



**Institute for Advanced Computing And
Software Development (IACSD)**
Akurdi, Pune

HTML

**Dr. D.Y. Patil Educational Complex, Sector 29, Behind Akurdi Railway Station,
Nigdi Pradhikaran, Akurdi, Pune - 411044.**

Definitions

- WWW – World Wide Web.
- HTML – HyperText Markup Language – The Language of Web Pages on the WorldWide Web.
- HTML is a text formatting language.
- URL – Uniform Resource Locator.
- Browser – A software program which is used to show web pages.
- “Normal text” surrounded by bracketed tags that tell browsers
- How to display web pages
Pages end with “.htm” or “.html”
- HTML Editor – A word processor that has been specialized to make the writing of HTML documents more effortless.

Tags

- Codes enclosed in brackets
- Usually paired
- <TITLE>My Web Page</TITLE>
- *Not* case sensitive
- <TITLE> = <title> = <TITLE>

Choosing Text Editor

There are many different programs that you can use to create web documents.

HTML Editors enable users to create documents quickly and easily by pushing a few buttons. Instead of entering all of the HTML codes by hand.

These programs will generate the HTML Source Code for you.

HTML Editors are excellent tools for experienced web developers; however, it is important that you learn and understand the HTML language so that you can edit code and fix “bugs” in your pages.

IACSD

HTML

For this Course, we will focus on using the standard Microsoft Windows text editors, NotePad. We may use also textpad.

Starting NotePad

NotePad is the standard text editor that comes with the Microsoft Windows operating system. To start NotePad in Windows 9x or XP follow the steps below:

Click on the "Start" button located on your Windows task bar.

Click on "Programs" and then click on the directory menu labeled "Accessories".

Locate the shortcut "NotePad" and click the shortcut once.

HTML Page Creation & Editing

In this chapter you will learn to create HTML pages with a standard text editor.

Objectives

Upon completing this section, you should be able to

1. Choose a Text Editor.
2. Create a Basic Starting Document.
3. Understand and set Document Properties.
4. View Your Results in a Browser.

Creating a Basic Starting Document

```
<HTML>
<HEAD>
    <TITLE>Al al-Bayt University</TITLE>
</HEAD>
<BODY>
    This is what is displayed.
</BODY>
</HTML>
```

IACSD

HTML

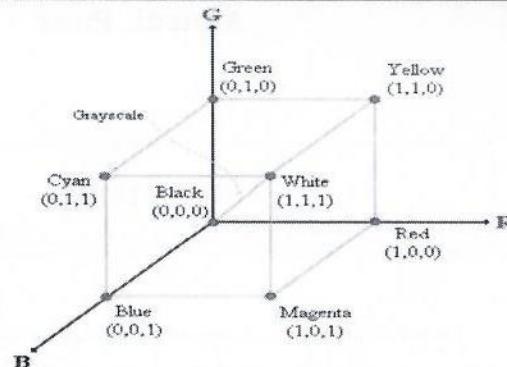
The HEAD of your document points to above window part. The TITLE of your document appears in the very top line of the user's browser. If the user chooses to "Bookmark" your page or save as a "Favorite"; it is the TITLE that is added to the list.

The text in your TITLE should be as descriptive as possible because this is what many search engines, on the internet, use for indexing your site.

Document properties are controlled by attributes of the BODY element. For example, there are color settings for the background color of the page, the document's text and different states of links.

Color Codes

Colors are set using "RGB" color codes, which are represented as hexadecimal values. Each 2-digit section of the code represents the amount, in sequence, of red, green or blue that forms the color. For example, a RGB value with 00 as the first two digits has no red in the color.



The Body Element

The BODY element of a web page is an important element in regards to the page's appearance. Here are the attributes of the BODY tag to control all the levels:

TEXT="#RRGGBB" to change the color of all the text on the page (full page text color.)

This element contains information about the page's background color, the background image, as well as the text and link colors.

IACSD

HTML

Background Color

It is very common to see web pages with their background color set to white or some other colors.

To set your document's background color you need to edit the <BODY> element by adding the BGCOLOR attribute. The following example will display a document with a background color.

```
<BODY BGCOLOR="#FFFFFF"></BODY>
```

TEXT Color

The TEXT attribute is used to control the color of all the normal text in the document. The default color for text is black. The TEXT attribute would be added as follows:

```
<BODY BGCOLOR="#FFFFFF" TEXT="#000000"></BODY>
```

LINK, VLINK, and ALINK

These attributes control the colors of the different linkstates:

1. LINK – initial appearance – default = Blue.
2. VLINK – visited link – default = Purple.
3. ALINK – active link being clicked – default = Yellow. The Format for setting these attributes is:

```
<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF"  
      VLINK="#FF00FF" ALINK="#FFFF00"></BODY>
```

Using Image Background

The BODY element also gives you ability of setting an image as the document's background.

An example of a background image's HTML code is as follows:

```
<BODY BACKGROUND="big.gif"  
      BGCOLOR="#FFFFFF"></BODY>
```

- Once you have created your basic starting document and set your document properties it is a good idea to save your file.
- To save a file, in NotePad, follow these steps:

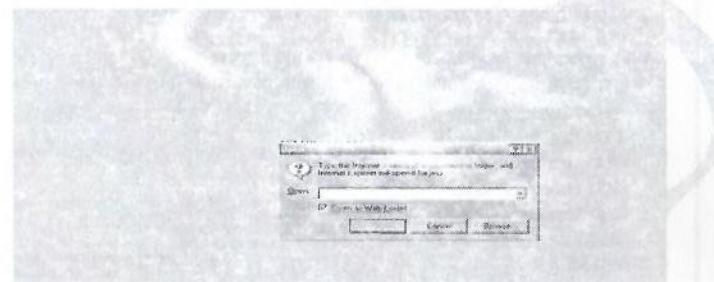
IACSD

HTML

1. Locate and click on the menu called "File".
2. Select the option under File Menu labeled "Save As".
3. In the "File Name" text box, type in the entire name of your file (including the extension name itself).

Edit, Save and View Cycle

- To preview Your Work, open a web browser and do the following:
 1. Click on the menu labeled "File".
 2. Locate the menu option, "Open".



- In the "Open" dialog box, click on the "Browse" button and locate your web document.
- Click "OK" once you have selected your file.
- The web browser will load the same document but with the new revisions. This process is the Edit, Save and View Cycle.

Headings, Paragraphs, Breaks & Horizontal Rules

In this chapter you will add headings to your page, add paragraphs, add so horizontal rules.

Objectives

Upon completing this section, you should be able to

1. List and describe the different Heading elements.
2. Use Paragraphs to add text to a document.
3. Insert breaks where necessary.
4. Add a Horizontal Rule.

Headings, <Hx> </Hx>

- Inside the BODY element, heading elements H1 through H6 are generally used for major

IACSD**HTML**

divisions of the document. Headings are permitted to appear in any order, but you will obtain the best results when your documents are displayed in a browser if you follow these guidelines:

1. H1: should be used as the highest level of heading, H2 as the next highest, and so forth.
2. You should not skip heading levels: e.g., an H3 should not appear after an H1, unless there is an H2 between them.

```
<HTML>
<HEAD>
<TITLE> Example Page</TITLE>
</HEAD>
<BODY>
<H1> Heading 1 </H1>
<H2> Heading 2 </H2>
<H3> Heading 3 </H3>
<H4> Heading 4 </H4>
<H5> Heading 5 </H5>
<H6> Heading 6 </H6>
</BODY>
</HTML>
```

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Paragraphs, <P> </P>

- Paragraphs allow you to add text to a document in such a way that it will automatically adjust the end of line to suite the window size of the browser in which it is being displayed. Each line of text will stretch the entire length of the window.

Paragraphs, <P> </P>

```
<HTML><HEAD>
<TITLE> Example Page</TITLE>
```

IACSD**HTML**

```
</HEAD>
<BODY><H1> Heading 1 </H1>
<P> Paragraph 1, ....</P>
<H2> Heading 2 </H2>
<P> Paragraph 2, ....</P>
<H3> Heading 3 </H3>
<P> Paragraph 3, ....</P>
<H4> Heading 4 </H4>
<P> Paragraph 4, ....</P>
<H5> Heading 5 </H5>
<P> Paragraph 5, ....</P>
<H6> Heading 6</H6>
<P> Paragraph 6, ....</P>
</BODY></HTML>
```

Heading 1
Paragraph 1,....
Heading 2
Paragraph 2,....
Heading 3
Paragraph 3,....
Heading 4
Paragraph 4,....
Heading 5
Paragraph 5,.... Heading 6 Paragraph 6,....

Break,

- Line breaks allow you to decide where the text will break on a line or continue to the end of the window.
- A
 is an empty Element, meaning that it may contain attributes but it does not contain content.
- The
 element does not have a closing tag.

```
<HTML>
<HEAD>
```

IACSD**HTML**

```
<TITLE> Example Page</TITLE>
</HEAD>
<BODY>
<H1> Heading 1 </H1>
<P>Paragraph 1, <BR>
Line 2 <BR> Line 3 <BR>....
</P>
</BODY>
</HTML>
```

Horizontal Rule, <HR>

- The <HR> element causes the browser to display a horizontal line (rule) in your document.
- <HR> does not use a closing tag,

Horizontal Rule, <HR>

Attribute	Description	Default Value
SIZE	Height of the rule in pixels	2 pixels
WIDTH	Width of the rule in pixels or percentage of screen width	100%
NOSHADe	Draw the rule with a flat look instead of a 3D look	Not set(3D look)
ALIGN	Aligns the line (Left, Center, Right)	Center
COLOR	Sets a color for the rule (IE 3.0 or later)	Not set

Horizontal Rule, <HR>

IACSD**HTML**

```
<TITLE> Example Page</TITLE>
</HEAD>
<BODY>
<H1> Heading 1 </H1>
<P>Paragraph 1, <BR>
Line 2 <BR>
<HR> Line 3 <BR>
</P>
</BODY>
</HTML>
```

Heading 1
Paragraph 1,.... Line 2

Line 3

Character Formatting

In this chapter you will learn how to enhance your page with Bold, Italics, and other character formatting options.

Objectives

Upon completing this section, you should be able to

1. Change the color and size of your text.
2. Use Common Character Formatting Elements.
3. Align your text.
4. Add special characters.
5. Use other character formatting elements.

Bold, Italic and other Character Formatting Elements

 Two sizes bigger

The size attribute can be set as an absolute value from 1 to 7 or as a relative value using the "+" or "-" sign. Normal text size is 3 (from -2 to +4).

 Bold

<I> Italic </I>

<U> Underline </U>

Color = "#RRGGBB" The COLOR attribute of the FONT element. E.g., this text has color

<PRE> Preformatted </PRE> Text enclosed by PRE tags is displayed in a mono-spaced font. Spaces and line breaks are supported without additional elements or special characters.

 Emphasis Browsers usually display this as italics.

 STRONG Browsers display this as bold.

IACSD

HTML

<TT> **TELETYPE** </TT> Text is displayed in a mono-spaced font. A typewriter text, e.g. fixed-width font.
<CITE> Citation </CITE> represents a document citation (*italics*). For titles of books, films, etc. Typically displayed in *italics*. (*A Beginner's Guide to HTML*)

<P> One
Size Larger - Normal
-

 One Size

Smaller

 Bold - <i> italics </i> -
<u> Underlined </u> -

Colored

 Emphasized -
 Strong
 - <TT> Tele Type
</TT>

One Size Larger - Normal – One Size Smaller

Bold - italics - Underlined - Colored

Emphasized - Strong - Tele Type

Alignment

- Some elements have attributes for alignment (ALIGN) e.g. Headings, Paragraphs and Horizontal Rules.
- The Three alignment values are : LEFT, RIGHT, CENTER.
- <CENTER></CENTER> Will center elements.

Alignment

IACSD

HTML

<DIV ALIGN="value"></DIV> Represents a division in the document and can contain most other element type. The alignment attribute of the DIV element is well supported.

<TABLE></TABLE> Inside a TABLE, alignment can be set for each individual cell.

Special Characters & Symbols

□ These Characters are recognized in HTML as they begin with an ampersand and end with a semi-colon e.g. &value; The value will either be an entity name or a standard ASCII character number. They are called escape sequences.

□ The next table represents some of the more commonly used special characters. For a comprehensive listing, visit the W3C's section on special characters at:

http://www.w3.org/MarkUp/HTML_Plus/htmlplus_13.html

Special Characters & Symbols

Special Character	Entity Name	Special Character	Entity Name
Ampersand	& &	Greater-than sign	> >
Asterisk	∗ *	Less-than sign	< <
Cent sign	¢ ¢	Non-breaking space	
Copyright	© ©	Quotation mark	" "
Fraction onequarter	¼ ¼	Registrationmark	® ®
Fraction onehalf	½ ½	Trademark sign	™ ™

43

Special Characters & Symbols

- Additional escape sequences support accented characters, such as:
- ö
- a lowercase o with an umlaut: ö
- ñ
- a lowercase n with a tilde: ñ
- È

IACSD**HTML**

- an uppercase E with a grave accent: É

NOTE: Unlike the rest of HTML, the escape sequences are case sensitive. You cannot, for instance, use < instead of <.

Additional Character Formatting Elements

- <STRIKE> strike-through text</STRIKE>

DEL is used for STRIKE at the latest browsers

- <BIG> places text in a big font</BIG>
- <SMALL> places text in a small font</SMALL>
- _{places text in subscript position}
- ^{places text in superscript style position}

Example

<P><STRIKE> strike-through text </STRIKE></P>

<BIG>places text in a big font </BIG>

<SMALL> places text in a small font</SMALL>

_{places text in subscript position}Normal

<SUP> places text in superscript style position

</SUP>
 </P>

Lists

In this chapter you will learn how to create a variety of lists.

Objectives

Upon completing this section, you should be able to

- 1.Create an unordered list.
- 2.Create an ordered list.
- 3.Create a defined list.
- 4.Nest Lists.

List Elements

- HTML supplies several list elements. Most list elements are composed of one or more (List Item) elements.

UL : Unordered List. Items in this list start with a list mark such as a bullet. Browsers will usually change the list mark in nested lists.

 List item ...

 List item ...

•List item ...

•List item ...

- You have the choice of three bullet types: disc(default), circle, square.

These are controlled in Netscape Navigator by the "TYPE" attribute for the element.

IACSD**HTML**

```
<UL TYPE="square">
<LI> List item ...</LI>
<LI> List item ...</LI>
<LI> List item ...</LI>
</UL>
```

- List item ...
- List item ...
- List item ...

OL: Ordered List. Items in this list are numbered automatically by the browser.

```
<OL>
<LI> List item ...</LI>
<LI> List item ...</LI>
<LI> List item ...</LI>
</OL>
```

1. List item ...
2. List item ...
3. List item

You have the choice of setting the TYPE Attribute to one of five numbering styles

• TYPE	• Numb ering	Styles
• 1	• Arabic numbers	1,2,3,.....
a	Lower alpha	a, b, c,.....
A	Upper alpha	A, B, C,.....
i	Lower roman	i, ii, iii,.....
I	Upper roman	I, II, III,.....

You can specify a starting number for an ordered list.

```
<OL TYPE ="i">
<LI> List item ...</LI>
<LI> List item ...</LI>
</OL>
<P> text ....</P>
<OL TYPE="i" START="3">
<LI> List item ...</LI>
</OL>
```

i. List item ...

ii. List item ... Text

IACSD**HTML**

iii. List item ...

- DL: Definition List. This kind of list is different from the others. Each item in a DL consists of one or more Definition Terms (DT elements), followed by one or more Definition Description (DD elements).

```
<DL>
<DT> HTML </DT>
<DD> Hyper Text Markup Language </DD>
<DT> DOG </DT>
<DD> A human's best friend!</DD>
</DL>
```

HTML DOG

Hyper Text Markup Language

A human's best friend!

Nesting Lists

You can nest lists by inserting a UL, OL, etc., inside a listitem (LI).

Example

```
<UL TYPE = "square">
<LI> List item ...</LI>
<LI> List item ...
<OL TYPE="I" START="3">
<LI> List item ...</LI>
</OL>
</LI>
<LI> List item ...</LI>
</UL>
```

IACSD**HTML**

- List item ...
- List item ...
- iii. List item ...
- iv. List item ...
- v. List item ...
- vi. List item ...
- vii. List item ...
- List item ...

```
<H1 ALIGN="CENTER">SAFETY TIPS FOR CANOEISTS</H1>
<OL TYPE="a" START="2">
<LI>Be able to swim </LI>
<LI>Wear a life jacket at all times </LI>
<LI>Don't stand up or move around. If canoe tips,
<UL>
<LI>Hang on to the canoe </LI>
<LI>Use the canoe for support and </LI>
<LI>Swim to shore
</UL></LI>
<LI>Don't overexert yourself</LI>
<LI>Use a bow light at night </LI>
</OL>
```

The output....

SAFETY TIPS FOR CANOEISTS

- b. Be able to swim
- c. Wear a life jacket at all times
- d. Don't stand up or move around. If canoe tips,
 - Hang on to the canoe
 - Use the canoe for support and
 - Swim to shore
- e. Don't overexert yourself
- f. Use a bow light at night

```
<H1 ALIGN="CENTER">SAFETY TIPS FOR CANOEISTS</H1>
<OL TYPE="a" START="2">
<LI>Be able to swim </LI>
<LI>Wear a life jacket at all times </LI>
<LI>Don't stand up or move around. If canoe tips,
<UL>
<LI>Hang on to the canoe </LI>
<LI>Use the canoe for support
<OL type="I" start="4">
<LI> Be careful </LI>
<LI> Do not look around</LI>
</LI> </OL>
<LI>Swim to shore
</UL> </LI>
<LI>Don't overexert yourself </LI>
<LI>Use a bow light at night </LI>
</OL>
```

SAFETY TIPS FOR CANOEISTS

- b. Be able to swim
- c. Wear a life jacket at all times
- d. Don't stand up or move around. If canoe tips,
 - o Hang on to the canoe
 - o Use the canoe for support
- IV. Be careful
- V. Do not look around
- o Swim to shore
- e. Don't overexert yourself
- f. Use a bow light at night

Images

In this chapter you will learn about images and how to place images in your pages.

Objectives

Upon completing this section, you should be able to

1. Add images to your pages.

This element defines a graphic image on the page.

Image File (SRC:source): This value will be a URL (location of the image) E.g. [http://www.domain.com/dir/file.txt](http://www.domain.com/dir/file.ext).

Alternate Text (ALT): This is a text field that describes an image or acts as a label. It is displayed when they position the cursor over a graphic image.

Alignment (ALIGN): This allows you to align the image on your page.

Width (WIDTH): is the width of the image in pixels.

Height (HEIGHT): is the height of the image in pixels.

Border (BORDER): is for a border around the image, specified in pixels.

HSPACE: is for Horizontal Space on both sides of the image specified in pixels. A setting of 5 will put 5 pixels of invisible space on both sides of the image.

VSPACE: is for Vertical Space on top and bottom of the image specified in pixels. A setting of 5 will put 5 pixels of invisible space above and below the image.

1)

2)

3)

4)

5) blast blast blast blast

Anchors, URLs and Image Maps

In this chapter you will learn about Uniform Resource Locator, and how to add them as Anchor or Links inside your web pages.

Objectives

Upon completing this section, you should be able to

1. Insert links into documents.
2. Define Link Types.
3. Define URL.
4. List some commonly used URLs.
5. Plan an Image Map.

HOW TO MAKE A LINK

1)The tags used to produce links are the <A>

and . The <A> tells where the link should start and the indicates where the link ends. Everything between these two will work as a link.

2)The example below shows how to make the word Here work as a link to iacsd.

Click here to go to iacsd.

```
<body LINK="#C0C0C0" VLINK="#808080"
ALINK="#FF0000">
```

LINK - standard link - to a page the visitor hasn't been to yet. (standard color is blue - #0000FF). VLINK - visited link - to a page the visitor has been to before. (standard color is purple - #800080).

ALINK - active link - the color of the link when the mouse is on it. (standard color is red #FF0000).

If the programmer what to change the color

Click here to go to yahoo.

Internal Links : Links can also be created inside large documents to simplify navigation. Today's world wants to be able to get the information quickly. Internal links can help you meet these goals.

1.Select some text at a place in the document that you would like to create a link to, then add an anchor to link to like this:

```
<A NAME="bookmark_name"></A>
```

The Name attribute of an anchor element specifies a location in the document that we link to shortly. All NAME attributes in a document must be unique.

2.Next select the text that you would like to create as a link to the location created above.

```
<A HREF="#bookmark_name">Go To Book Mark</A>
```

E-Mail (Electronic Mail)

E.g. mailto:kmf@yahoo.com

□ The type of service is identified as the mail client program. This type of link will launch the users mail client.

□ The recipient of the message is kmf@yahoo.com

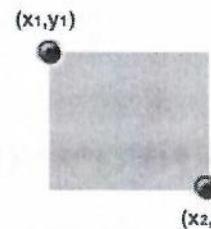
```
<A HREF="mailto:kmf@yahoo.com">Send me More Information </A>
```

Image Maps

□ Image maps are images, usually in gif format that have been divided into regions; clicking in a region of the image cause the web surfer to be connected to a new URL. Image maps are graphical form of creating links between pages.

□ There are two type of image maps: Client side and server side

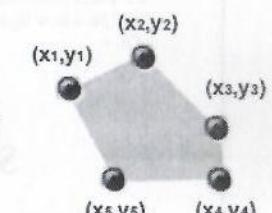
Both types of image maps involve a listing of co-ordinates that define the mapping regions and which URLs those coordinates are associated with. This is known as the map file.

Area Shapes Used**Finding the coordinates....**

Rectangle



Circle



Polygon

Client-Side Image Maps

□ Client-side image maps (USEMAP) use a map file that is part of the HTML document (in an element called MAP), and is linked to the image by the Web browser.

```
<IMG SRC="note.GIF" Width=200 Height=200 border="5" USEMAP="#map1">
<MAP NAME="map1">
<AREA SHAPE="RECT" COORDS="0,0,90,90"
HREF="hi.html" ALT="see me...">
<AREA SHAPE="RECT" COORDS="100,100,160,160"
HREF="divPara.html" ALT="see him...">
<AREA SHAPE="CIRCLE" COORDS="150,50,20"
HREF="house.html" ALT="see it...">
</MAP>
```

We can use Poly as well as Rect.....

Shapes, Coords

- Types of Shapes
 - Rect used for squares and ordered shapes.
 - Circle used for circles.
 - Poly used for unordered shapes.
- Number of coodenations for each shape:
 - Rect 4 numbers for two corners
 - Circle 3 numbers for the center & R

Poly depends on the number of corners of the shape(2 numbers for each corner)

Tables

In this chapter you will learn that tables have many uses in HTML.

Objectives:

Upon completing this section, you should be able to:

1. Insert a table.
2. Explain a table's attributes.
3. Edit a table.
4. Add a table header.

The `<TABLE></TABLE>` element has four sub-elements:

1. Table Row `<TR></TR>`.
 2. Table Header `<TH></TH>`.
 3. Table Data `<TD></TD>`.
 4. Caption `<CAPTION></CAPTION>`.
- The table row elements usually contain table header elements or table data elements.

```
<table border="1">
<tr>
<th> Column 1 header </th>
<th> Column 2 header </th>
</tr>
<tr>
<td> Row1, Col1 </td>
<td> Row1, Col2 </td>
</tr>
<tr>
<td> Row2, Col1 </td>
<td> Row2, Col2 </td>
</tr>
</table>
```

Column 1 Header	Column 2 Header
Row1, Col1	Row1, Col2
Row2, Col1	Row2, Col2

Tables Attributes

- BGColor: Some browsers support background colors in a table.
- Width: you can specify the table width as an absolute number of pixels or a percentage of the document width. You can set the width for the table cells as well.
- Border: You can choose a numerical value for the border width, which specifies the border in pixels.
- CellSpacing: Cell Spacing represents the space between cells and is specified in pixels.
- CellPadding: Cell Padding is the space between the cell border and the cell contents and is specified in pixels.
- Align: tables can have left, right, or center alignment.
- Background: Background Image, will be titled in IE3.0 and above.
- BorderColor, BorderColorDark.

Table Caption

A table caption allows you to specify a line of text that will appear centered above or below the table.

```
<TABLE BORDER=1 CELLPADDING=2>
<CAPTION ALIGN="BOTTOM"> Label For My Table
</CAPTION>
```

The Caption element has one attribute ALIGN that can be either TOP (Above the table) or BOTTOM (below the table).

Table Header

Table Data cells are represented by the TD element. Cells can also be TH (Table Header) elements which results in the contents of the table header cells appearing centered and in bold text.

Table Data and Table Header

Attributes

- Colspan: Specifies how many cell columns of the table this cell should span.
- Rowspan: Specifies how many cell rows of the table this cell should span.
- Align: cell data can have left, right, or center alignment.
- Valign: cell data can have top, middle, or bottom alignment.
- Width: you can specify the width as an absolute number of pixels or a percentage of the document width.
- Height: You can specify the height as an absolute number of pixels or a percentage of the document height.

IACSD**HTML**

```
Basic Table Code
<TABLE BORDER=1 width=50%>
<CAPTION><h1>Spare Parts </h1></Caption>
<TR><TH>Stock Number</TH><TH>Description</TH><TH>List Price</TH></TR>
<TR><TD bgcolor=red>3476-AB</TD><TD>76mm Socket</TD><TD>45.00</TD></TR>
<TR><TD>3478-AB</TD><TD><font color=blue>78mm Socket</font></TD>
<TD>47.50</TD></TR>
<TR><TD>3480-AB</TD><TD>80mm Socket</TD><TD>50.00</TD></TR>
</TABLE>
```

Spare Parts

Stock Number	Description	List Price
3476-AB	76mm Socket	45.00
3478-AB	78mm Socket	47.50
3480-AB	80mm Socket	50.00

Table Data and Table Header Attributes

```
<Table border=1 cellpadding =2>
<tr><th> Column 1 Header</th> <th> Column 2 Header</th> </tr>
<tr><td colspan=2> Row 1 Col 1</td> </tr>
<tr><td rowspan=2>Row 2 Col 1</td>
<td> Row 2 Col2</td> </tr>
<tr><td> Row 3 Col2</td> </tr>
</table>
```

Table Data and Table Header Attributes

Column 1 Header	Column 2 Header
Row 1 Col 1	
Row 2 Col 1	Row 2 Col 2
	Row 3 Col 2

IACSD**HTML**

- TH, TD and TR should always have end tags.

Although the end tags are formally optional, many browsers will mess up the formatting of the table if you omit the end tags. In particular, you should always use end tags if you have a TABLE within a TABLE – in this situation, the table parser gets hopelessly confused if you don't close your TH, TD and TR elements.

- A default TABLE has no borders

By default, tables are drawn without border lines. You need the BORDER attribute to draw the lines.

- By default, a table is flush with the left margin

TABLEs are plopped over on the left margin. If you want centered tables, You can either: place the table inside a DIV element with attribute ALIGN="center".

Most current browsers also supports table alignment, using the ALIGN attribute. Allowed values are "left", "right", or "center", for example: <TABLE ALIGN="left">. The values "left" and "right" float the table to the left or right of the page, with text flow allowed around the table. This is entirely equivalent to IMG alignment

```
<TABLE BORDER width="750">
<TR> <TD colspan="4" align="center">PageBanner</TD></TR>

<TR> <TD rowspan="2" width="25%">NavLinks</TD>
      <TD colspan="2" width="25%">Feature Article</TD>
      <TD rowspan="2" width="25%">Linked Ads</TD>
</TR>

<TR><TD width="25%">News Column 1 </TD>
      <TD width="25%"><News Column 2 </TD>
</TR>
</TABLE>
```



Frames

- relatively new addition to the HTML standard. First introduced in Netscape Navigator
-
- completing this section, you should be able to:
- based page.
- Frameset, Frame, and Noframes elements.
- attributes of the Frames elements to control the display.
- appropriately.

Frames are a

Objectives:
Upon

Create a Frame

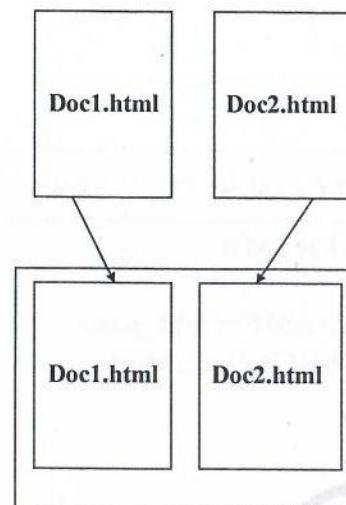
Work with the

Use the

Set Targets

A framed page is actually made up of multiple HTML pages. There is one HTML document that describes how to break up the single browser window into multiple windowpanes. Each windowpane is filled with an HTML document.

For Example to make a framed page with a windowpane on the left and one on the right requires three HTML pages. Doc1.html and Doc2.html are the pages that contain content. Frames.html is the page that describes the division of the single browser window into two windowpanes.



- A <FRAMESET> element is placed in the htmldocument before the <BODY> element. The <FRAMESET> describes the amount of screen realestate given to each windowpane by dividing the screen into ROWS or COLS.
- The <FRAMESET> will then contain <FRAME> elements, one per division of the browser window.
- Note: Because there is no BODY container, FRAMESET pages can't have background imagesand background colors associated with them.

Frame Page Architecture

```

<HTML>
<HEAD>
<TITLE> Framed Page <TITLE>
<FRAMESET COLS="23%,77%">
<FRAME SRC="Doc1.html">
<FRAME SRC="Doc2.html">
</FRAMESET >
</HEAD>
  
```

</HTML>

The Diagram below is a graphical view of the document described above

FRAMESET COLS="23%, 77%"

FRAME
NAME=
left_pane
SRC=Doc1.h
tml

FRAME
NAME=right_pane
SRC= Doc2.html

<FRAMESET> Container

<FRAMESET> : The FRAMESET element creates divisions in the browser window in a single direction. This allows you to define divisions as either rows or columns.

□ ROWS : Determines the size and number of rectangular rows within a <FRAMESET>. They are set from top of the display area to the bottom.

Possible values are:

- Absolute pixel units, i.e. "360,120".
- A percentage of screen height, e.g. "75%,25%".
- Proportional values using the asterisk (*). This is often combined with a value in pixels , e.g. "360,*".
- <FrameSet cols="200,20%,*,2%">

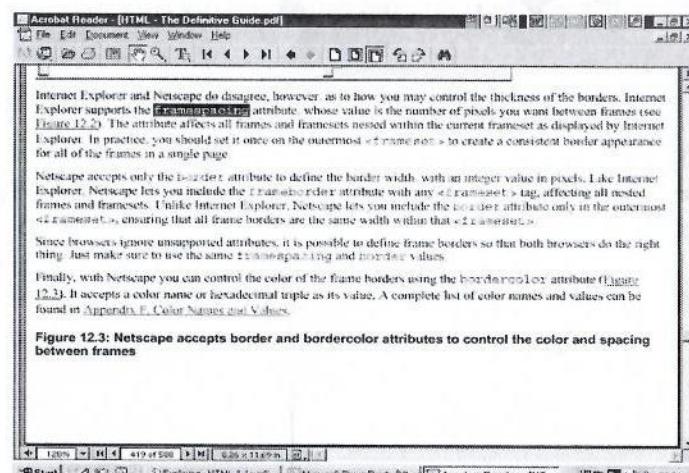
Creating a Frames Page

COLS: Determines the size and number of rectangular columns within a <FRAMESET>. They are set from left to right of the display area.

Possible values are:

- Absolute pixel units, i.e. "480,160".
- A percentage of screen width, e.g. "75%,25%".
- Proportional values using the asterisk (*). This is often combined with a value in pixels , e.g. "480,*".

GHT: Optional attribute stated in pixels. Determines vertical space between the <FRAME> contents and the frame's borders.



1

4ARGINWID
: between the

MARGINHEI

IACSD**HTML**

■ Displays a scroll bar(s) in the frame.

Possible values are:

- | | |
|--|--|
| <ol style="list-style-type: none"> 1. display scroll bar(s). 2. display scroll bar(s). 3. will decide based on frame contents. | SCROLLING: <ul style="list-style-type: none"> Yes – always No – never Auto – browser |
|--|--|

By default: scrolling is auto.

NORESIZE: Optional – prevents viewers from resizing the frame. By default the user can stretch or shrink the frame's display by selecting the frame's border and moving it up, down, left, or right.

<NOFRAMES>

<NOFRAMES>: Frame – capable browsers ignore all HTML within this tag including the contents of the BODY element. This element does not have any attributes.

```
<HTML>
<HEAD>
<TITLE> Framed Page </TITLE>
</HEAD>
<FRAMESET COLS="23%,77%">
<FRAME SRC=""NAME="left_pane">
<FRAME SRC=""NAME="right_pane">
<NOFRAMES>
<P> This is a Framed Page. Upgrade your browser to support frames.</P>
</NOFRAMES></FRAMESET>
```

Compound FRAMESET Divisions

In this case a second FRAMESET element will be inserted in the place of the FRAME element that would describe the second row.

The second FRAMESET element will divide the remaining screen real estate into 2 columns.

This nested FRAMESET will then be followed by 2 FRAME elements to describe each of the subsequent frame divisions created.

IACSD**HTML**

Banner File

```
<html>
<head>
<title> Compound Frames Page</title>
</head>
<frameset rows="120,*">
<frame src="banner_file.html" name="banner">
<frameset cols="120,*">
<frame src="links_file.html" name="links">
<frame src="content_file.html" name="content">
```

```
<noframes>
<p>
Default message
</p>
</noframes>
</frameset>
</frameset>
</head>
```

Compound FRAMESET Divisions

You may want to create a frames design with a combination of rows and columns.

Link
sFile

Contents File

```
<HEAD>
<FRAMESET ROWS="25%,50%,25%">
<FRAME SRC="">
<FRAMESET COLS="25%,*"
<FRAME SRC="">
<FRAME SRC="">
</FRAMESET>
<FRAME SRC="">
</FRAMESET>
</HEAD>
```

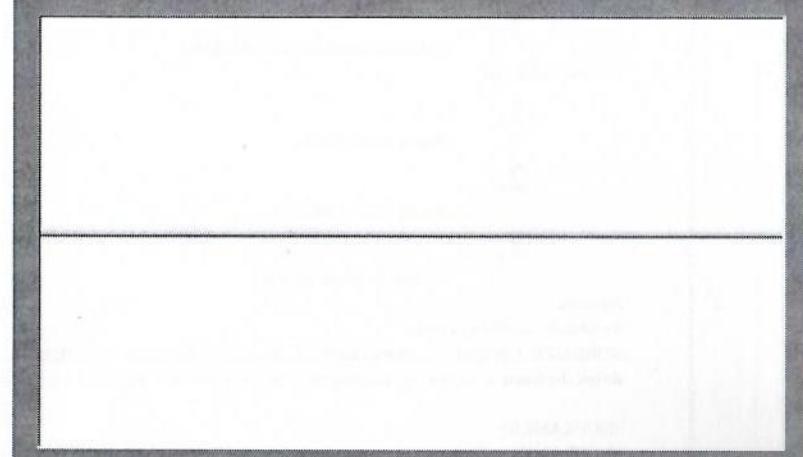


Figure 5-14: Frames created with <FRAMESET ROWS="50%, 50%">

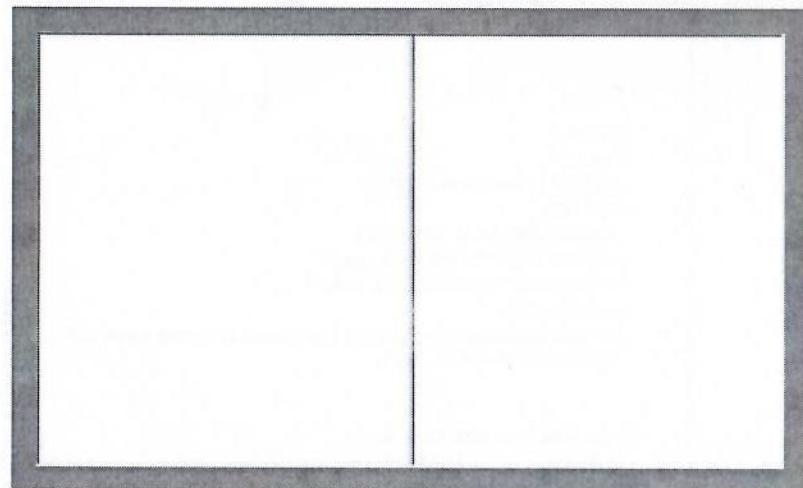


Figure 5-15: Frames created with <FRAMESET COLS="50%, 50%">

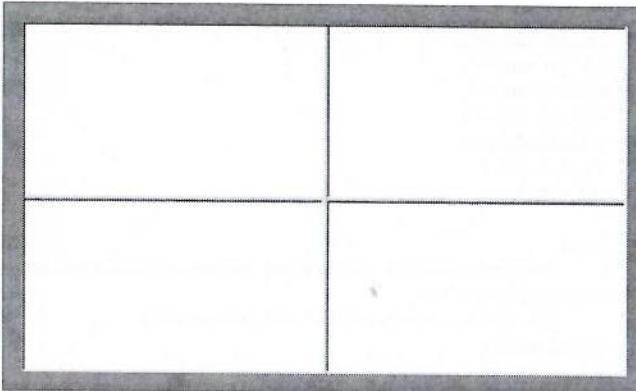


Figure 5-13: Frames created with <FRAMESET ROWS="50%, 50%" COLS="50%, 50%">

- Example:

Frame Formatting

```
<frameset rows="20%, *, 20%">
<frame src="header.html" noresize scrolling=no>
<frame src="body.html">
<frame src="navigationbar.html" noresize scrolling=no>
</frameset>
```

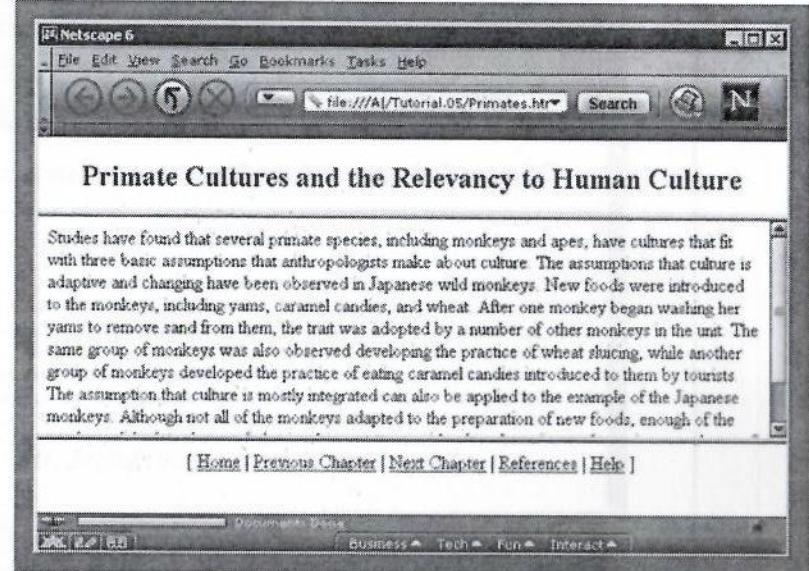


Figure 5-24: Output of program that includes NORESIZE and SCROLLING attributes

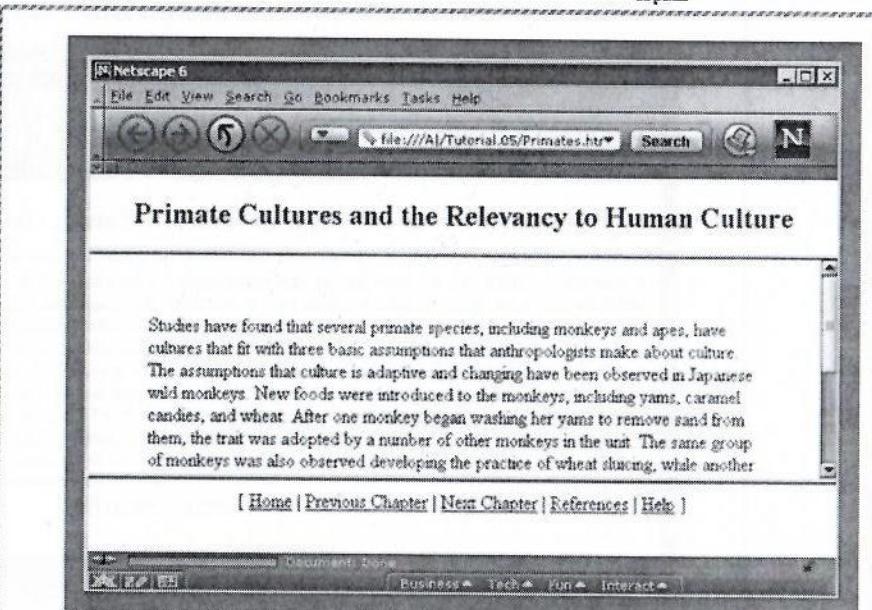


Figure 5-25: Middle frame changed to `<FRAME SRC="body.html" MARGINHEIGHT=50 MARGINWIDTH=50>`

What do the following mean?

- 1) `<FRAMESET COLS="2*, 3*, 5*>`
- 2) `<FRAMESET COLS="150, 20%, *, 3*>`

So what are the space-allocation priorities?

Absolute pixel values are always assigned space first, in order from left to right. These are followed by percentage values of the total space. Finally, proportional values are divided based upon what space is left.

Generic Frame Formula

- The `<FRAME>` tag has six associated attributes: SRC, NAME, MARGINWIDTH, MARGINHEIGHT, SCROLLING, and NORESIZE. Here's a complete generic FRAME:
 - `<FRAME SRC="url" NAME="window_name" SCROLLING=YES|NO|AUTO MARGINWIDTH="value" MARGINHEIGHT="value" NORESIZE>`

What will be the Output?

```
<FRAMESET ROWS="*, 2*, *"    COLS="2*, *">
<FRAME SRC="">
<FRAME SRC="">
<FRAME SRC="">
<FRAME SRC="">
<FRAME SRC="">
<FRAME SRC="">
</FRAMESET>
```

Targets

- When you use links for use in a frames environment you will need to specify an additional attribute called TARGET.
 - The TARGET attribute uses the NAME attribute of the FRAME element.
 - If we were to place a link in doc1.html that linked to doc3.html and we wanted doc3.html to be displayed in the right windowpane; the HTML code would appear in doc1.html as follows:

```
<A HREF="doc3.html" TARGET="right_pane">Link to Document 3
</A>
```

Special Targets

*There are 4 special target names that cannot be assigned by the NAME attribute of the FRAME tag.

1.TARGET="`_top`" : This loads the linked document into the full browser window with the URL specified by the HREF attribute. All frames disappear, leaving the new linked page to occupy the entire window. The back is turned on.

2.TARGET="`_blank`" : Opens an unnamed new browser window and loads the document specified in the URL attribute into the new window (and your old window stays open). The back is turned off. Other windows remains on.

3.TARGET="`_self`" : Loads the document in the same window where the anchor was {Clicked}. This is the default setting for linking elements.

4.TARGET="`_parent`" : the `_parent` frame is a prior frameset that the current frameset was "spawned" from. If there isn't one it is the browser window. The document is loaded into the area occupied by the columns or rows frameset containing the frame that contains the link. The back is turned on. All windows disappear.

If a frame contains the following link, then clicking the link launches a new, unnamed browser display window that contains the content defined in stuff.HTM. This can be a simple HTML document, or an entirely new FRAMESET definition.

1.

If a frame contains the following link, then clicking the link will simply cause the frame which contains the link to clear, and its content will be replaced with whatever is in stuff.htm.

2.

If a frame contains the following link, the frameset that contains the frame that contains this link will be replaced by stuff.HTM.

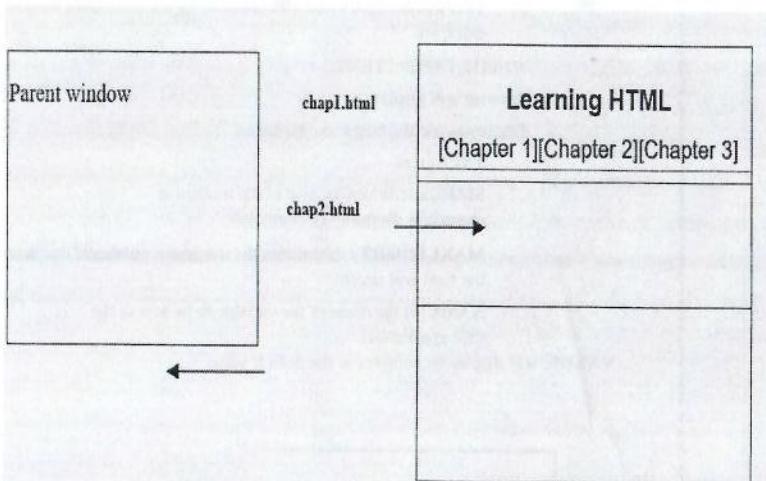
3. Finally, if a frame contains the following link, clicking the link replaces the entire browser window with the contents of stuff.HTM.

4.

Targeting links to frames

The TARGET attribute allows you to specify the frame into which a page is to be loaded into in a frames setting.

```
<A HREF="chap1.html" TARGET="_self"> [Chapter 1]</A>
<A HREF="chap1.html" TARGET="_parent"> [Chapter 2]</A>
```

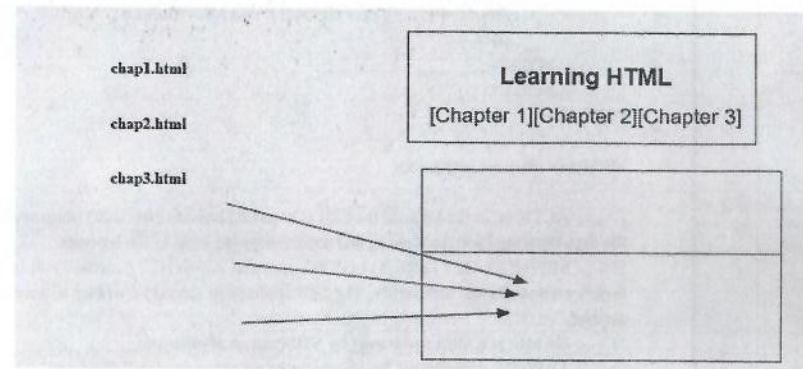


Targeting links to frames

 [Chapter 1]

 [Chapter 2]

 [Chapter 3]



FORMS

- Forms add the ability to web pages to not only provide the person viewing the document with dynamic information but also to obtain information from the person viewing it, and process the information.

Objectives:

Upon completing this section, you should be able to

1. Create a FORM.
 2. Add elements to a FORM.
 3. Specify an action for the FORM.
- Forms work in all browsers.
 - Forms are Platform Independent.

- To insert a form we use the <FORM></FORM> tags. The rest of the form elements must be inserted in between the form tags.

IACSD**HTML**

```
<HTML>
  <HEAD>
    <TITLE> Sample Form</TITLE>
  </HEAD>
  <BODY BGCOLOR="#FFFFFF">
    <FORM ACTION = "http://www.xnu.com/formtest.asp">
      <P> First Name: <INPUT TYPE="TEXT" NAME="fname" MAXLENGTH="50">
      </P>
      <P> <INPUT TYPE="SUBMIT" NAME="fsubmit1" VALUE="Send Info">
      </P>
    </FORM>
  </BODY>
</HTML>
```

<FORM> element attributes

- ACTION: is the URL of the CGI (Common Gateway Interface) program that is going to accept the data from the form, process it, and send a response back to the browser.
- METHOD: GET (default) or POST specifies which HTTP method will be used to send the form's contents to the web server. The CGI application should be written to accept the data from either method.
- NAME: is a form name used by VBScript or JavaScripts.
- TARGET: is the target frame where the response page will show up.

Form Elements

- Form elements have properties: Text boxes, Password boxes, Checkboxes, Option(Radio) buttons, Submit, Reset, File, Hidden and Image.
- The properties are specified in the TYPE Attribute of the HTML element <INPUT></INPUT>.

Name: Sami Ali

Student No.: 123456789

Address: Yarmouk University
Faculty of IT

City: Amman
Arman
Irbid
Karak

is foreign?

Male:

Female:

Submit Reset

126

IACSD**HTML****<INPUT> Element's Properties**

TYPE= Type of INPUT entry field.

NAME = Variable name passed to CGI application

VALUE= The data associated with the variable
name to be passed to the CGI application

CHECKED= Button/box checked

SIZE= Number of visible characters in text field

MAXLENGTH= Maximum number of characters
accepted.**Text Box****Text boxes:** Used to provide input fields for text, phone numbers, dates, etc.

<INPUT TYPE="TEXT">

Browser will display

Textboxes use the following attributes:

TYPE: text.

SIZE: determines the size of the textbox in characters. Default=20 characters.

MAXLENGTH: determines the maximum number of characters that the field will accept.

NAME: is the name of the variable to be sent to the CGI application.

VALUE: will display its contents as the default value.

IACSD**HTML**

Example on Text Box

```
<HTML>
<TITLE>Form_Text_Type</TITLE>
<HEAD>
<BODY>
<h1> <font color=blue>Please enter the following bioData</font></h1>
<FORM name="fome1" Method=" get " Action=" URL " >
First Name:
<INPUT TYPE="TEXT" NAME="FName" SIZE="15" MAXLENGTH="25"><BR>
Last Name:
<INPUT TYPE="TEXT" NAME="LName" SIZE="15" MAXLENGTH="25"><BR>
Nationality:
<INPUT TYPE="TEXT" NAME="Country" SIZE="25" MAXLENGTH="25"><BR>
The Phone Number:
<INPUT TYPE="TEXT" NAME="Phone" SIZE="15" MAXLENGTH="12"><BR>
</FORM>
</BODY>
</HTML>
```

Output

Please enter the following bioData

First Name:

Last Name:

Nationality:

The Phone Number:

Password

Password: Used to allow entry of passwords.

<INPUT TYPE="PASSWORD" >

Browser will display

Text typed in a password box is starred out in the browser display.

Password boxes use the following attributes:

TYPE: password.

IACSD**HTML**

- SIZE: determines the size of the textbox in characters.
- MAXLENGTH: determines the maximum size of the password in characters.
- NAME: is the name of the variable to be sent to the CGI application.
- VALUE: is usually blank.

```
<HTML>
<HEAD>
<TITLE>Form_Password_Type</TITLE>
</HEAD>
<BODY>
<h1> <font color=red>To Access, Please enter:</font></h1>
<FORM name="fome2" Action="url" method="get">
User Name:
<INPUT TYPE="TEXT" Name="FName" SIZE="15" MAXLENGTH="25"><BR>
Password:
<INPUT TYPE="PASSWORD" NAME="PWord" value="" SIZE="15" MAXLENGTH="25"><BR>
</FORM>
</BODY>
</HTML>
```

Output

To Access, Please enter:

User Name:

Password:

Hidden

Hidden: Used to send data to the CGI application that you don't want the web surfer to see, change or have to enter but is necessary for the application to process the form correctly.

<INPUT TYPE="HIDDEN">

Nothing is displayed in the browser.

Hidden inputs have the following attributes:

- TYPE: hidden.
- NAME: is the name of the variable to be sent to the CGI application.
- VALUE: is usually set a value expected by the CGI application.

Check Box

- Check Box: Check boxes allow the users to select more than one option.

<INPUT TYPE="CHECKBOX">

Browser will display

Checkboxes have the following attributes:

- TYPE: checkbox.
- CHECKED: is blank or CHECKED as the initial status.
- NAME: is the name of the variable to be sent to the CGI application.
- VALUE: is usually set to a value.

```
<HTML>
<HEAD>
  <TITLE>CheckBoxType</TITLE>
</HEAD>
<BODY>
  <h1>
    <font color=green>Please check one of the following</font></h1>
    <FORM name="fome3" Action="url" method="get">
      <font color=red> Select Country: </font><BR>
      jordan:<INPUT TYPE="CheckBox" Name="country" CHECKED><BR>
      Yemen:<INPUT TYPE="CheckBox" Name="country"><BR>
      Qatar:<INPUT TYPE="CheckBox" Name="country"><BR><BR>
      <font color=blue>Select Language:</font><BR>
      Arabic:<INPUT TYPE="CheckBox" Name="language" CHECKED><BR>
      English:<INPUT TYPE="CheckBox"
      Name="language"><BR>
      French:<INPUT TYPE="CheckBox" Name="language"><BR>
    </FORM>
  </BODY>
</HTML>
```

Output

Please check one of the following

Select Country:

jordan:

Yemen:

Qatar:

Select Language:

Arabic:

English:

French:

Radio Button

- Radio Button: Radio buttons allow the users to select only one option.

<INPUT TYPE="RADIO">

Browser will display

Radio buttons have the following attributes:

- TYPE: radio.
- CHECKED: is blank or CHECKED as the initial status. Only one radio button can be checked
- NAME: is the name of the variable to be sent to the CGI application.
- VALUE: usually has a set value.
- VALUE: usually has a set value.

```
<HTML>
<HEAD><TITLE>CheckBoxType</TITLE> </HEAD>
<BODY>
```

```
  <h1> <font color=green>Please check one of the following</font></h1>
  <FORM name="fome3" Action="url" method="get">
    <font color=red> Select Country: </font><BR>
    jordan:<INPUT TYPE="RADIO" Name="country" CHECKED><BR>
    Yemen:<INPUT TYPE="RADIO" Name="country"><BR>
    Qatar:<INPUT TYPE="RADIO" Name="country"><BR>
    <BR>
    <font color=blue>Select Language:</font><BR>
```

```
Arabic:<INPUT TYPE="RADIO" Name="language" CHECKED><BR>
English:<INPUT TYPE=" RADIO " Name="language"><BR>
French:<INPUT TYPE=" RADIO " Name="language">
<BR>
</FORM>
</BODY>
</HTML>
```

Please check one of the following

Select Country:

jordan:

Yemen

Qatar:

Select Language:

Arabic:

English:

French:

```
<HTML><HEAD>
<TITLE>RADIOBox</TITLE> </HEAD>
<BODY>
Form #1:
<FORM>
<INPUT TYPE="radio" NAME="choice" VALUE="one"> Yes.
<INPUT TYPE="radio" NAME="choice" VALUE="two"> No.
</FORM>
<HR color="red" size="10" >
Form #2:
<FORM>
<INPUT TYPE="radio" NAME="choice" VALUE="three" CHECKED> Yes.
<INPUT TYPE="radio" NAME="choice" VALUE="four"> No.
</FORM>
</BODY></HTML>
```

Output

Form #1:

Yes. No.

Form #2:

Yes. No.

Push Button

- Push Button: This element would be used with JavaScript to cause an action to take place.
- <INPUT TYPE="BUTTON">
- Browser will display

Push Button has the following attributes:

- TYPE: button.
- NAME: is the name of the button to be used in scripting.
- VALUE: determines the text label on the button.

```
<DIV align="center"><BR><BR>
<FORM>
<FONT Color=red>
<h1>Press Here to see a baby crying:<BR>
<INPUT TYPE="button" VALUE="PressMe"><BR><BR>
<FONT Color=blue>
Click Here to see a baby shouting:<BR>
<INPUT TYPE="button" VALUE="ClickMe" >
<BR><BR>
<FONT Color=green>
Hit Here to see a baby eating:<BR>
<INPUT TYPE="button" VALUE="HitME" > <BR><BR>
<FONT Color=yellow>
</FORM>
</DIV>
```

Press Here to see a baby crying:

Click Here to see a baby shouting:

Hit Here to see a baby eating:

Submit Button

- Submit: Every set of Form tags requires a Submit button. This is the element causes the browser to send the names and values of the other elements to the CGI Application specified by the ACTION attribute of the FORM element.

<INPUT TYPE="SUBMIT">

The browser will display

Submit has the following attributes:

- TYPE: submit.
- NAME: value used by the CGI script for processing.
- VALUE: determines the text label on the button,

usually Submit Query.

```
<FORM Action="URL" method="get">
First Name: <INPUT TYPE="TEXT" Size=25 name="firstName"><BR>
Family Name: <INPUT TYPE="TEXT" Size=25 name="lastName"><BR>
<BR>
<FONT Color=red> Press Here to submit the data:<BR>
<INPUT TYPE="submit" VALUE="SubmitData " >
</FORM>
```

First Name:
 Family Name:

Press Here to submit the data:

Reset Button

- Reset: It is a good idea to include one of these for each form where users are entering data. It allows the surfer to clear all the input in the form.

▪ <INPUT TYPE="RESET">

▪ Browser will display

▪ Reset buttons have the following attributes:

▪ TYPE: reset.

▪ VALUE: determines the text label on the button, usually Reset.

```
<FORM Action="URL" method="get"> First Name: <INPUT TYPE="TEXT" Size=25
name="firstName"><BR>
Family Name: <INPUT TYPE="TEXT" Size=25 name="lastName"><BR>
<BR>
<FONT Color = red>
<STRONG><font size=5>Press Here to submit the data:</font></STRONG><BR>
<INPUT TYPE="submit" VALUE="SubmitData " >
<INPUT TYPE="RESET" VALUE="Reset">
</FORM>
```

First Name:
 Family Name:

Press Here to submit the data:

Image Submit Button

- Image Submit Button: Allows you to substitute an image for the standard submit button.

```
<INPUT TYPE="IMAGE" SRC="jordan.gif">
```

Image submit button has the following attributes:

- TYPE: Image.
- NAME: is the name of the button to be used in scripting.
- SRC: URL of the Image file.

```
<form>
<H1><font color=blue> Click to go Jordan's Map:
<INPUT TYPE="IMAGE" SRC="jordan.gif">
</form>
```

Click to go Jordan's Map: 

File

- File Upload: You can use a file upload to allow surfers to upload files to your web server.
- <INPUT TYPE="FILE">
- Browser will display

- File Upload has the following attributes:
- TYPE: file.
- SIZE: is the size of the text box in characters.
- NAME: is the name of the variable to be sent to the CGI application.
- MAXLENGTH: is the maximum size of the input in the textbox in characters.

```
<BODY bgcolor=lightblue>
<form>
<H3><font color=forestgreen>
Please attach your file here to for uploading to My
<font color =red>SERVER...<BR>
<INPUT TYPE="File" name="myFile" size="30">
<INPUT TYPE="Submit" value="SubmitFile">
</form>
</BODY>
```

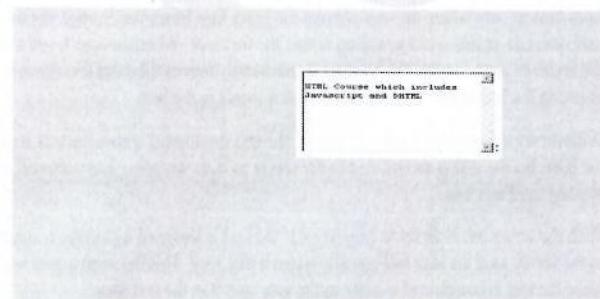
Other Elements used in Forms

- <TEXTAREA></TEXTAREA>: is an element that allows for free form text entry.

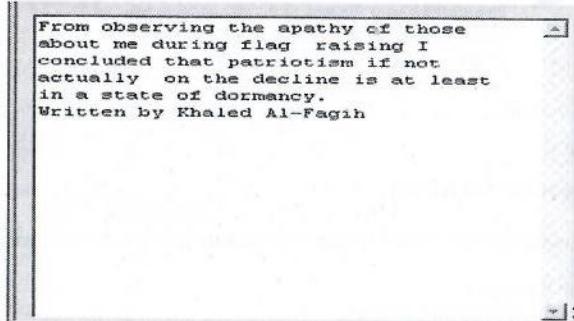
Browser will display

Textarea has the following attributes:

- NAME: is the name of the variable to be sent to the CGI application.
- ROWS: the number of rows to the textbox.
- COLS: the number of columns to the textbox.



```
<BODY bgcolor=lightblue>
<form>
<TEXTAREA COLS=40 ROWS=20
Name="comments">
From observing the apathy of those about me during flag raising I concluded that patriotism if not
actually on the decline is at least in a state of dormancy.
Written by Khaled Al-Fagih
</TEXTAREA>
</form>
</BODY>
```



10.6.1.2 The **wrap** attribute

Normally, text typed in the text area by the user is transmitted to the server exactly as typed, with lines broken only where the user pressed the Enter key. Since this is often not the desired action by the user, you can enable word wrapping within the text area. When the user types a line that is longer than the width of the text area, the browser automatically moves the extra text down to the next line, breaking the line at the nearest point between words in the line.

With the **wrap** attribute set to **virtual**, the text is wrapped within the text area for presentation to the user, but the text is transmitted to the server as if no wrapping had occurred, except where the user pressed the Enter key.

With the **wrap** attribute set to **physical**, the text is wrapped within the text area and is transmitted to the server as if the user had actually typed it that way. This is the most useful way to use word wrap, since the text is transmitted exactly as the user sees it in the text area.

To obtain the default action, set the **wrap** attribute to **off**.

As an example, consider the following 60 characters of text being typed into a 40-character-wide text area:

Word wrapping is a feature that makes life easier for users.

With **wrap=off**, the text area will contain one line and the user will have to scroll to the right to see all of the text. One line of text will be transmitted to the server.

With **wrap=virtual**, the text area will contain two lines of text, broken after the word "makes." Only one line of text will be transmitted to the server: the entire line with no embedded newline characters.

With **wrap=physical**, the text area will contain two lines of text, broken after the word "makes." Two lines of text will be sent to the server, separated by a newline character after the word "makes."

Other Elements used in Forms

- The two following examples are

<SELECT></SELECT> elements, where the attributes are set differently.

The Select elements attributes are:

- NAME: is the name of the variable to be sent to the CGI application.
- SIZE: this sets the number of visible choices.
- MULTIPLE: the presence of this attribute signifies that the user can make multiple selections.

By default only one selection is allowed.

```
<BODY bgcolor=lightblue>
<form>
Select the cities you have visited:
<SELECT name="list" size=5>
<option> London</option>
<option> Tokyo</option>
<option> Paris</option>
<option> New York</option>
<option> LA</option>
<option> KL</option>
</SELECT>
</form>
</BODY>
```

IACSD

HTML

Select the cities you have visited:

The screenshot shows a dropdown menu with the following options: London, Tokyo, Paris, New York, and LA. The menu has scroll arrows at the top and bottom.

- **Drop Down List:**



- **Name:** is the name of the variable to be sent to the CGI application.

The screenshot shows a dropdown menu with two items: 'Item 1' and 'Item 2'.

IACSD

HTML

- **Option**

The list items are added to the <SELECT> element by inserting <OPTION></OPTION> elements.

The Option Element's attributes are:

- **SELECTED:** When this attribute is present, the option is selected when the document is initially loaded. It is an error for more than one option to be selected.
- **VALUE:** Specifies the value the variable named in the select element.

```
</HEAD>
<BODY>
<h2><font color=blue>What type of Computer do you have?</font></h2>
<FORM>
<SELECT NAME="ComputerType" size=4>
    <OPTION value="IBM" SELECTED> IBM</OPTION>
    <OPTION value="INTEL"> INTEL</OPTION>
    <OPTION value=" Apple"> Apple</OPTION>
    <OPTION value="Compaq"> Compaq</OPTION>
</SELECT>
</FORM>
</BODY>
</HTML>
```

What type of Computer do you have?



IACSD

HTML

```
<HEAD>
  <TITLE>SELECT with Multiple </TITLE>
</HEAD>
<BODY>
  <h2><font color=blue>What type of Computer do you have?</font></h2>
  <FORM>
    <SELECT NAME="ComputerType" size=5 multiple>
      <OPTION value="IBM" > IBM</OPTION>
      <OPTION value="INTEL"> INTEL</OPTION>
      <OPTION value=" Apple"> Apple</OPTION>
      <OPTION value="Compaq" SELECTED> Compaq</OPTION>
      <OPTION value=" other"> Other</OPTION>
    </SELECT>
  </FORM>
</BODY>
</HTML>
```

What type of Computer do you have?

IBM
INTEL
Apple
Compaq
Other

55

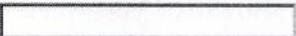
IACSD

HTML

There are eleven different types of form elements:

Button 

Checkbox 

FileUpload 

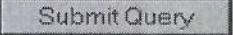
Hidden

Password 

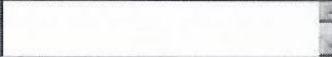
Radio 

Reset object 

Select object 

Submit object 

Text 

Textarea 

56



**Institute for Advanced Computing And
Software Development (IACSD)
Akurdi, Pune**

HTML5

Dr. D.Y. Patil Educational Complex, Sector 29, Behind Akurdi Railway Station,
Nigdi Pradhikaran, Akurdi, Pune - 411044.

Content

- What is HTML5
- Components of HTML 5
- Detecting HTML 5 capabilities
- HTML5 Form tags
- Media tags
- Web worker
- Canvas
- Geolocation
- Web Storage
- Introduction to CSS3
- Introduction to Media element
- CSS3 validations
- Many more....

What is HTML 5

- HTML5 will be the new standard for HTML..

IACSD**React JS**

- The previous version of HTML, HTML 4.01, came in 1999. The web has changed a lot since then.
- HTML5 is still a work in progress. However, the major browsers support many of the new HTML5 elements and APIs.
- HTML5 introduces a number of new elements and attributes that helps in building a modern websites
- HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

What is New in HTML 5

- New Semantic Elements: These are like `<header>`, `<footer>`, and `<section>`.
- Forms 2.0: Improvements to HTML web forms where new attributes have been introduced for `<input>` tag.
- Persistent Local Storage: To achieve without resorting to third-party plugins.
- Canvas: This supports a two-dimensional drawing surface that you can program with JavaScript.
- Audio & Video: You can embed audio or video on your web pages without resorting to third-party plugins.
- Geolocation: Now visitors can choose to share their physical location with your web application.

HTML 5 Syntax

- The DOCTYPE:
- DOCTYPEs in older versions of HTML were longer because the HTML language was SGML based and therefore required a reference to a DTD.
- HTML 5 authors would use simple syntax to specify DOCTYPE as follows:
- `<!DOCTYPE html>` All the above syntax is case-insensitive.
- Character Encoding:
- HTML 5 authors can use simple syntax to specify Character Encoding as follows:
`<meta charset="UTF-8">` All the above syntax is case-insensitive.

IACSD**React JS**

- The `<script>` tag:
It's common practice to add a type attribute with a value of "text/javascript" to script elements as follows:
- `<script type="text/javascript" src="scriptfile.js"></script>` HTML 5 removes extra information required and you can use simply following syntax:
`<script src="scriptfile.js"></script>`
- The `<link>` tag:
So far you were writing `<link>` as follows:
`<link rel="stylesheet" type="text/css" href="stylefile.css">` HTML 5 removes extra information required and you can use simply following syntax:
`<link rel="stylesheet" href="stylefile.css">`

HTML 5 Elements

New types for `<input>` tag:
The input element's type attribute now has the following new values:

Type	Description
color	Color selector, which could be represented by a wheel or swatch picker
date	Selector for calendar date
datetime-local	Date and time display, with no setting or indication for time zones
datetime	Full date and time display, including a time zone.
email	Input type should be an email.
month	Selector for a month within a given year
number	A field containing a numeric value only
range	Numeric selector within a range of values, typically visualized as a slider
search	Term to supply to a search engine. For example, the search bar atop a browser.
tel	Input type should be telephone number.
time	Time indicator and selector, with no time zone information
url	Input type should be URL type.
week	Selector for a week within a given year

HTML 5 Removed Elements

Removed Elements
The following HTML 4.01 elements are removed from HTML5:

IACSD

```
<acronym>
<applet>
<basefont>
<big>
<center>
<dir>
<font>
<frame>
<frameset>
<noframes>
<strike>
<tt>
```

React JS**Tags - <header>:**

"a group of introductory or navigational aids. A header element typically contains the section's heading (an h1–h6 element or an hgroup element), but can also contain other content, such as a table of contents, a search form, or any relevant logos."

```
<header>
<h1>Main Header</h1>
<p class="tagline">Tagline</p>
</header>
```

Tags - <nav>:

"Not all groups of links on a page need to be in a nav element only sections that consist of major navigation blocks are appropriate for the nav element."

```
<nav role="navigation">
<ul>
<li>
<a href="#" title="link">link</a>
</li>
</ul>
</nav>
```

Tags - <article>:

"a composition that forms an independent part of a document, page, application, or site. This could be a forum post, a magazine or newspaper article, a Web log entry ..."

```
<article>
<h2>Item</h2>
<p>Some content here.</p>
</article>
```

IACSD**React JS****Tags - <section>:**

"section is a blob of content that you could store as an individual record in a database."

```
<section id="foo">
<h2>Foo</h2>
<p>
Content here
</p>
</section>
```

Tags - <article> VS Tags - <section>:

"Authors are encouraged to use the article element instead of the section element when it would make sense to syndicate the contents of the element."

"Section is used when there is naturally a heading at the start of the section."

Tags - <footer>:

"The footer element represents a footer for its nearest ancestor sectioning content or sectioning root element. A footer typically contains information about its section such as who wrote it, links to related documents, copyright data, and the like."

```
<footer role="contentinfo">
<p>
Footer
</p>
<nav>
<h5>Quick Links</h5>
<ul role="navigation">
<li>
<a href="#">link</a>
</li>
</ul>
</nav>
</footer>
```

Canvas

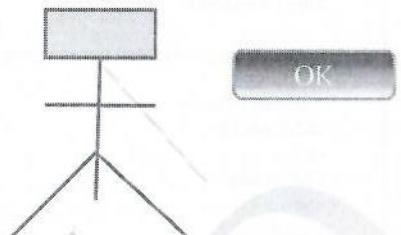
HTML5 element `<canvas>` gives you an easy and powerful way to draw graphics using JavaScript.

It can be used to draw graphs, make photo compositions or do simple (and not

IACSD

so simple) animations.

```
<!DOCTYPE HTML>
<html>
<head>
<style>
#mycanvas{
    border:1px solid red;
}
</style>
</head>
<body>
    <canvas id="mycanvas" width="100" height="100"></canvas>
</body>
</html>
```

**HTML5 <video>**

- HTML5 has DOM methods, properties, and events for the <video> and <audio> elements.
- There are methods for playing, pausing, and loading, for example and there are properties (like duration and volume).
- There are also DOM events that can notify you when the <video> element begins to play, is paused, is ended, etc.

<Audio>

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

<audio controls>
<source src="horse.ogg" type="audio/ogg">
<source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>

</body>
</html>
```

Geolocation

HTML5 Geolocation API lets you share your location with your favorite web sites.

A Javascript can capture your latitude and longitude and can be sent to backend web server and do

React JS**IACSD**

fancy location-aware things like finding local businesses or showing your location on a map.

Today most of the browsers and mobile devices support Geolocation API. The geolocation APIs work with a new property of the global navigator object ie. Geolocation object which can be created as follows: var geolocation = navigator.geolocation; The geolocation object is a service object that allows widgets to retrieve information about the geographic location of the device.

```
<script>
var x=document.getElementById("demo");
function getLocation()
{
    if(navigator.geolocation)
    {
        navigator.geolocation.getCurrentPosition(showPosition);
    }
    else{x.innerHTML="Geolocation is not supported by this browser.";}
}
function showPosition(position)
{
    x.innerHTML="Latitude: " +
    position.coords.latitude + "<br>Longitude: " +
    position.coords.longitude;
}
</script>
```

Web Workers

A web worker is a JavaScript running in the background, without affecting the performance of the page.

What is a Web Worker?

When executing scripts in an HTML page, the page becomes unresponsive until the script is finished.

A web worker is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page.

You can continue to do whatever you want: clicking, selecting things, etc., while the web worker runs in the background.

```
function startWorker()
{
if(typeof(Worker)!=="undefined")
```

IACSD

```
if(typeof(w)=="undefined")
{
w=new Worker("demo_workers.js");
}
w.onmessage = function (event) { document.getElementById("result1").innerHTML=event.data;
};

else {
document.getElementById("result").innerHTML="Sorry, your browser doesnot
support Web Workers...";
}

function stopWorker()
{
w.terminate();
}
```

React JS

Web Session

localStorage and sessionStorage

There are two new objects for storing data on the client:
localStorage - stores data with no expiration date
sessionStorage - stores data for one session

Before using web storage, check browser support for localStorage and sessionStorage:

```
if(typeof(Storage)!=="undefined")
{
// Yes! localStorage and sessionStorage support!
// Some code.....
}
else
{
// Sorry! No web storage support..
}
```

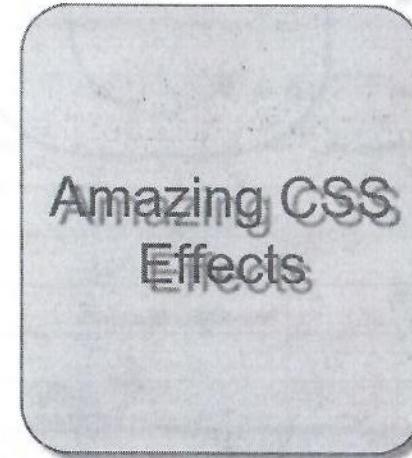
CSS3

IACSD

- Rounded corners
- Gradients
- Box and text shadows
- Fonts
- Transparencies
- Multiple background images and border images
- Multiple columns and grid layout
- Box sizing
- Stroke and outlines
- Animation, movement and rotation
- Improved selectors

React JS

CSS Effect Example



```
.amazing {
border: 1px solid blue;
color: red;
background-color: gold;

-webkit-border-radius: 40px;
-moz-border-radius: 40px;
border-radius: 40px;

-webkit-box-shadow: 8px 8px 6px
#474747;
-moz-box-shadow: 8px 8px 6px
#474747;
box-shadow: 8px 8px 6px #474747;

text-shadow: 8px 8px 2px #595959;
filter: dropshadow(color=#595959,
offx=8, offy=8);
}
```

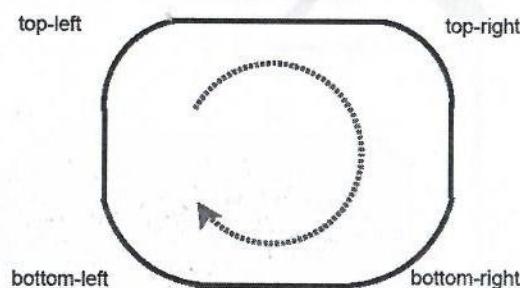
Create Rounded Corners

border-radius: [x] {1,4} [/ [y] {1,4}]?

IACSD

React JS

```
border-radius: 5px 10px 10px  
10px / 10px 10px 5px 5px;  
border-radius: 5px 10px / 10px;  
border-radius: 10px;
```



W3C Specification	Mozilla Implementation
border-radius	-moz-border-radius
border-top-left-radius	-moz-border-radius-topleft
border-top-right-radius	-moz-border-radius-topright
border-bottom-right-radius	-moz-border-radius-bottomright
border-bottom-left-radius	-moz-border-radius-bottomleft

Drop Shadow

```
box-shadow: x y b color
```

Examples:

11

IACSD

React JS

```
-webkit-box-shadow: -3px 3px 3px #999999;  
-moz-box-shadow: -3px 3px 3px #999999; text-shadow: -4px 4px 3px #999999;
```

Create Gradient Buttons

- `rgba(r, g, b, opacity)`

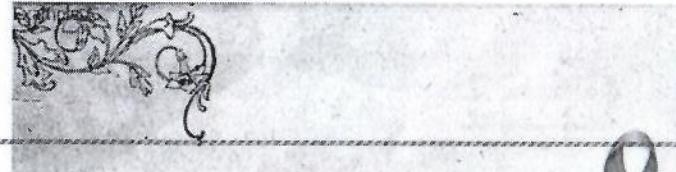


- `background:-webkit-gradient(linear,0% 0%,0% 100%, from(rgba(255,255,255,1)),to(rgba(185,185,185,1)));`
- `background:-moz-linear-gradient(top,rgba(255,255,255,1),rgba(185,185,185,1));`

Multiple background images

background:
<image> <position> <size> <repeat> <attachment> <box>,
<image> <position> <size> <repeat> <attachment> <box>,
<image> <position> <size> <repeat> <attachment> <box>,
<image> <position> <size> <repeat> <attachment> <box> <color>;
OR
background-image: <image>, <image>, <image>, <image>; background-repeat: <repeat>,
<repeat>, <repeat>, <repeat>; background-position: <position>, <position>,
<position>, <position>;
/* plus any background attachment and/or box properties as needed */

```
#example { width: 500px; height: 250px;  
background-image: url(decoration.png), url(ribbon.png), url(old_paper.jpg);  
background-repeat: no-repeat;  
background-position: left top, right bottom, left top;  
}
```



Multi-Column Layout

```
#multi-column {
/* For Mozilla: */
-moz-column-width: 13em;
-moz-column-gap: 1em;
/* For WebKit: */
-webkit-column-width: 13em;
-webkit-column-gap: 1em;
}
```

Lorem ipsum dolor sit amet, Nunc ut leo vel magna adipiscing
consectetur adipiscing elit. tempor. Donec pretium, ligula et
Aenean egestas blandit ipsum. Mauris in elit in dolor imperdiet
Morbi nulla metus, luctus et, interdum. Vivamus egestas
ullamcorper sit amet, commodo sagittis justo. Sed lorem. Sed vel
quis, nisl. Ut blandit lacus nec odio quis porta nonummy, mauris
nibh. Phasellus eleifend enim et
risus. Nam condimentum. arcu gravida odio, quis aliquam
Praesent euismod auctor dui. lacus elit non libero. Proin elit.
aliquam augue accumsan augue.

Multi-Column Layout

```
#multi-column {
-moz-column-count: 3;
-moz-column-gap: 1em;
```

```
-moz-column-rule: 1px solid black;
-webkit-column-count: 3;
-webkit-column-gap: 1em;
-webkit-column-rule: 1px solid black;
}
```

Cras urna metus, aliquam sed, condimentum eget, pellentesque scelerisque massa. Nullam et est id augue blandit tincidunt. Ut consectetur, justo eleifend varius facilisis, tortor lorem pharetra nunc, ac sodales purus nunc semper tortor. Integer nec urna. Praesent scelerisque, ipsum nec aliquet volutpat, sem ante sagittis risus, sed condimentum magna libero luctus elit. Donec pede purus, hendrerit non, laoreet vel, porta ut, neque. Cras eu lacus. Pellentesque tempus mattis magna. Nullam id nisi. Quisque dolor lorem, commodo ac, pharetra sed, nonummy nec, nulla. Mauris purus. Suspendisse eget mauris nec justo eleifend vestibulum. Nunc ut eros sed pede pretium congue. Etiam in elit ut nisi ultrices hendrerit. Cras vulputate ultrices quam. Curabitur venenatis. Cras sed nisi.

Flex box

It allows to arrange flex items very easily

In flex box parent element behave like a flex-container and child elements behave like flex-items To make parent behave like flex-container

Add display:flex property

Container properties

Flex-direction	Row column row-reverse column-column-reverse, by default it is row
Flex-wrap	Wrap nowrap wrap-wrap
Flex-flow	It is shortcut for flex-direction and flex-wrap
Justify-content	Space-between center flex-start flex-end space-around by default flex-start
Align-items	Center top bottom baseline

Flex items properties

order	The items will be arranged in ascending order based on value of order property
Align-self	Center top bottom
Flex-grow	A number which determine how fast the size should grow compared to other elements
Flex-shrink	A number which determine how fast the size should shrink compared to other elements

IACSD

To make your page responsive In
CSS use media query
@media not|only mediatype and (expression){
 Rules for media query
}

Mediatype can be either screen or print scanner In
expression specify min-width and max-width

```
@media screen and (min-width:480px and max-width:764px){  
    body{  
        background-color:blue;  
        color:white;  
    }  
}  
  
@media screen and (min-width:765px){  
    body{  
        background-color:cyan;  
        color:blue;  
    }  
}
```

BootstrapJS

Download bootstrapjs from the following link
<https://getbootstrap.com/docs/4.6/getting-started/download/>

Javascript DOM

