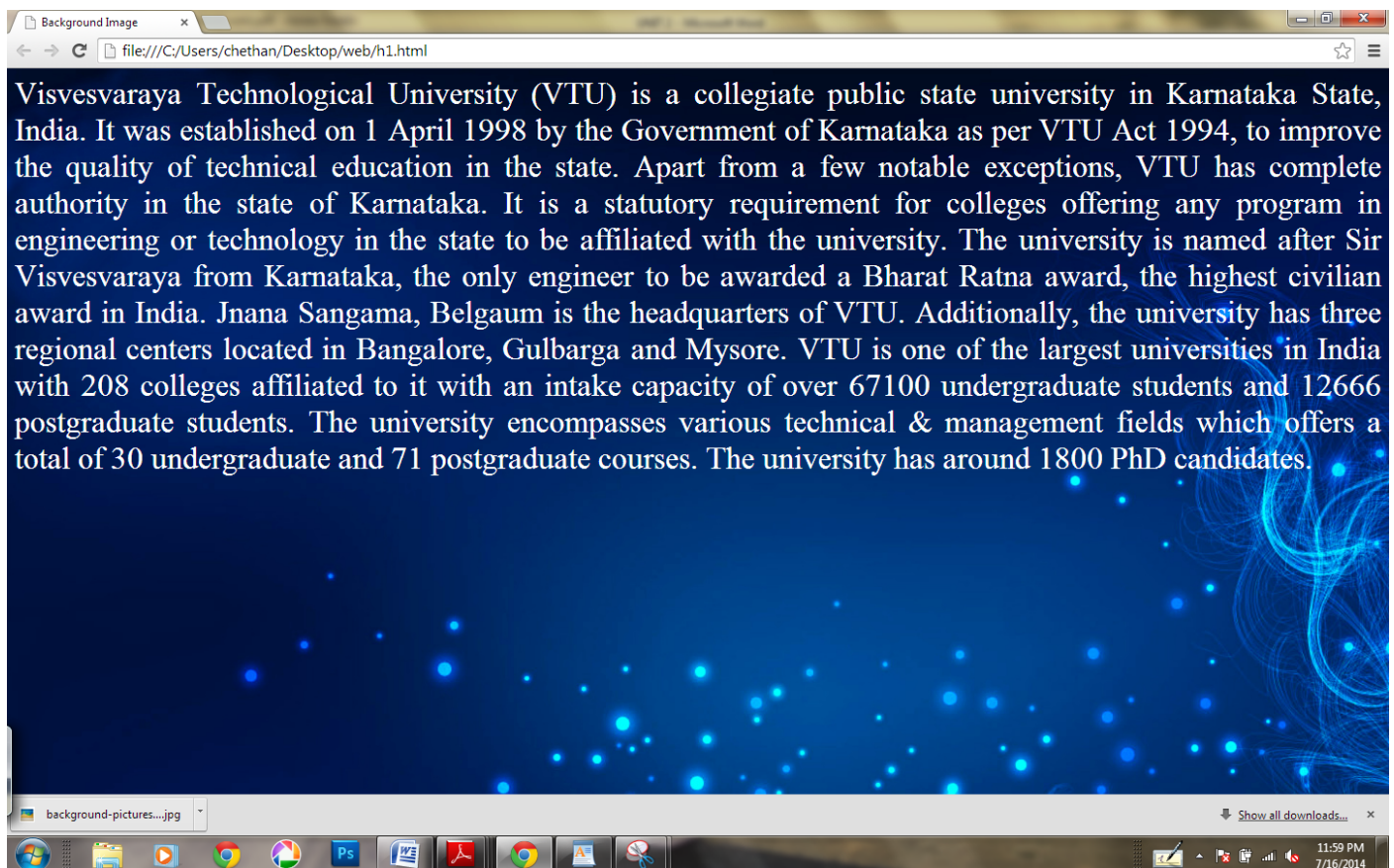
has complete authority in the state of Karnataka. It is a statutory requirement for colleges offering any program in engineering or technology in the state to be affiliated with the university.

The university is named after Sir Visvesvaraya from Karnataka, the only engineer to be awarded a Bharat Ratna award, the highest civilian award in India. Jnana Sangama, Belgaum is the headquarters of VTU. Additionally, the university has three regional centers located in Bangalore, Gulbarga and Mysore.

VTU is one of the largest universities in India with 208 colleges affiliated to it with an intake capacity of over 67100 undergraduate students and 12666 postgraduate students. The university encompasses various technical & management fields which offers a total of 30 undergraduate and 71 postgraduate courses. The university has around 1800 PhD candidates.

</body>

</html>



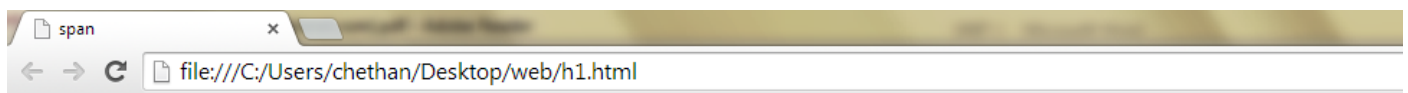
In some time, the background image is replicated as necessary to fill the area of the element. This replication is called *tiling*. Tiling can be controlled with the background-repeat property, which can take the value repeat (the default), no-repeat, repeat-x, or repeat-y. The no-repeat value specifies that just one copy of the image is to be displayed. The repeat-x value means that the image is to be repeated horizontally; repeat-y means that the image is to be repeated vertically. In addition, the position of a non-repeated background image can be specified with the background-position property, which can take a large number of different values. The keyword values are top, center, bottom, left, and right, all of which can be used in many different combinations.

2.16 THE AND <div> TAGS

In many situations, we want to apply special font properties to less than a whole paragraph of text.

The tag is designed for just this purpose.

```
<html>
<head>
    <title>span</title>
<style type = "text/css">
    .spanviolet { font-size:25pt;font-family:'lucida calligraphy';color:violet;}
</style>
</head>
<body>
<p> The university is named after
    <span class = "spanviolet"> Sir Visvesvaraya </span>
, from Karnataka, the only engineer to be awarded a Bharat Ratna award. </p>
</body>
</html>
```



The university is named after *Sir Visvesvaraya*, from Karnataka, the only engineer to be awarded a Bharat Ratna award.

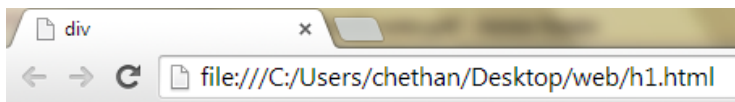
It is more convenient, however, to be able to apply a style to a section of a document rather than to each paragraph. This can be done with the <div> tag. As with , there is no implied layout for the content of the <div> tag, so its primary use is to specify presentation details for a section or division of a document.

```
<html>
<head>
    <title>div</title>
<style type = "text/css">
    .one { font-size:20pt;font-family:'lucida calligraphy';color:violet;}
    .two { font-size:25pt;font-family:'comic sans ms';color:green;}
</style>
</head>
<body>
    <div class = "one">
        <p>Paragraph 1 under division 1</p>
        <p>Paragraph 2 under division 1</p>
        <p>Paragraph 3 under division 1</p>
```

```

</div>
<div class = "two">
    <p>Paragrah 1 under division 2</p>
    <p>Paragrah 2 under division 2</p>
    <p>Paragrah 3 under division 2</p>
</div>
</body>
</html>

```



Paragrah 1 under division 1

Paragrah 2 under division 1

Paragrah 3 under division 1

Paragrah 1 under division 2

Paragrah 2 under division 2

Paragrah 3 under division 2

2.17 CONFLICT RESOLUTION

- Sometimes on a web page, there can be two different values for the same property on the same element leading to conflict.
- h3 {color: blue;}
body h3 {color: red;}
- The browser has to resolve this conflict.
- There can be one or more type of conflict: i.e. when style sheets at 2 or more levels specify different value for same property on some element.
- This conflict is resolved by providing priority to the different levels of style sheets.
- The inline level gets the highest priority over the document level.
- The document level gets the higher priority over the external level
- But the browser must be able to resolve the conflict in the first example using same technique.

- There can be several different origins of the specification of property values.
- One of the value may come from a style sheet created by the author or it can be specified by the user using the options provided by the browser.
- The property values with different origin have different precedence.
- The precedence can also be set for a property by marking it as important.
- `p.special { font-style: italic !important; font-size: 14 }`
- This means that `font-style:italic` is important [this is known as weight of specification]
- The process of conflict resolution is a multi-stage sorting process.
- The first step is to gather information about levels of style sheet.
- Next, all the origins and weights are sorted. The following rules are considered:
 1. Important declarations with user origin
 2. Important declarations with author origin
 3. Normal declarations with author origin
 4. Normal declarations with user origin
 5. Any declarations with browser (or other user agent) origin
- If there are other conflicts even after sorting, the next step is sorting by specificity. Rules are:
 1. id selectors
 2. Class and pseudo class selectors
 3. Contextual selectors (more element type names means that they are more specific)
 4. Universal selectors
- If there still conflicts, they are resolved by giving precedence to most recently seen specification.

Question paper questions :

- a) Write an XHTML document to describe an ordered list of your five favorite movies. Each element of the list must have a nested list of at least two actors in your favorite movies. (05 M)
- b) With examples, explain a style class selector. (05 M)
- c) Write an XHTML document that has six short paragraphs of text. Define three different paragraph styles p1, p2 and p3. The p1 style must use left and right margins of 20 pixels, a background colour of yellow, and a foreground color of blue . The p2 style must use font size of 18 points, font name 'Arial' and a font style in italic form. The p3 style must use a text indent of 1 centimeter , a background color of green, and a foreground color of white. The 1st and the 4th paragraph must use p1, the 2nd and 5th must use p2 and the 3rd and 6th must use p3. (10 M)
- d) Explain the following with respect to table creation in XHTML documents: Align and valign attributes tr, th and td attributes Rowspan and Colspan attributes Cell padding and Cell spacing attributes (10 M)

- e) Create XHTML document to describe a table with the following contents: The columns of the table must have the headings pine, maple, Oak and fir. The row must have the labels average height, average width, typical lifespan, and leaf type. Fill the data cells with some values. (10 M)
- f) Explain the syntactic differences between HTML and XHTML (05 M)
- g) What tag and attribute are used to describe a link? Discuss about it. (04 M)
- h) Explain all controls that are created with the <input> tag with examples, which are used for text collection. (08 M)
- i) Explain the XHTML tags used for lists in documents (08 M)
- j) Write an XHTML program to create a link within a document (04 M)
- k) Create XHTML document that defines a table with 5 rows and 5 columns. The first row should contain country name, gold, silver, bronze (all three indicating the type of medals) and total in each column respectively. Fill in the information details in the table with appropriate values. After filling the details, set red color to the background for the first row, blue for the second, yellow for the third, purple for the fourth and green for the fifth row. Use of align and valign attributes for this table has to be made at the appropriate places (10 M)
- l) How lists are handled in XHTML? Design an XHTML code to illustrate nested lists (10 M)
- m) Design an XHTML code to construct a simple class time table to illustrate table handling (10 M)
- n) Write a XHTML program to create a table with 2 levels of column label an overall label, meals and 3 secondary labels of much breakfast, lunch and dinner. There must be 2 level of row labels: an overall label, food and 4 secondary labels, bread, main cause, vegetables and dessert. The cell of table must contain a number if grams for each category of the food.(12 M)
- o) Explain the different levels of style sheets available in CSS (04 M)
- p) Create XHTML document that contain student information via name, USN, subject 1, subject 2 and subject 3. Insert values for each student in five rows. Also row background of each student should be in the different color (08 M)
- q) Explain the following tags
 - i. <select> ii. <Frame> ii. <text area> (08 M)
- r) What are selector forms? Explain the different levels of selector forms available in CSS with example (08 M)
- s) Write document level style sheet to illustrate pseudo classes. Discuss the conflict resolution in CSS(08 M)
- t) Create an XHTML document that includes atleast 2 images and enough text to precede the images, flow around them (one on left and one on right) and continue after the last image (Note : Use CSS tags) (04 M)
- u) Write an XHTML document to describe an ordered list of four states. Each element of the list must have an unordered list of at least two cities in the state. (05 M)

- v) Explain the following , with respect to table creation in XHTML documents.
- i. <table>
 - ii. tr, th and td attributes
 - iii. rowspan and colspan attributes
 - iv. align and valign attributes
 - v. cell padding and cell spacing (10 M)
- w) Create XHTML document that has two frames. The left frame displays contents.html and the right frame displays cars.html where the second frame is a target of link from the first frame. [Note: contents.html is a list of links to the cars description.] (05 M)
- x) Create , test and validate a XHTML document that has a form with
- i. Three text boxes to collect user name and address.
 - ii. Tables with the headings product name , price and quantity and the values are
 - 100—watts light bulb, \$2.39 , 4
 - 200—watts light bulb, \$4.29 , 8
 - 100—watts long life light bulbs, \$3.95 , 4
 - 200—watts long life light bulbs, \$7.49 , 8
 - iii. A collection of 4 radio buttons that are labeled as Visa Master card Discover Check
 - iv. A submit and a reset button (10 M)