

HTML

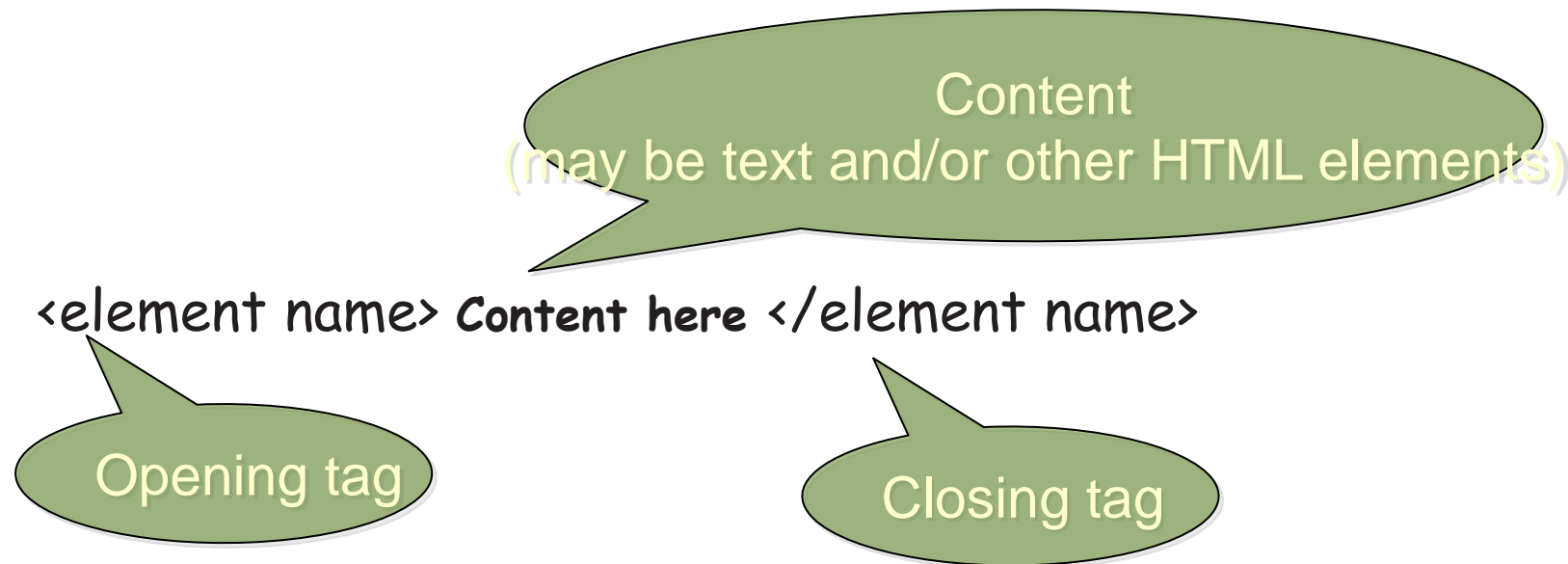
HTML (Hyper Text Markup Language)

- 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

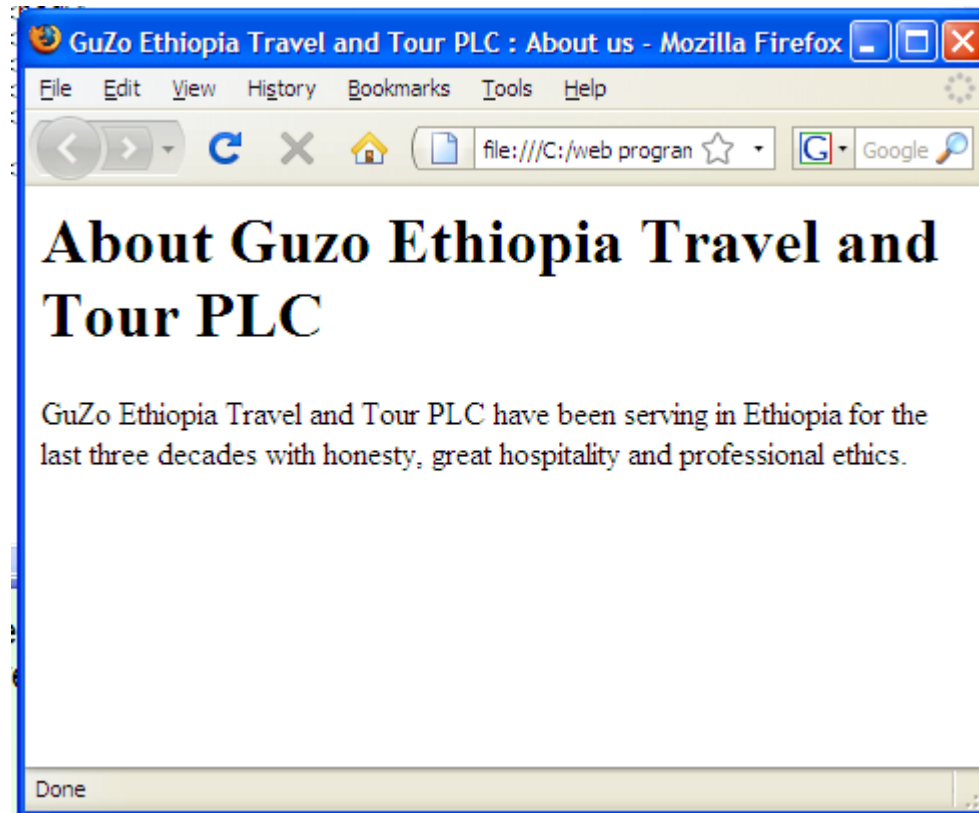
Simple page

```
<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 opening tag

`<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 closing tag

`</html>`

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>

This is the beginning of a heading

This is the end of a heading

element

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

elem
ent

The main body 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>
```

Header one starts
here

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

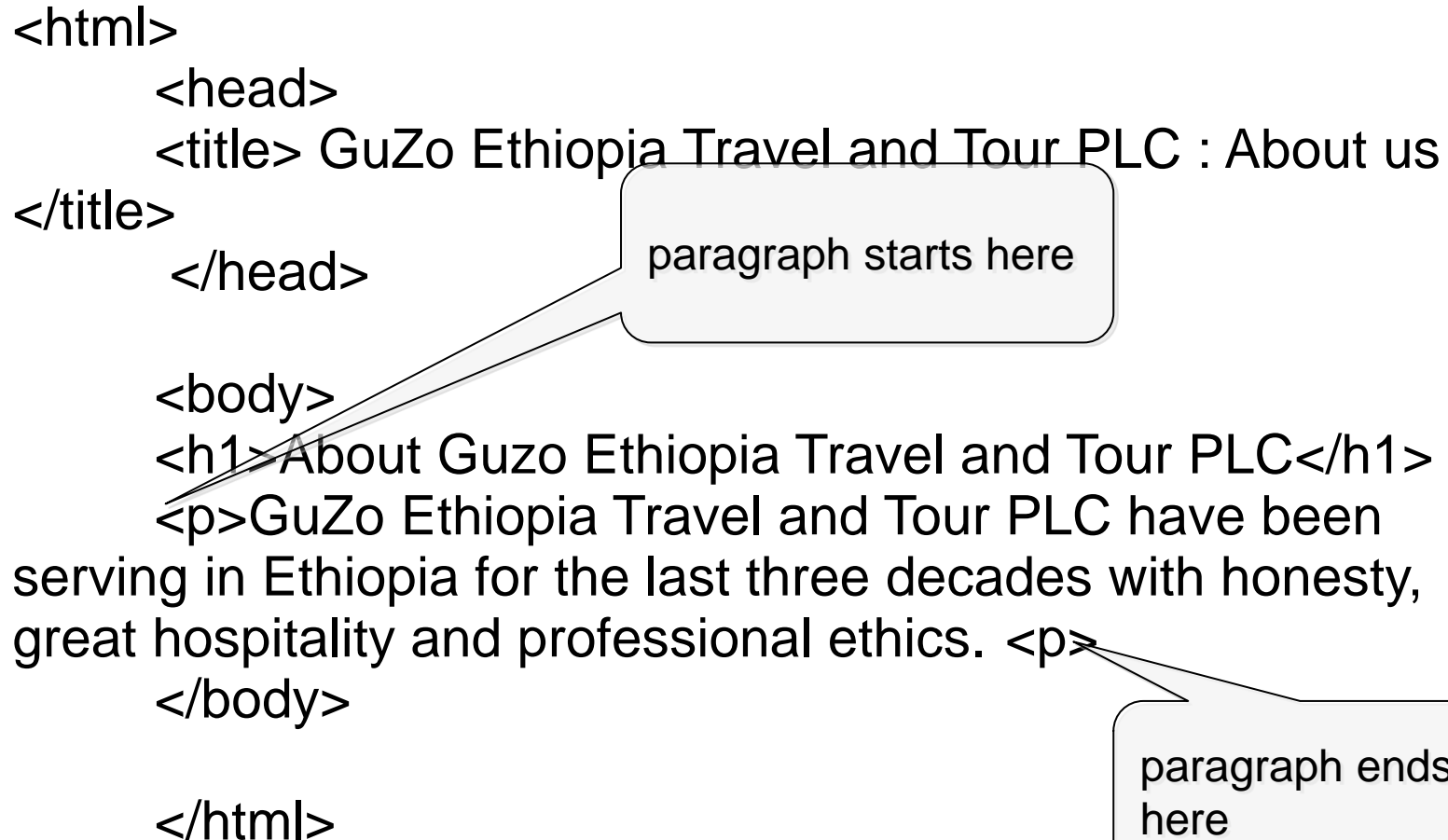
Header one ends here

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>
```



The diagram illustrates the structure of an HTML document. It shows the opening and closing tags for the document, head, body, title, h1, and p. Two callout boxes highlight specific parts of the code: one pointing to the opening <p> tag with the text 'paragraph starts here', and another pointing to the closing <p> tag with the text 'paragraph ends here'.

paragraph starts here

paragraph ends here

Tags

- `<html> . . . </html>`
 - contains the whole page
- `<head> . . . </head>`
 - the head of the page
 - contains information *about* the page which is not displayed
 - Title
 - Description of the page
 - Keywords that search engines can use to index the page
(Meta)

Cont'd . . . **Tags**

- `<body> . . . </body>`
 - the body of the page, main content
 - the information seen in the main browser window

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>  
<head>  
  <title> 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>
```

Specifies the type of font for the body

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

```
<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>

The Structure of a Page



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

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 <html> Element

- It comes after optional XML declaration and required DOCTYPE declaration
- If you are writing **Strict XHTML 1.0**, the opening tag must also include something known as a *namespace identifier* (this indicates that the markup in the document belongs to the XHTML 1.0 namespace).

<html xmlns="<http://www.w3.org/1999/xhtml>">

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"

Basic text formatting

- Whitespace and Flow
 - several consecutive spaces between two words
 - a new line in your source document
 - consecutive empty lines,
 - Tab characters
 - These all will be ignored and simply treated as **one space**
- Useful to indent the code for better readability

example on the next slide

Example:

<html>

<body>

***<p> Several consecutive spaces between two words
, a new line in the source document***

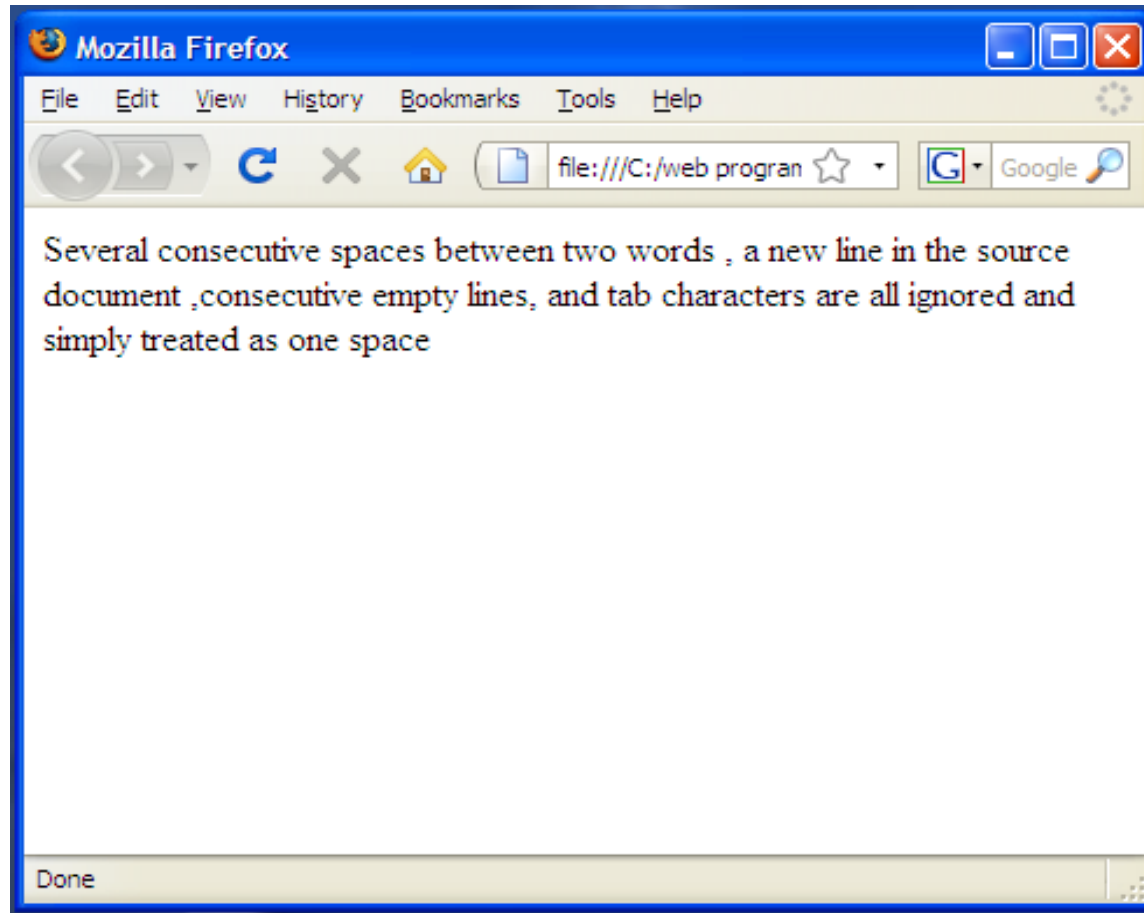
,consecutive empty lines,

and tab characters are all ignored and simply treated as one space </p>

</body>

</html>

Cont'd . . . Example

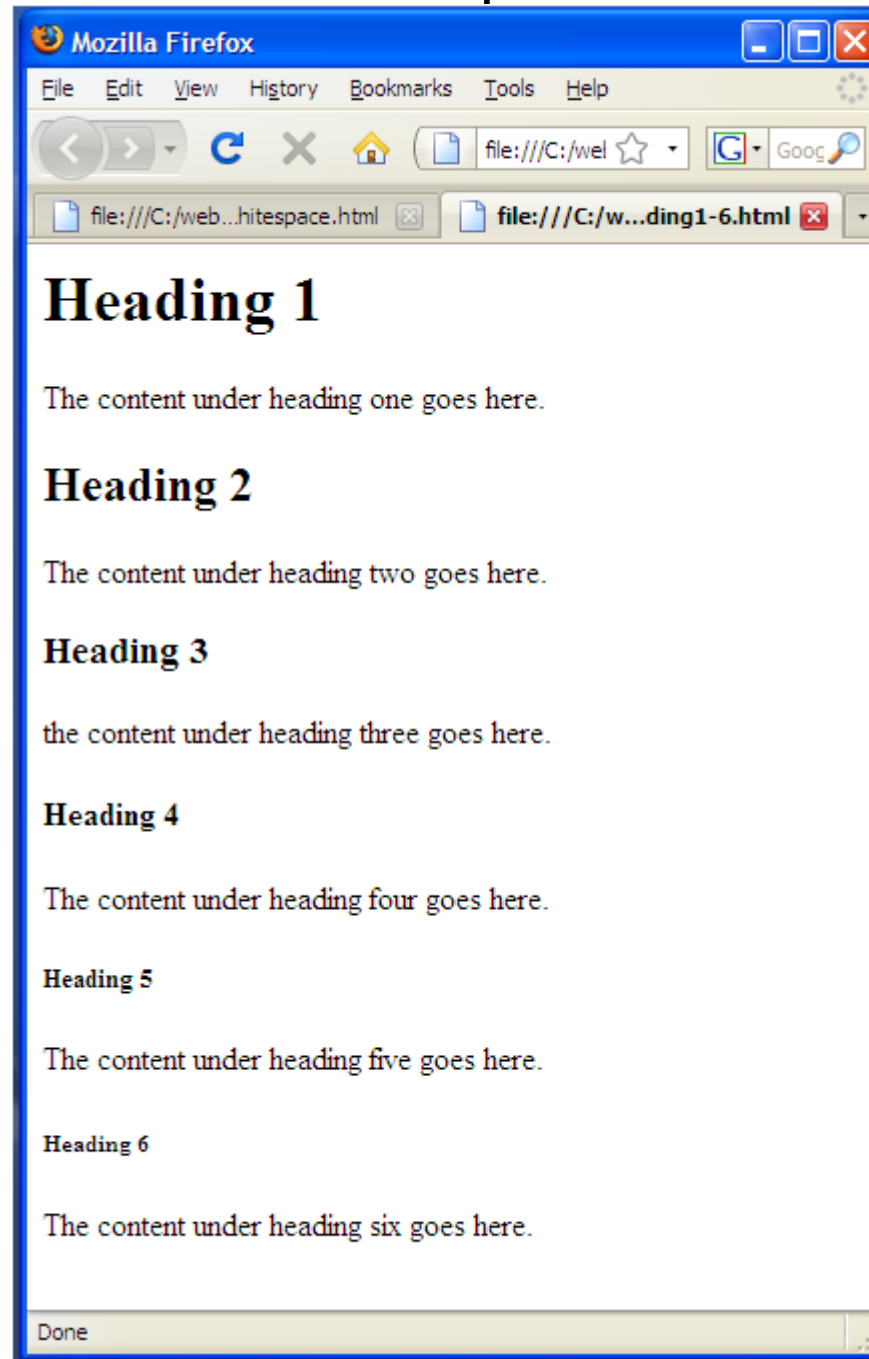


Creating Headings using Hn Element

- 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



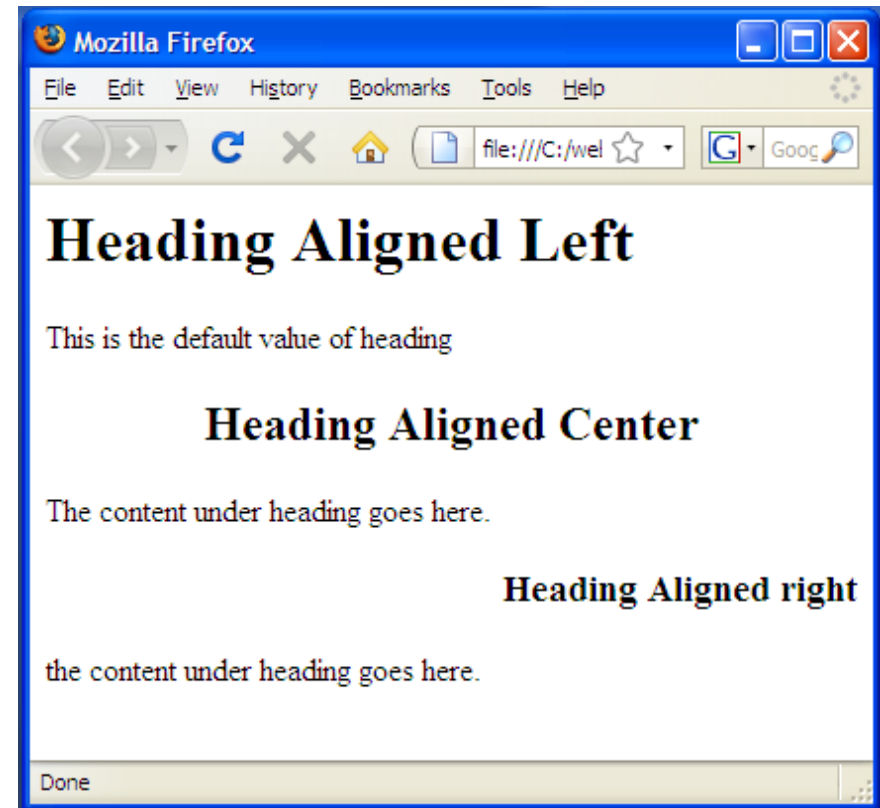
Heading

- 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>`

left	heading is displayed to the left of the browser window
center	heading is displayed in the center of the browser window
right	heading is displayed to the right of the browser window

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.

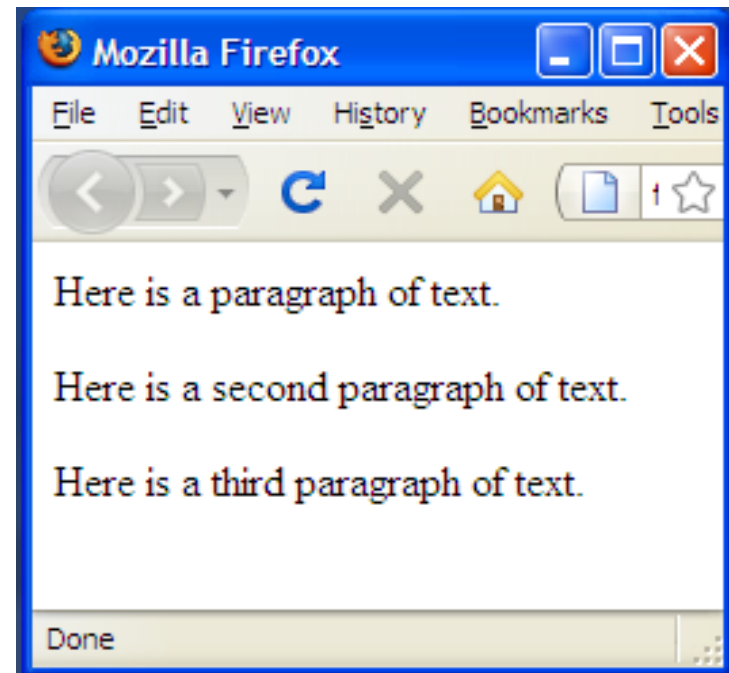
. . .

`<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.

`
` (XHTML)

`
` (HTML)

- It does not need opening and closing tags

eg.

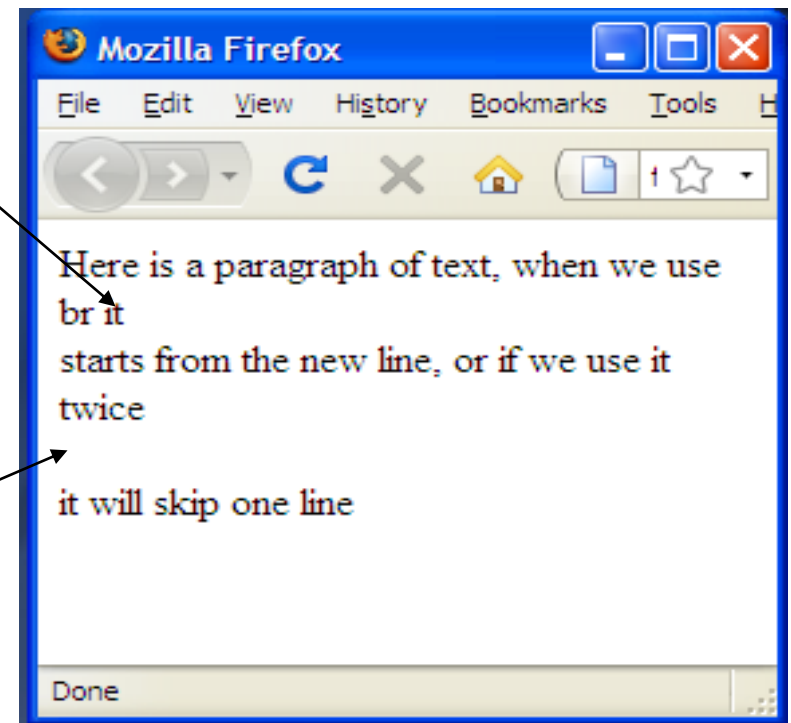
```
<html>
<body>
```

```
<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 formatted 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>

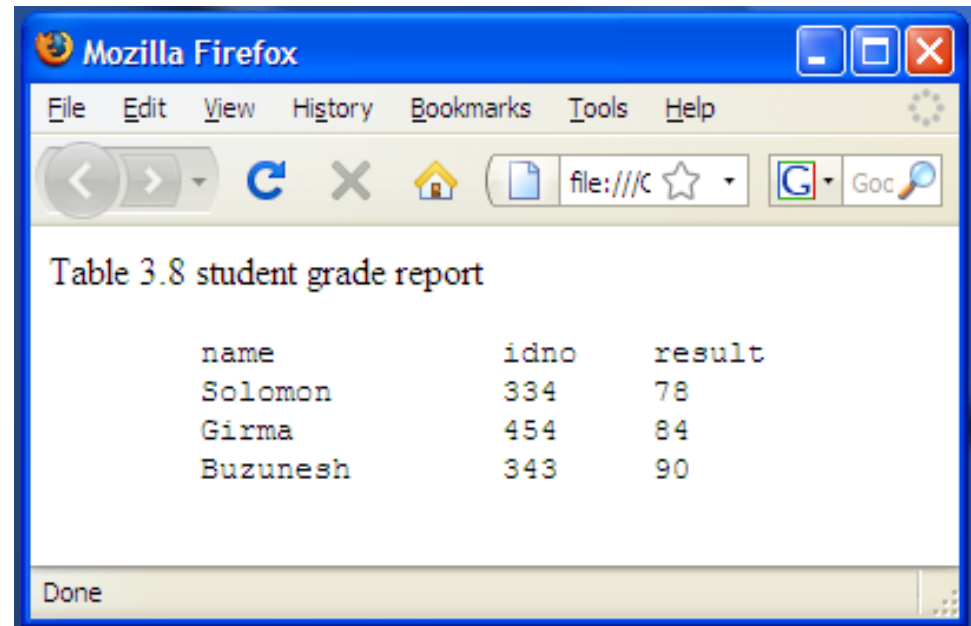
| name | idno | result |
|----------|------|--------|
| Solomon | 334 | 78 |
| Girma | 454 | 84 |
| Buzunesh | 343 | 90 |

</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;
```

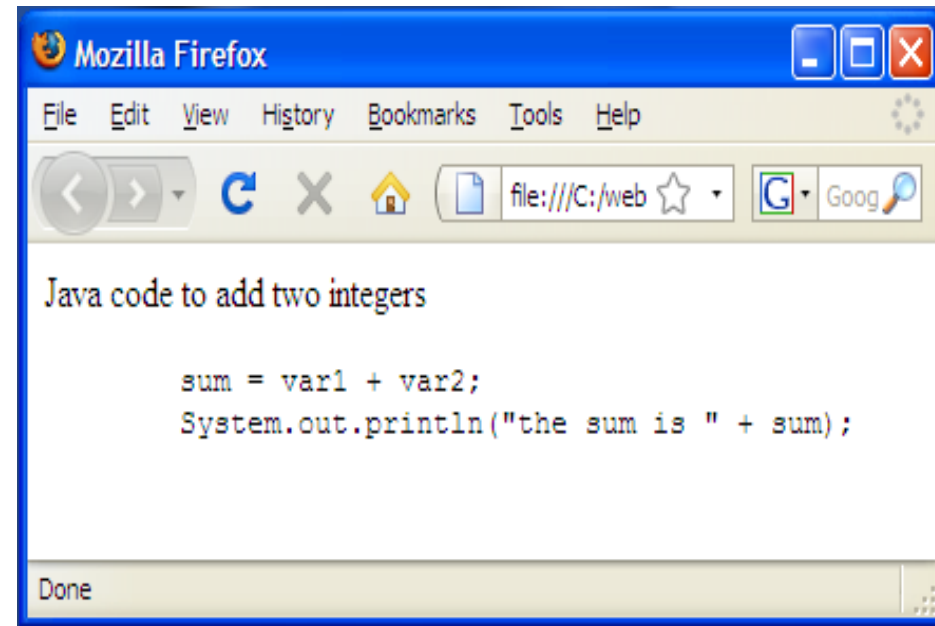
```
System.out.println("the sum is " + sum);
```

</pre>

</p>

</body>

</html>



Horizontal Rules

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

```
<body>
```

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

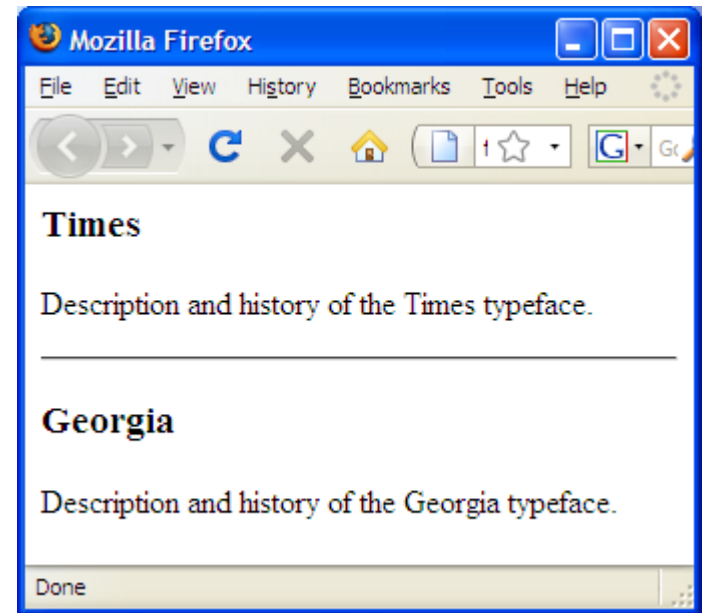
```
<p>Description and history of the Times  
typeface.</p>
```

```
<hr />
```

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

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

```
</body>
```



Address

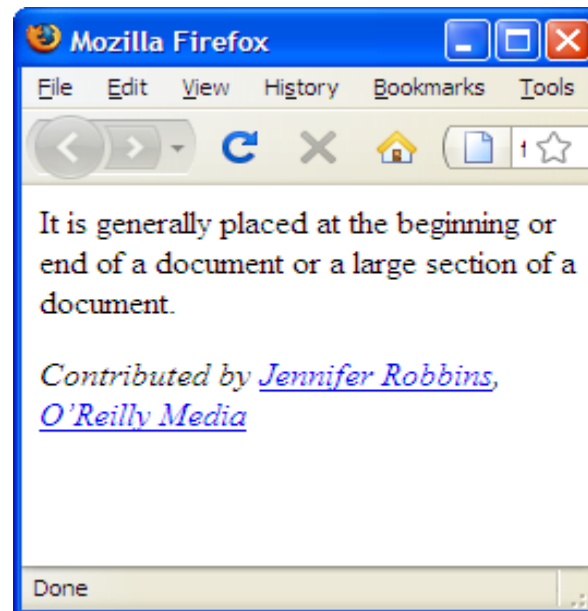
- the **address** element that is used to provide contact information for the author or maintainer of the document.

<address>

Contributed by [Jennifer Robbins](../authors/robbins/),

[O'Reilly Media](http://www.oreilly.com/)

</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

syntax:

``

` . . . `

` . . . `

``

• eg.

`<p>`This is how unordered list show:

``

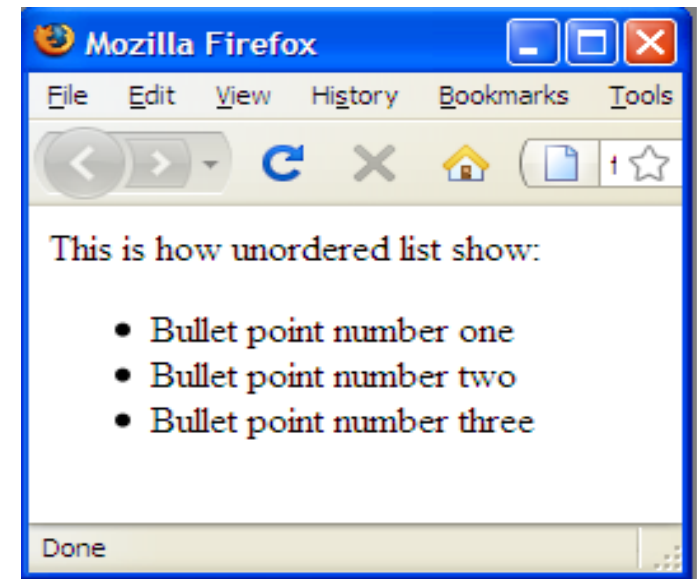
``Bullet point number one``

``Bullet point number two``

``Bullet point number three``

``

`</p>`



Ordered lists

syntax:

``

` . . . `

` . . . `

``

- eg.

`<p>`This is how unordered list show:

``

``Bullet point number one``

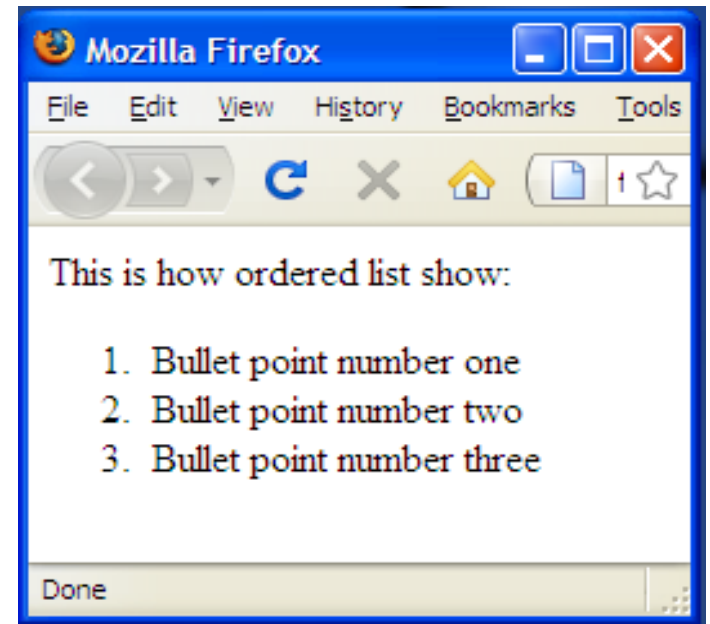
``Bullet point number two``

``Bullet point number three``

``

`</p>`

- For letters or Roman numerals , it is better to use CSS list-style-type property which will be discussed later



Definition lists

Definition
list

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

• eg.

Defined
term

<dl>

<dt>Unordered List</dt>

Definition

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

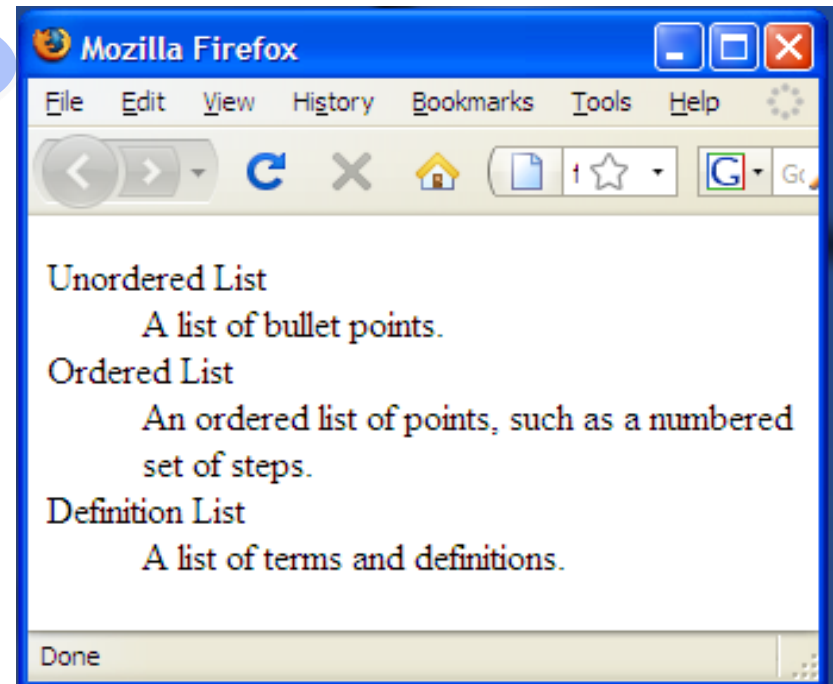
<dt>Ordered List</dt>

<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>



Summary

Table 5-1. Block-level elements for text content

Type	Element(s)
Headings	<code>h1</code> , <code>h2</code> , <code>h3</code> , <code>h4</code> , <code>h5</code> , <code>h6</code>
Paragraphs	<code>p</code>
Block (long) quotes	<code>blockquote</code>
Preformatted text	<code>pre</code>
Various list elements	<code>ol</code> , <code>ul</code> , <code>li</code> , <code>dl</code> , <code>dt</code> , <code>dd</code>
Horizontal rules (lines)	<code>hr</code>

Inline Text Element

- Most (X)HTML text elements are inline elements, which means they just stay in the flow of text and do not cause line breaks.
- Inline text elements fall into two general categories:
 - semantic elements and
 - presentational elements.
- semantic elements describe the meaning of the text; for example, an acronym or emphasized text.
- presentational inline elements are discouraged from use in contemporary web design where style information should be kept separate from the markup.

Semantic inline elements

Semantic inline text elements

Element	Description
abbr	abbreviation
acronym	acronym
cite	citation; a reference to another document, such as a book title
code	program code sample
del	deleted text; indicates an edit made to a document
dfn	the defining instance or first occurrence of a term
em	emphasized text
ins	inserted text; indicates an insertion in a document
kbd	keyboard; text entered by a user (for technical documents)
q	short, inline quotation
samp	sample output from programs
strong	strongly emphasized text
var	a variable or program argument (for technical documents)

Adding emphasis to text

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

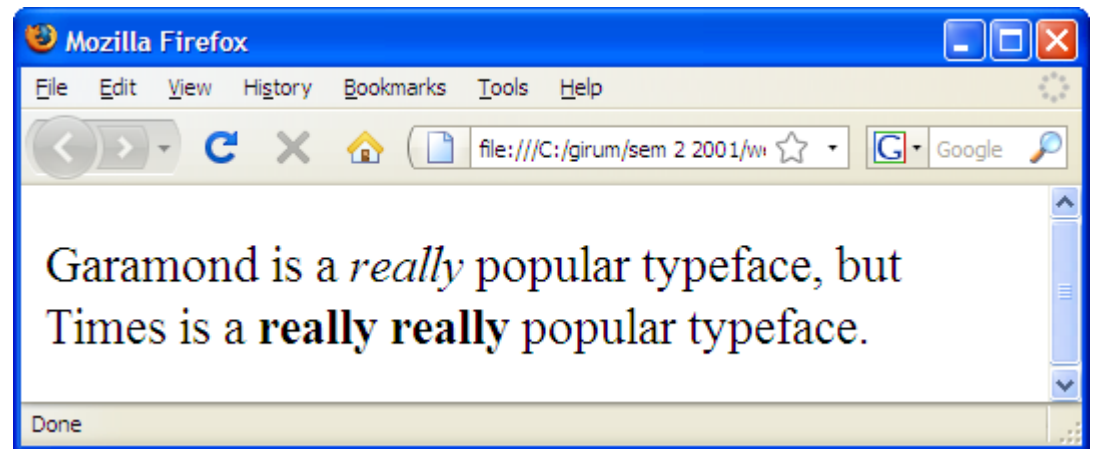
- **em** for emphasized text and

`...`

- **strong** for strongly emphasized text.

`...`

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

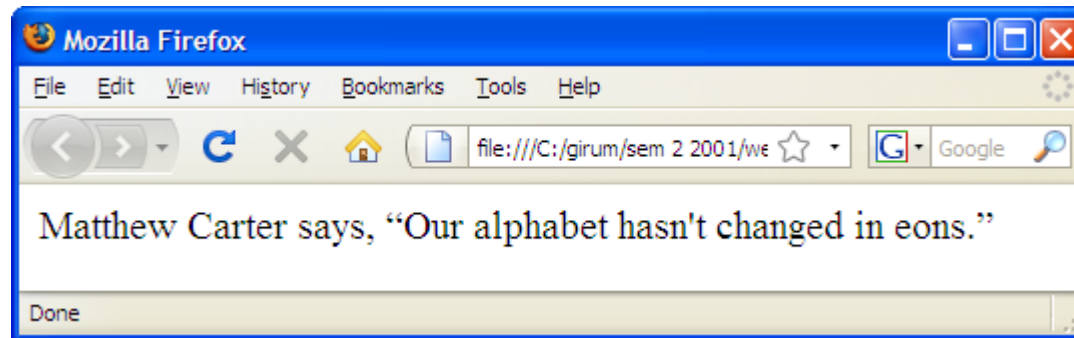


Screen readers may use a different tone of voice to deliver emphasized text, which is why you should use an **em** or **strong** element only when it makes sense semantically, not just to achieve italic or bold text.

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

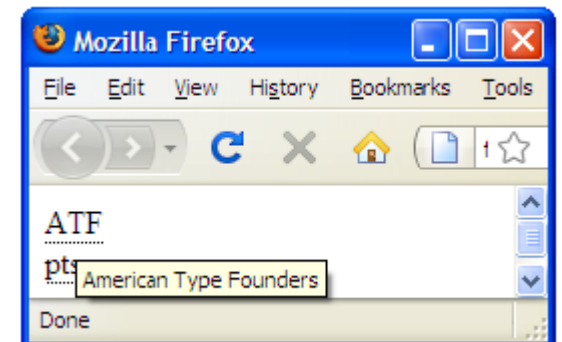
- Marking up shorthand terms as acronyms and abbreviations **provides useful information for search engines, screen readers, and other devices.**
- Abbreviations, indicated by the **abbr** element, are shortened versions of a word ending in a period (Conn. for Connection, for example).

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

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

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

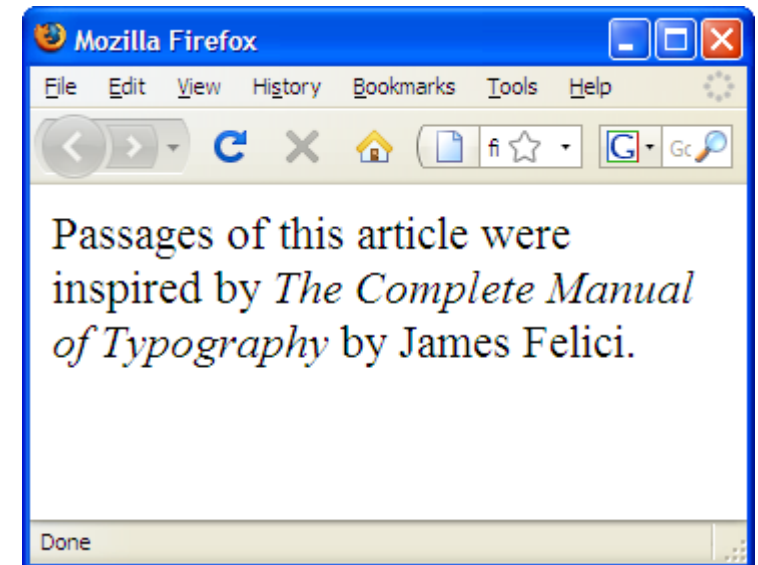
Both elements use the title attribute to provide the long version of the shortened term



Citations

- The **cite** element is used to identify a **reference to another document**, such as a book, magazine, article title, and so on.

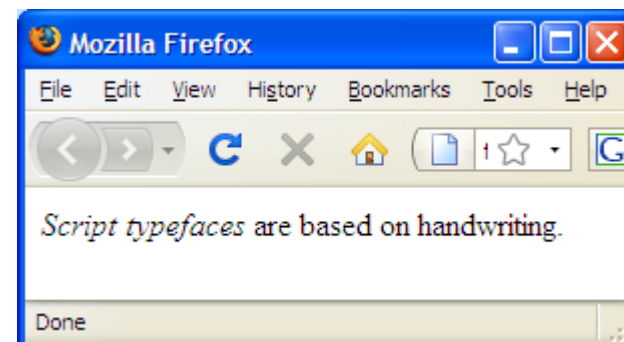
<p>Passages of this article were inspired by
citeThe Complete Manual of Typography
by James Felici.</p>



Defining terms

- Used when defining instance of a word

<p>dfnScript typefaces
are based on handwriting.</p>



Program code elements

- are used for describing the parts of technical documents, such as:
 - code (**code**),
 - variables (**var**),
 - program samples (**samp**), and
 - user-entered keyboard strokes (**kbd**).
- **Code**, **sample**, and **keyboard** elements typically render in a **constant-width (also called monospace)** font such as Courier by default.
- **Variables** usually render in *italics*.

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>
```



Presentational inline elements

- Be careful to keep style information like this out of the (X)HTML document.
- Find alternate solutions - using CSS

Presentational Elements

- Bold ` . . . `
- Italic `<i> . . . </i>`
- Monospaced `<tt> . . . </tt>` teletype (constant width font)
- Underlined `<u> . . . </u>`
- strikethrough `<s> . . . </s>`
- larger
bigger `<big> . . . </big>` makes the text one size
- Smaller `<small> . . . </small>` makes the text one size smaller
- Superscripted `^{. . .}`
- subscripted text `_{. . .}`
- Font ``

Example: Presentational elements

```
<html>
```

```
<body>
```

```
<p>The following word uses a bold typeface.</p>
```

```
<p>The following word uses a italic typeface.</p>
```

```
<p>The following word uses a monospaced typeface.</p>
```

```
<p>The following word is underlined .</p>
```

```
<p>The following word uses a strikethrough.</p>
```

```
<p>The following word has larger fonts.</p>
```

```
<p>The following word has smaller fonts.</p>
```

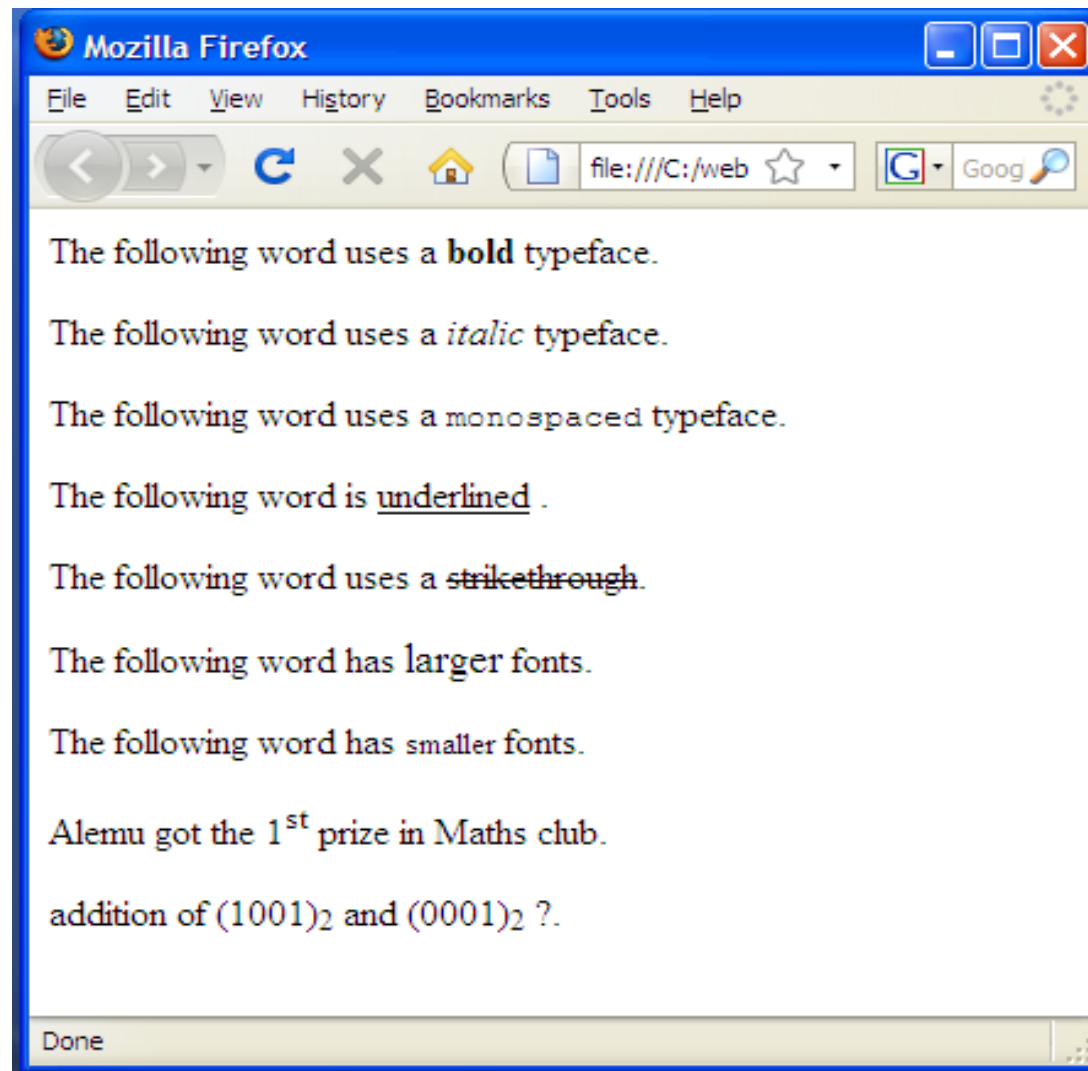
```
<p>Alemu got the 1st prize in Maths club.</p>
```

```
<p>Addition of (1001)2 and (0001)2 ?.</p>
```

```
</body>
```

```
</html>
```

Cont'd . . . Example



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. -->`

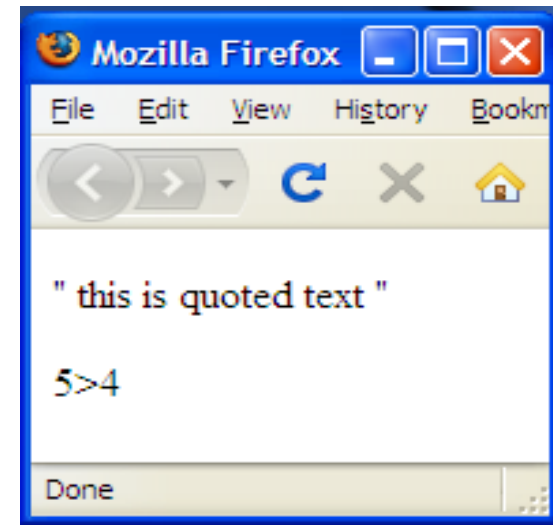
Special Characters

- some characters that have special meaning in XHTML
- use a set of different characters known as a *character entity* to represent these special characters

Character	Numeric Entity	Named Entity
"	"	"
&	&	&
<	<	<
>	>	>
®	®	®
©	©	©

eg. <p> " this is quoted text ";</p>

<p>5>4</p>



Attributes

- **id**

- used to uniquely identify any element within a page or style sheet.
 - to link to a specific point in a document,
 - to select one specific element's content,
 - to associate a CSS style with a particular element,
 - or identify that element using a script
- If you have two elements of the same name within a Web page or style sheet, you can use the id attribute to distinguish between elements
- Syntax : id="idname"

It should be unique within the document
and should start with alphabet

<p id="faculty">This paragraph explains the role of the faculty.</p>
<p id="department">This paragraph explains the role of the department.</p>

Cont'd ... Attributes

- **Title**
 - gives a suggested title for the element
 - Syntax: `title="string"`
- **class**
 - used to associate an element with a style sheet, and specifies the *class* of element
 - Syntax: `class="className"`
- **Style**
 - attribute allows you to specify CSS rules within the element
 - `<p style="font-family:arial;color:#FF0000;">Some text...</p>`
 - The use of this attribute should be avoided since it mixes up the stylistic markup with structural and semantic markup

Generic Elements

- The `<div>` and `` elements allow you to group together several elements to create sections or subsections of a page.
- Commonly used with CSS to allow you to attach a style to a section of a page
 - `<div>` used to group block-level elements together.
 - `` is used to group inline elements only.
- Both use `id` or `class` to give names to the generic `div` and `span` elements

```
<div class="footnotes">
```

```
<h2>Footnotes</h2>
```

```
<p><span class="inventor"><b>1</b> The World Wide Web was invented by Tim  
Berners Lee</span></p>
```

```
<p><b>2</b> The W3C is the World Wide Web Consortium who maintain many Web  
standards</p>
```

```
</div>
```

```
<div>...</div>
```

Generic block-level element

```
<span>...</span>
```

Generic inline element

Questions?

[Links and Navigations](#)

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.



Basics

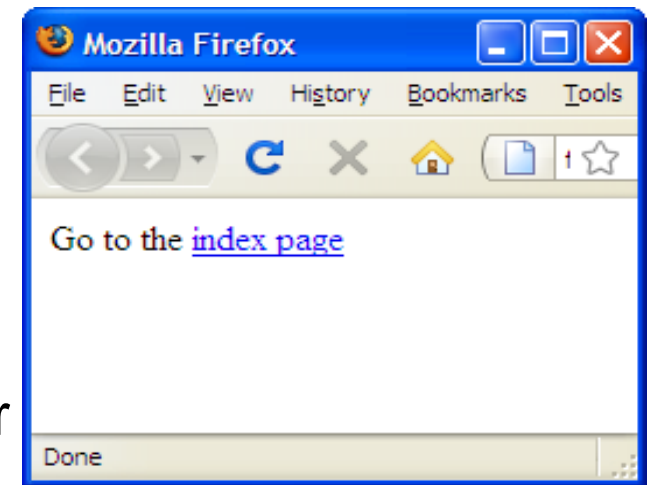
- A link is specified using the **anchor** `<a>` element.
- Anything between the opening `<a>` tag and the closing `` tag becomes part of the link a user can click in a browser.
- To link to another document, the opening `<a>` tag must carry an attribute called **href**, whose value is the page you are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

`<body>`

Go to the ``index page ``

`</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>`

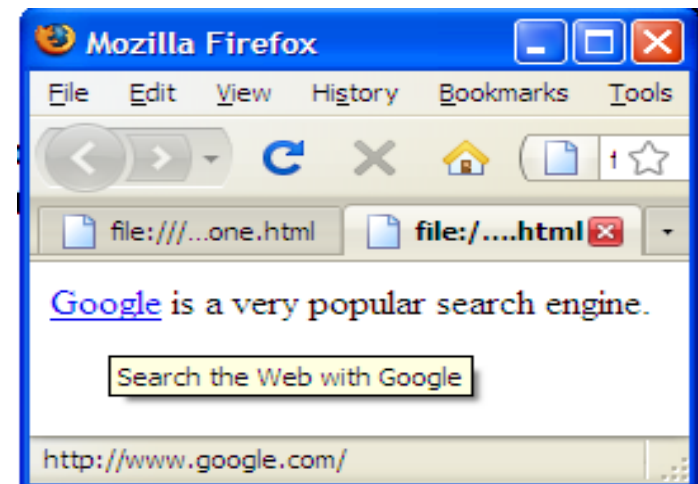
- You can include **title** attribute to links - when the mouse is over the link additional information could be delivered to the user.

`<body>`

`Google`

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.

Absolute and Relative URLs

- No two files on the Internet share the same URL
- An **absolute URL** uniquely identify a particular file on the Internet.

<http://www.exampleNewsSite.com/Entertainment/Film/index.htm>

- A **relative URL** indicates where the resource is in relation to the current page.
- If you are at the index page for the entertainment section you can link to film index page in the following way:
 - Film/index.htm
- the browser changes the relative URLs into full absolute URLs.

(Cont'd. . .) Absolute and Relative URLs

- Another key of using relative URLs within your site is that it means you can **change your .domain name**

Same Directory

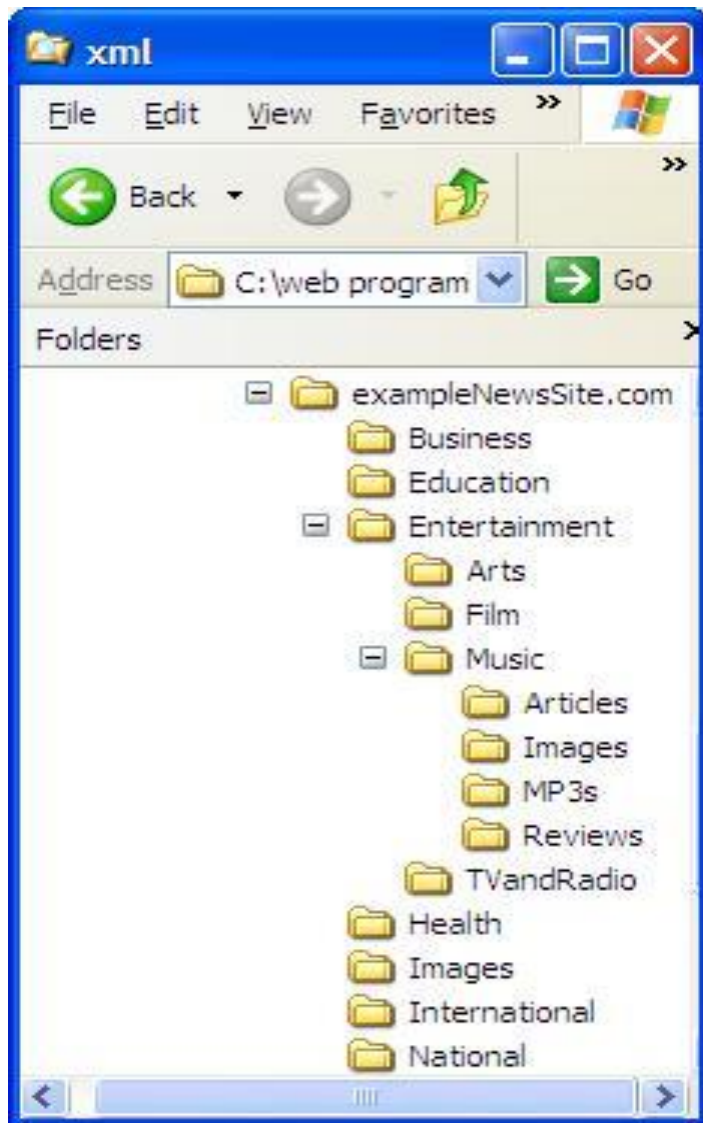
- When you want to link to or include a resource from the same directory, you can just use the name of that file.
- For example, to link from the home page (index.html) to the contact us page (contactUs.htm),
 - contactUs.htm

Linking Within Your Own Site

Directories and Directory Structures

- Directory is another name for a folder on the Web server
- The main directory that holds the whole of your Website is known as the *root folder* of your Website.
- A directory that is within another directory is known as a *subdirectory*.
- A directory that contains another directory is known as the *parent directory* of the subdirectory.

Example



- Root directory : exampleNewSite.com
- Film is a subdirectory of Entertainment
- Entertainment is the parent directory of
 - Film,
 - Television and radio
 - Arts, and
 - Music.

Example

- Linking within a directory
 - `About the site...`
- Linking to a lower directory
 - From root-directory(exampleNewSite.com) to index in Film directory
 - `Visit Film's `
- Linking to a higher directory
 - Use `../` to link one level up in the directory
 - Use `../ ../` to link two level up in the directory and so on
 - To link from **Film directory** to the **Entertainment's index page**
 - `Visit Film's `

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>
```

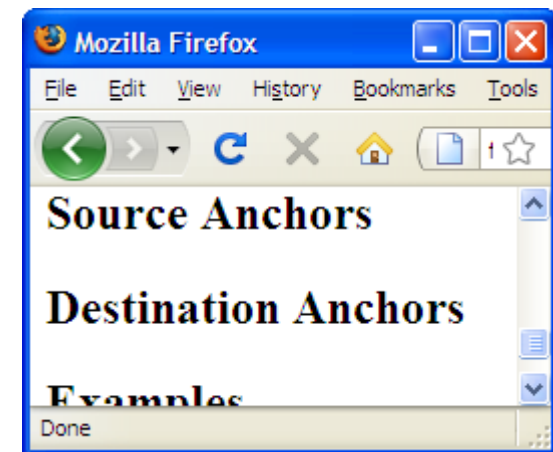
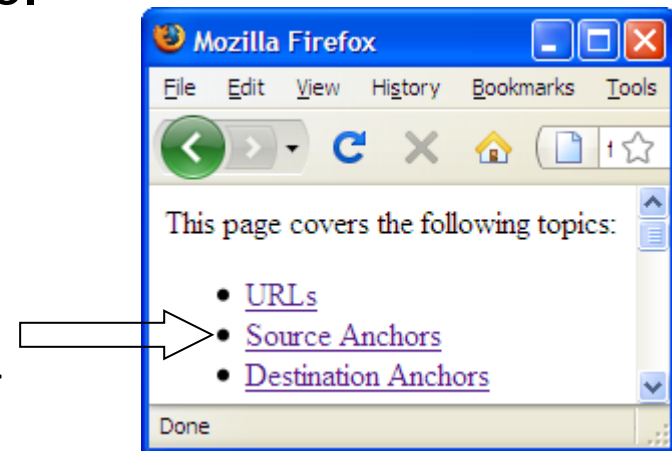
```
<h2><a name="URL">URLs</a></h2>
```

```
<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 a fragment in another document

- You can link to a fragment in another document by adding the **fragment name** to the end of the URL (absolute or relative).

`See the Examples`

Targeting a New Browser Window

- Problem with putting links on your page is that when people click on them, they may never come back.
- Solution to this Problem is to have the linked page open in a new browser window
- To open a new window using (X)HTML markup,
 - use the **target** attribute in the anchor (**a**) element ,
 - Set the value of target to **_blank** (**target="_blank"**)

Opens a fresh window to display the link

For example:

```
<a href="http://www.oreilly.com" target="_blank">O'Reilly</a>
```

Or to open one window and to use it repeatedly:

```
<a href="http://www.oreilly.com" target="display">O'Reilly</a>
```

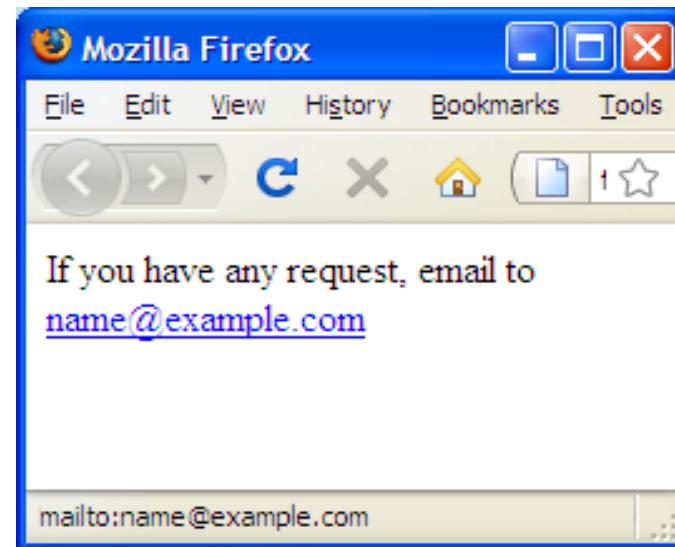
Could be any name

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

E-mail us.



(Cont'd. . .) Linking to E-mail Addresses

- You can also specify some other parts of the message, too, such as the
 - subject,
 - body, and
 - people that it should be cc'd or blind cc'd to.

Tables

- Are used
 - displaying tabular data
 - to format whole pages and create visually attractive layouts (recently CSS is becoming the standard for layout and tables are not recommended)
- 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>
```

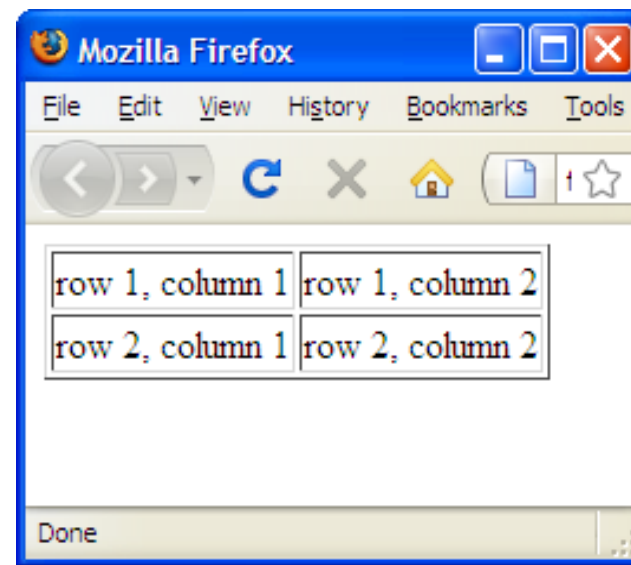


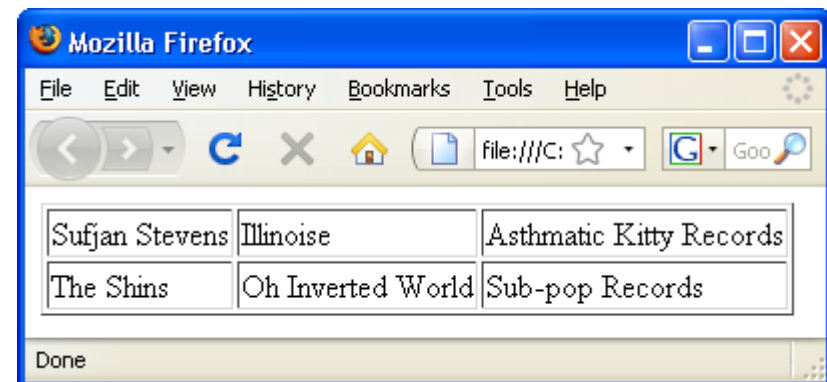
Table Headers

- 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

Exercise

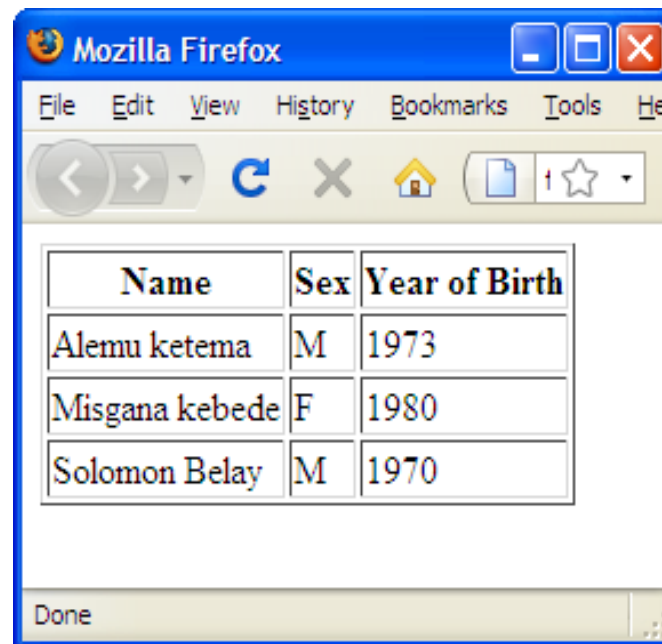
In the code fragment below how many columns and rows are there?

```
<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



The screenshot shows a Mozilla Firefox browser window. The address bar is empty. The main content area displays a table with the following data:

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

Spanning Cells

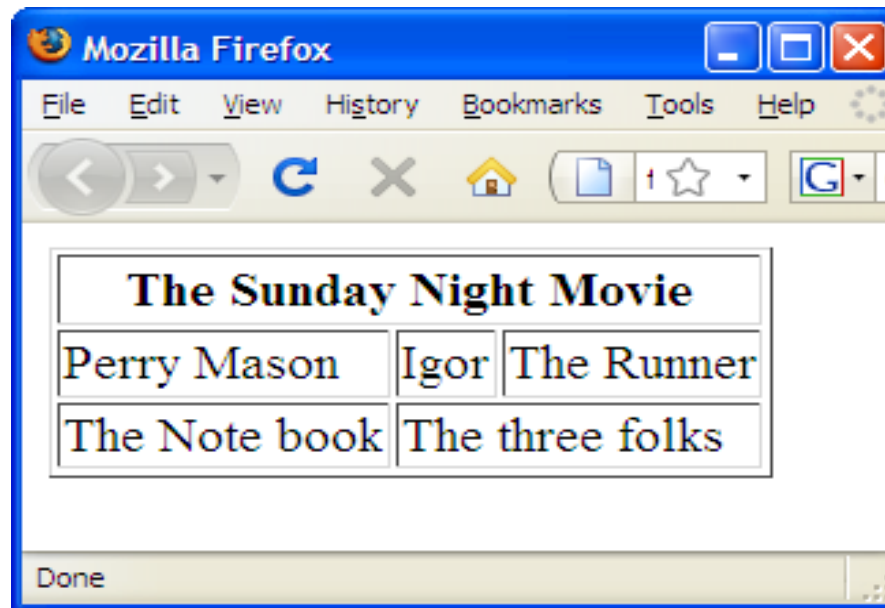
- is the stretching of a cell to cover several rows or columns
- You make a header or data cell span by adding the **colspan** or **rowspan** attributes
- **Column spans**, created with the **colspan** attribute in the **td** or **th** element, stretch a cell to the right to span over the subsequent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

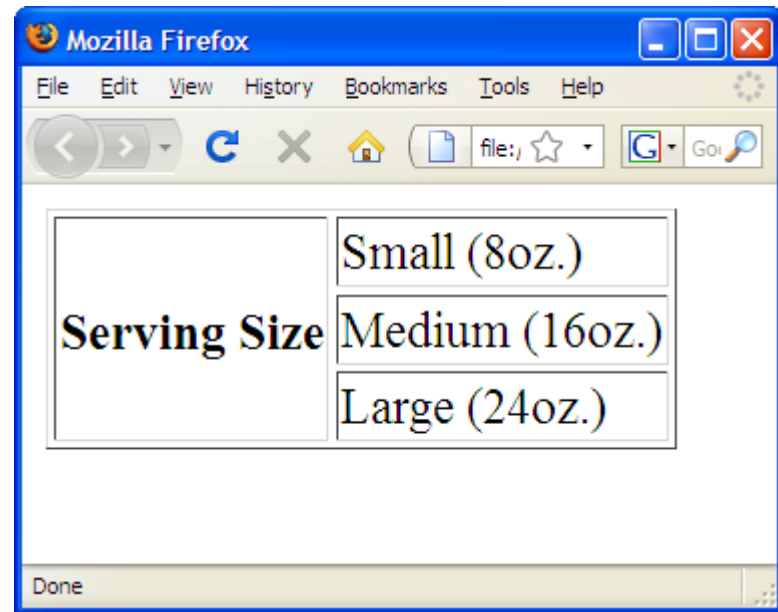
- write the markup for the table shown



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



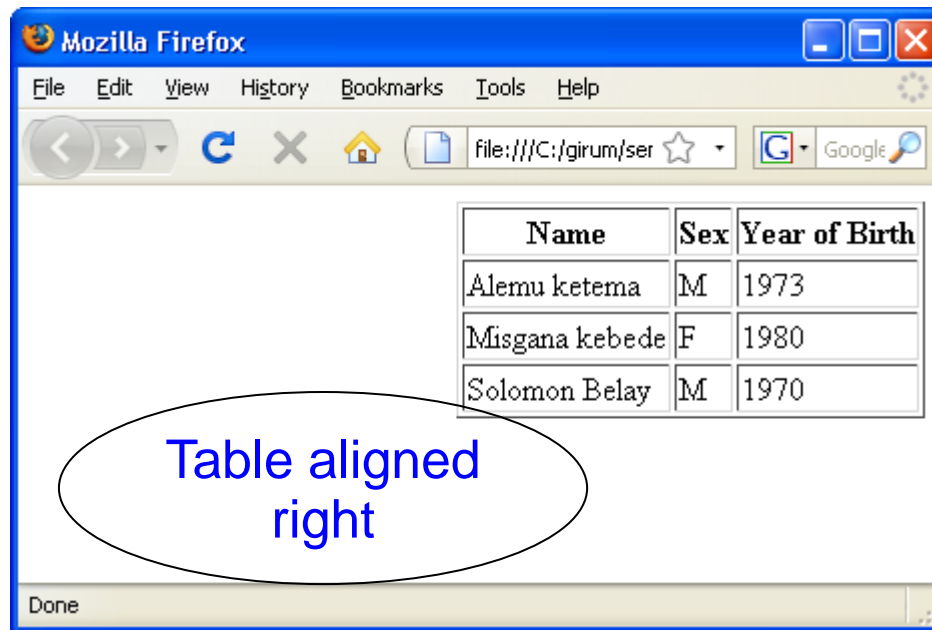
A screenshot of a Mozilla Firefox browser window. The window title is "Mozilla Firefox". The menu bar includes "File", "Edit", "View", "History", "Bookmarks", "Tools", and "Help". The address bar shows "file:///". The search bar contains the Google logo and the text "Go". The main content area displays a table with three rows and three columns. The first column contains "Apples", "Bananas", and "Lychees". The second column contains "Oranges". The third column contains "pears" and "Pineapple". The status bar at the bottom shows "Done".

Apples		pears
Bananas	Oranges	
Lychees		Pineapple

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">

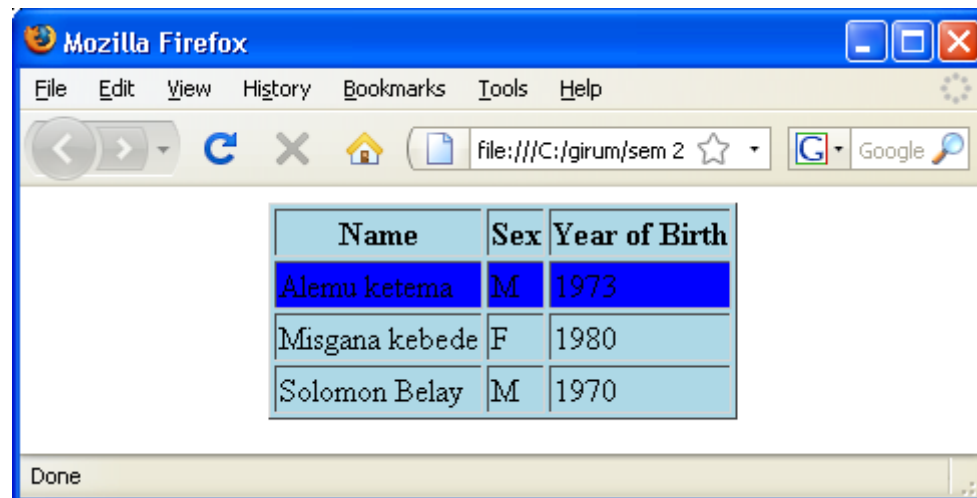


The bgcolor and background attributes

- 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">



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

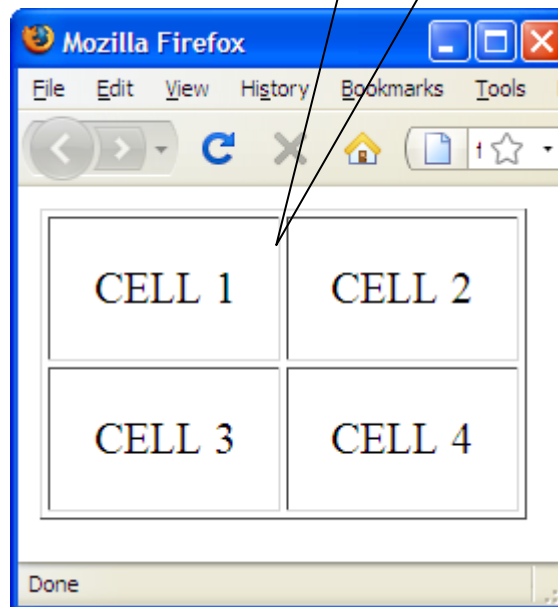
Cell Padding and Spacing

- 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.

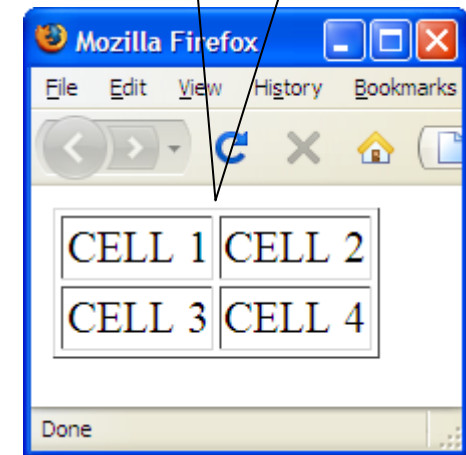
– Example:

```
<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>
```

With cellpadding



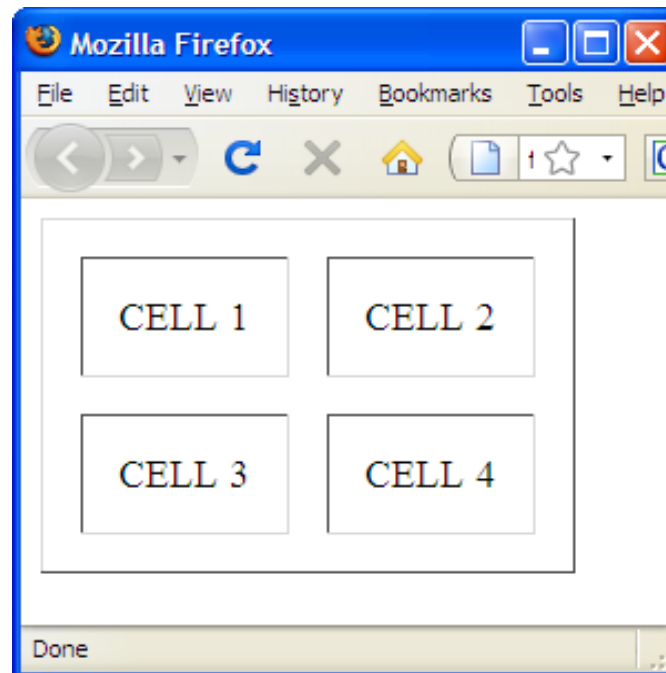
With out cellpadding



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

The screenshot displays the HTML code in Notepad++ and its rendered output in Mozilla Firefox. The code defines a table with three columns: Menu item, Calories, and Fat (g). The first row contains 'Chicken noodle soup', '120', and '2'. The second row contains 'Caesar salad', '400', and '26'. The 'Menu item' header is aligned top, 'Calories' is centered, and 'Fat (g)' is aligned bottom.

```
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>
```

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

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

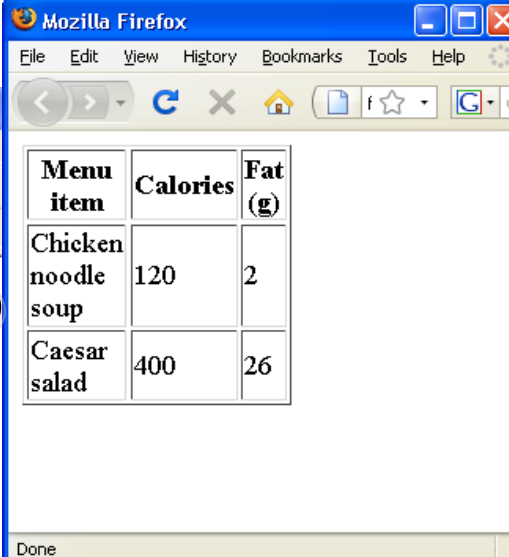
C:\giram\sem 2 2001\web programing\code\tables\table_simple2.html - Notepad++

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

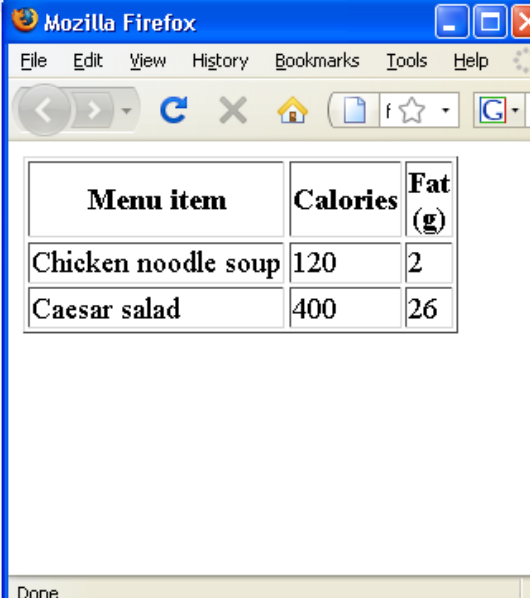
table_simple2.html

```
2 <body>
3 <table border=1 width=20>
4 <tr>
5     <th>Menu item</th>
6     <th>Calories</th>
7     <th>Fat (g)</th>
8 </tr>
9 <tr>
10    <td nowrap>Chicken noodle soup</td>
11    <td>120</td>
12    <td>2</td>
13 </tr>
14 <tr>
15    <td>Caesar salad</td>
16    <td>400</td>
17    <td>26</td>
18 </tr>
19 </table>
20 </body>
21 </html>
```

Without
nowrap



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

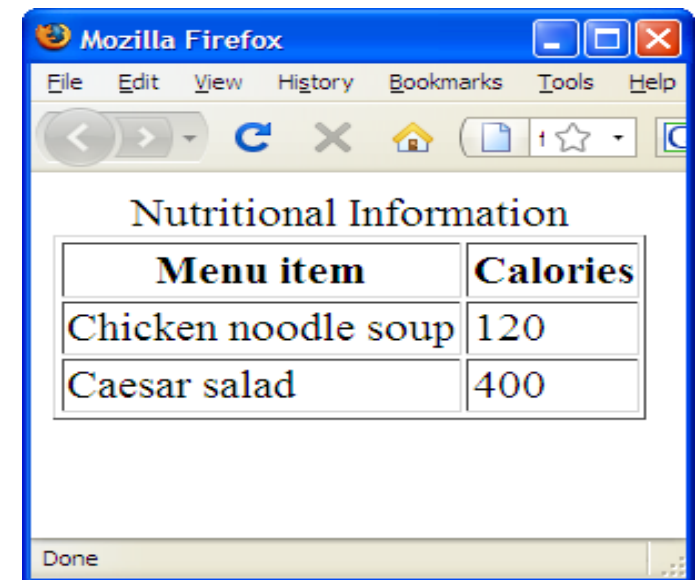


Menu item	Calories	Fat (g)
Chicken noodle soup	120	2
Caesar salad	400	26

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>
```

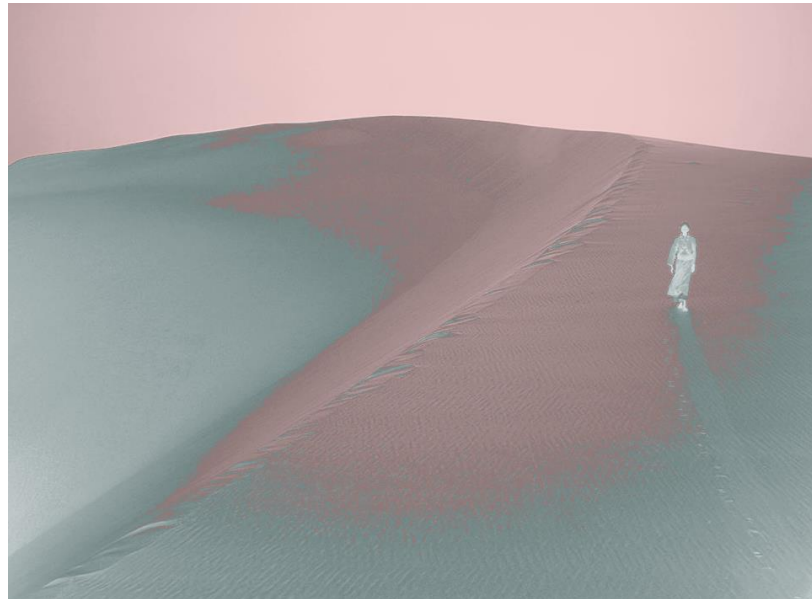
Exercise:

- 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

Questions ?

Colors, Images, and Objects



Adding Color to Your Site

In XHTML there are two key ways of specifying a color:

- **Hex codes:** A six-digit code representing the amount of red, green, and blue that make up the color, preceded by a pound or hash sign # example, #333333.
- **Color names:** A set of names that represent over 200 colors, such as red, blue

Color Hexadecimal Code

Black	#000000
White	#FFFFFF
Red	#FF0000
Green	#008000
Blue	#0000FF
Purple	#800080

Cont'd ... Adding Color to Your Site

- The numbers that follow the *#* sign is the amount of red, green, and blue that make up the color.
 - The format for hex codes is: *# rrggbb*
 - *Each letter represent a hexadecimal number and*
 - *each basic color has a value of 0 to 255 (00000000 - 11111111)or (00 - ff)*
 - *If all values of red, green, and blue are 0 the color = black*
 - *If all have 255 then the color would be white*

Using Color Names to Specify Colors

- you can also use color names such as red, green, and white to specify the color.
- Example:
 - aqua, beige, coral, darkcyan, firebrick, green, honeydew, indianred,
- But usually it is better to use hex codes since it gives more color options.
- Most graphic softwares also provide a color-picking tool

Choosing colors for website

- In order to choose the colors you are going to use on the site you must have an understanding of the following:
 - How colors are made up, and some key terms that describe colors
 - What a color wheel is, and how it can help you choose your color schemes
 - Different types of color scheme
 - What the Web safe color palette is and why you can largely ignore it
 - Factors that make colors look different on different computers
 - Some issues regarding usability and color

Adding Images to Your Site

- Adding images can really slow down the speed of a site—and slow sites frustrate users. Watch out!
- Types of Image Formats
 - There are two main ways in which graphics are created for computers:
 - *Bitmapped graphics* divide a picture into a grid of pixels and specify the color of each pixel,
 - Example: BMP, JPEG, GIF, TIFF, and PNG
 - *Vector graphics* break the image into lines and shapes (like a wireframe drawing), and store the lines as coordinates, then fill the spaces between the lines with color.
 - AI, SVG, Flash

Bitmap Images

Most static images on the Web will be bitmapped images.

- **GIF:** Graphics Interchange format
 - Uses one of one of the 256 colors
- **JPEG:** Joint Photographic Experts Group format (pronounced "jay peg")
- **PNG:** Portable Network Graphics (pronounced "ping")

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

Cont ... Adding Images

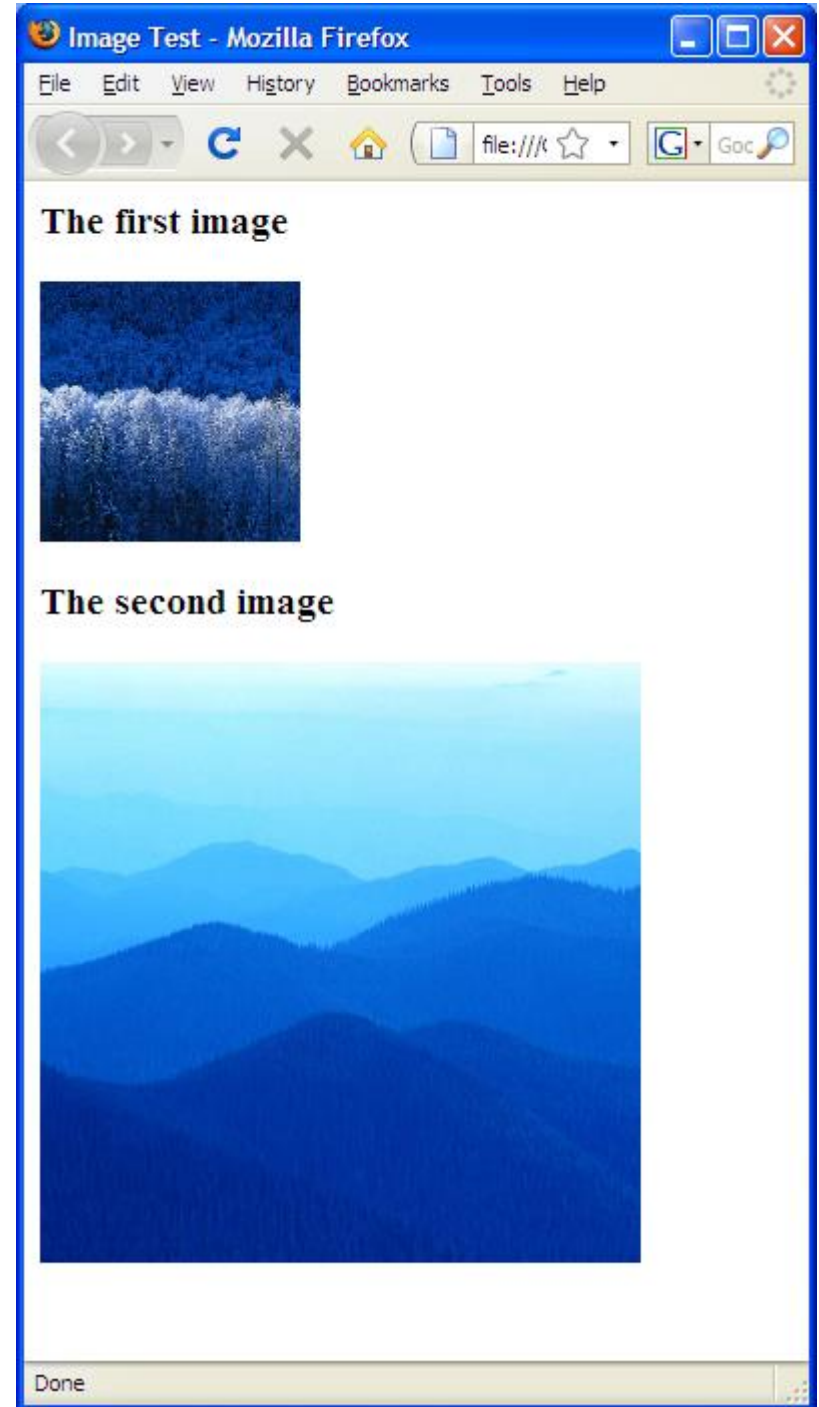
- **ismap usemap** – used with image maps (will be discussed later)
- **Name**- to specify a name for the image so that it can then be referenced from script code.to specify a name for the image so that it can then be referenced fromscript code. Now it is substituted by **id**.

Example

<h3>The first image </h3>

<h3>The second image</h3>

</body>



Using Images as Links

- Images are often used to create graphical buttons or links to other pages.
- place the `` element inside an `<a>` element:

```
<a href="../../../index.html" title="Click here to return  
to the home page">
```

```
</a>
```

Image Maps

- Image maps allow you to specify different areas of an image in your code, so that when users click different parts of the image, they get taken to different pages.
- It uses two specific tags : **map** and **area**

```

```

```
<map name="color">
```

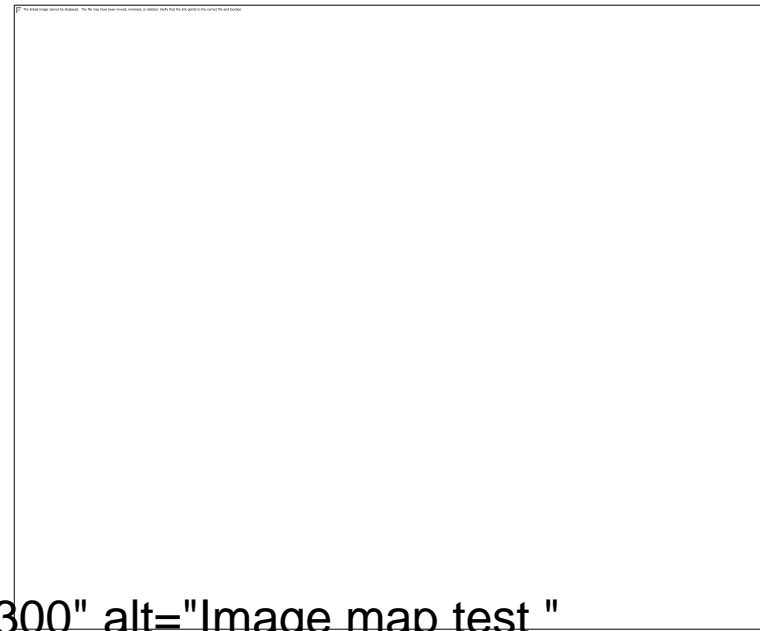
```
<area shape="rect" coords ="25,25,125,125" href="blueRectangle.html"  
alt="blue rectangle" title="Blue rectangle" />
```

```
</map>
```

Image Map Attributes

Attributes	Value	Description
shape	rect,poly or circle	defines the shape of the hot spot
coords	rect: x1,y1,x2,y2 lower-right points	defined by upper-left and
	poly: x1,y1,x2,y2,x3,y3 . . .	defined by each of their points
	circle: x,y,z	(x,y) center coordinates and the radius (r)
href	filename.html	defines the linked page
alt	text string	defines the alternate text
target	target="_self"	specifies window the page should be loaded into.

Example



```
<body>
```

```

```

```
<map name="color">
```

```
<area shape="rect" coords="25,25,125,125" href="blueRectangle.html" alt="blue  
    rectangle" title="Blue rectangle" />
```

```
<area shape="circle" coords="150,150,25" href="circle.html" alt="circle" title="circle" />
```

```
<area shape="poly" coords="95,300,85,165,200,265,300,161,300,300" href="poly.html"  
    alt="green polygon" title="Green area"/>
```

```
</map>
```

```
</body>
```

Questions?

Frames

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*.

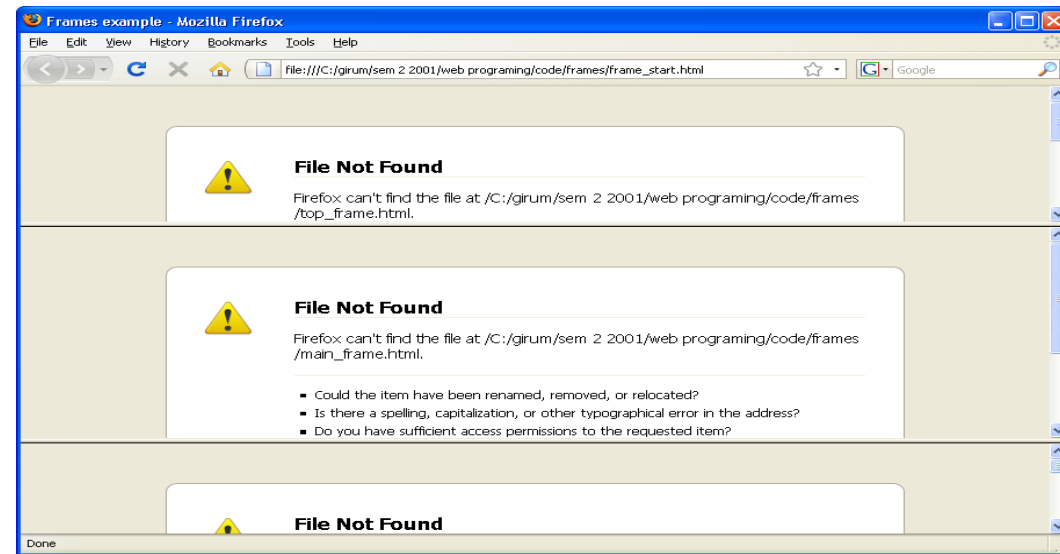
(Cont'd . . .) Frames

- 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"

Additional <frameset> Element

- 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>

Creating Links Between Frames

The most popular uses of frames is to place **navigation bars** in one frame and then load the pages with the content into a separate frame. This is particularly helpful in three situations:

- When your **navigation bar is rather large in size** (such as thumbnails of photographs in a gallery). By using frames, the user does not need to reload the navigation bar each time she views a new page.
- When your **main document is very long** and the navigation bar provides shortcuts to parts of the main document.
- When you do not want to reload the whole page.

Creating Links Between Frames

In `frame_link.html`

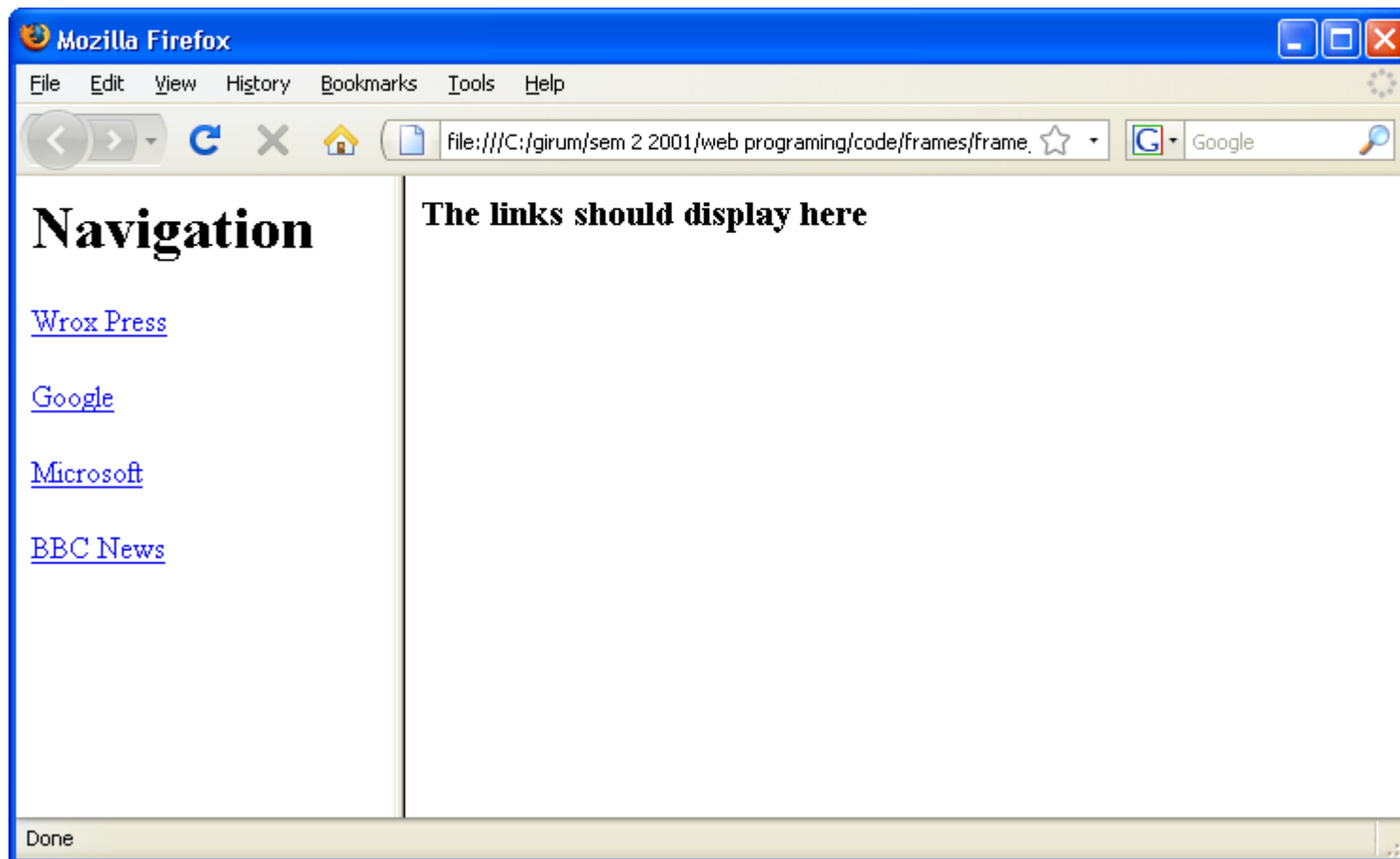
```
<frameset cols="200, *"
<frame src="frames/linksNav.html" />
<frame src="frames/linksMain.html" name="main_page" />
</frameset>
```

- In `linksNav.html`

```
<body>
<h1> Navigation </h1>
<a href="http://www.wrox.com" target="main_page">Wrox Press</a><br />
<a href="http://www.google.com" target="main_page">Google</a><br />
<a href="http://www.microsoft.com" target="main_page">Microsoft</a><br />
<a href="http://news.bbc.co.uk/" target="main_page">BBC News</a><br />
</body>
```

- In `linksMain.html`

```
<body>
<h3> The links should display here </h3>
</body>
```



target

- The target attribute can also take the attribute values

Value	Purpose is to load a page
_self	into the current frame.
_blank	into a new browser window
_parent	into the parent window
_top	into the browser window, replacing any current frames

Default Target Frames

- **Setting a Default Target Frames Using the <base> Element**
 - You can set a default target frame using the <base> element in any page that contains links that should open in another frame.
 - The <base> element should carry an attribute called target, whose value is the name for the frame you want the content to be loaded into.
- So, you could add the following to linksNav.html to specify a default frame target:

```
<head>
```

```
<base target="main_page" />
```

```
</head>
```

Inline Frames

- define a frame that exists within a conventional document, displayed as part of that document's text flow.

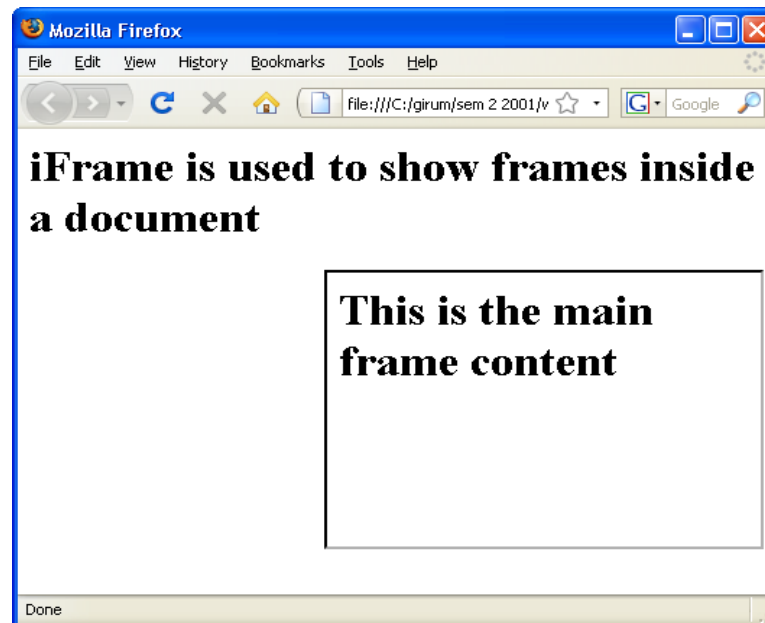
Uses **<iframe>** Tag

`<h1> iFrame is used to show frames inside a document</h1>`

`<iframe src="main_frame.html", frameborder = "1",width=175 height=200 align=right>`

Your browser does not support inline frames. To view this `` document`` correctly, you need to install a more recent browser on your computer.

`</iframe>`



When to Use Frames

- use frames to compare a returned form with its original to verify the content the user submitted.
- to hold a table of contents and various searching tools for the collection.

?

Animated Text

- The `<marquee>` Tag

Function: Creates a scrolling text marquee

Attributes: align, behavior, bgcolor, class, controls, direction, height, hspace, loop, scrollamount, scrolldelay, style, vspace, width

End tag: `</marquee>`; never omitted

Contains: plain_text

Used in: body_content

Example

The behavior, direction, and loop attributes

- The behavior attribute accepts three values:
 - scroll (default)
 - Slide
 - Alternate

If you do not specify a marquee behavior, the default behavior is scroll.

- The **direction** attribute sets the direction for marquee text scrolling.
- The **loop** attribute determines how many times the marquee text scrolls.
- The **bgcolor** attribute lets you change the background color of the marquee area.

Example Slide

Example Alternate

The scrollamount and scrolldelay attributes

- The **scrollamount** attribute value is the number of pixels needed to move text each successive movement during the scrolling process.
 - Lower values mean smoother but slower scrolling;
 - higher numbers create faster, jerkier text motion.
 - The **scrolldelay** attribute lets you set the number of milliseconds to wait between successive movements during the scrolling process.
 - The smaller this value, the faster the scrolling.
- ```
<marquee scrollamount=1 scrolldelay=1>
```

example



# 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 )

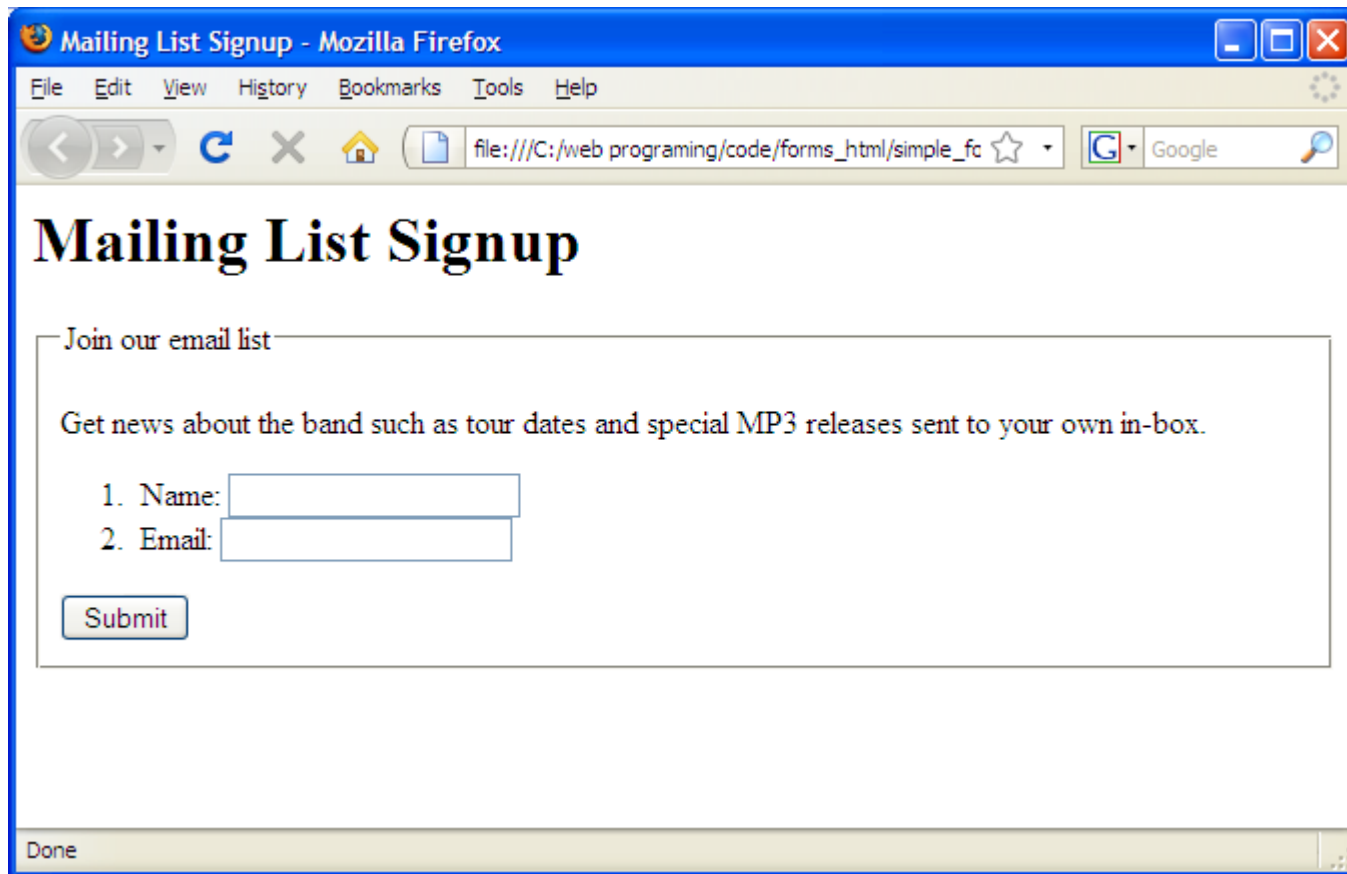
# Example

```
<html>
<head>
<title>Mailing List Signup</title>
</head>
<body>
<h1>Mailing List Signup</h1>
<form action="/cgi-bin/maillinglist.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>

<label for="name">Name:</label>
<input type="text" name="name" id="name" />
<label for="name">Email:</label>
<input type="text" name="email" id="email" />

<input type="submit" value="Submit" />
</fieldset>
</form>
</body>
</html>
```

# Cont'd ... Example



The screenshot shows a Mozilla Firefox browser window with the title 'Mailing List Signup - Mozilla Firefox'. The address bar displays a local file path: `file:///C:/web programming/code/forms_html/simple_fc`. The page content includes a heading 'Mailing List Signup', a subheading 'Join our email list', and a description: 'Get news about the band such as tour dates and special MP3 releases sent to your own in-box.' Below this, there are two numbered input fields: '1. Name:' and '2. Email:'. A 'Submit' button is located at the bottom left of the form area. The status bar at the bottom of the browser window shows the word 'Done'.

**Mailing List Signup**

Join our email list

Get news about the band such as tour dates and special MP3 releases sent to your own in-box.

1. Name:

2. Email:

Done



# The action attribute

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

```
<form action="/cgi-bin/maillinglist.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!
```

# Labels

- is used to associate descriptive text with its respective form field. This provides important context for users with speech-based browsers.
- Each **label** element is associated with exactly one form control.
- There are two ways to use it.
  - 1) **implicit association**, nests the control and its description within a **label** element:
- `<label>Male: <input type="radio" name="gender" value="M" /></label>`
- `<label>Female: <input type="radio" name="gender" value="F" /></label>`

## Cont'd . . . Labels

2) **explicit association**, matches the label with the control's **id** reference, The **for** attribute tells which control the label is for.

- useful when the control is not directly next to its descriptive text in the source
- and when we use CSS to align them

```
<label for="form-login-username">Login account:</label>
```

```
<input type="text" name="login" id="form-login-username" />
```

```
<label for="form-login-password">Password:</label>
```

```
<input type="password" name="password" id="form-login-password"/>
```

# 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>**

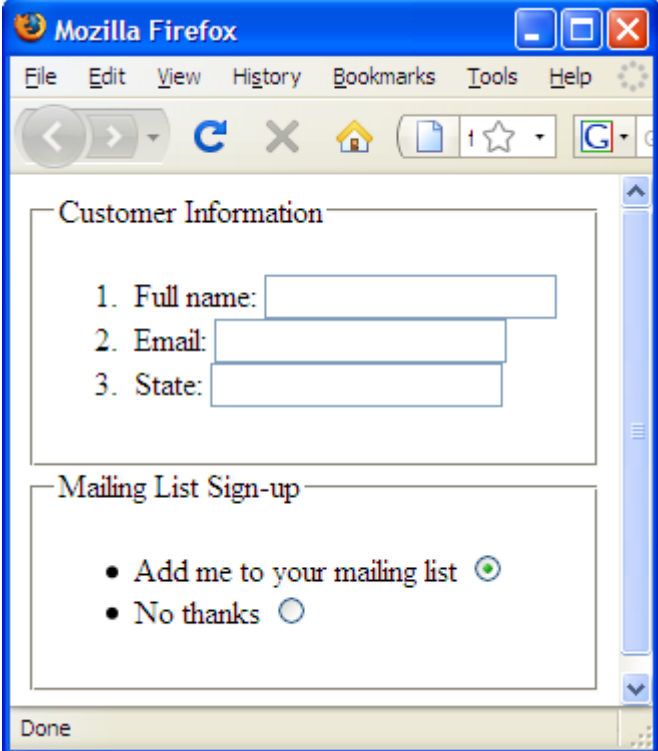
**<ul>**

**<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>**



The screenshot shows a Mozilla Firefox browser window with a web form. The form contains two fieldsets. The first fieldset is titled "Customer Information" and contains three text input fields labeled "1. Full name:", "2. Email:", and "3. State:". The second fieldset is titled "Mailing List Sign-up" and contains two radio button options: "Add me to your mailing list" (which is selected) and "No thanks". The browser's address bar shows a Google search page.

# Text entry controls

- There are three basic types:
  - single-line text fields,
  - password entry fields, and
  - multiline text entry fields.

# Single-line text field

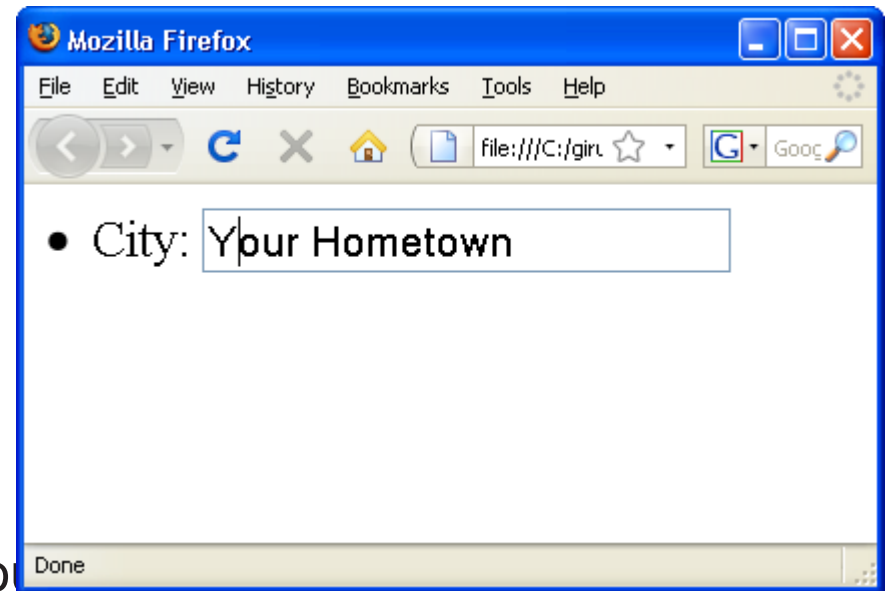
- used for entering a single word or line of text

syntax : `<input type="text" />`

Example:

```
<label for="form-city">City:</label>
```

```
<input type="text" name="city" value="Your Hometown"
maxlength="50" id="form-city" />
```



The **value** attribute specifies default text that appears in the field when the form is loaded. When you reset a form, it returns to this value.

## Cont'd . . . Single-line text field

### Size

- By default, browsers display a text-entry box that is 20 characters wide, but you can change the number of characters using the size attribute. eg. `size="25"`

### maxlength

- By default, users can type an unlimited number of characters in a text field regardless of its size.
- You can set a maximum character limit using the **maxlength** attribute. Eg `maxlength="50"`



# Password text entry field

- A password field works just like a text entry field, except the characters are obscured from view using asterisk (\*) or bullet (•) characters,

**Syntax:** `<input type="password" />`

**Example:**

```
<label for="form-pswd">Log in:</label> <input type="password"
 name="pswd" size="8" maxlength="8" id="form-pswd" />
```



# Multiline text entry field

- Syntax: `<textarea>...</textarea>`

- Example:

```
<label for="form-entry">

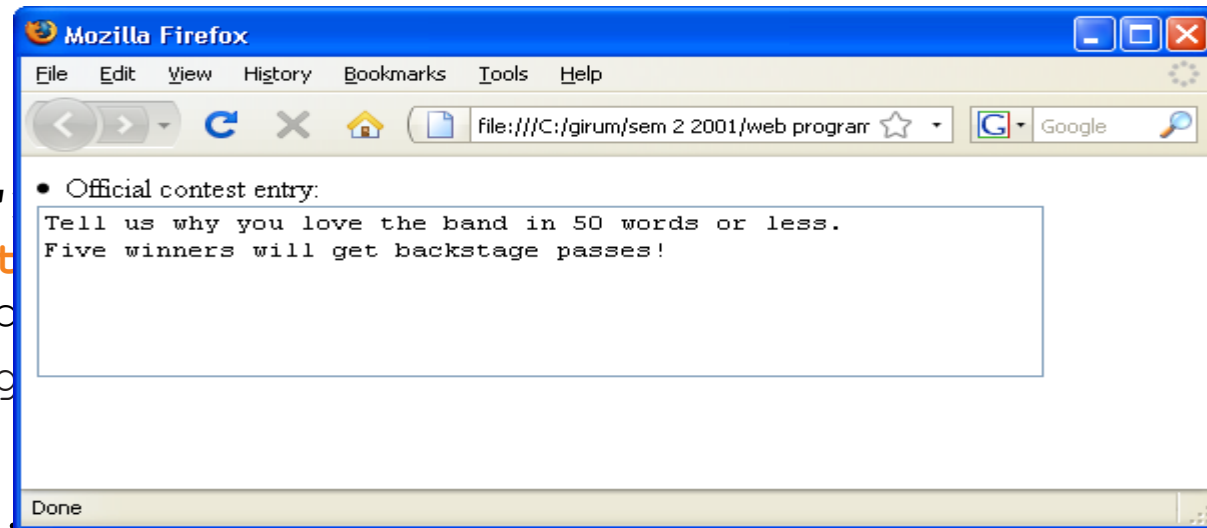
<textarea name="contest
entry">Tell us why you love
winners will get backstage
```

rows

- Specifies the number of lines of text the area should display. Scrollbars will be provided if the user types more text than fits in the allotted space.

cols

- Specifies the width of the text area measured in number of characters



# 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

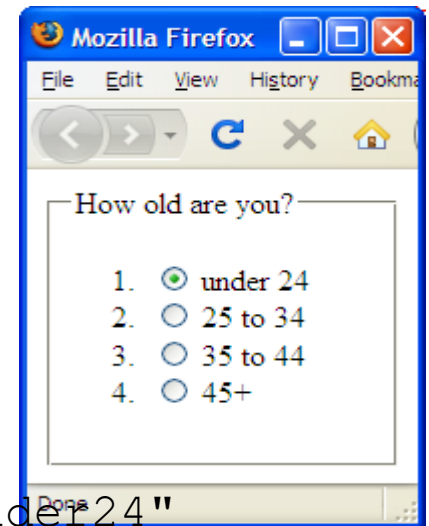
Syntax: `<input type="radio" name="variable" />`

Example:

```
<fieldset>
<legend>How old are you?</legend>

<label><input type="radio" name="age" value="under24"
checked="checked" /> under 24</label>
<label><input type="radio" name="age" value="25-34" /> 25 to
34 </label>
<label><input type="radio" name="age" value="35-44" /> 35 to
44 </label>
<label><input type="radio" name="age" value="over45" /> 45+
</label>

</fieldset>
```



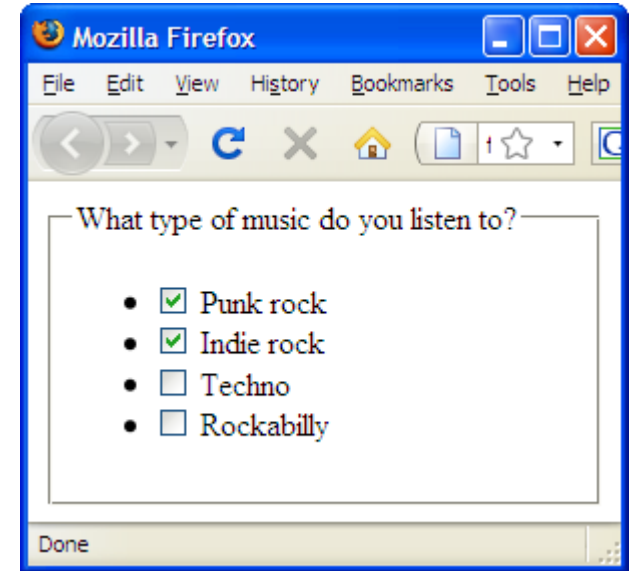
## Cont'd ... Radio and checkbox buttons

- 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>

<label><input type="checkbox" name="genre" value="punk"
checked="checked" /> Punk rock</label>
<label><input type="checkbox" name="genre" value="indie"
checked="checked" /> Indie rock</label>
<label><input type="checkbox" name="genre" value="techno" />
Techno </label>
<label><input type="checkbox" name="genre" value="rockabilly"
/> Rockabilly</label>

</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

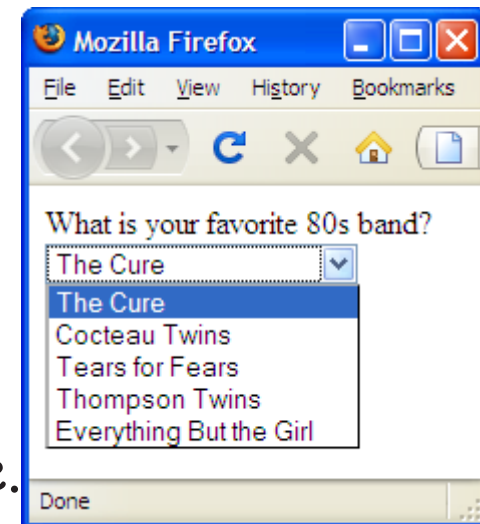
- only one item may be selected.
- **select** element displays as a pull-down menu by default when
  - no size is specified or
  - if the **size** attribute is set to 1

## Example:

```
<label for="form-fave">What is your favorite 80s band?</label>

<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>
```

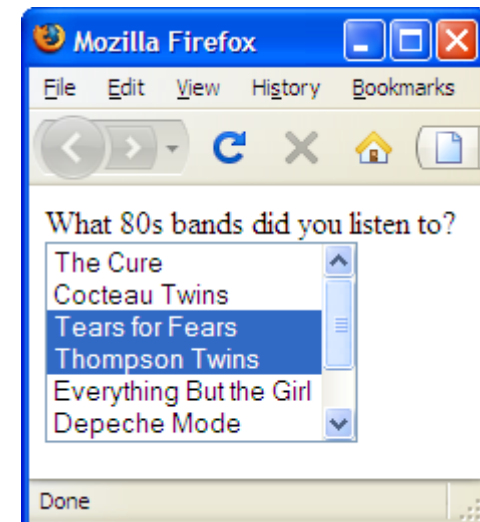
- Use the **value** attribute to provide an overriding value.



# 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>
```

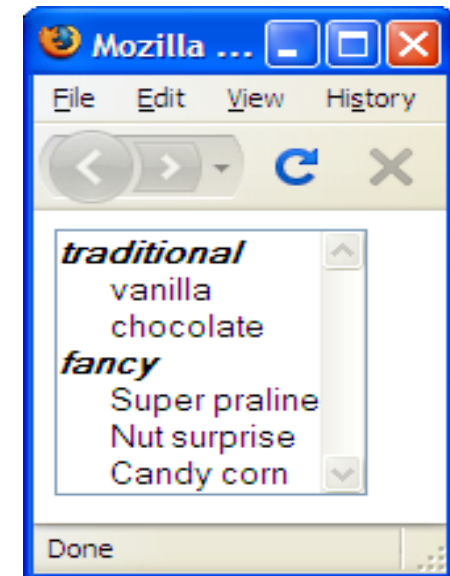




# Grouping menu options

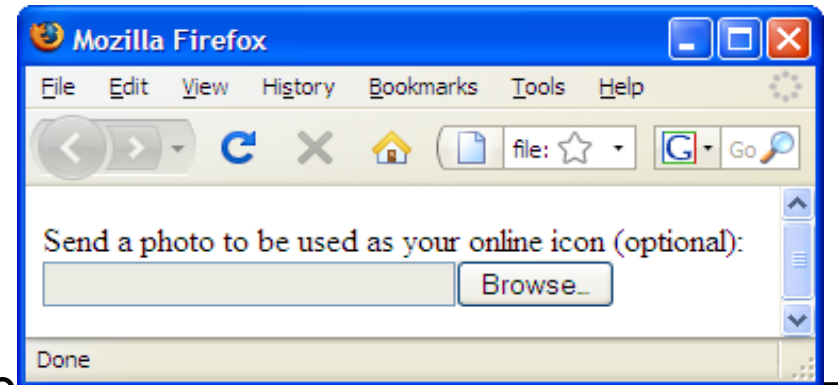
- You can use the **optgroup** element to create conceptual groups of options.
- Example :

```
<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>

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

