

Chapter Two

Introduction to Hyper Text Markup
Language (HTML) and XML?

Markup language

- **Markup languages** are computer languages that are used to structure, format, or define relationships between different parts of text documents with the help of symbols or **tags** inserted in the document.
- These languages are more readable than usual programming languages. Basic Example of markup language HTML,XML,XHTML etc.

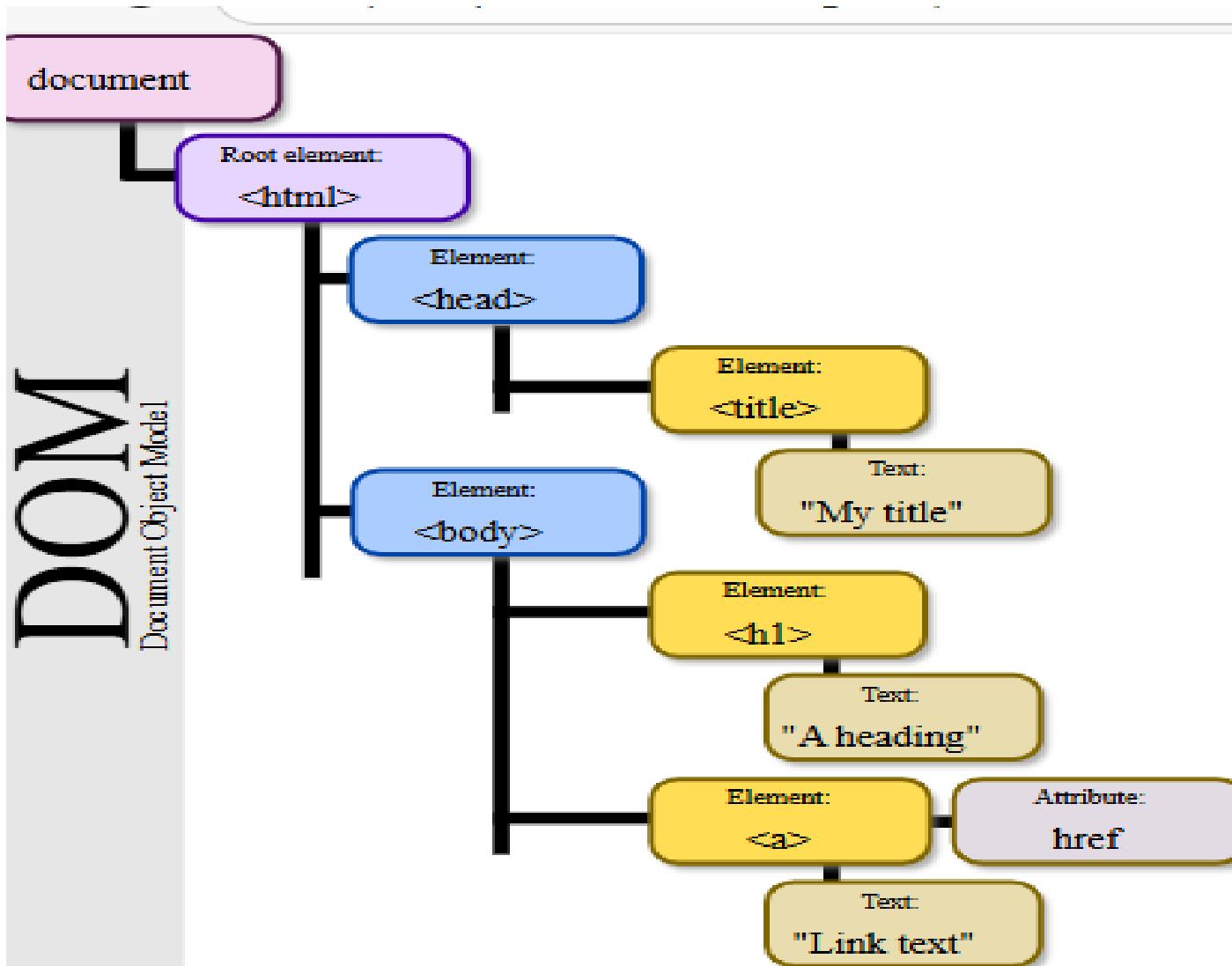
Document Object Model (DOM) of HTML

- The HTML Document Object Model (HTML DOM) defines a standard way for accessing and manipulating HTML documents.
- It presents an HTML document as a tree-structure (a node tree), with elements, attributes, and text.
- Each HTML tag becomes a node in the hierarchy. At the top, the `<html>` tag is the root element, containing both `<head>` and `<body>` as child elements.

Cont,

```
<html>
<head>
<title> My title </title>
</head>
<body>
<h1> A heading </h1>
<a href="http://....> link text </a>
</body>
</html>
```

Cont.



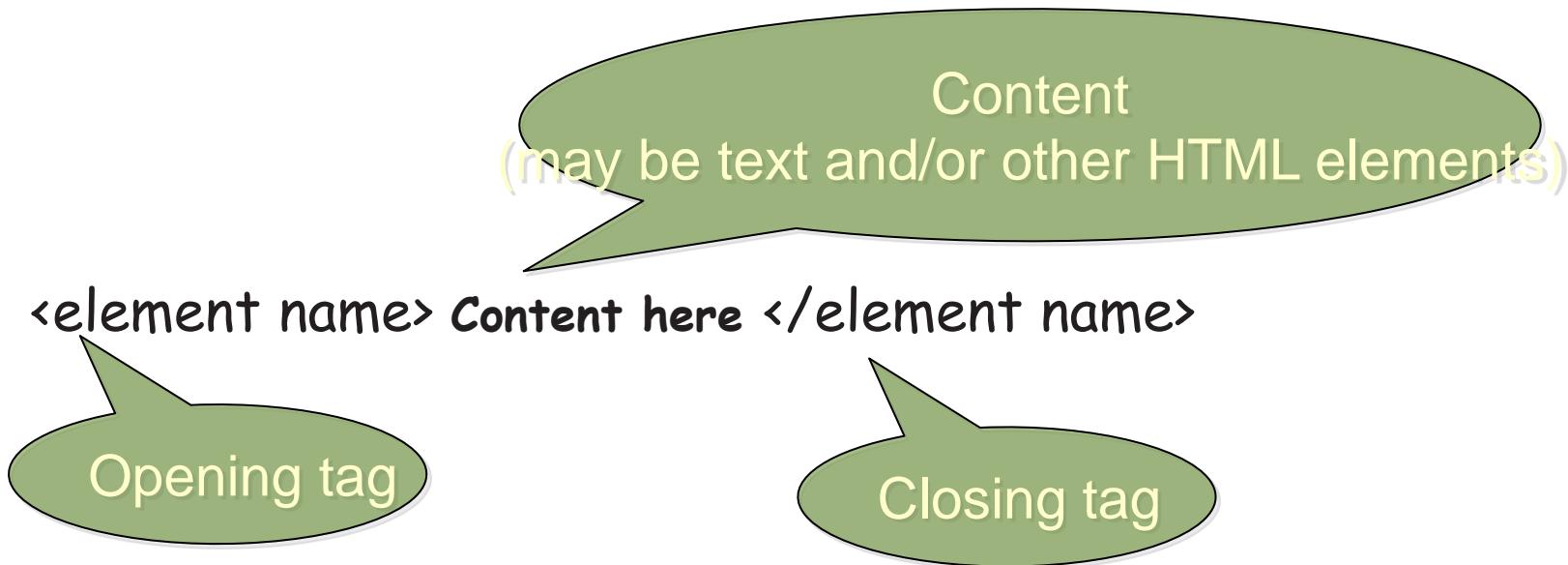
HTML (Hyper Text Markup Language)

- is a markup language (markup -instructions for layout and style)
 - is used to tell a Web browser
 - where the heading is for a Web page,
 - what is a paragraph,
 - what is part of a table and so on, so it can structure your document and render it properly.
- is not programming

Syntax:

`<element name> Content here </element name>`

The structure of an HTML element



It is recommended to use **lowercase** in all elements since required for XHTML documents.

The structure of an HTML element



The minimal structure of an (X)HTML document:

- ① Identifies the document as written in HTML or XHTML
- ② The head provides information about the document
- ③ A descriptive title is required
- ④ The body contains the content that displays in the browser

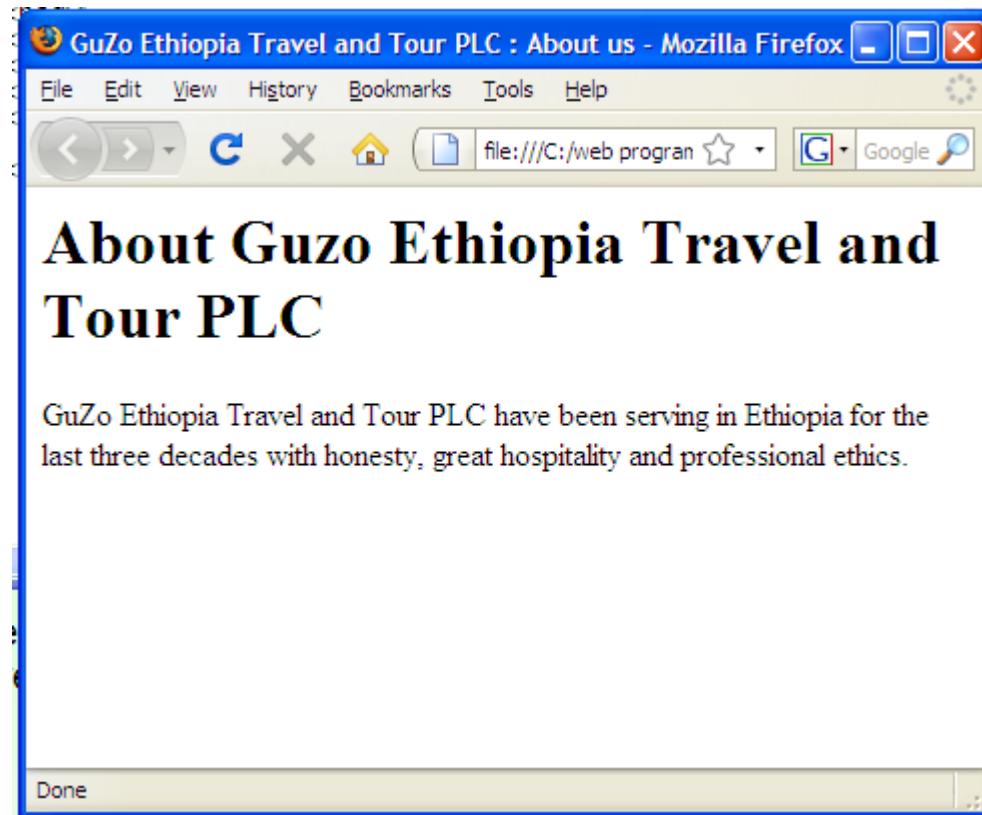
Simple page

```
<html>
  <head>
    <title> GuZo Ethiopia Travel and Tour PLC : About us
  </title>
  </head>

  <body>
    <h1>About Guzo Ethiopia Travel and Tour PLC</h1>
    <p>GuZo Ethiopia Travel and Tour PLC have been
    serving in Ethiopia for the last three decades with honesty, great
    hospitality and professional ethics.  <p>
  </body>

</html>
```

Simple page



Looking at the code

```
<html>
  <head>
    <title> GuZo Ethiopia Travel and Tour PLC : About us
  </title>
  </head>

  <body>
    <h1>About Guzo Ethiopia Travel and Tour PLC</h1>
    <p>GuZo Ethiopia Travel and Tour PLC have been
serving in Ethiopia for the last three decades with honesty,
great hospitality and professional ethics. <p>
  </body>

</html>
```

Html opening tag

elem
ent

Html closing tag

Looking at the code

The diagram illustrates the structure of an HTML document. A red curly brace on the left groups the opening and closing tags of the title element. A callout bubble points to the start of the heading section with the text "This is the beginning of a heading". Another callout bubble points to the end of the heading section with the text "This is the end of a heading". The text within the body is partially faded.

```
<html>
  <head>
    <title> GuZo Ethiopia Travel and Tour PLC : About us
  </head>
  <body>
    <h1>About GuZo Ethiopia Travel and Tour PLC</h1>
    <p>GuZo Ethiopia Travel and Tour PLC have been
      serving in Ethiopia for the last three decades with honesty,
      great hospitality and professional ethics. <p>
  </body>
</html>
```

Looking at the code

```
<html>
  <head>
    <title> GuZo Ethiopia Travel and Tour PLC : About us
  </title>
  </head>
  <body>
    <h1>About Guzo Ethiopia Travel and Tour PLC</h1>
    <p>GuZo Ethiopia Travel and Tour PLC have been
    serving in Ethiopia for the last three decades with honesty,
    great hospitality and professional ethics. <p>
  </body>
</html>
```

The main body starts here

The main body ends here

elem
ent

Looking at the code

```
<html>
  <head>
    <title> GuZo Ethiopia Travel and Tour PLC : About us
  </title>
  </head>
  <body>
    <h1>About Guzo Ethiopia Travel and Tour PLC</h1>
    <p>GuZo Ethiopia Travel and Tour PLC have been
serving in Ethiopia for the last three decades with honesty,
great hospitality and professional ethics. <p>
  </body>

</html>
```

Header one starts here

*In HTML and XHTML
there are
six levels of
headings.*

Header one ends here

Looking at the code

```
<html>
    <head>
        <title> GuZo Ethiopia Travel and Tour PLC : About us
</title>
    </head>
    <body>
        <h1>About Guzo Ethiopia Travel and Tour PLC</h1>
        <p>GuZo Ethiopia Travel and Tour PLC have been
serving in Ethiopia for the last three decades with honesty,
great hospitality and professional ethics. </p>
    </body>
</html>
```

paragraph starts here

paragraph ends here

Adding style

- Adding styles include:
 - **Adding font types**
 - Making fonts **Blod** , *italic* or underline
 - **Colors**
 - Alignment
 - Change background color or picture
 - etc

Adding styles

Colors are identified by six digit number

Specifies the back ground color of the whole page (green)

```
<html> GuZo Ethiopia Travel and Tour PLC : About us </title>
```

```
</head>
```

```
<body bgcolor="#71ec90">
```

Specifies the type of font for the heading

```
<font face="arial">
```

```
<h1>About Guzo Ethiopia Travel and Tour PLC</h1>
```

```
</font>
```

```
<font face="arial" color="cc2400">
```

Specifies the type of font for the body

```
<p>GuZo Ethiopia Travel and Tour PLC have been serving in
```

Ethiopia for the last three decades with honesty, great hospitality and professional ethics.<p>

```
</font>
```

```
</body>
```

```
</html>
```

The output of the style

Title displayed here



Elements and their Attributes

- - element
 - face - attribute
 - color - attribute
- Attributes say something about elements.
- Attributes have two parts
 - Name - the property to be set eg. face, color
 - Value - is the value of the property eg. Arial, 71ec90

An element should never have two attributes of the same name!! it will ignore the second one.

Keeping Style Separate from Structure and Semantics

- Structural markup
 - indicating the structure of a document
 - the paragraphs and headings
- Semantic markup
 - telling us something about the content of the data,
 - like the `<title>` element.
- Stylistic markup
 - indicates how the document should look
 - Eg ``, ``, and `<i>`

Structure of A page

- XHTML documents are contained between the opening <html> and closing </html> called **root element**
- Inside the root element the document is divided in to two
 - The **<head> element**, which contains information about the document
 - such as a title or a link to a style sheet
 - The **<body> element**, which contains the real content of the document that you see.

The <head> Element

The <head> Element can contain one of the following in any order

- <title> - to display the title of the page
- <object>, which is designed to include images, JavaScript objects, Flash animations, MP3 files, QuickTime movies and other components of a page.
- <link> to link to an external file, such as a style sheet or JavaScript file
- <style> to include CSS rules inside the document
- <script> for including script in the document
- <meta>, which includes information about the document such as keywords and a description, which are particularly helpful for search applications;

The <title> Element

- a title is needed for every page that you write inside the <title> element
- The title of a web page is used for
 - At the very top of a browser window
 - As the default name for a bookmark
 - By search engines that use its content to help index pages
- So it is important to write the title of the page which can describe it well.
 - Eg. "Guzo Ethiopia Travel and Tour : Contact us" describes the page more than just "Contact us"

Creating Headings using Hn Element

- In XHTML you have six levels of headings, which use the elements `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, and `<h6>`. Example:

```
<html>
<body>
<h1> Heading 1 </h1>
<p> The content under heading one goes here. </p>
<h2> Heading 2 </h2>
<p> The content under heading two goes here. </p>
<h3> Heading 3 </h3>
<p> the content under heading three goes here. </p>
<h4> Heading 4 </h4>
<p> The content under heading four goes here. </p>
<h5> Heading 5 </h5>
<p> The content under heading five goes here. </p>
<h6> Heading 6 </h6>
<p> The content under heading six goes here. </p>
</body>
</html>
```

Example

A screenshot of a Mozilla Firefox browser window. The title bar says "Mozilla Firefox". The menu bar includes "File", "Edit", "View", "History", "Bookmarks", "Tools", and "Help". The toolbar includes icons for back, forward, search, and other functions. The address bar shows two tabs: "file:///C:/wel...hitespace.html" and "file:///C:/w...ding1-6.html". The main content area displays the following text:

Heading 1
The content under heading one goes here.

Heading 2
The content under heading two goes here.

Heading 3
the content under heading three goes here.

Heading 4
The content under heading four goes here.

Heading 5
The content under heading five goes here.

Heading 6
The content under heading six goes here.

Heading

- The six heading elements can all carry the universal attributes as well as a deprecated attribute called align.
 - align, class, id, style, title, dir
- **The align Attribute (deprecated)**
 - <h1 align="left">Left-Aligned Heading</h1>

Example

```
<html>  
<body>  
<h1 align="left"> Heading Aligned Left </h1>  
<p> This is the default value of heading </p>  
<h2 align="center"> Heading Aligned Center </h2>  
<p> The content under heading goes here. </p>  
<h3 align="right"> Heading Aligned right</h3>  
<p> the content under heading  goes here. </p>  
</body>  
</html>
```

Paragraph <p> . . . </p>

- When a browser displays a paragraph it usually inserts a
 - new line before the next paragraph and
 - adds a little bit of extra vertical space
- eg.

...

```
<p>Here is a paragraph of text.</p>
```

```
<p>Here is a second paragraph of text.</p>
```

```
<p>Here is a third paragraph of text.</p>
```

...

Creating Line Breaks

- Whenever you use the `
` element, anything following it starts on the next line.

`
` (HTML)

- It does not need opening *and* closing tags

eg.

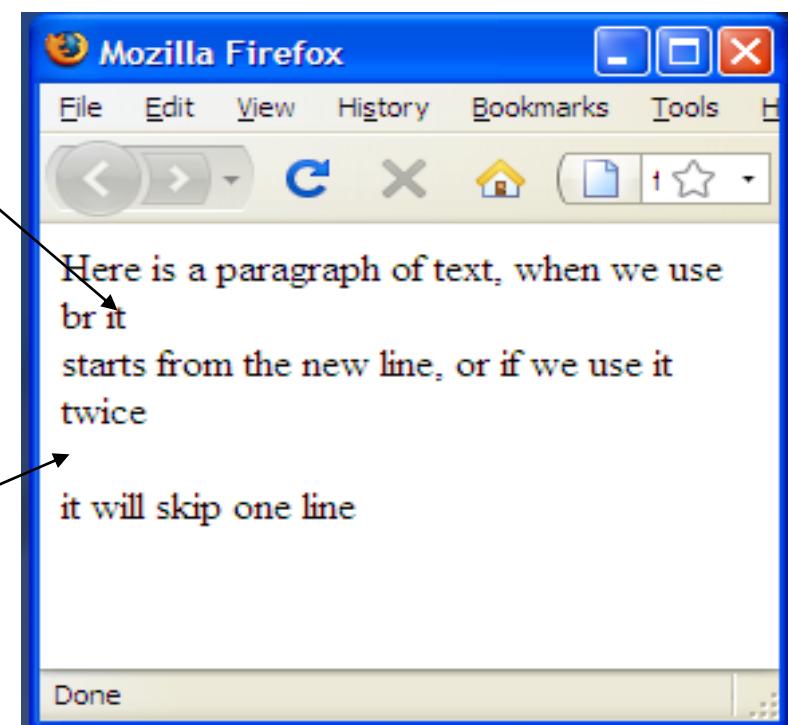
```
<html>
<body>

<p>Here is a paragraph of text,
when we use br it <br/>starts from
the new line, or if we use it twice
<br/><br/> it will skip one line </p>

</body>
</html>
```

`
`
used
once

`
`
used
twice



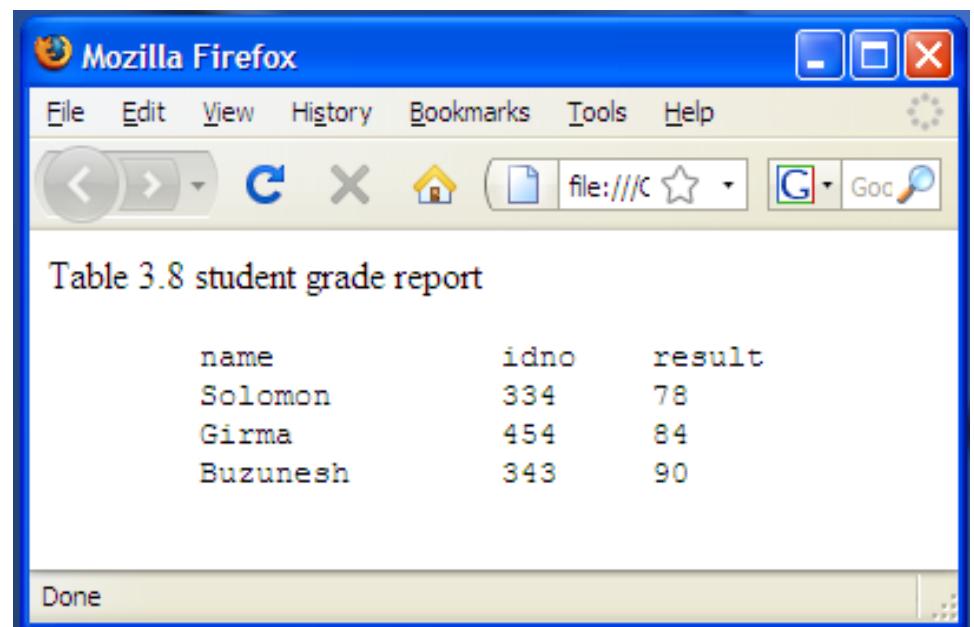
Creating Preformatted Text

- It will display the content as it is formated in the source document
- <pre> . . . </pre>
- most browsers would display this text in a monospaced font by default
 - Used to display tabular data without the use of a table(in which case you must use the monospaced font or columns will not align correctly) and
 - to represent computer source code.

Example on the next slide

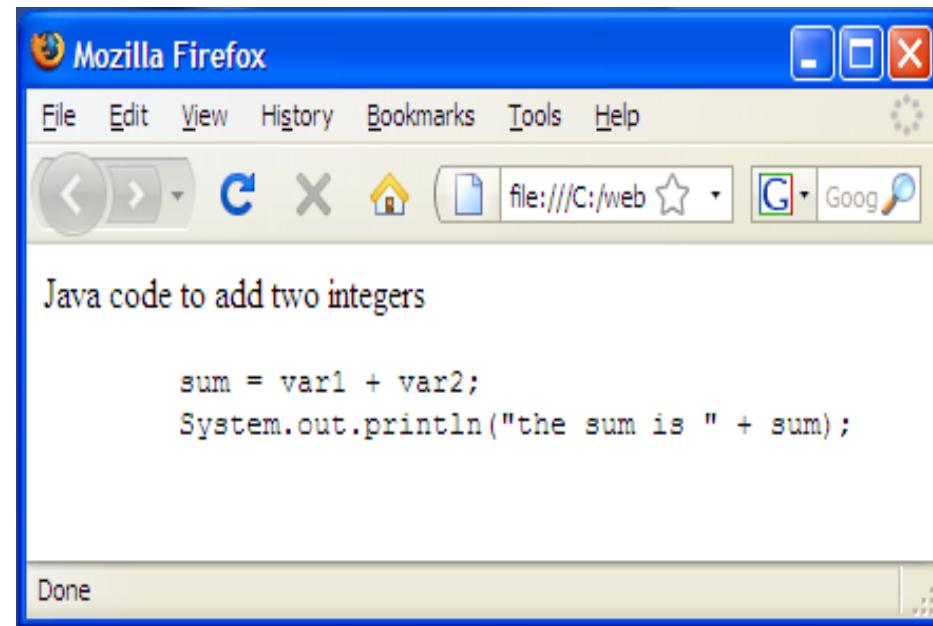
Example: tabular data using <pre> tag

```
<html>
<body>
<p> Table 3.8 student grade report
<pre>
name      idno      result
Solomon   334       78
Girma     454       84
Buzunesh  343       90
</pre>
</p>
</body>
</html>
```



Example: using <pre> tag to display source code

```
<html>
<body>
<p> Java code to add two integers
<pre>
sum = var1 + var2;
System.out.println("the sum is " + sum);
</pre>
</p>
</body>
</html>
```



Horizontal Rules

- Is a divider between sections
- Use `<hr>` html or `<hr />` xhtml

- eg.

```
<body>
```

```
<h3>Times</h3>
```

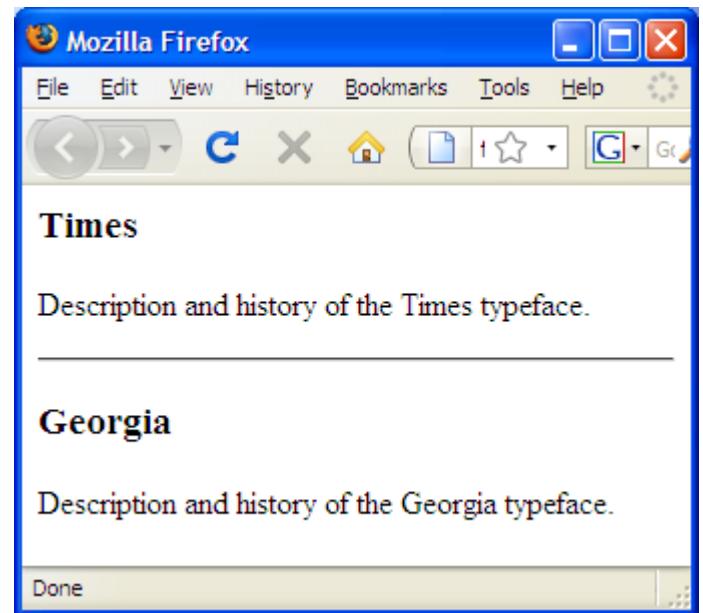
```
<p>Description and history of the Times  
typeface.</p>
```

```
<hr />
```

```
<h3>Georgia</h3>
```

```
<p>Description and history of the Georgia  
typeface.</p>
```

```
</body>
```



Address

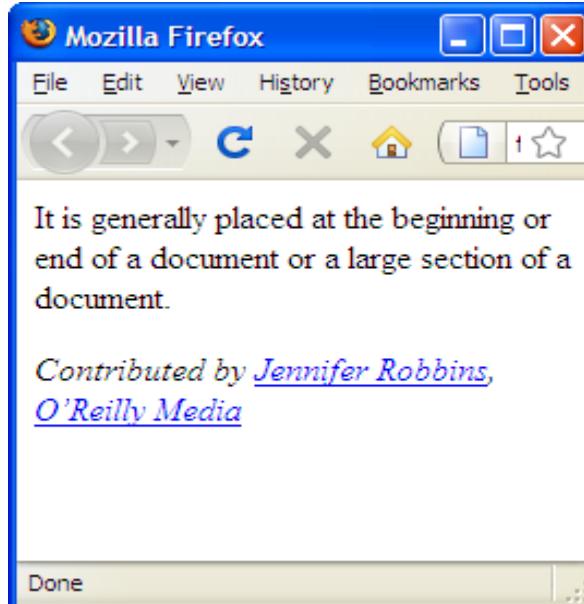
- the **address** element that is used to provide contact information for the author or maintainer of the document.

<address>

Contributed by Jennifer Robbins,

O'Reilly Media

</address>



Lists

- You can create three types of lists in XHTML:
 - **Unordered lists**, which are like lists of bullet points
 - **Ordered lists**, which use a sequence of numbers or letters instead of bullet points
 - **Definition lists**, which allow you to specify a term and its definition

Unordered lists

Each item in the list is marked with a bullet

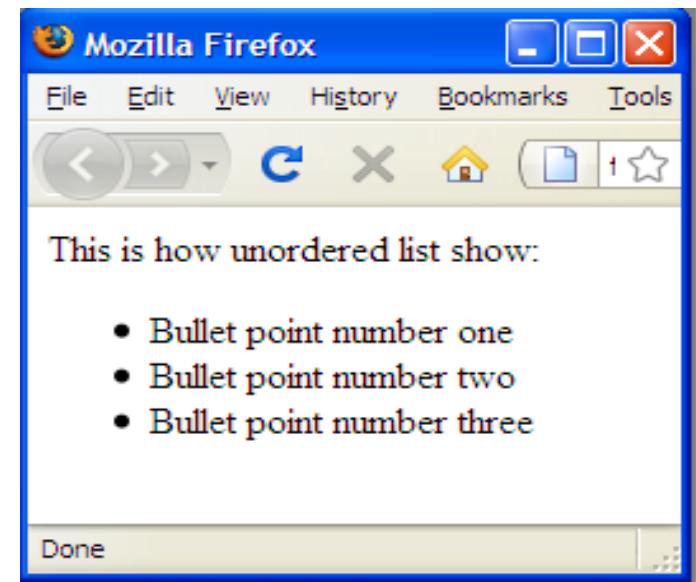
syntax:

```
<ul>  
<li> . . . </li>  
<li> . . . </li>  
</ul>
```

- eg.

<p>This is how unordered list show:

```
<ul>  
<li>Bullet point number one</li>  
<li>Bullet point number two</li>  
<li>Bullet point number three</li>  
</ul>  
</p>
```



Type Attribute:

- You can use type attribute for tag to specify the type of bullet you like. By default it is a disc.
Following are the possible options:

- <ul type="square">
- <ul type="disc">
- <ul type="circle">

example: Following is an example where we used <ul type="square">

```
<!DOCTYPE html>

<html>
<head>
<title>HTML Unordered List</title>
</head>
<body>
```

```
<ul type="square">  
<li>Beetroot</li> <li>Ginger</li>  
<li>Potato</li> <li>Radish</li>  
</ul> </body>  
</html>
```

- This will produce following result:
 - Beetroot
 - Ginger
 - Potato
 - Radish

Ordered lists

If you are required to put your items in a numbered list instead of bulleted then HTML ordered list will be used.

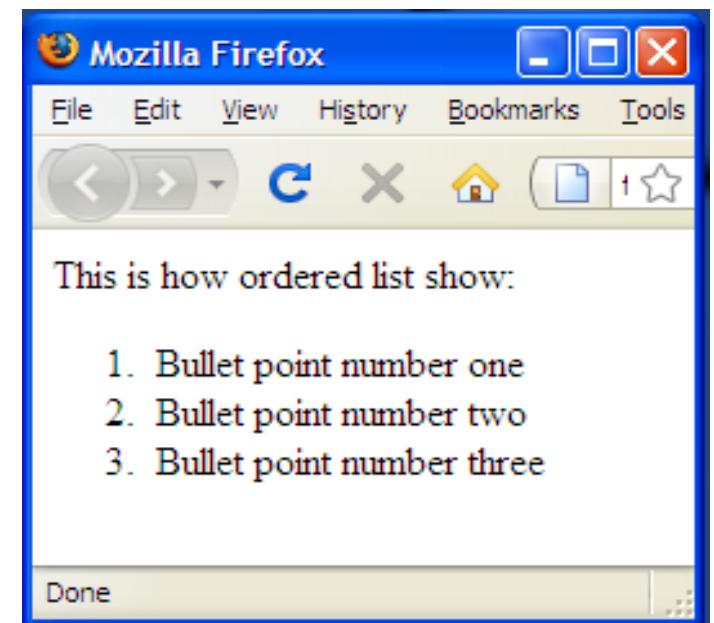
syntax:

```
<ol>  
<li> . . . </li>  
<li> . . . </li>  
</ol>
```

- eg.

<p>This is how unordered list show:

```
<ol>  
<li>Bullet point number one</li>  
<li>Bullet point number two</li>  
<li>Bullet point number three</li>  
</ol>  
</p>
```



The type Attribute:

- You can use type attribute for tag to specify the type of numbering you like. By default it is a number. Following are the possible options:
 - <ol type="1">- Default-Case Numerals.
 - <ol type="I"> - Upper-Case Numerals.
 - <ol type="i">- - Lower-Case Numerals.
 - <ol type="a">- Lower-Case Letters.
 - <ol type="A">- Upper-Case Letters

Example:

```
<!DOCTYPE html>  
<html>  
<head>  
<title>HTML Ordered List</title>  
</head> <body>  
<ol type="I">  
<li>Beetroot</li>  
<li>Ginger</li>  
<li>Potato</li>  
<li>Radish</li>  
</ol> </body>  
</html>
```

This will produce following result:

- I. Beetroot
- II. Ginger
- III. Potato
- IV. Radish

- The **start** Attribute You can use start attribute for tag to specify the starting point of numbering you need. Following are the possible options:
 - <ol type="1" start="4"> number start from 4
 - <ol type="I" start="4"> letter start from IV
 - <ol type="i" start="4"> letter start from iv
 - <ol type="a" start="4"> letter start from d
 - <ol type="A" start="4"> letter start from D

```
<!DOCTYPE html>  
<html>  
<head>  
<title>HTML Ordered List</title>  
</head> <body>  
<ol type="I" start ="4">  
<li>Beetroot</li>  
<li>Ginger</li>  
<li>Potato</li>  
<li>Radish</li>  
</ol> </body>  
</html>
```

This will produce following result:

- IV.** Beetroot
- V.** Ginger
- VI.** Potato
- VII.** Radish

Definition lists

Definition
list

- is a special kind of list for providing terms followed by a short text definition or description for them.

- eg.

<dl>

<dt>Unordered List</dt>

Defined
term

Definition

<dd>A list of bullet points.</dd>

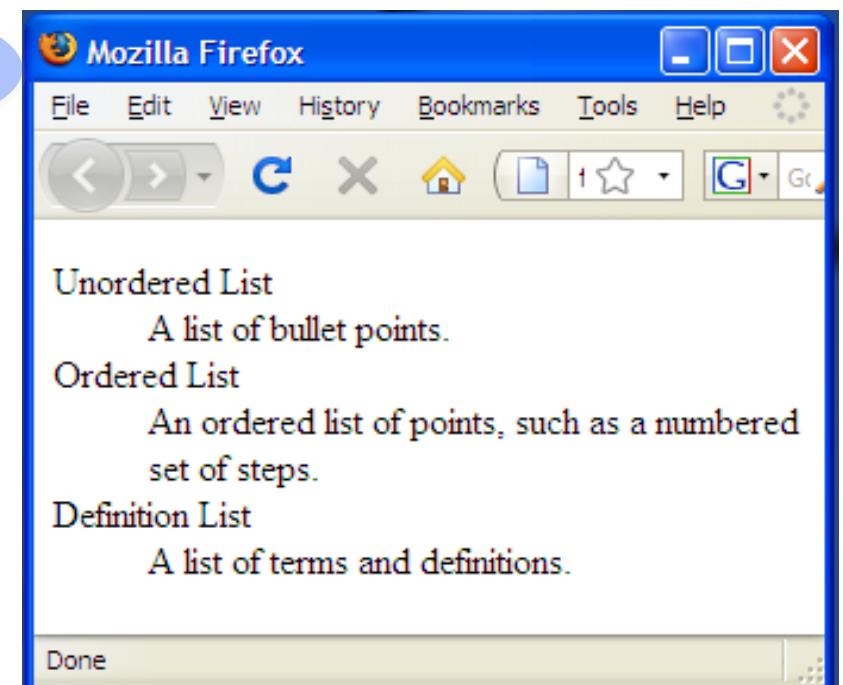
<dt>Ordered List</dt>

<dd>An ordered list of points, such as a numbered set of steps.</dd>

<dt>Definition List</dt>

<dd>A list of terms and definitions.</dd>

</dl>



Marquee HTML:

- The Marquee HTML tag is a non-standard HTML element which is used to scroll a image or text horizontally or vertically.
- In simple words, you can say that it scrolls the image or text up, down, left or right automatically

Example:

```
<marquee width="100%" behavior="scroll"  
bgcolor="pink">  
wellcome to html  
</marquee>
```

Adding emphasis to text

There are two elements that indicate that text should be emphasized:

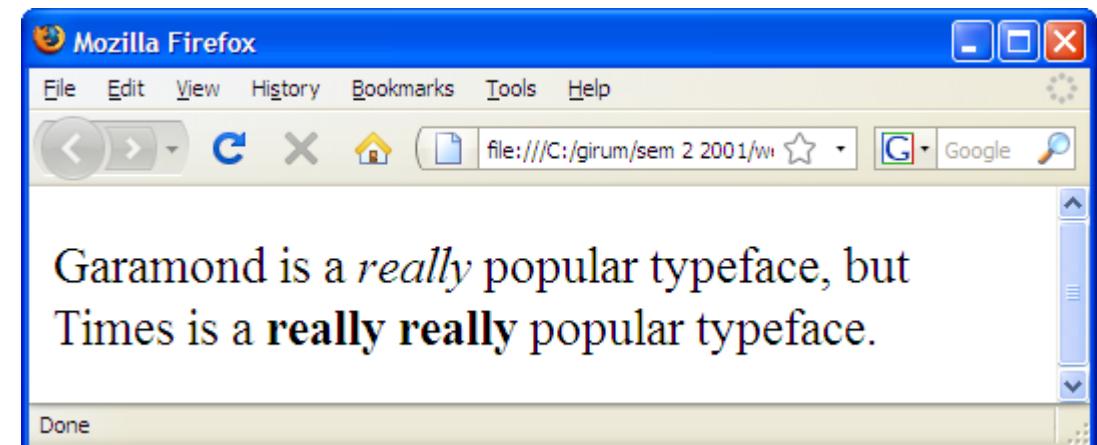
- **em** for emphasized text and

...

- **strong** for strongly emphasized text.

...

```
<p>Garamond is a <em>really</em>  
popular typeface, but Times is a  
<strong>really really</strong> popular  
typeface.</p>  
</p>
```

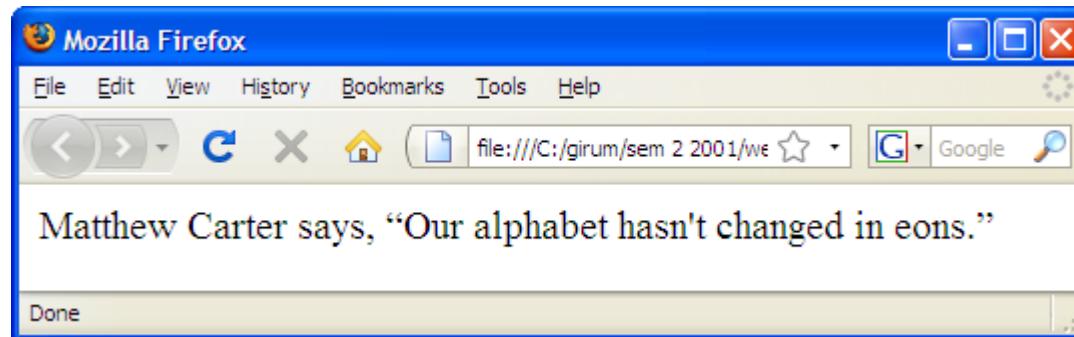


Screen readers may use a different tone of voice to deliver emphasized text, which is why you should use an **em** or **strong** element only when it makes sense semantically, not just to achieve italic or bold text.

Short quotations

- The quotation (**q**) element is used to mark up short quotations,

Matthew Carter says, < q > Our alphabet hasn't changed in eons. < / q >



Abbreviations and acronyms

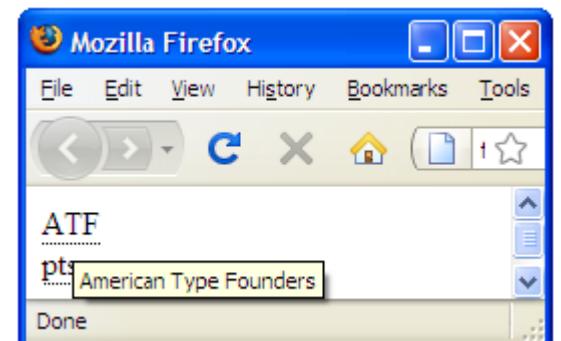
- Marking up shorthand terms as acronyms and abbreviations **provides useful information for search engines, screen readers, and other devices.**
- Abbreviations, indicated by the **abbr** element, are shortened versions of a word ending in a period (Conn. for Connection, for example).

`<abbr title="Points">pts.</abbr>`

- Acronyms, indicated by the **acronym** element, are abbreviations formed by the first letters of the words in a phrase (such as WWW or USA).

`<acronym title="American Type Founders">ATF</acronym>`

Both elements use the title attribute to provide the long version of the shortened term



Inserted and deleted text

- The **ins** and **del** elements are used to mark up changes to the text and indicate parts of a document that have been inserted or deleted

```
<p><Chief Executive Officer:  
<del title="retired">Peter Pan</del>  
<ins>Pippi Longstockings</ins></p>
```



Comments

- You can put comments between any tags in your XHTML documents. Comments use the following syntax:
- Syntax: <!-- comment goes here -->

```
<!-- This is a comment -->
```

```
<!-- This is a  
multiple-line comment  
that ends here. -->
```

Links and Navigation

- The average Web site is a group of pages users navigate between using hypertext links.
- These pages often include links to
 - other Web sites
 - other pages in the same site.

Link Basics

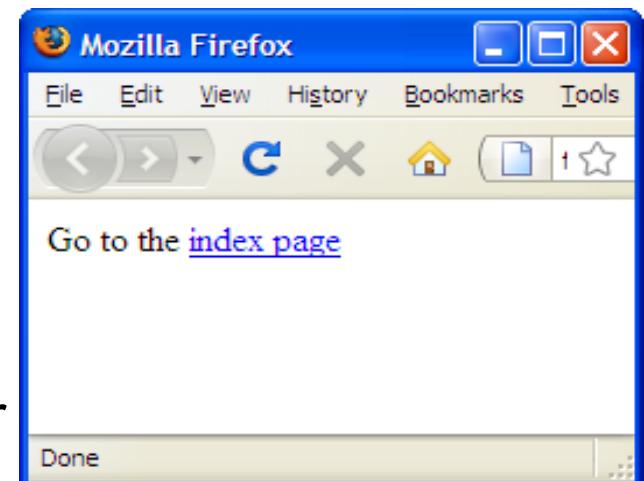
- A link is specified using the **anchor <a>** element.
- Anything between the opening **<a>** tag and the closing **** tag becomes part of the link a user can click in a browser.
- To link to another document, the opening **<a>** tag must carry an attribute called **href**, whose value is the page you are linking to.

```
<body>
```

Go to the [index page](index.html)

```
</body>
```

- The file index.html should be in the same folder



Linking to a website

- to link to a different site you need to write the URL between the `<a>` and `` tags - source anchor,

```
<body>
```

You can also `search Google`?

```
</body>
```

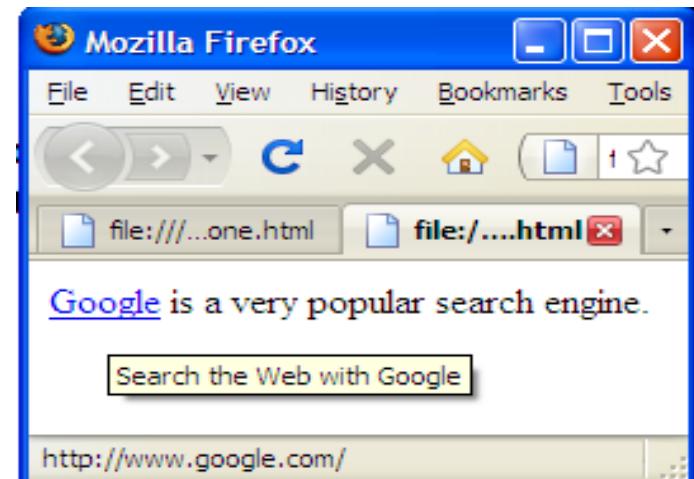
- You can include **title** attribute to links - when the mouse is over the link additional information could be delivered to the user.

```
<body>
```

```
<a href="http://www.Google.com/" title="Search the Web  
with Google">Google</a>
```

is a very popular search engine.

```
</body>
```



Using an image as a link

- put the `img` element in the anchor element:

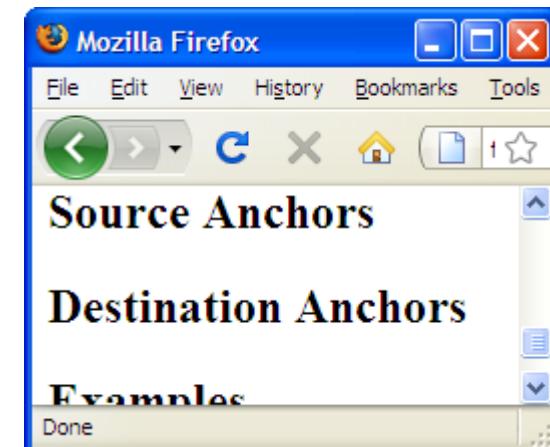
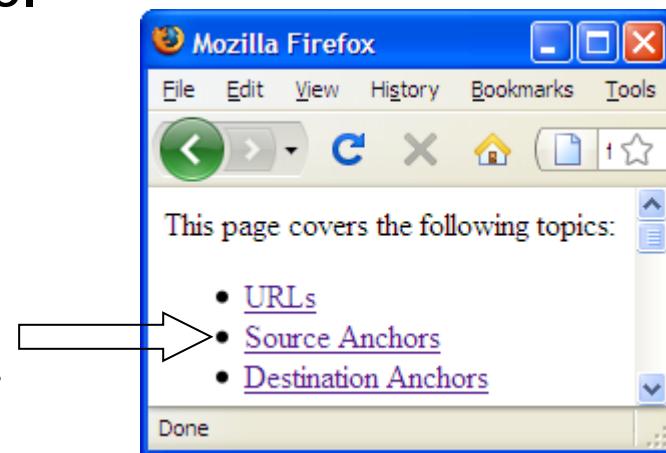
```
<a href="http://www.oreilly.com"></a>
```

- Most browsers display linked text as blue and underlined, and linked images with a blue border.
- Visited links generally display in purple.
- One can change the color of links by using *CSS* and it is recommended that you keep them **consistent** throughout your site so as not to confuse your users.

Linking to a specific part of a page

- If you have a long Web page, you might want to link to a specific part of that page.
- The *destination anchor* allows the page author to mark specific points in a page that a source link can point to.

```
<body>  
<p>This page covers the following topics:  
<ul>  
<li><a href="#URL">URLs</a></li>  
<li><a href="#SourceAnchors">Source Anchors</a></li>  
<li><a href="#DestinationAnchors">Destination Anchors</a></li>  
<li><a href="#Examples">Examples</a></li>  
</ul>  
</p>  
<h1>Linking and Navigation</h1>  
<h2><a name="URL">URLs</a></h2>  
<h2><a name="SourceAnchors">Source Anchors</a></h2>  
<h2><a name="DestinationAnchors">Destination Anchors</a></h2>  
<h2><a name="Examples">Examples</a></h2>  
</body>
```

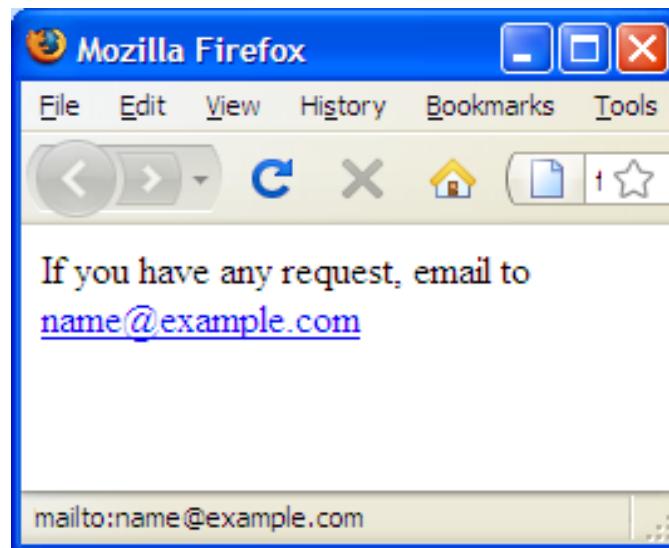


Linking to E-mail Addresses

- To create a link to an e-mail address you need to use the following syntax with the `<a>` element:

```
<a href="mailto:name@example.com">name@example.com</a>
```

```
<a href="mailto:name@example.com">E-mail us</a>.
```



Tables

- Are used
 - displaying tabular data
 - to format whole pages and create visually attractive
 - We use `<table> ... </table>`

(Cont'd . . .) Tables

- to create a table in XHTML use the `<table>` element.
- A row is contained inside a `<tr>` element— *table row*
- each cell is then written inside the row element using a `<td>` element for “*table data*” and `<th>` for “*table headers*,”

```
<table border="1">
<tr>
    <td>row 1, column 1 </td>
    <td>row 1, column 2 </td>
</tr>
<tr>
    <td>row 2, column 1 </td>
    <td>row 2, column 2 </td>
</tr>
```

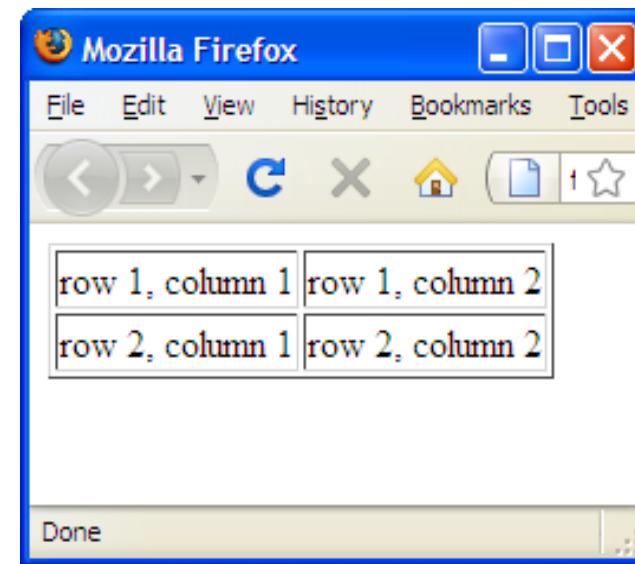


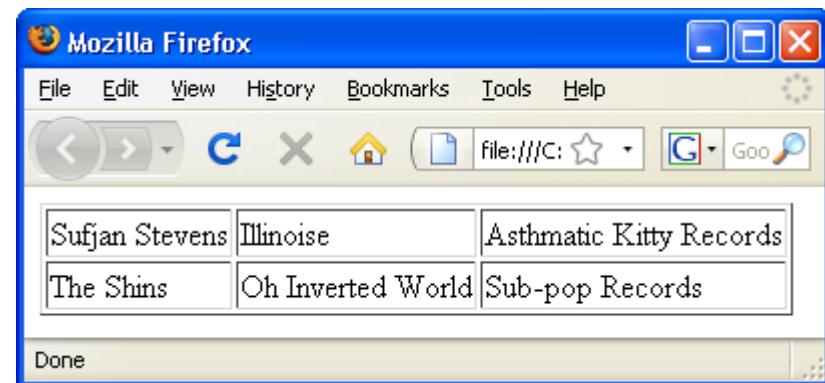
Table Headers

- the text marked up as headers (**th** elements) are displayed differently from the other cells in the table (**td** elements).
 - Bold and
 - Centric
- are important because they provide information or context about the cells in the row or column

Example

In the code fragment below how many columns and rows are there?

```
<table>
<tr>
  <td>Sufjan Stevens</td>
  <td>Illinoise</td>
  <td>Asthmatic Kitty Records</td>
</tr>
<tr>
  <td>The Shins</td>
  <td>Oh Inverted World</td>
  <td>Sub-pop Records</td>
</tr>
</table>
```



Exercise -1

- write the markup for the table shown



A screenshot of a Mozilla Firefox browser window. The title bar says "Mozilla Firefox". The menu bar includes "File", "Edit", "View", "History", "Bookmarks", "Tools", and "Help". Below the menu is a toolbar with icons for back, forward, search, and bookmarks. The main content area displays a table with three rows and three columns. The columns are labeled "Name", "Sex", and "Year of Birth". The rows contain the following data:

Name	Sex	Year of Birth
Alemu ketema	M	1973
Misgana kebede	F	1980
Solomon Belay	M	1970

Spanning Cells

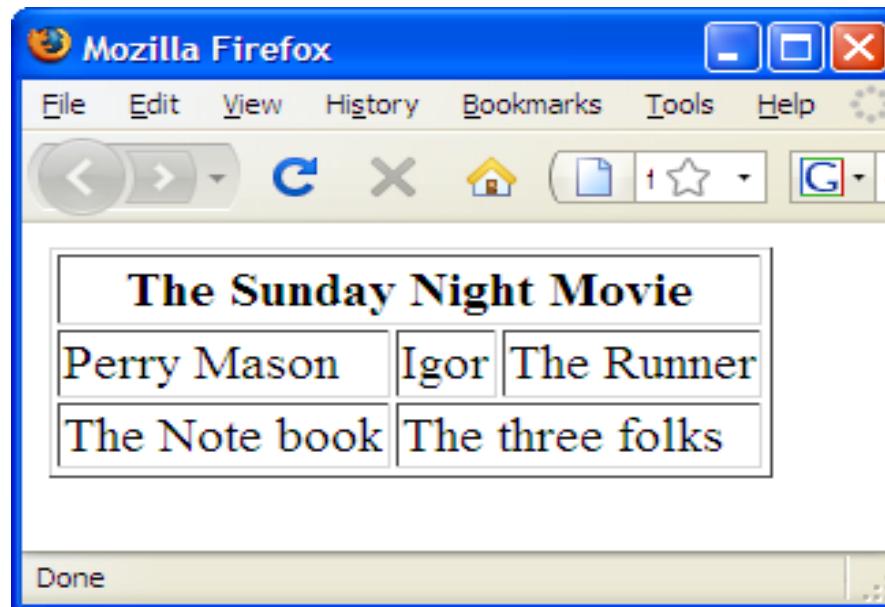
- is the stretching of a cell to cover several rows or columns
- You make a header or data cell span by adding the **colspan** or **rowspan** attributes
- **Column spans**, created with the **colspan** attribute in the **td** or **th** element, stretch a cell to the right to span over the subsequent columns
 - Example :

```
<table border="1">
<tr>
<th colspan="2">Fat</th>
</tr>
<tr>
<td>Saturated Fat (g)</td>
<td>Unsaturated Fat (g)</td>
</tr>
</table>
```



Exercise -2

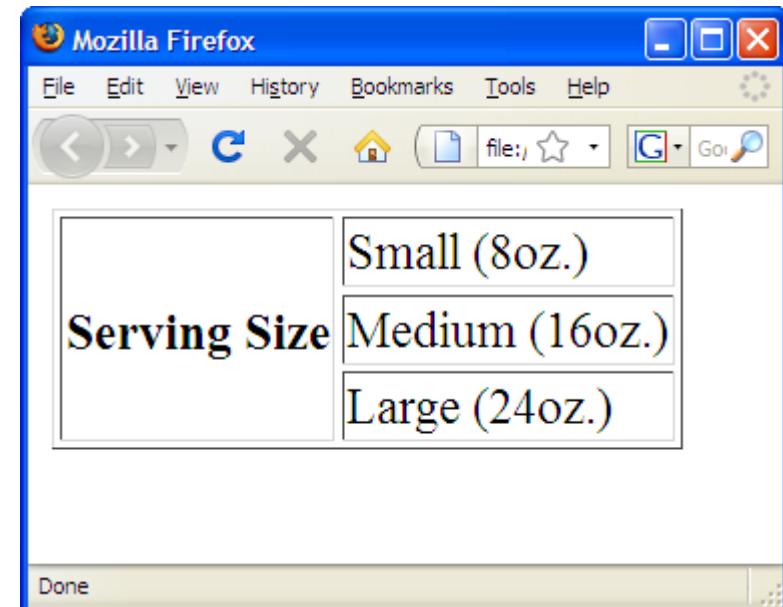
- write the markup for the table shown



Row spans

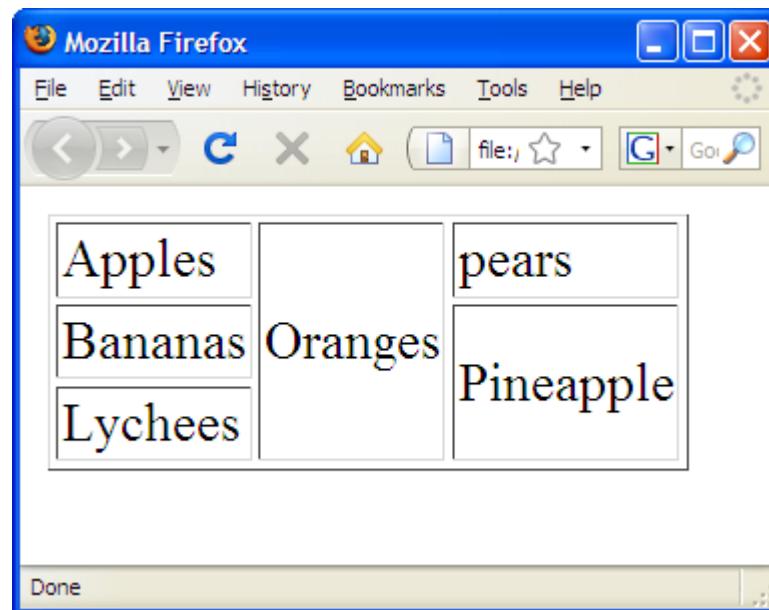
- Row spans, created with the **rowspan** attribute
 - cause the cell to span downward over several rows.
 - Example:

```
<table>
<tr>
<th rowspan="3">Serving Size</th>
<td>Small (8oz.)</td>
</tr>
<tr>
<td>Medium (16oz.)</td>
</tr>
<tr>
<td>Large (24oz.)</td>
</tr>
</table>
```



Exercise:

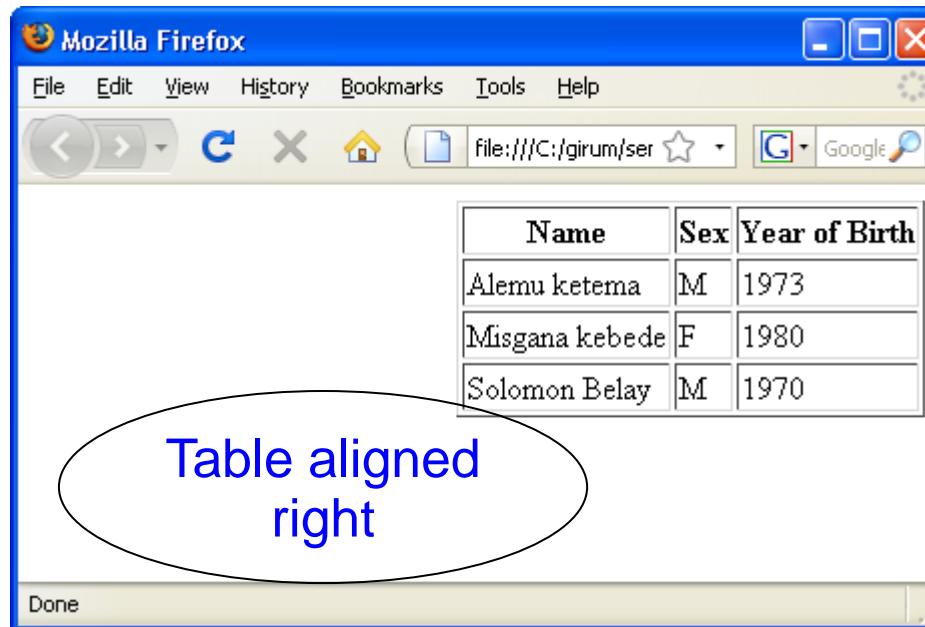
- write the markup for the table shown



The align attribute (deprecated)

- alignment of the table within the containing body-text flow
- accepts a value of either **left**, **right**, or **center**,

```
<table align="right" , border="1">
```

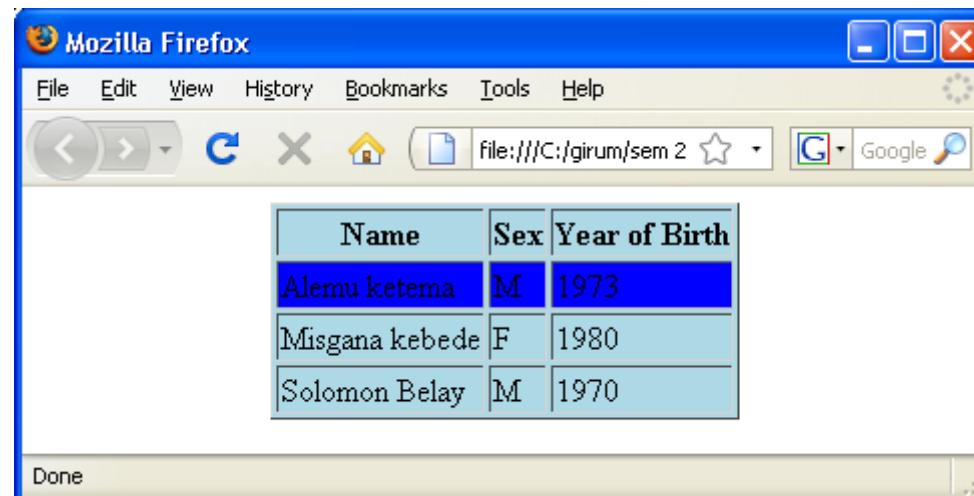


The bgcolor and background attributes

- To make the **background** of a table a different color than the document's background
- You may also set individual row and cell colors by providing the **bgcolor** attribute

```
<table align="center" , bgcolor ="lightblue" , border="1">
```

```
<tr bgcolor="blue">
```

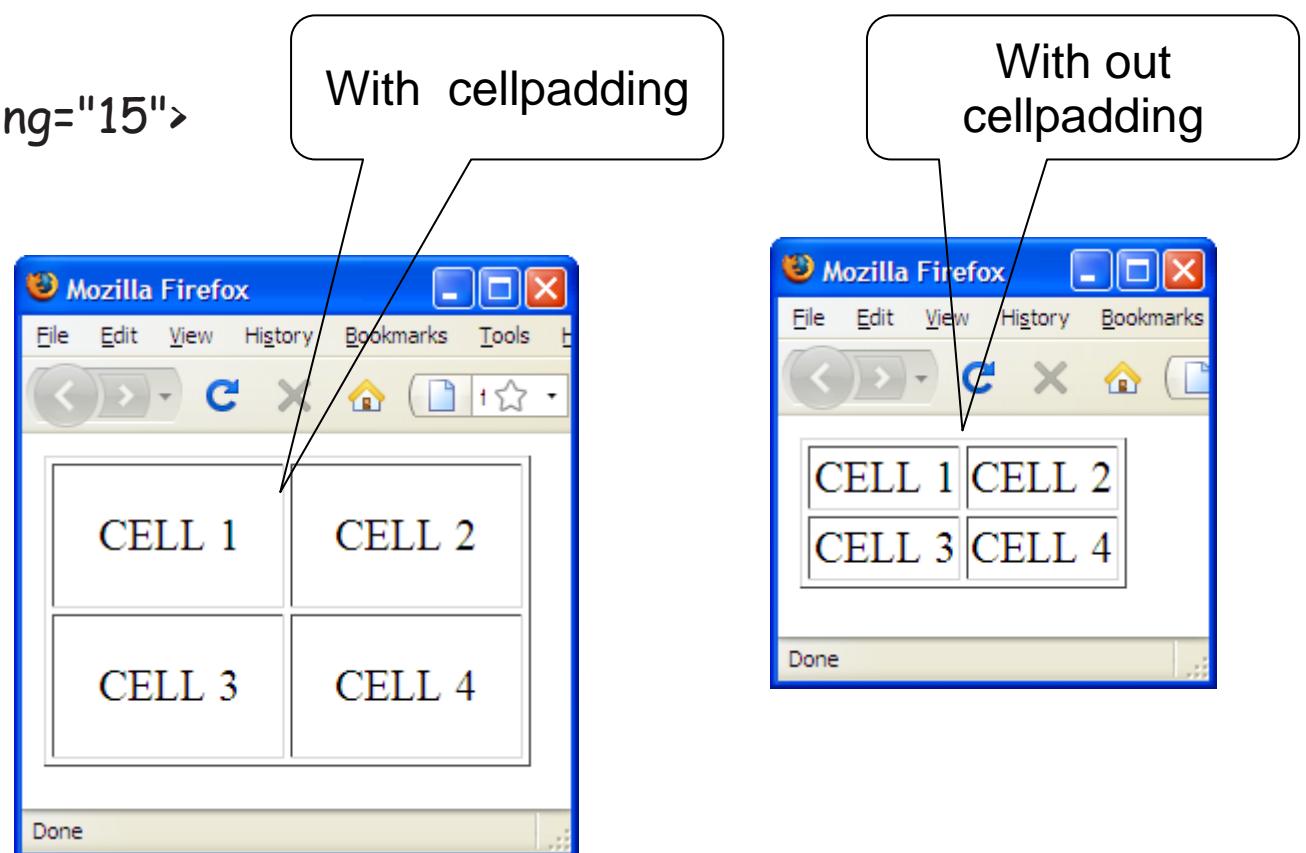


Cell Padding and Spacing

- By default, cells are sized just large enough to fit their contents
- **Cell padding** is the amount of space held between the contents of the cell and the cell border.
- if you don't specify any cell padding, the cells will have the default value of one pixel of padding.

- Example:

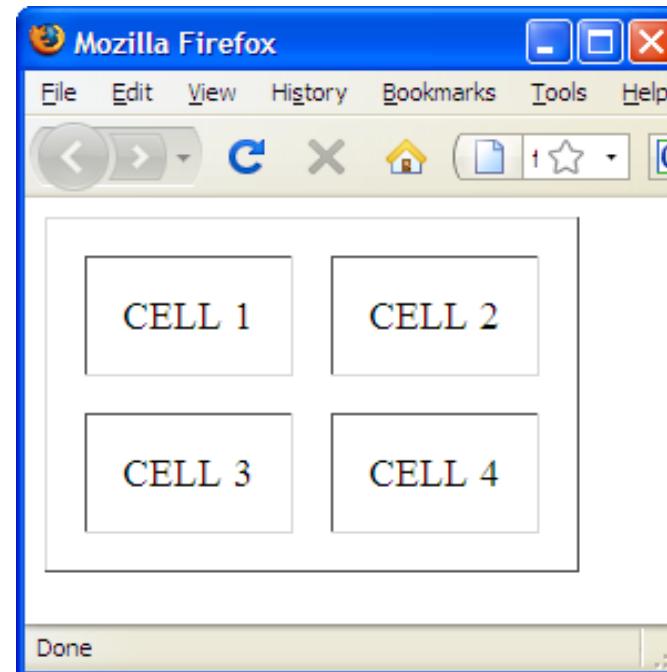
```
<table border="1" cellpadding="15">
<tr>
  <td>CELL 1</td>
  <td>CELL 2</td>
</tr>
<tr>
  <td>CELL 3</td>
  <td>CELL 4</td>
</tr>
</table>
```



Cont'd ... Cell Padding and Spacing

- **cellpadding** attribute may be used with the **table** element only, the **cellpadding** value applies to all the cells in the table.
- **Cell spacing** is the amount of space held between cells, specified in number of pixels.

```
<table border="1" cellpadding="15" cellspacing="15">
<tr>
<td>CELL 1</td>
<td>CELL 2</td>
</tr>
<tr>
<td>CELL 3</td>
<td>CELL 4</td>
</tr>
</table>
```



The width and height attributes

- the **width** attribute is either
 - an integer number of pixels
`<table width=400>`
 - a relative percentage of the screen width
`<table width="50%">`
- **height** attribute to suggest a recommended height for the table

The align and valign attributes in <tr> tag

- The **align** attribute for the <table> tag may be **deprecated** in the HTML and XHTML but not in <tr> tags
- **align** attributes could be **left, right, center, and justify**
- the **valign** attribute in the <tr> tag with a value of **top, bottom, center, middle, or baseline** (Internet Explorer only),
 - Normally, browsers render cell contents centered vertically

align example

```
C:\girum\sem 2 2001\web programing\code\tables\table_simple2.html - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
valign.html table_simple2.html

1 <html>
2 <body>
3 <table border=1 width=500>
4 <tr height=50>
5   <th valign=top>Menu item</th>
6   <th valign=center>Calories</th>
7   <th valign=bottom>Fat (g)</th>
8 </tr>
9 <tr height=100>
10  <td valign= top align=left>Chicken noodle soup</td>
11  <td align=middle>120</td>
12  <td align=right>2</td>
13 </tr>
14 <tr>
15  <td align=right>Caesar salad</td>
16  <td>400</td>
17  <td>26</td>
18 </tr>
19 </table>
20 </body>
21 </html>
```

Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///C:/girum/sem 2 2001/web programin... Google

Menu item	Calories	Fat (g)
Chicken noodle soup	120	2
Caesar salad	400	26

Done

valign example

C:\girum\sem 2 2001\web programing\code\tables\valign.html - Notepad++

File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?

valign.html

```
1 <html>
2 <body>
3 <table border="border">
4   <tr>
5     <th>Alignment</th>
6     <th>Top</th>
7     <th>Baseline</th>
8     <th>Center</th>
9     <th>Middle</th>
10    <th>Bottom</th>
11  </tr>
12  <tr align="center">
13    <th><h1>Baseline _ _ <br />Another line</h1></th>
14    <td valign="top">AAyy</td>
15    <td valign="baseline">_AAyy_</td>
16    <td valign="center">AAyy</td>
17    <td valign="middle">AAyy</td>
18    <td valign="bottom">AAyy</td>
19  </tr>
20 </table>
21 </body>
22 </html>
```

Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///C:/girum/sem 2 2001/we

Alignment	Top	Baseline	Center	Middle	Bottom
Baseline Another line	AAyy	_AAyy_		AAyy	AAyy
					AAyy

Done

nowrap

- stops that normal word wrapping

```
C:\girum\sem 2 2001\web programming\code\tables\table_simple2.html - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?


| Menu item           | Calories | Fat (g) |
|---------------------|----------|---------|
| Chicken noodle soup | 120      | 2       |
| Caesar salad        | 400      | 26      |


```

Without nowrap

Mozilla Firefox

File Edit View History Bookmarks Tools Help

Done

Menu item	Calories	Fat (g)
Chicken noodle soup	120	2
Caesar salad	400	26

Mozilla Firefox

File Edit View History Bookmarks Tools Help

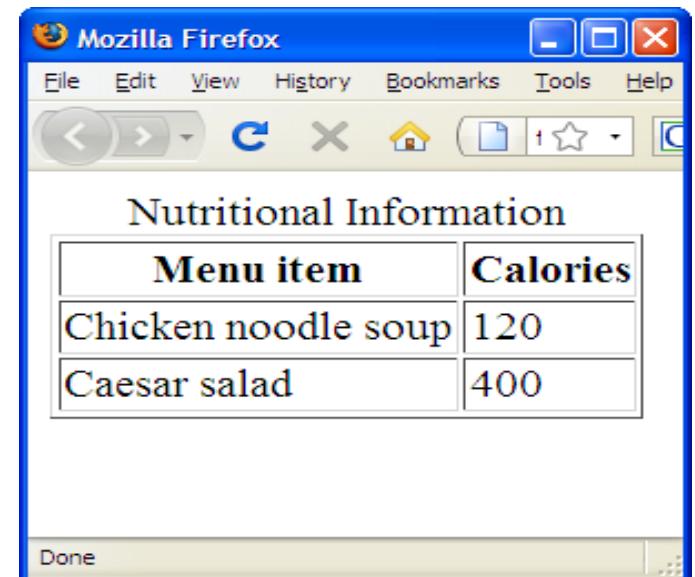
Done

Menu item	Calories	Fat (g)
Chicken noodle soup	120	2
Caesar salad	400	26

Captions and Summaries

- are two methods for providing additional information about a table
- **caption** is displayed with the table in visual browsers,
 - is used to give a table a title or brief description
 - Example:

```
<table>
<caption>Nutritional Information</caption>
<tr>  <th>Menu item</th>
<th>Calories</th>
</tr>
<tr>      <td>Chicken noodle soup</td>
<td>120</td>
</tr>
<tr>      <td>Caesar salad</td>
          <td>400</td>
</tr>
</table>
```



Cont'd... Captions and Summaries

- **Summary** are used to provide a more lengthy description of the table and its contents.
- They are added using the **summary** attribute in the **table** element, and they not displayed but may be used by **assistive devices**

```
<table summary="A listing of the calorie and fat content for each  
of the most popular menu items">
```

```
<caption>Nutritional Information</caption>
```

```
...table continues...
```

```
</table>
```

Exercise:

- write the markup for the table shown

Your Content Here			
	A common header for two subheads		Header 3
	Header 1	Header 2	
Thing A	data A1	data A2	data A3
Thing B	data B1	data B2	data B3
Thing C	data C1	data C2	data C3

Adding Images

- use the `` (XHTML) and `` (HTML)

`` -- works fine if logo.gif and the file are in the same folder.

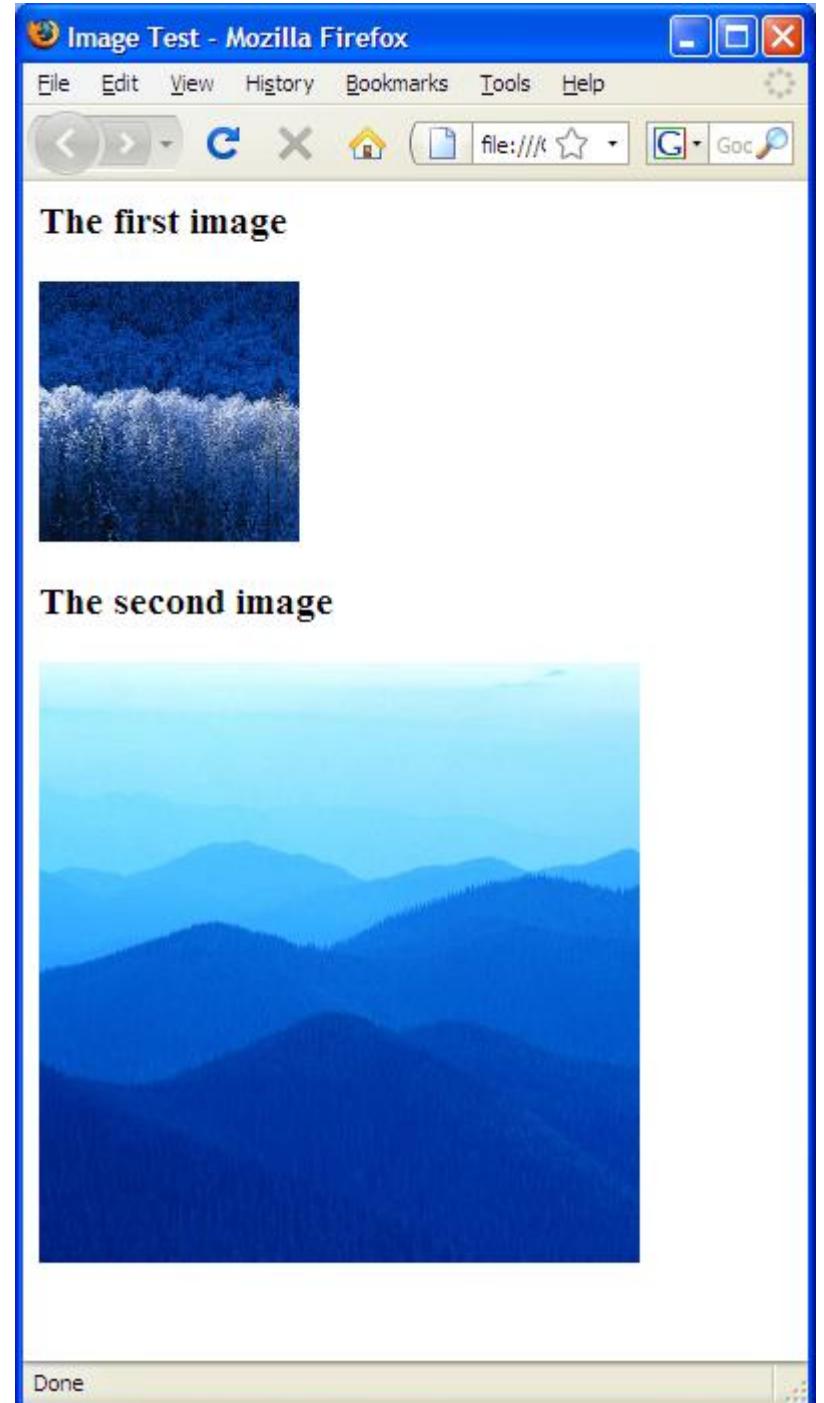
- `<image>` carry the following attributes:
 - `src` - is required to specify the URL of the image to load. `src="url"` the url could be absolute or relative.
 - `alt` - to specify a text alternative for the image in case the browser cannot display. `alt="Company logo"`
 - `align` - used to align the image within the page or the element that contains the image(such as a table cell). Eg `align="right"` . The values could be: top, middle, bottom, left, right

Cont'd ... Adding Images

- **border** - specifies the width of the border around the image in pixels. Eg. border="2" , the default value is 0 unless it is a link.
- **height and width** - specify the height and width of the image: Eg. height="120" width="180"
 - The values can either be pixels or a percentage (%)of the page or containing element.
 - It is a good practice to specify the height and width of an image, because it will allow the browser to render the rest of the page before the image.
- **hspace vspace** - are used to control the amount of whitespace around an image. Eg. Hspace="10" vspace="14"
 - helpful because text can flow around an image

Example

```
<h3>The first image </h3>  
  
  
  
<h3>The second image</h3>  
  
  
  
</body>
```



Frames

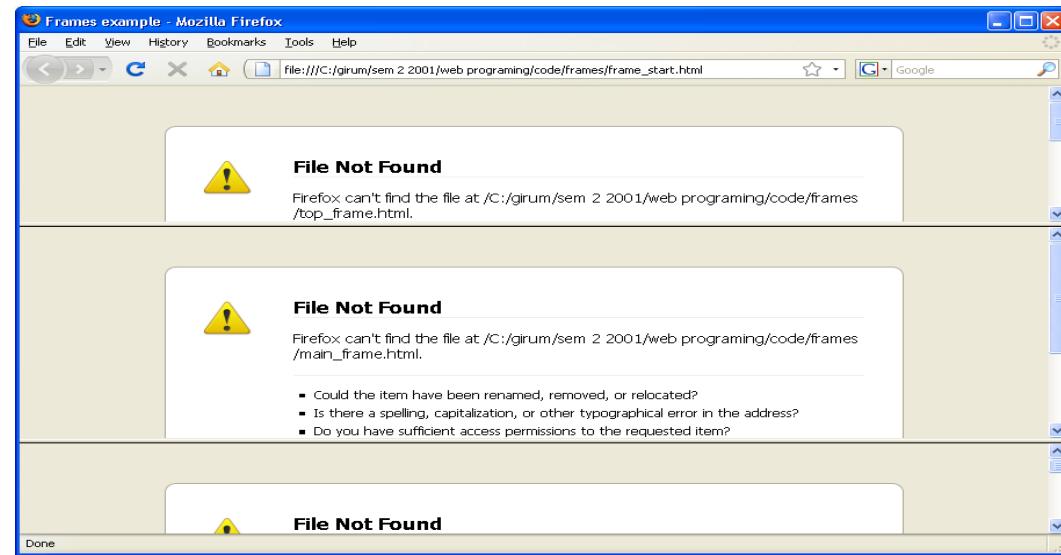
- *Frames* divide a browser window into several pieces or panes, each pane containing a separate XHTML page.
- One of the key advantages that frames offer is that you can then load and reload single panes without having to reload the entire contents of the browser window.
- A collection of frames in the browser window is known as a *frameset*.

(Cont'd . . .) Frames

- To create a frameset document,
 - first you need the `<frameset>` element, which is used instead of the `<body>` element.
 - The frameset defines the rows and columns your page is divided into, including where each individual frame will go.
 - Each frame is then represented by a `<frame>` element.

Example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html>
<head>
<title>Frames example</title>
</head>
<frameset rows="150, *, 100">
<frame src="top_frame.html" />
<frame src="main_frame.html" />
<frame src="bottom_frame.html" />
<noframes><body>
This site uses a technology called frames. Unfortunately, your
browser does not support this technology. Please upgrade
your browser and visit us again!
</body></noframes>
</frameset>
</html>
```



The <frameset> Element

- The <frameset> element replaces the <body> element in frameset documents.
- the attributes of the <frameset> element
 - **cols** specifies how many columns are in the frameset
 - **rows** specifies how many rows are in the frameset
 - <frame> element for each frame of the document
 - <noframes> elements to indicate what should be displayed to the user if their browser does not load frames.

The cols Attribute

- specifies
 - how many columns are contained in the frameset and
 - the size of each column.
 - Eg `cols="20%, 60%, 20%"` – three columns and their relative size
- If you do not specify a `cols` attribute then the default value is 100 percent
- You can specify the width of each column in one of four ways:
 - Absolute values in pixels
 - A percentage of the browser window (or parent frame if you are nesting framesets)
 - Using a wildcard symbol (*)
 - As relative widths of the browser window (or parent frame)

Width of column

- A Percentage of the Browser Window or Parent Frame
 - use a number followed by the percent sign
 - `Cols="40%, 60%"`
 - . The first frame takes 40% of the browser window
- The second frame takes 60% of the browser window
- The Wildcard Symbol (*)
 - The asterisk, or wildcard symbol, indicates the "rest of the window,"
 - `Cols="400, *"`
 - `<frameset cols="10%,3*,*,*">` - ??

Width of column

- **Relative Widths Between Columns**

- `Cols="3*, 2*, 1*"`
 - It is divided into six and the first frame takes half of it

- **Absolute Values in Pixels**

- use a number
 - `Cols="100, 500, *"` - - three columns 100, 500 and the rest
 - `Cols="100, 100, 100"` - will produce three frames of size 100, but if the screen size is 600px it will produce three 200 px size frames
 - `Cols= "100,100,100,*"` will produce three exactly 100px frames and one frame to fill the rest of the space

Additional <frameset> Element

- The border Attribute
 - specifies the width of the border of each frame in pixels.
 - border="10"
- The frameborder Attribute
 - Used to display a three-dimensional border between frames.
 - frameborder="0" – the border will not show
 - frameborder="1" – to display border
- The framespacing Attribute
 - specifies the amount of space between frames in a frameset
 - should be given in pixels and the default value is 2

The <frame> Element

- indicates what goes in each frame of the frameset.
- **The src Attribute**
 - indicates the file that should be used in the frame
 - `src="main_page.html"`
- **The name Attribute**
 - it is used to indicate which frame a document should be loaded into.
 - to create links in one frame that load pages into a second frame
 - `name="main_frame"`

The <frame> Element

- The **marginwidth** and **marginheight** Attributes
 - Specifies the space between the three-dimensional border of a frame and its contents.
 - `marginheight="10" marginwidth="10"`
- The **noresize** Attribute
 - prevents a user from being able to resize the frame
 - `noresize="noresize"`

The <frame> Element

- The **scrolling** Attribute
 - `scrolling="yes"` - to always have the scrollbar
 - or "`no`" - the frameset will not contain a set of scrollbars
 - or "`auto`" - browser should include scrollbars when the content does not fit in the frame
- The **<noframes>** Element
 - If a user's browser does not support frames the contents of the **<noframes>** element should be displayed to the user.
 - place a **<body>** element inside the **<noframes>**

Forms

- There are two parts to a working form.
 - The first part is the form , which are made up of buttons, text fields, and pull-down menus
 - Forms are added to web pages using the **form** element.
 - The other component of a web form is an application or script on the server that processes the information collected by the form and returns an appropriate response.(CGI, PHP, ASP etc)

Example

```
<html>
<head>
<title>Mailing List Signup</title>
</head>
<body>
<h1>Mailing List Signup</h1>
<form action="/cgi-bin/mailinlist.pl" method="post">
<fieldset>
<legend>Join our email list</legend>
<p>Get news about the band such as tour dates and special MP3 releases sent to your own in-box.</p>
<ol>
<li><label for="name">Name:</label>
<input type="text" name="name" id="name" /></li>
<li><label for="email">Email:</label>
<input type="text" name="email" id="email" /></li>
</ol>
<input type="submit" value="Submit" />
</fieldset>
</form>
</body>
</html>
```

Cont'd . . . Example

A screenshot of a Mozilla Firefox browser window titled "Mailing List Signup - Mozilla Firefox". The address bar shows the URL "file:///C:/web programing/code/forms_html/simple_fc". The main content area displays a "Mailing List Signup" page. The page has a heading "Join our email list" and a descriptive text: "Get news about the band such as tour dates and special MP3 releases sent to your own in-box." Below this, there are two input fields labeled "1. Name:" and "2. Email:", each with a corresponding text input box. A "Submit" button is located below the input fields. At the bottom of the page, a "Done" button is visible.

The action attribute

- provides the location (URL) of the application or script that will be used to process the form.

```
<form action="/cgi-bin/mailinlist.pl"
      method="post">...</form>
```

Variables and Content

The name attribute

- The **name** attribute identifies the variable name for the control.

```
<textarea name="comment" rows="4" cols="45">Would you  
like to add a comment?</textarea>
```

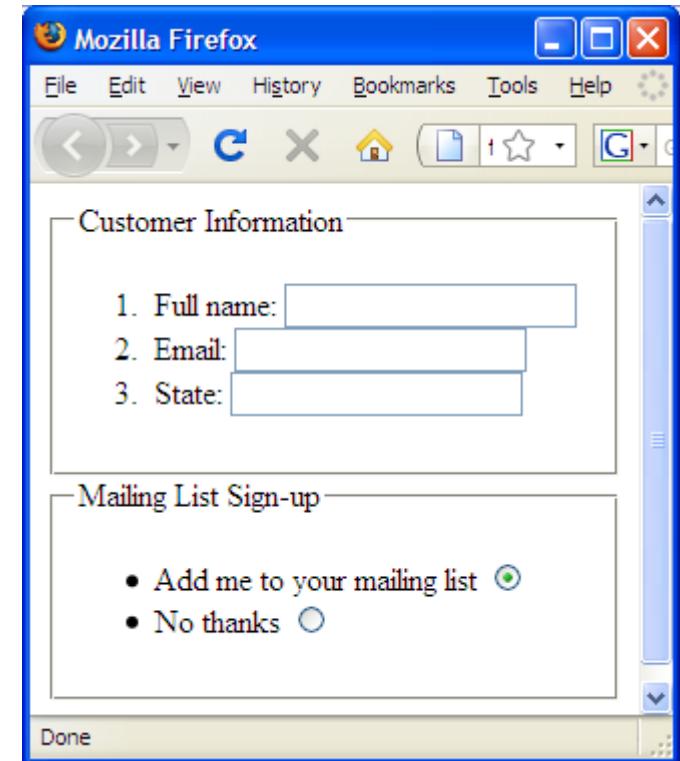
- When a user enters a comment in the field ("This is the best band ever!"), it would be passed to the server as a name/value (variable/content) pair like this:

```
comment=This%20is%20the%20best%20band%20ever!
```

Fieldset and legend

- **fieldset** element is used to indicate a logical group of form controls.
- A **fieldset** may also include a **legend** element that provides a caption for the enclosed fields.

```
<fieldset>
<legend>Customer Information</legend>
<ol>
<li><label>Full name: <input type="text" name="name" /></label></li>
<li><label>Email: <input type="text" name="email" /></label></li>
<li><label>State: <input type="text" name="state" /></label></li>
</ol>
</fieldset>
<fieldset>
<legend>Mailing List Sign-up</legend>
<ul>
<li><label>Add me to your mailing list <input type="radio" name="list" value="yes" checked="checked" /></label></li>
<li><label>No thanks <input type="radio" name="list" value="no" /></label></li>
</ul>
</fieldset>
```



Submit and reset buttons

- **submit** button immediately sends the collected form data to the server for processing.
- **reset** button returns the form controls to the state they were in when the form loaded.
- submit and reset buttons are added using the **input** element.

```
<input type="submit" />
```

```
<input type="reset" />
```

eg. <p><input type="submit" /> <input type="reset" value="Start over" /></p>

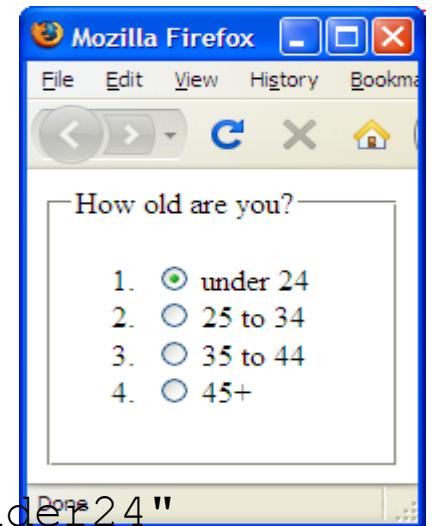
Radio and checkbox buttons

- radio buttons is appropriate when only one option from the group is permitted

Syntax: <input type="radio" name="variable" />

Example:

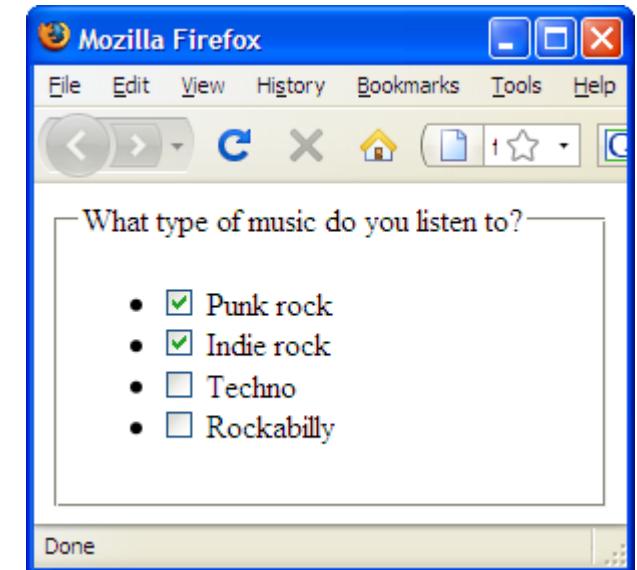
```
<fieldset>
<legend>How old are you?</legend>
<ol>
<li><label><input type="radio" name="age" value="under24"
    checked="checked" /> under 24</label></li>
<li><label><input type="radio" name="age" value="25-34" /> 25 to
    34 </label></li>
<li><label><input type="radio" name="age" value="35-44" /> 35 to
    44 </label></li>
<li><label><input type="radio" name="age" value="over45" /> 45+
    </label></li>
</ol>
</fieldset>
```



Cont'd... Radio and checkbox buttons

- Checkboxes allow users to select as many or as few from the group as desired.
- Syntax: <input type="checkbox" />
- Example:

```
<fieldset>
<legend>What type of music do you listen to?</legend>
<ul>
<li><label><input type="checkbox" name="genre" value="punk" checked="checked" /> Punk rock</label></li>
<li><label><input type="checkbox" name="genre" value="indie" checked="checked" /> Indie rock</label></li>
<li><label><input type="checkbox" name="genre" value="techno" /> Techno </label></li>
<li><label><input type="checkbox" name="genre" value="rockabilly" /> Rockabilly</label></li>
</ul>
</fieldset>
```



Menus

- Menus tend to be more compact than groups of buttons and checkboxes.
 - Pull-down
 - scrolling menu

Menu control

`<select>...</select>`

An option within a menu

`<option>...</option>`

A logical grouping of options within a menu

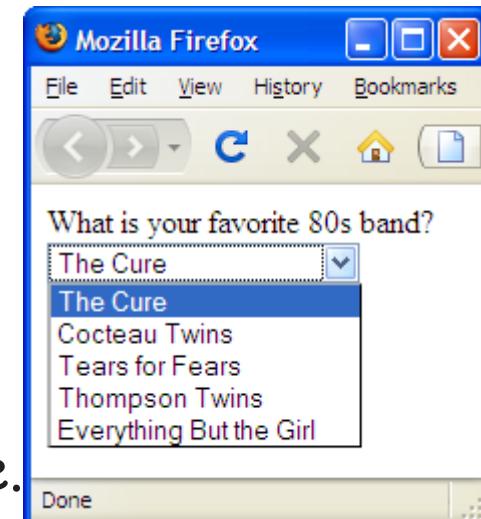
`<optgroup>...</optgroup>`

Pull-down menus

- only one item may be selected.
- **select** element displays as a pull-down menu by default when
 - no size is specified or
 - if the **size** attribute is set to 1

Example:

```
<label for="form-fave">What is your favorite 80s band?<label><br />
<select name="EightiesFave" id="form-fave">
<option>The Cure</option>
<option>Cocteau Twins</option>
<option>Tears for Fears</option>
<option>Thompson Twins</option>
<option value="EBTG">Everything But the Girl</option>
</select>
```

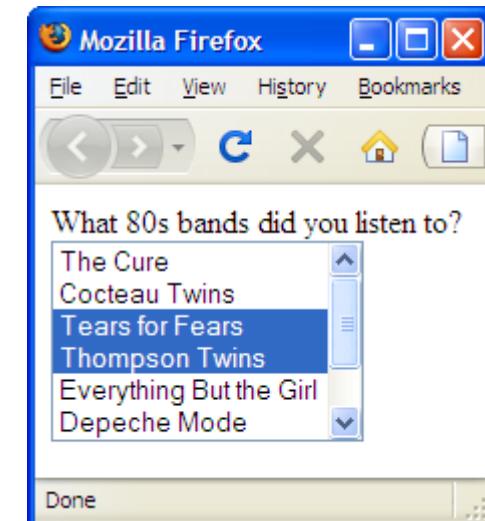


- Use the **value** attribute to provide an overriding value.

Scrolling menu

- specify the number of lines you'd like to be visible using the **size** attribute.
- The **multiple** attribute allows users to make more than one selection from the scrolling list.

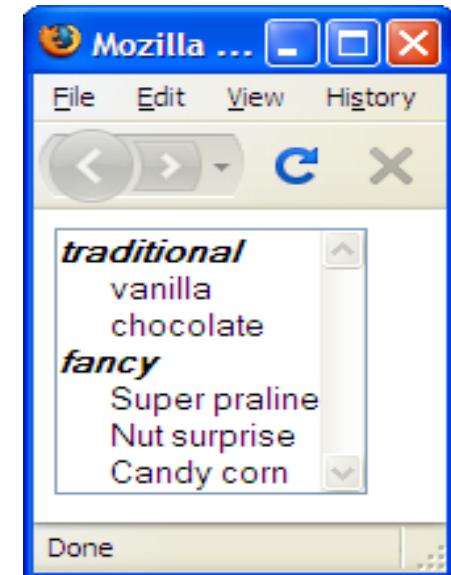
```
<label for="EightiesBands">What 80s bands did you listen  
to?</label>  
  
<select name="EightiesBands" size="6" multiple="multiple"  
for="EightiesBands">  
  <option>The Cure</option>  
  <option>Cocteau Twins</option>  
  <option selected="selected">Tears for Fears</option>  
  <option selected="selected">Thompson Twins</option>  
  <option value="EBTG">Everything But the Girl</option>  
  <option>Depeche Mode</option>  
  <option>The Smiths</option>  
  <option>New Order</option>  
</select>
```



Grouping menu options

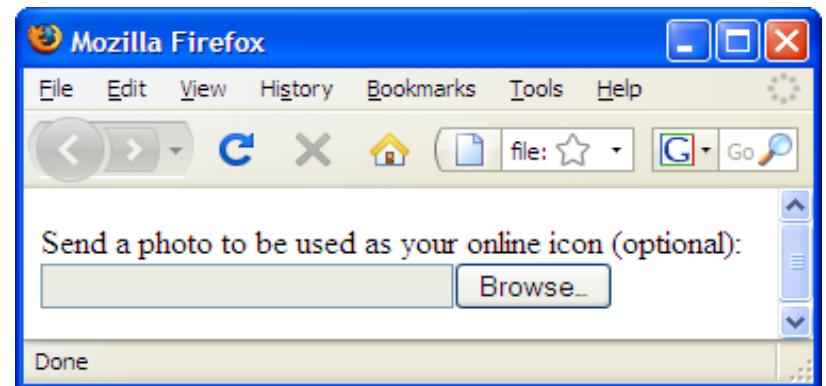
- You can use the **optgroup** element to create conceptual groups of options.
- Example :

```
<select name="icecream" multiple="multiple">  
<optgroup label="traditional">  
    <option>vanilla</option>  
    <option>chocolate</option>  
</optgroup>  
<optgroup label="fancy">  
    <option>Super praline</option>  
    <option>Nut surprise</option>  
    <option>Candy corn</option>  
</optgroup>  
</select>
```



File selection control

- Forms can be used to transmit external documents from a user's hard drive.
- ~ is used to select a document from the hard drive to be submitted with the form data.
- Syntax: <input type="file" />
- Eg:



```
<form action="/client.php" method="post" enctype="multipart/form-data">

<p><label for="form-photo">Send a photo to be used as your online icon (optional):</label><br />

<input type="file" name="photo" size="28" id="form-photo" /></p>

</form>
```