Web Technology

Introduction to Web Technology

What is WWW?

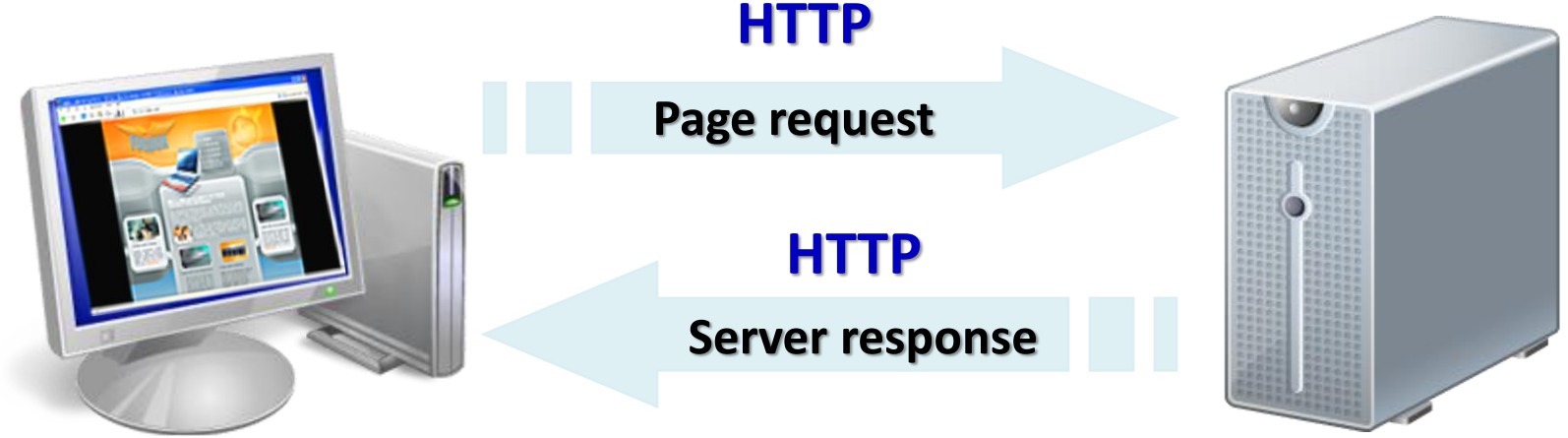
- WWW stands for **W**orld **W**ide **W**eb.
- A technical definition of the WWW is – All the resources and users on the Internet that are using HTTP.
- HTTP stands for **H**yper**t**ext **T**ransfer **P**rotocol.
- This is the protocol being used to transfer hypertext documents that makes the World Wide Web possible.

How the Web Works?

- World Wide Web (WWW) use classical client / server architecture
  - **H**yper **T**ext **T**ransfer **P**rotocol is text-based request-response protocol

Client running a Web Browser

Server running Web Server Software (IIS, Apache, etc.)



What is the Internet?

- The Internet is a massive **network of networks**, a networking infrastructure.
- It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the Internet.
- Information that travels over the Internet uses many different set of rules which are known as **protocols**.

Web Browsers

- A web browser or Internet browser is a **software application** for **retrieving**, **presenting**, and **traversing** information resources on the **World Wide Web**.
- Example :


Mozila Firefox


Google Chrome


Opera


Internet Explorer

- An *information resource* is identified by a Uniform Resource Identifier (**URI**) and may be a web page, image, video, or other piece of content.
- Hyperlinks present in resources enable users to easily navigate their browsers to related resources

Web Technology

Unit-1
HTML

Outline

1. Introduction to HTML
   - What is a Web Page?
   - My First HTML Page
   - HTML Code Formatting
2. Basic HTML Tags
   - Heading
   - Paragraph
   - Color
   - Font
   - List
   - Anchor
   - Image
3. HTML Tables
4. HTML Forms
5. XHTML
6. Introduction to HTML 5

What is a Web Page?

- Web page is text file containing HTML
- HTML – Hyper Text Markup Language
  - A notation for describing
    - document structure (semantic markup)
    - formatting (presentation markup)
- The markup tags provide information about the page content structure

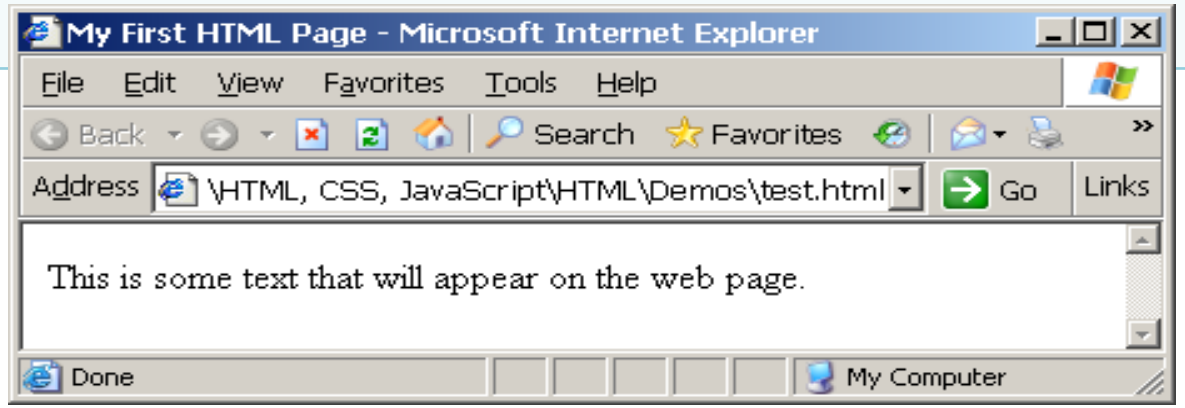
Creating HTML Pages

- An HTML file must have an `.htm` or `.html` file extension
- HTML files can be created with text editors:
  - NotePad, NotePad ++, PSPad
- Or HTML editors (WYSIWYG Editors):
  - Microsoft FrontPage
  - Macromedia Dreamweaver
  - Netscape Composer
  - Visual Studio

First HTML Page

test.html

```
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
     <p>This is some text...</p>
  </body>
</html>
```



HTML Structure

- HTML is comprised of “elements” and “tags”
  - Begins with `<html>` and ends with `</html>`
- Elements (tags) are nested one inside another:

```
<html> <head></head> <body></body> </html>
```

- Tags have attributes:

```
<img src="logo.jpg" alt="logo" />
```

- HTML describes structure using two main sections: `<head>` and `<body>`

HTML Code Formatting

- The HTML source code should be formatted to increase readability and facilitate debugging.
  - Every block element should start on a new line.
  - Every nested (block) element should be indented.
  - Browsers ignore multiple whitespaces in the page source, so formatting is harmless.
- For performance reasons, formatting can be sacrificed

First HTML Page: Tags

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
     <p>This is some text...</p>
  </body>
</html>
```

Opening tag

Closing tag

An HTML element consists of an opening tag, a closing tag and the content inside.

First HTML Page: Header

HTML header

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
     <p>This is some text...</p>
  </body>
</html>
```

First HTML Page: Body

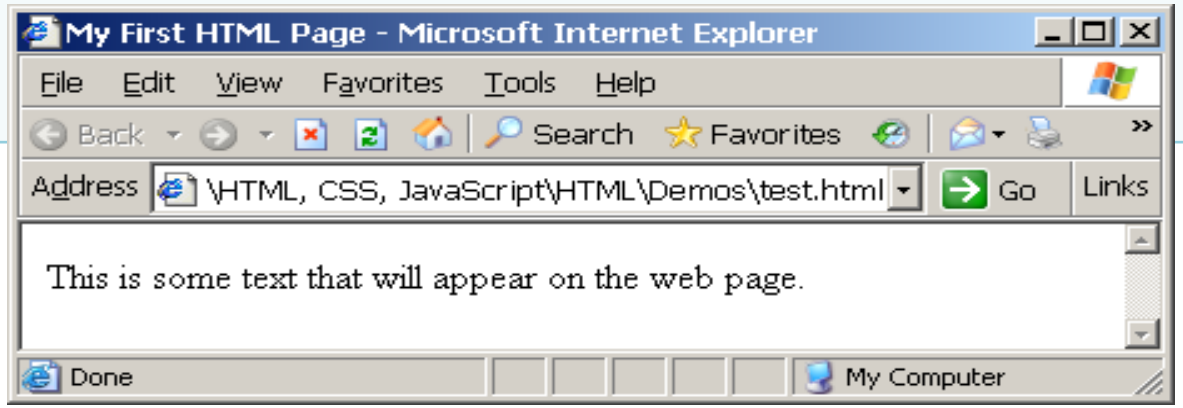
```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is some text...</p>
  </body>
</html>
```

HTML body

First HTML Page

test.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
     <p>This is some text...</p>
  </body>
</html>
```



Basic HTML Tags

1. Headings
2. Paragraph
3. Colors
4. Fonts
5. List
6. Anchor Tag
7. Image
8. Table
9. Form

1) Headings

- Headings are important because search engines use the headings to index the structure and content of your web pages.

<h1> text </h1> -- largest of the six
<h2> text </h2>
<h3> text </h3>
<h4> text </h4>
<h5> text </h5>
<h6> text </h6> -- smallest of the six

align=*"position"* --left (default), center or right

2) paragraph

- defines a paragraph
- Add **align**=*"position"* (left, center, right)
- Multiple 's do not create blank lines
- Use <br> for blank line
- Fully-specified text uses and , but is optional

3) Colors

- Values for **bgcolor** and **color**
  - many are predefined (red, blue, green, ...)
  - all colors can be specified as a six character hexadecimal value: #RRGGBB
  - #FF0000 – red
  - #888888 – gray
  - #00FF00 –green
  - #000000 – black

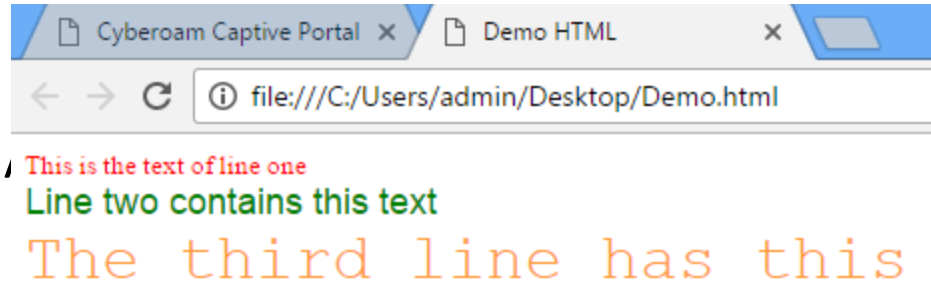
4) Fonts

- The <font> tag specifies the font face, font size, and color of text.
- The <font> tag is not supported in HTML5.

```
<font color="red" size="2" face="Times Roman">
This is the text of line one </font>

<font color="green" size="4" face="
Line two contains this text </font>

<font color="#FF9933" size="6" face="Courier“>
The third line has this additional text </font>
```



5) List

Ordered List

1. Block-A
2. Block-B
3. Block-C
4. Block-D

a) Block-A
b) Block-B
c) Block-C
d) Block-D

A. Block-A
B. Block-B
C. Block-C
D. Block-D

i. Block-A
ii. Block-B
iii. Block-C
iv. Block-D

I. Block-A
II. Block-B
III. Block-C
IV. Block-D

Unordered List

- Block-A
- Block-B
- Block-C
- Block-D

- Block-A
- Block-B
- Block-C
- Block-D

- Block-A
- Block-B
- Block-C
- Block-D

5.1) Ordered List

```
<ol>
  <li> Item one </li>
  <li> Item two </li>
  <ol type="I" >
    <li> Sublist item one </li>
    <li> Sublist item two </li>
    <ol type="i">
      <li> Sub-sub list item one </li>
      <li> Sub-sub list item two </li>
    </ol>
  </ol>
</ol>
```

Types:

Type = 1 (default)
Type = a
Type = A
Type = I
Type = i

Output

1. Item one
2. Item two
    I. Sublist item one
    II. Sublist item two
        i. Sub-sub list item one
        ii. Sub-sub list item two

5.2) Unordered List

```
<ul>
  <li> One </li>
  <li> Two </li>
  <ul type="circle">
    <li> Three </li>
    <li> Four </li>
    <ul type="square">
      <li> Five </li>
      <li> Six </li>
    </ul>
  </ul>
</ul>
```

Types:

Type = disc (default)
Type = circle
Type = square

Output

- One
- Two
  - Three
  - Four
    - Five
    - Six

6) <a> Anchor Tag (Hyperlinks)

- The <a> tag defines a hyperlink, which is used to link from one page to another.

Link to an absolute URL:
If you get spam, contact
Microsoft to report the problem.

Link to a relative URL:
See these references
concerning our fine products.

Link to a section within a URL:

Reference Section.

7) Images

- Syntax :

- **src** is required
- **alt** will specify the text to display if the Image not found
- **width**, **height** may be in units of pixels or percentage of page or frame
  - width="357"
  - height="50%"

Images (cont.)

<img src="dolphin.jpg" align="left" width="150" height="150" alt="dolphin jump!">

align=*position*	Image/Text Placement
Left	Image on left edge; text flows to right of image
Right	Image on right edge; text flows to left
Top	Image is left; words align with top of image
Bottom	Image is left; words align with bottom of image
Middle	Words align with middle of image

Image (cont.) **=> align="bottom"**

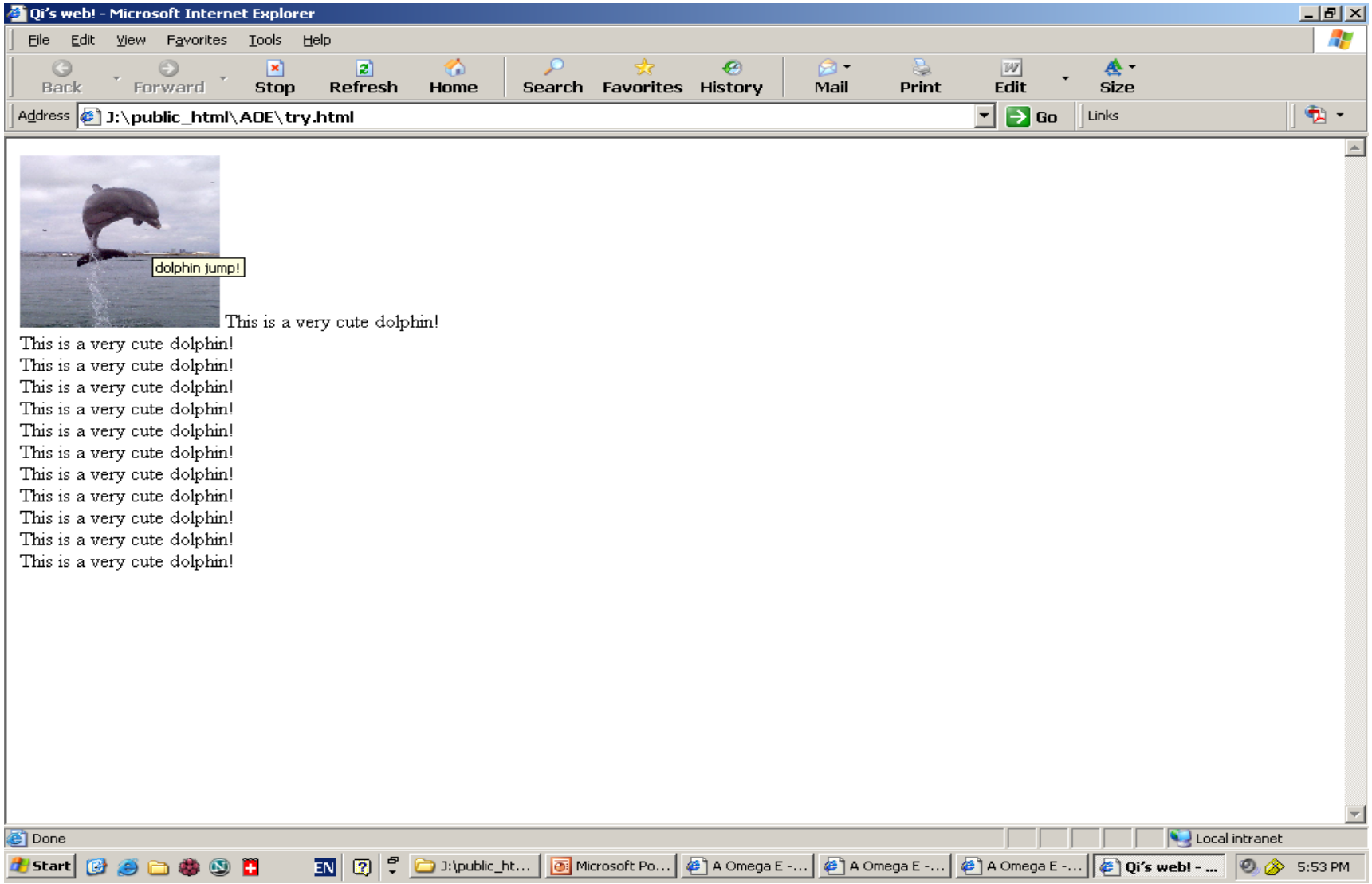
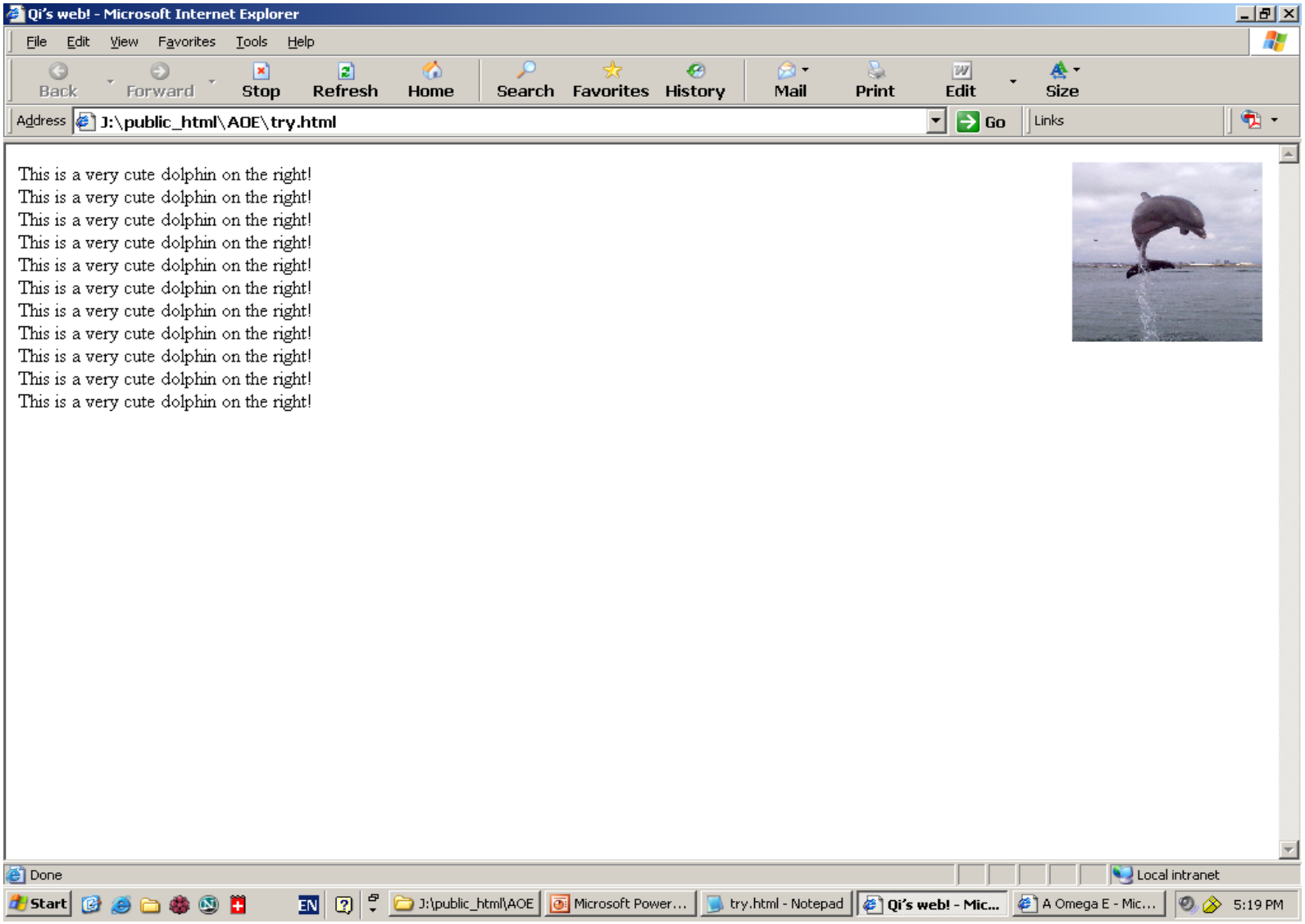


Image (cont.) => align="right"



8) Table

<table> table tag
<caption> optional table title
<tr> table row
olumn header
ata element

```
<table border=1>
 <caption>
 <tr>
     <th>H
     <th>H
 </tr>
 <tr>
     <td>R
     <td>R
 </tr>
 <tr>
     <td>Row2 Col1 Data</td>
     <td>Row2 Col2 Data</td>
 </tr>
</table>
```

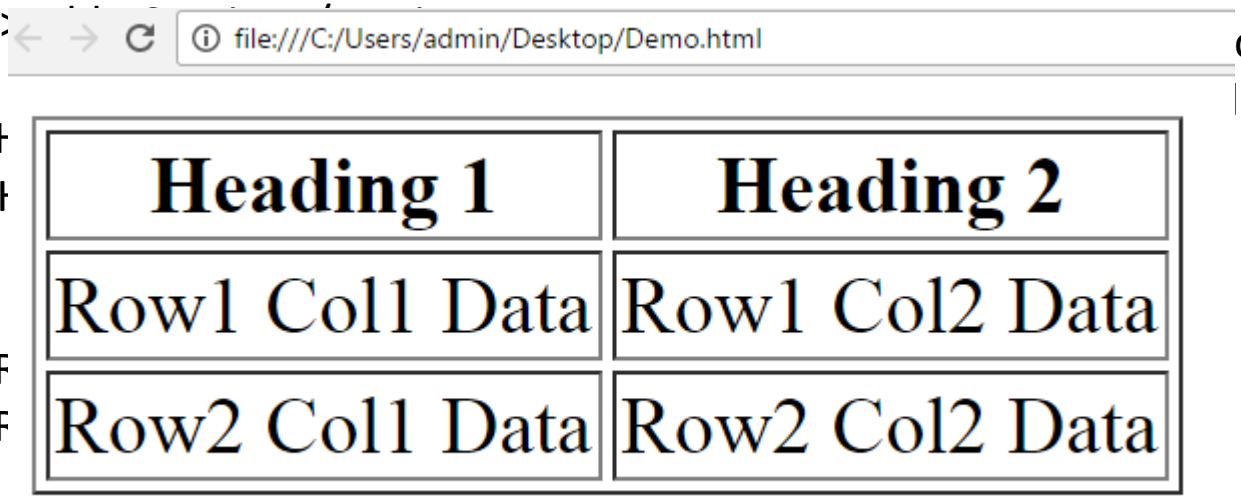
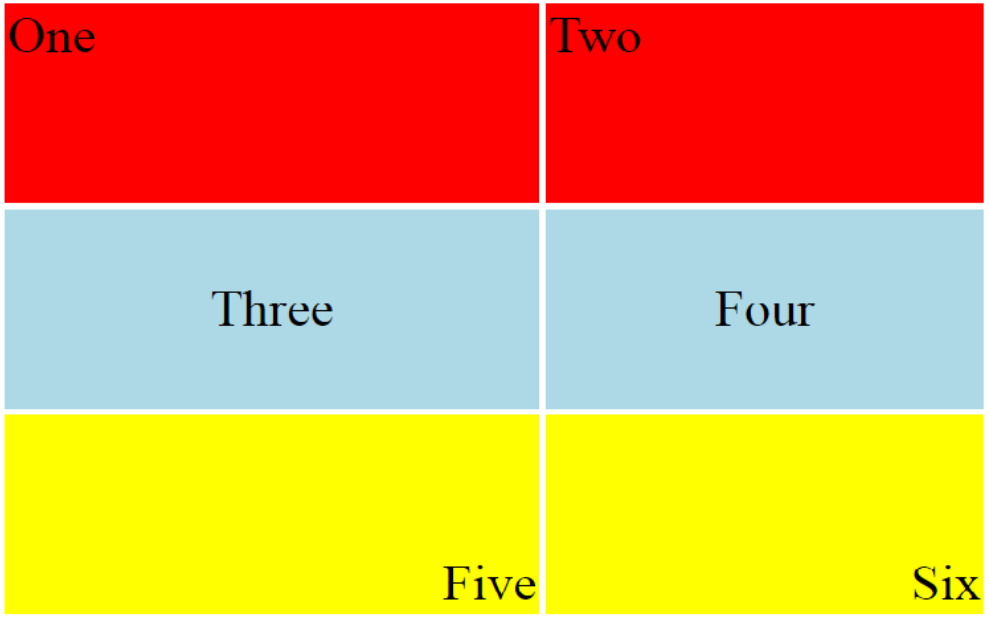


Table Element Attributes

- **align=*position*** -- left, center, right for table
- **border=*number*** -- width in pixels of border (default 0)
- **cellspacing=*number*** -- spacing in pixels between cells, default about 3
- **cellpadding=*number*** -- space in pixels between cell border and table element, default about 1
- **width=*number[%]*** -- width in pixels or percentage of page/frame width

Table Row <tr> Attributes

Valid for the table row:
align -- left, center, right
valign -- top, middle, bottom
bgcolor -- background color




```
<table align="center" width="300" height="200">
        <tr align="left" valign="top" bgcolor="red">
                <td>One</td>
                <td>Two</td>
        </tr>
        <tr align="center" valign="middle" bgcolor="lightblue">
                <td>Three</td>
                <td>Four</td>
        </tr>
        <tr align="right" valign="bottom" bgcolor="yellow">
                <td>Five</td>
                <td>Six</td>
        </tr>
</table>
```

Irregular Table

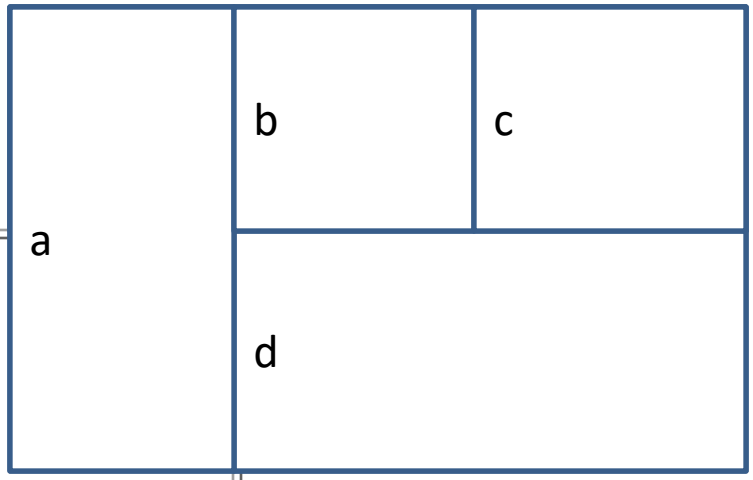
Valid for the table cell:

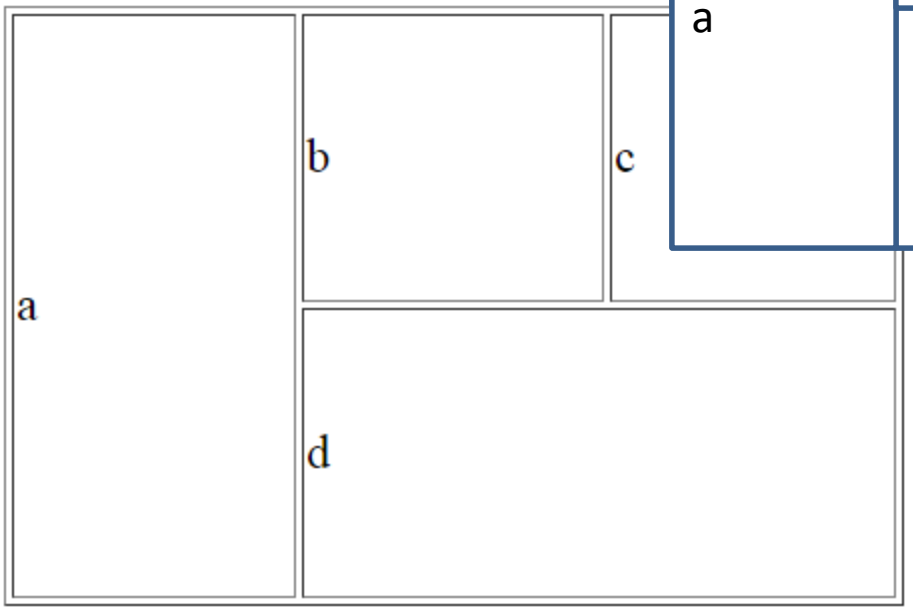
colspan

- how many columns this cell occupies

rowspan

- how many rov

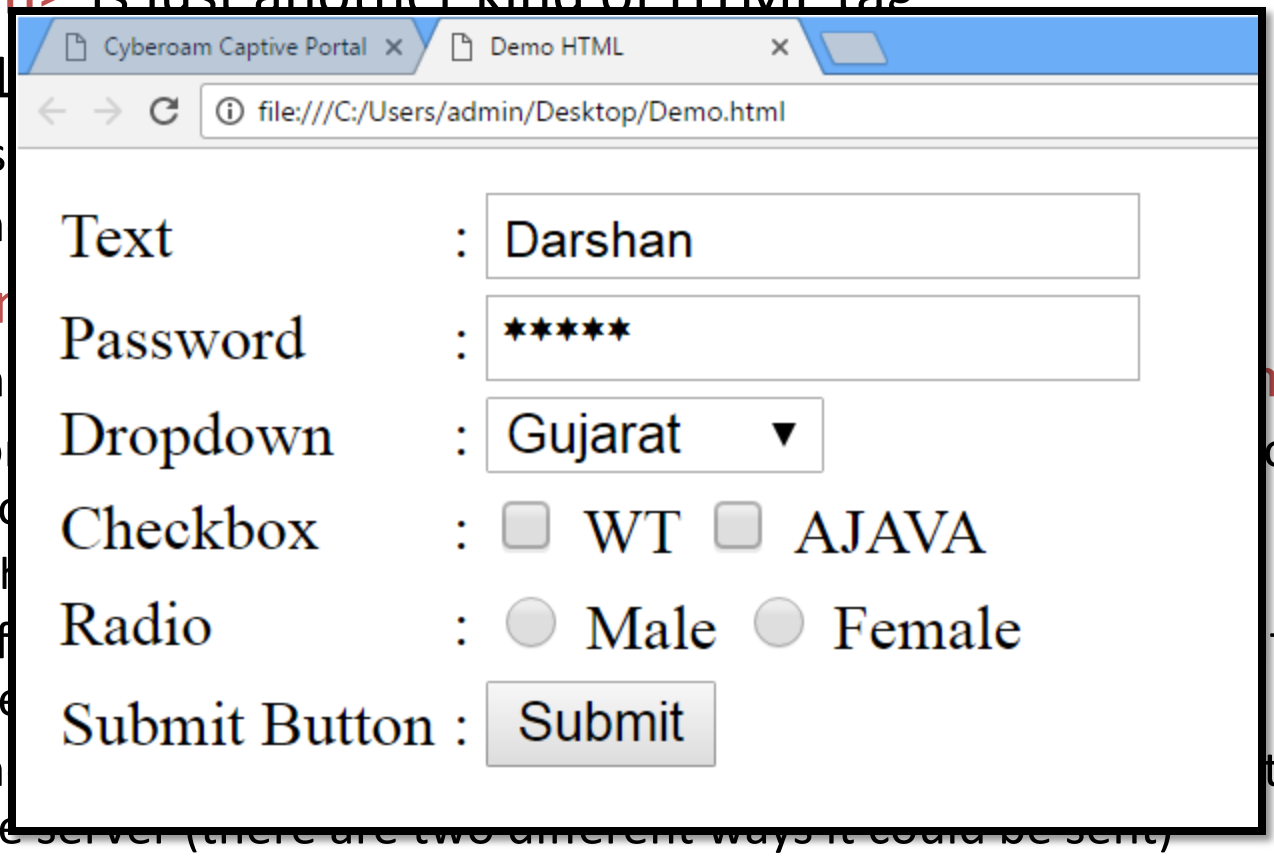





```
<table align="cente
<tr>
        <td colsp
        <td colsp
        <td colsp
</tr>
<tr>
        <td colspan=“2"  rowspan="1">d</td>
</tr>
</table>
```

9) HTML Form

- <form> is just another kind of HTML tag
- HTML
  - Us
  - Th
- A forr
  - Th
  - Fo
  - Oth
  - A f
  - Th



The <form> Tag

- The <form ***arguments***> … </form> tag encloses form elements (and probably other HTML as well)
- The arguments to form tell what to do with the user input
  - **action="*url*" (required)**
    - Specifies where to send the data when the Submit button is clicked
  - **method="get" (default)**
    - Form data is sent as a URL with ?form_data info appended to the end
    - Can be used *only* if data is all ASCII and not more than 100 characters
  - **method="post"**
    - Form data is sent in the body of the URL request
    - Cannot be bookmarked by most browsers
  - **target="*target*"**
    - Tells where to open the page sent as a result of the request
    - ***target*= _blank** means open in a new window
    - ***target*= _top** means use the same window

Input tags

- Text field
  - Example: <input type="text" name="inputname"/>

- Password field
  - Example: <input type="password" name="inputname"/>

- Radio buttons
  - Example:

    <input type="radio" name="gender"> Male

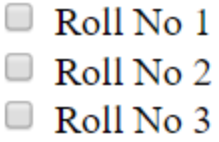
    <input type="radio" name="gender"> Female

- Check boxes
  - Example:

    <input type="checkbox" name="Roll1"> Roll No 1

    <input type="checkbox" name="Roll2"> Roll No 2

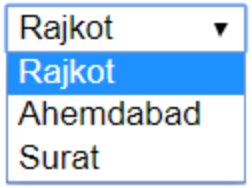
    <input type="checkbox" name="Roll3"> Roll No 3



Input tags (cont.)

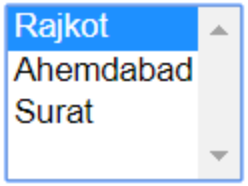
- Dropdown list
  - <select> tag is used to create a drop-down list in HTML.
  - <option> tags inside the <select> tag define the available options in the list.
  - Example:

```
<select>
    <option value="1">Rajkot</option>
    <option value="2">Ahemdabad</option>
    <option value="3">Surat</option>
</select>
```



  - Example (multiple select):

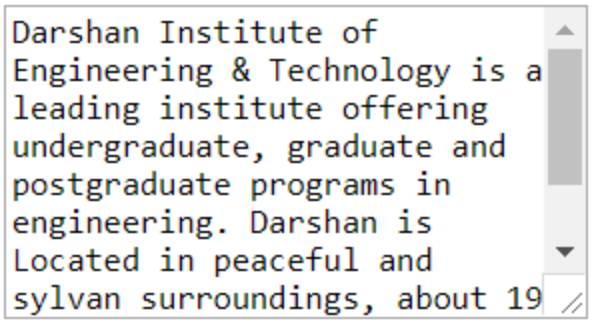
```
<select multiple="multiple">
    <option value="1">Rajkot</option>
    <option value="2">Ahemdabad</option>
    <option value="3">Surat</option>
</select>
```



Input tags (cont.)

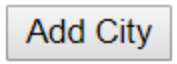
- Text area
  - <textarea> tag defines a multi-line text input control.
  - Example :

```
<textarea rows="8" cols="30">
    Darshan Institute of Engineering & Technology is a leading institute ….
</textarea>
```



- Submit Button
  - Submit button is used to submit the data to the form action url.
    - Example :

```
<input type="submit" value="Add City">
```



HTML Images

- Images can improve the design and the appearance of a web page.
- The HTML <img> tag is used to embed an image in a web page.
- Images are not technically inserted into a web page; images are linked to web pages. The <img> tag creates a holding space for the referenced image.
- The <img> tag is empty, it contains attributes only, and does not have a closing tag.
- The <img> tag has two required attributes:
  - src - Specifies the path to the image
  - alt - Specifies an alternate text for the image
- Syntax

<img src="url" alt="alternatetext">

HTML Images

- **Image Size - Width and Height**

You can use the style attribute to specify the width and height of an image.

<img src="img.jpg" alt="jacket“ style="width:500px;height:600px;">

Alternatively, you can use the width and height attributes:

<img src="img.jpg" alt="jacket" width="500" height="600">

- **Image as a Link**

To use an image as a link, put the <img> tag inside the <a> tag:

<img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;">

HTML Image Maps

- With HTML image maps, we can create clickable areas on an image.
- The HTML <map> tag defines an image map. An image map is an image with clickable areas. The areas are defined with one or more <area> tags.
- Example:
- Here is the HTML source code for the image map :

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">
<map name="workmap">
<area shape="rect" coords="34,44,270,350" alt="Computer“ href="computer.htm">
<area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
<area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">
</map>
```

HTML Image Maps

- **How Does it Work?**

- The idea behind an image map is that you should be able to perform different actions depending on where in the image you click.
- To create an image map you need an image, and some HTML code that describes the clickable areas.
- The image is inserted using the <img> tag. The only difference from other images is that you must add a usemap attribute:

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">
```

- The usemap value starts with a hash tag # followed by the name of the image map, and is used to create a relationship between the image and the image map.

HTML Image Maps

- **Create Image Map**

➢ add a <map> element.

➢ The <map> element is used to create an image map, and is linked to the image by using the required name attribute:

<map name="workmap">

➢ The name attribute must have the same value as the <img>'s usemap attribute .

- **The Areas**

➢ Then, add the clickable areas.

➢ A clickable area is defined using an <area> element.

- Shape

➢ You must define the shape of the clickable area, and you can choose one of these values:

- rect - defines a rectangular region
- circle - defines a circular region
- poly - defines a polygonal region
- default - defines the entire region

➢ You must also define some coordinates to be able to place the clickable area onto the image.

HTML Image Maps

- **Shape="rect"**

- The coordinates for shape="rect" come in pairs, one for the x-axis and one for the y-axis.
- For example:
- So, the coordinates 34,44 is located 34 pixels from the left margin and 44 pixels from the top:
- The coordinates 270,350 is located 270 pixels from the left margin and 350 pixels from the top:

```
<area shape="rect" coords="34, 44, 270, 350" href="computer.html">
```

- This is the area that becomes clickable and will send the user to the page "computer.htm":

HTML Image Maps

- **Shape="circle"**

- To add a circle area, first locate the coordinates of the center of the circle:
- For example: 337,300
- Then specify the radius of the circle: 44 pixels

<area shape="circle" coords="337, 300, 44" href="coffee.html">

- **Shape="poly"**

- The shape="poly" contains several coordinate points, which creates a shape formed with straight lines (a polygon).
- This can be used to create any shape.

<area shape="poly"
coords="140,121,181,116,204,160,204,222,191,270,140,329,85,355,58,352,37,322,40,259,103,161,128,147" href="croissant.html">

Image Map and JavaScript

- A clickable area can also trigger a JavaScript function.
- Add a click event to the <area> element to execute a JavaScript function:

```
<map name="workmap">
<area shape="circle" coords="337,300,44" href="coffee.htm"
onclick="myFunction()">
</map>
<script>
function myFunction() {
    alert("You clicked the coffee cup!");
}
</script>
```

FRAMES IN HTML

index

- Introduction
- Disadvantages of Frames
- Create the frames in HTML
- Example
- <Frameset> tag Attributes
- <Frame> tag Attributes

Introduction

- HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Disadvantages of Frames

- There are few drawbacks with using frames, so it's never recommended to use frames in your webpages:
- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's *back button* might not work as the user hopes. There
- are still few browsers that do not support frame technology.

Creating Frames

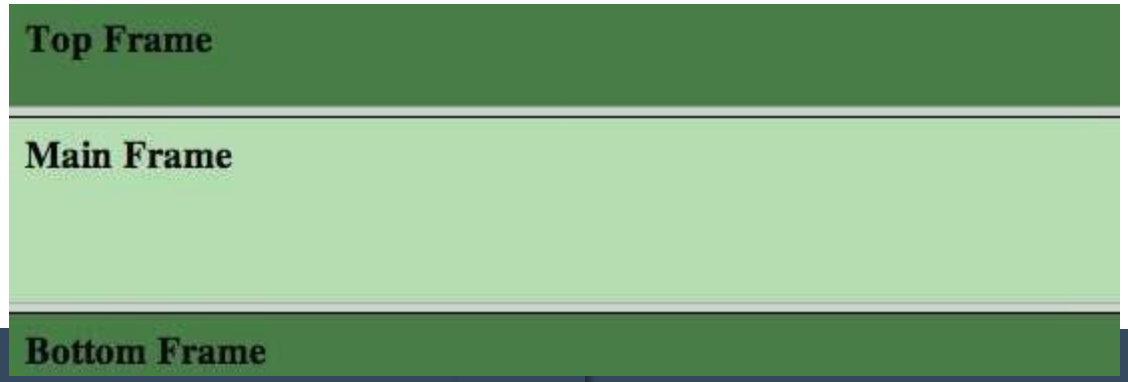
- To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines how to divide the window into frames. The **rows** attribute of <frameset> tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Example

- Following is the example to create three horizontal frames:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset rows="10%,80%,10%">
      <frame name="top" src="/html/top_frame.htm" />
      <frame name="main" src="/html/main_frame.htm" />
      <frame name="bottom" src="/html/bottom_frame.htm" />
      <noframes>
      <body>
            Your browser does not support frames.
      </body>
      </noframes>
</frameset>
</html>
```

This will produce following result:



The <frameset> Tag Attributes

- Following are important attributes of the <frameset> tag
- **Cols attribute:**
- specifies how many columns are contained in the frameset and the size of each column.You can specify the width of each column in one of four ways:
- Absolute values in pixels. For example to create three vertical frames, use
- *cols="100, 500,100"*.
- A percentage of the browser window. For example to create three vertical frames, use *cols="10%, 80%,10%"*.
- Using a wildcard symbol. For example to create three vertical frames, use
- *cols="10%,*,10%"*. In this case wildcard takes remainder of the window.
- As relative widths of the browser window. For example to create three vertical frames, use *cols="3*,2*,1*"*.This is an alternative to percentages.You can use relative widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.
- **Rows attribute:**
- This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows in the frameset. For example to create two horizontal frames, use *rows="10%, 90%"*.You can specify the height of each row in thes ame way as explained above for columns.

- **Border**
- This attribute specifies the width of the border of each frame in pixels. For example border="5". Avalue of zero means no border.
- **frameborder**
- This attribute specifies whether a three-dimensional border should be displayed between frames.This attrubute takes value either 1 *yes* or 0 *no*. For example frameborder="0" specifies no border.
- **framespacing**
- This attribute specifies the amount of space between frames in a frameset.This can take any integer value. For example framespacing="10" means there should be 10 pixels spacing between each frames.

The <frame> Tag Attributes

- src
- This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL. For example, src="/html/top_frame.htm" will load an HTML file available in html directory.
- name
- This attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into.This is especially important when you want to create links in one frame that load pages into an another frame, in which case the second frame needs a name to identify itself as the target of the link.
- frameborder
- This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> tag if one is given, and this can take values either 1 *yes* or 0 *no*.

- **marginwidth**
- This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example marginwidth="10".
- **marginheight**
- This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents.The value is given in pixels. For example marginheight="10".
- **noresize**
- By default you can resize any frame by clicking and dragging on the borders of a frame.The noresize attribute prevents a user from being able to resize the frame. For example noresize="noresize".

- **scrolling**
- This attribute controls the appearance of the scrollbars that appear on the frame.This takes values either "yes", "no" or "auto". For example scrolling="no" means it should not have scroll bars.
- **Longdesc**
- This attribute allows you to provide a link to another page containing a long description of the contents of the frame. For example longdesc="framedescription.htm"

?