

# **CSE479**

## **Web Programming**

**Nishat Tasnim Niloy**

Lecturer

Department of Computer Science and Engineering

Faculty of Science and Engineering

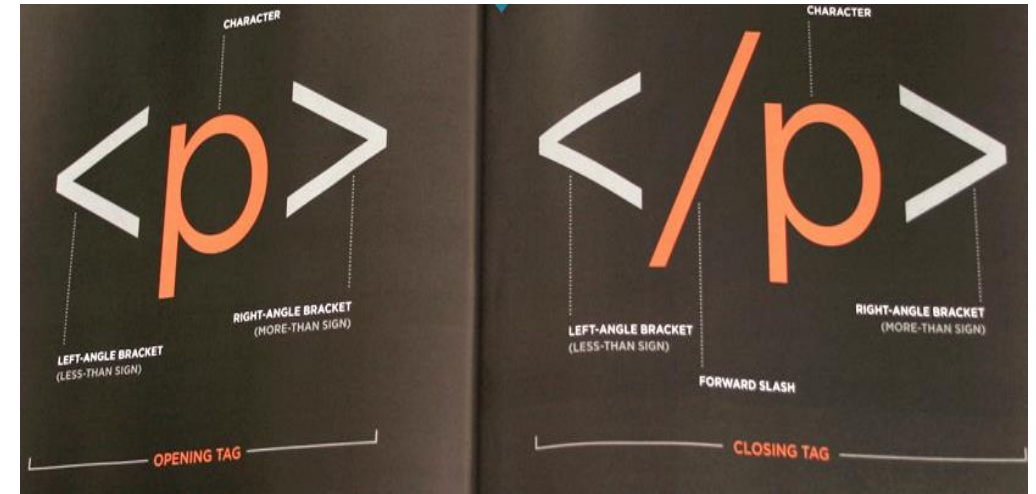
# Topic 1

## Introduction to HTML

(Structure, Text, Lists, Links, images, and extra markup)

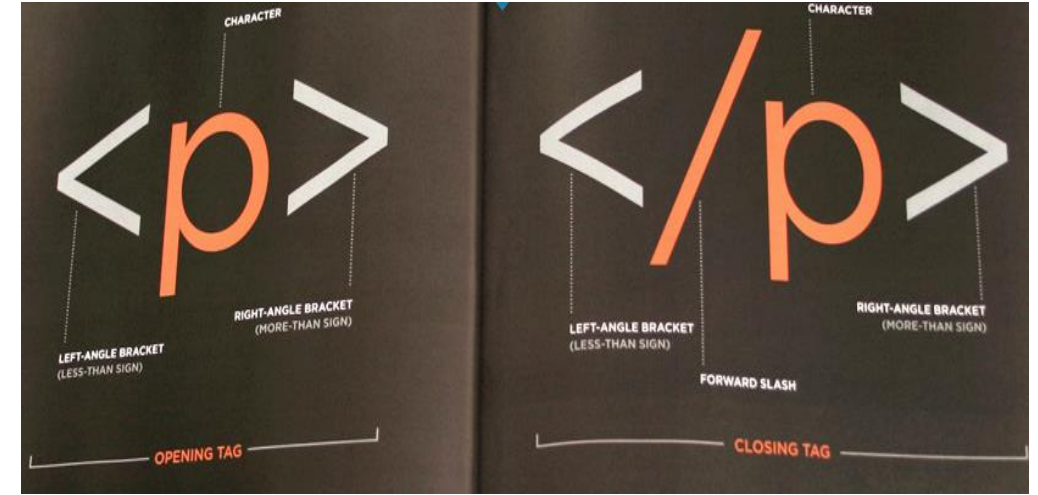
# By the end of this unit you should be able to...

- ❑ Explain why HTML is a markup language
- ❑ Explain how HTML provides structure to a Web document
- ❑ Differentiate between elements and tags
- ❑ Describe the components of opening tags
- ❑ Differentiate between empty and nonempty elements
- ❑ Build a simple web page that uses
  - ❑ Structured tags
  - ❑ Text elements including paragraphs and headings
  - ❑ List elements



# By the end of this unit you should be able to...

- ❑ Provide comments in HTML code
- ❑ Differentiate between block and inline elements
- ❑ Use Chrome DevTools to view page source
- ❑ Use div and span elements to group elements and text
- ❑ Use HTML entities appropriately
- ❑ Use meta elements to provide information about a Web page
- ❑ Create and use links within and between HTML documents
- ❑ Use image elements appropriately in a Web document



# HTML describes the structure of pages

## **DO NOW:**

- ☐ Point your browser to your favorite website or blog post.
- ☐ With your partner, describe some of the structural elements that you see.

**How is this structure provided?**

# HTML is a markup language

HTML pages are text documents

When creating a page, you add tags (aka markup) to the content of the page

These tags are used to provide extra meaning to the page:

- Describe elements
- Allow browsers to show users appropriate structure

# HTML uses elements to describe structure to Web pages

Let's look closer at the code from the last page.  
There are several different elements. Each element has an opening tag and a closing tag.

Tags act like containers. They tell you something about the information that lies between their opening and closing tags.

## CODE

## DESCRIPTION

<code>&lt;html&gt;</code>	The opening <code>&lt;html&gt;</code> tag indicates that anything between it and a closing <code>&lt;/html&gt;</code> tag is HTML code.
<code>&lt;body&gt;</code>	The <code>&lt;body&gt;</code> tag indicates that anything between it and the closing <code>&lt;/body&gt;</code> tag should be shown inside the main browser window.
<code>&lt;h1&gt;This is the Main Heading&lt;/h1&gt;</code>	Words between <code>&lt;h1&gt;</code> and <code>&lt;/h1&gt;</code> are a main heading.
<code>&lt;p&gt;This text might be an introduction to the rest of the page. And if the page is a long one it might be split up into several sub-headings.&lt;/p&gt;</code>	A paragraph of text appears between these <code>&lt;p&gt;</code> and <code>&lt;/p&gt;</code> tags.
<code>&lt;h2&gt;This is a Sub-Heading&lt;/h2&gt;</code>	Words between <code>&lt;h2&gt;</code> and <code>&lt;/h2&gt;</code> form a sub-heading.
<code>&lt;p&gt;Many long articles have sub-headings to help you follow the structure of what is being written. There may even be sub-sub-headings (or lower-level headings).&lt;/p&gt;</code>	Here is another paragraph between opening <code>&lt;p&gt;</code> and closing <code>&lt;/p&gt;</code> tags.
<code>&lt;h2&gt;Another Sub-Heading&lt;/h2&gt;</code>	Another sub-heading inside <code>&lt;h2&gt;</code> and <code>&lt;/h2&gt;</code> tags.
<code>&lt;p&gt;Here you can see another sub-heading.&lt;/p&gt;</code>	Another paragraph inside <code>&lt;p&gt;</code> and <code>&lt;/p&gt;</code> tags.
<code>&lt;/body&gt;</code>	The closing <code>&lt;/body&gt;</code> tag indicates the end of what should appear in the main browser window.
<code>&lt;/html&gt;</code>	The closing <code>&lt;/html&gt;</code> tag indicates that it is the end of the HTML code.

# HTML uses elements to describe structure to Web pages

Attributes provide additional information about the contents of an element. They appear on the opening tag of the element and are made up of two parts: a **name** and a **value**, separated by an equals sign.

HTML5 allows you to use uppercase attribute names and omit the quotemarks, but this is not recommended.

Diagram illustrating the structure of an HTML attribute:

ATTRIBUTE NAME

ATTRIBUTE VALUE

`<p lang="en-us">Paragraph in English</p>`

Diagram illustrating the structure of an HTML attribute:

ATTRIBUTE NAME

ATTRIBUTE VALUE

`<p lang="fr">Paragraphe en Français</p>`

The attribute name indicates what kind of extra information you are supplying about the element's content. It should be written in lowercase.

The value is the information or setting for the attribute. It should be placed in double quotes. Different attributes can have different values.

Here an attribute called `lang` is used to indicate the language used in this element. The value of this attribute on this page specifies it is in US English.

The majority of attributes can only be used on certain elements, although a few attributes (such as `lang`) can appear on any element.

Most attribute values are either pre-defined or follow a stipulated format. We will look at the permitted values as we introduce each new attribute.

The value of the `lang` attribute is an abbreviated way of specifying which language is used inside the element that all browsers understand.



# Two types of markup

## Structural markup:

- ❑ Elements that describe structure of a page
  - ❑ e.g. headings, paragraphs

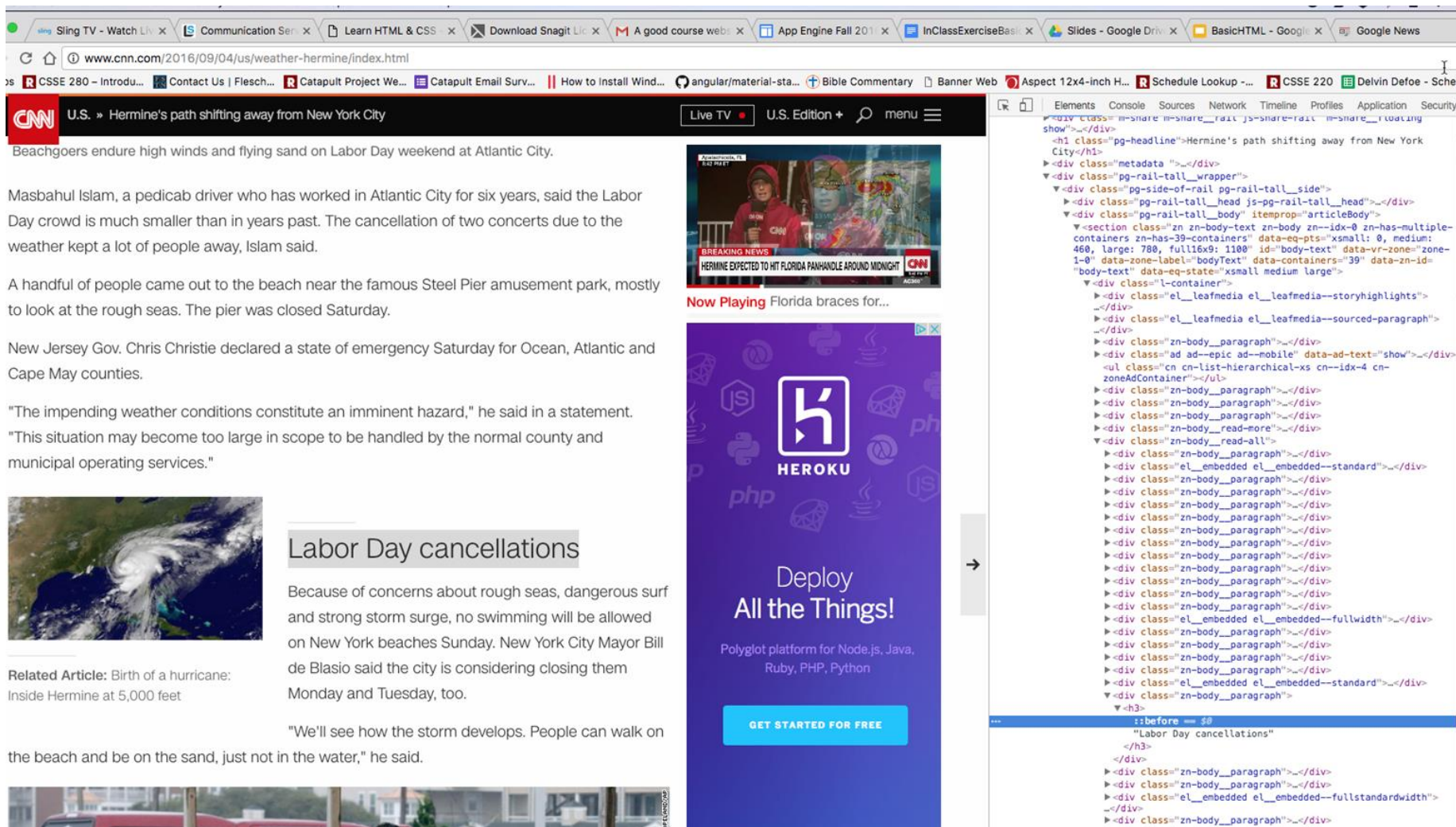
## Semantic markup:

- ❑ Elements that provide extra information, e.g.
  - ❑ Where **emphasis** is *placed* in a sentence
  - ❑ Something you have written is a quotation
  - ❑ Who said what
  - ❑ The meaning of an acronym

# Body, head, and title: structural markup

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>This is the Title of the Page</title>
5      </head>
6      <body>
7          <h1>This is the Body of the Page</h1>
8          <p> Anything within the body of a web page is
9              displayed in the main browser window.</p>
10     </body>
11 </html>
12
```

# HTML has 6 levels of heading



# Headings: separate major areas of a page

Note the placement of the heading inside the body element

Headings are **block elements**:

*Always start on a new line and take up the full width available*

Other examples:

[http://www.w3schools.com/tags/tryit.asp?filename=tryhtml\\_headers](http://www.w3schools.com/tags/tryit.asp?filename=tryhtml_headers)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Introduction to HTML </title>
  </head>

  <body>
    <p>This is      a paragraph of text </p>
    <h1>University of Smart People</h1>
    <h2>Department of Computer Science</h2>
    <h3>Sponsored by Big Rich Corporation</h3>
    <h6>We teach the best stuff here!</h6>
  </body>
</html>
```

# Paragraph: <p> describes a paragraph of text

To create a paragraph, surround the words that make up the paragraph with the opening tag <p> and the closing tag </p>.

Paragraphs are **block elements**.

*(Again: they start on a new line and take up the full width available)*

```
<!DOCTYPE html>
<html>
  <head>
    <title>Introduction to HTML </title>
  </head>

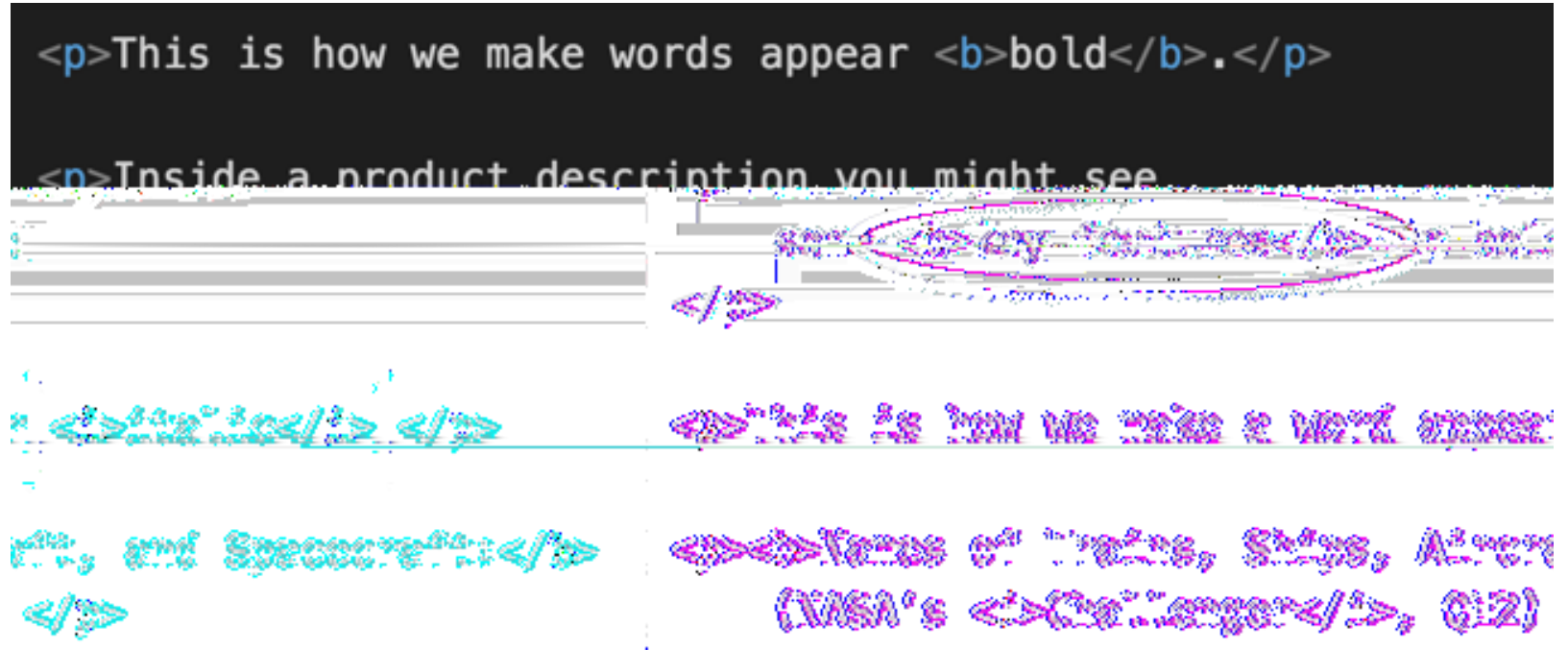
  <body>
    <p>This is      a paragraph of text </p>
  </body>
</html>
```

# Bold and italics

Bold elements are presented in a visual way - no additional meaning.

Better to use the **<strong>** semantic element

Instead of the italic element, use the *<em>* semantic element



**Semantic elements are better for screen readers**

# Superscript and subscript

**<sup>** elements contain characters that should be Superscript.

E.g.: suffixes of dates or mathematical concepts like raising a number to a power

**<sub>** elements contain text that should be subscript.

E.g.: In chemical formulas and footnotes

```
<p>Classes began on the 1<sup>st</sup> of September.</p>  
<p>The chemical formular for baking soda or sodium  
bicarbonate is NaHCO<sub>3</sub></p>
```

# White space collapsing

In order to make code easier to read, Web page authors often add extra space or start some elements on new lines.

Browser only displays 1 space when it encounters  $\geq 2$  spaces in a row.

Good for indentation and for making code easier to read.

```
<p>
    Notice that the layout    of the
    input text    makes    little difference in a
    paragraph.                The browser collapses multiple

|
spaces and treats line breaks as spaces.
</p>
```



# Line breaks and horizontal rule: empty elements

- ❑ Browser automatically shows each new paragraph, heading, or other block element on a new line
- ❑ If you wish to add a line break inside a paragraph, you can use the `<br />` tag.
- ❑ To create a break between themes you can add a horizontal rule between sections using the `<hr />` tag. E.g.,
  - ❑ Change of topic in a book
  - ❑ New scenes in a play
- ❑ These elements have no words between the opening and closing tags
- ❑ They are called empty elements

# Semantic markup

Comprises elements that are not intended to affect the structure of your Web pages.

BUT they all extra information to the pages.

These are known as semantic markup.

**Examples:** `<em>`, `<strong>`, `<blockquote>`

Should not use them to change how your text look. Their purpose is to more accurately describe the content of your page.

**Reasons:** so that other programs, such as screen readers and search engines, can use this extra information

# Strong and emphasis

`<strong>` element indicates that its content has strong importance.

Browsers show the content of strong element in **bold**.

`<em>` element indicates emphasis that subtly changes the sentence's meaning.

Browsers show the content of `<em>` element in *italic*

```
<p>  
    <strong>Beware:</strong> Pickpockets operate  
    in this area.  
</p>  
<p>This toy has many small pieces and is  
    <strong>not suitable for children under five  
    years old.</strong>  
</p>
```

```
<p>I <em>think</em> Ivy was the first.</p>  
<p>I think <em>Ivy</em> was the first.</p>  
<p>I think Ivy was the <em>first</em>.</p>
```

# Quotation and citation elements

**<blockquote>**: used for long quotes that take up an entire paragraph. NOTE: **<p> element is nested INSIDE** <blockquote> element.

Browsers tend to indent the content of the <blockquote> element.  
Don't use solely for indentation.

**<q>**: used for shorter quotations that sit within a <p> element.

Browsers are supposed to put quotes around the <q> element, but older browsers (IE) do not.

Both elements may use the **cite** attribute, with a URL value, to indicate where the quotation is from.

## Quotation and citation elements (2)

Examples for quotation and citation elements:

[http://www.w3schools.com/html/html\\_quotation\\_elements.asp](http://www.w3schools.com/html/html_quotation_elements.asp)

**<abbr>**: If you use an abbreviation or acronym, this element can be used. A title attribute on the opening is used to specify the full term

**<cite>**: When referencing a piece of work such as a book, film, etc., the <cite> element can be used to indicate where the citation is from.

Browsers render the content of the <cite> element in *italics*.

**<dfn>**: Used to indicate the definition instance of a new term.

## Author details -- very specific meaning

**<address>**: contains author details **for the author of the page**.

It can contain a physical address, but does not have to.

Browsers typically display the content of the <address> element in *italics*.

Might use hCard microformat to add physical address information to your markup - <https://en.wikipedia.org/wiki/HCard>

# Changes to content

**<ins>**: can be used to show content that has been inserted into a document

**<del>**: can be used to show content that has been deleted from a document

Browsers typically underline the content of <ins> elements and draw a ~~line through~~  
~~the content~~ of <del> elements

**<s>**: indicates something that is no longer accurate or relevant, but should not be deleted

Visually, the content of <s> elements are displayed with a ~~line through the center~~

# Lists, lists, lists, and more lists ...

There are lots of occasions when we need to use lists.

HTML provides us with different types of lists:

- ❑ Unordered lists
- ❑ Ordered lists
- ❑ Definition lists

Lists are block elements

[http://www.w3schools.com/html/html\\_lists.asp](http://www.w3schools.com/html/html_lists.asp)



# Ordered lists

Created with the `<ol>` element.

Each list item is placed between an opening `<li>` tag and a closing `</li>` tag.

Browsers indent lists by default.

Sometimes you may see a *type* attribute in the `<ol>` element that specifies the type of numbering; e.g.:

- ❑ Numbers
- ❑ Letters,
- ❑ Roman numeral

It is better to use the CSS *list-style-type* property instead. (we will learn how later)

```
<ol>
  <li>Chop potatoes into quarters</li>
  <li>Simmer in salted water for 15-20
    minutes until tender</li>
  <li>Heat milk, butter and nutmeg</li>
  <li>Drain potatoes and mash</li>
  <li>Mix in the milk mixture</li>
</ol>
```

1. Chop potatoes into quarters
2. Simmer in salted water for 15-20 minutes until tender
3. Heat milk, butter and nutmeg
4. Drain potatoes and mash
5. Mix in the milk mixture

# Unordered lists

Created with the `<ul>` element.

Each list item is placed between an opening `<li>` and a closing `</li>` tag.

Browsers indent lists by default.

Sometimes you may see a *type* attribute in the `<ul>` element that specifies the type of bullet point; e.g.:

- ❑ Circles
- ❑ Squares,
- ❑ Diamonds

It is better to use the CSS *list-style-type* property instead.

```
<ul>
  <li>1kg King Edward potatoes</li>
  <li>100ml milk</li>
  <li>50g salted butter</li>
  <li>Freshly grated nutmeg</li>
  <li>Salt and pepper to taste</li>
</ul>
```

- 1kg King Edward potatoes
- 100ml milk
- 50g salted butter
- Freshly grated nutmeg
- Salt and pepper to taste

# Definition lists

Created with the **<dl>** element. Usually  
Consists of a series of terms and their  
definitions.

Inside a **<dl>** element you will usually see  
a pair of **<dt>** and **<dd>** elements.

**<dt>**: contains the definition term

**<dd>**: contains the definition

You might see a list of 2 terms with the same  
definition or 2 different definitions for the  
same term

```
<dl>
  <dt>Sashimi</dt>
  <dd>Sliced raw fish that is served with condiments such
      as shredded daikon radish or ginger root, wasabi and
      soy sauce</dd>
  <dt>Scale</dt>
  <dd>A device used to accurately measure the weight of
      ingredients</dd>
  <dd>A technique by which the scales are removed from the
      skin of a fish</dd>
  <dt>Scamorze</dt>
  <dt>Scamorzo</dt>
  <dd>An Italian cheese usually made from whole cow's milk
      (although it was traditionally made from buffalo milk)</dd>
</dl>
```

Sashimi  
Sliced raw fish that is served with condiments such as shredded  
daikon radish or ginger root, wasabi and soy sauce

Scale  
A device used to accurately measure the weight of ingredients  
A technique by which the scales are removed from the skin of a fish

Scamorze

Scamorzo  
An Italian cheese usually made from whole cow's milk (although it  
was traditionally made from buffalo milk)

# Nested lists

A list can contain other lists.

Notice the proper nesting of tags.

Indentation should make it clearer

- Simpsons:
  - Homer
  - Marge
- Family Guy:
  - Peter
  - Lois

```
<ul>
  <li>Simpsons:
    <ul>
      <li>Homer</li>
      <li>Marge</li>
    </ul>
  </li>
  <li>Family Guy:
    <ul>
      <li>Peter</li>
      <li>Lois</li>
    </ul>
  </li>
</ul>
```

# Comments in HTML

**<!-- Comments go here  
and can span multiple lines -->**

Used to add comments to code. Comments will not be visible in the main browser window

Good to add comments because:

- ☐ Helps you remember what's going when you visit code later
- ☐ Makes it easier for someone else to understand your code
- ☐ Can be used to indicate where long sections start and end
- ☐ Can be used around blocks of code you want to stop displaying in browser

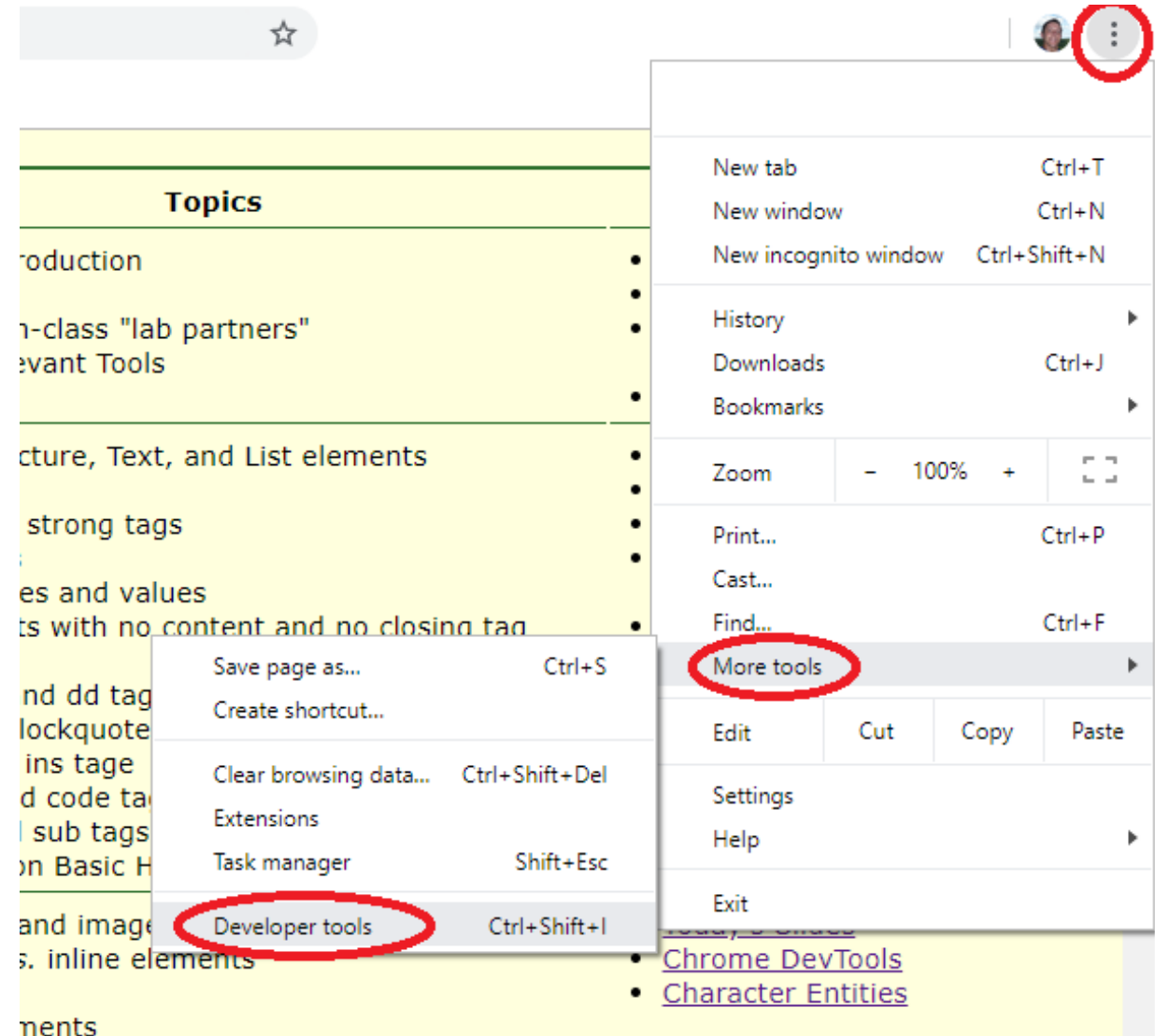
# Chrome Developer tools

Chrome Developer tools allow developers to explore, debug, and get insight for code optimization.

See <https://developer.chrome.com/devtools> to learn how to use it

Use it to view page source

Keyboard shortcut: Control-Shift-i



# Global attributes id and class

These are global attributes because they can be used on any element.

## The id attribute

- ❑ Uniquely identifies the containing element from others on page
- ❑ Value starts with a letter or underscore
- ❑ Uniqueness allows you to style it differently using CSS
- ❑ When defining style declared with a #

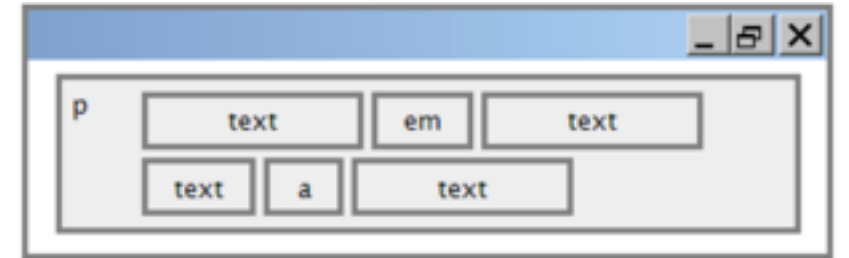
## The class attribute

- ❑ Provides a way to identify and style several elements
- ❑ Can share same value as class attribute on another element
- ❑ Can separate class names by space for an element
- ❑ When defining style declared with a .

# Review: Block elements vs inline elements

## Block elements:

- ❑ Some element always appear to start on a new line
  - ❑ E.g., <h2>, <p>, <ul>, <li>



## Inline elements:

- ❑ Some elements always appear to continue on the same line as their neighboring elements
  - ❑ E.g., <a>, <img>, <b>, <em>, <strong>, <i>



# <div>: Grouping text and elements in a block

Group a set of elements together in one block-level box.

Using an id or class attribute on a div element:

- ❑ Can create CSS style rules for the <div> element
- ❑ Can determine how much space the element occupies
- ❑ Can change the appearance of all the elements contained within it

Using a <div> can make it easier to follow your code, especially if you add comments after the closing tag.

# <span>: grouping text and elements inline

Acts like an inline equivalent of the <div> element. [See example here](#)

Uses:

- ❑ To contain a section of text where there is no other suitable element to differentiate from surrounding text
- ❑ To contain a number of inline elements

Most common reason for use:

- ❑ To control the appearance of the content of the enclosed elements using CSS

Use with **class** or **id** attribute:

- ❑ To explain the meaning of the span
- ❑ To style with CSS

# <meta>: Information about your pages

Lives inside the <head> element and contains all kinds of information about your page.

See Meta example [here](#).

Not visible to users, BUT:

- ❑ Fulfills a number of purposes
- ❑ E.g.:
  - ❑ Tells search engines about your Web page
  - ❑ Who created it
  - ❑ Whether your page is time sensitive (when will it expire?)

Is an empty element:

- ❑ Has no closing tag
- ❑ Uses attributes to carry information
- ❑ **name** and **content** attributes used together

## <meta>: Information about your pages (2)

```
<meta charset="utf-8" />
<meta charset="utf-8" http-equiv="refresh" content="5" />  <!-- Thanks, Dax! -->
<meta name="description"
      content="Authors' web site for Building Java Programs." />
<meta name="keywords" content="java, textbook" />
```

HTML

- ❑ Some <meta> elements use the **http-equiv** attribute instead of **name**
- ❑ <meta> elements with the **charset** attribute indicates language/character encodings
- ❑ **charset** is important because not setting **utf-8** may result in HTML files with **utf-8** content being incorrectly processed as ASCII

# Character Entities (a.k.a. Escape characters)

Some characters are used in and reserved by HTML code.

A way of representing Unicode characters within a page

Complete list of HTML entities ⇒  
[http://www.w3schools.com/charsets/ref\\_html\\_symbols.asp](http://www.w3schools.com/charsets/ref_html_symbols.asp)

character(s)	entity
< >	&lt; &gt;
é è ñ	&eacute; &egrave; &ntilde;
™ ©	&trade; &copy;
π δ Δ	&pi; &delta; &Delta;
И	&#1048;
" &	&quot; &amp;

# Links

The defining feature of the Web:

- ❑ They allow you to move from one Web page to another.
- ❑ They enable the very idea of browsing and surfing

Types of links you will commonly come across:

- ❑ From one website to another
- ❑ From one page to another on the same website
- ❑ From one part of a page to another part of the same website
- ❑ That open in a new browser window
- ❑ That start up your email program and address a new email to someone



## Links created using the <a> element



Link text should explain where user will be taken if they click on it

# Absolute URLs: Linking to other sites

When linking to a **different website** the value of the **href** attribute should be the **full Web address or Uniform Resource Locator** for the site.

If no page is specified in Web address, the site will display the homepage.

```
<p>Movie Reviews:
  <ul>
    <li><a href="http://www.empireonline.com">Empire</a></li>
    <li><a href="http://www.metacritic.com">Metacritic</a></li>
    <li><a href="http://www.rottentomatoes.com">Rotten Tomatoes</a></li>
    <li><a href="http://www.variety.com">Variety</a></li>
  </ul>
</p>
```



# Relative URLs: Linking to other parts of the same site

You do not have to specify the domain name in the URL, just the **relative path**.

If all the files are in the same folder, just the filename is needed.

If you have different pages of a site in different folders, then you need to specify the file path in relation the current page -- see pages **81 - 84 of HTML & CSS book** for details.

Relative URLs help when building a site on your computer:

- ❑ You can create links between pages without having to setup a domain name (*e.g., structure of this course*)
- ❑ You don't need to host your site someplace else

## mailto: Email links

```
<a href="mailto:yoder1@rose-hulman.edu">Send email to Jason</a>
```

Use `<a>` element to startup the user's email program and address an email to a specified email address.

Note that the value of the `href` attribute starts with `mailto:` and is followed by the email address that you want the email sent to.

# Opening links in a new window

```
<p>  
  <a href="http://www.rose-hulman.edu" target="_blank">  
    Rose-Hulman Institute of Technology</a> (opens in new window)  
</p>
```

Use the **target** attribute on the opening <a> tag.

The value of this attribute should be **\_blank**.

Ideal when link points to another website.

Avoid if remaining on same website. It is advised that you inform user that the link will open in another window.

# Linking to a specific part of a page

On a long page you can add links to prevent the user from having to scroll a lot to get to a certain section of the page.

Before you can link to a specific part of a page, you need to identify where in the page the link will go to. use the **id** attribute to do so.

Use **<a>** element to link to an element that uses an **id** attribute

- ❑ On same page, the value of **href** attribute should start with **#** and followed by the **value of the id** attribute
  - ❑ E.g., **<a href="#top">Top</a>**
- ❑ On a different page, the URL to the file should precede the **#** symbol

# Images

Lots of reasons why you might want to add an image to a Web page.

You might want to include

- ☐ A logo
- ☐ An illustration
- ☐ A photograph
- ☐ A chart

Things to consider:

- ☐ Can add an image using HTML
- ☐ Should pick the right format
- ☐ Should show an image at the right size

# Choosing images for your site

Images should ...

- ☐ Be relevant
- ☐ Convey information
- ☐ Convey the right mood
- ☐ Be instantly recognizable
- ☐ Fit the color palette

**It is a good idea to create an images folder for all the images the site uses.**

As the site grows, you might consider adding subfolders to your images folder (e.g., logos, products, photos, etc.)

# Adding images to your page

```

```

Use an **<img>** element - an empty element

Must have **src** and **alt** attributes

- ❑ **src** tells the browser where to find the image file
- ❑ **alt** provides text description of the image for those who cannot see it
- ❑ **title** displayed when mouse placed over the image

Images take longer to load than HTML code.

Use size attributes to allow browser to render the rest of the page and leave the right amount of space for the image

# HTML5 <figure> and <figcaption> elements

These are new elements

**<figure>** contains images and their captions so that the two are associated.

You can have more than one image in a figure so long as they share the same caption.

Browsers may indent the figure element.

**<figcaption>** allows authors to add a caption to an image. There was no way to do that before HTML5



# Images as Buttons

- One of the most important aspects of images is how they can be combined with the **a** element to create buttons.
- To make an image "press able," simply enclose it within an anchor.
- Example

```
<a href="http://www.univdhaka.edu"></a>
```

# Tables in HTML

- The HTML table model allows authors to arrange data -- text, preformatted text, images, links, forms, form fields, other tables, etc. -- into rows and columns of cells.
- Each table may have an associated caption that provides a short description of the table's purpose.
- A table is divided into rows, and each row is divided into cells. Table cells may either contain "header" information (TH element) or "data" (TD element).

# Common HTML table elements and attributes

HTML Table Elements	Description
<code>&lt;TABLE&gt;...&lt;/TABLE&gt;</code>	The Table delimiter.
<code>&lt;TR ...&gt;...&lt;/TR&gt;</code>	Used to specify number of rows in a table.
<code>&lt;TD ...&gt;...&lt;/TD&gt;</code>	Specifies table data cells
<code>&lt;TH ...&gt;...&lt;/TH&gt;</code>	Table Header cell
<code>&lt;CAPTION ...&gt;...&lt;/CAPTION&gt;</code>	Specifies the table Caption.
<code>&lt;THEAD&gt;...&lt;/THEAD&gt;</code>	Specifies the Table head.
<code>&lt;TBODY&gt;...&lt;/TBODY&gt;</code>	Specifies the Table body.
<code>&lt;TFOOT&gt;...&lt;/TFOOT&gt;</code>	Specifies the Table footer.

# <TABLE>...</TABLE>

<TABLE>...</TABLE> Attributes	Description
BACKGROUND	Specifies a background image for the table. If used in the <TABLE> element, the image in question will be tiled behind all of the table cells.
BGCOLOR="#rrggbb color name"	Specifies the background color of the table to be specified, using either the specified color names, or a rrggbb hex triplet.
ALIGN="left right"	It allows a table to be aligned to the left or right of the page, allowing text to flow around the table.
BORDER	Control and set the borders to be displayed for the table
BORDERCOLOR="#rrggbb color name"	sets the border color of the table, using either the specified color names, or a rrggbb hex triplet. It is necessary for the BORDER attribute to be present in the main <TABLE> element for border coloring to work.
CELLPADDING=value	Specifies the amount of white space between the borders of the table cell and the actual cell data.

# <TR ...>...</TR>

- This stands for table row.

<b>&lt;TR&gt;...&lt;/TR&gt;</b> Attributes	Description
<b>BGCOLOR</b> ="#rrggbb color name"	Specifies the background color of the table row to be specified, using either the specified color names, or a rrggbb hex triplet.
<b>ALIGN</b> ="left center right"	Specifies whether text inside the table cell(s) is aligned to the left, right or center of the cell.
<b>VALIGN</b> ="top middle bottom baseline"	controls whether text inside the table cell(s) is aligned to the top, bottom, or vertically centered within the cell.
<b>BORDERCOLOR</b> ="#rrggbb color name"	sets the border color of the table row, using either the specified color names, or a rrggbb hex triplet. It is necessary for the BORDER attribute to be present in the main <TABLE> element for border coloring to work.

`<TD ...>...</TD>`

- This stands for table data, and specifies a standard table data cell.
- It must appear only within table rows.
- Each row need not have the same number of cells specified as short rows will be padded with blank cells on the right.
- Internet Explorer will allow the use of `<TD></TD>` to specify a blank cell, that will be rendered with a border (providing a border has been set). Other browsers will require some character within a data cell for it to be rendered with a border.

# <TD ...>...</TD>

<b>&lt;TD&gt;...&lt;/TD&gt; Attributes</b>	<b>Description</b>
BACKGROUND	Internet Explorer supports the placing of images inside the <TD> element. If used in the <TD> element, the image in question will be tiled behind the particular data cell.
BGCOLOR="#rrggbb color name"	Specifies the background color of the data cell to be specified, using either the specified color names, or a rrggbb hex triplet.
ALIGN="left center right"	Specifies text alignment inside the table cell(s).
VALIGN="top middle bottom baseline"	controls whether text inside the table cell(s) is aligned to the top, bottom, or vertically centered within the cell.
BORDERCOLOR="#rrggbb color name"	sets the border color of the data cell. Necessary for the BORDER attribute to be present in the main <TABLE> element for border coloring to work.
ROWSPAN="value"	This attribute can appear in any table cell (<TH> or <TD>) and specifies how many rows of the table this cell should span. The default ROWSPAN for any cell is 1.
COLSPAN="value"	This attribute can appear in any table cell (<TH> or <TD>) and it specifies how many columns of the table this cell should span. The default COLSPAN for any cell is 1.

<TH ...>...</TH>

- This stands for table header.
- Header cells are identical to data cells in all respects, with the exception that header cells are in a **bold font, and have a center default ALIGN.**
- <TH ...>...</TH> has the same attribute as the <TD ...>...</TD> element.



<CAPTION ...>...</CAPTION>

- This represents the caption for a table.
- <CAPTION> elements should appear inside the <TABLE> but not inside table rows or cells.
- The caption accepts an alignment attribute that defaults to ALIGN=top but can be explicitly set to ALIGN=bottom.
- Like table cells, any document body HTML can appear in a caption.
- Captions are, by default horizontally centered with respect to the table, and they may have their lines broken to fit within the width of the table.

# <CAPTION ...>...</CAPTION>

- The <CAPTION> element can accept the following attributes:
  - **ALIGN="top|bottom"**: Specifies whether the caption appears above or below the table, using the top and bottom values, defaulting to top.

- Example

```
<table border="1">
  <caption align="bottom">My savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>
```

# Parts of <TABLE>

- <THEAD>...</THEAD>
  - This element, which is Internet Explorer specific, is used to specify the head section of the table.
  - It does directly affect the rendering of the table on the screen, but is required if you want RULES to be set in the <TABLE> .
- <TBODY>...</TBODY>
  - This element, which is Internet Explorer specific, is used to specify the body section of the table.
  - It does directly affect the rendering of the table on the screen, but is required if you want RULES to be set in the <TABLE> .
- <TFOOT>...</TFOOT>
  - This element, which is Internet Explorer specific, is used to specify the footer section of the table. It does directly affect the rendering of the table on the screen, but is required if you want RULES to be set in the <TABLE>.

# Cellpadding and Cellspacing

- The space between cells in a table is controlled by the **cellspacing** attribute for **<table>**.
- The value is measured in pixels or percentage values.
- When using tables for layout, cells should jut up next to each other, so this attribute is often set to **0**.
- It is possible to give space between cells by setting this attribute to a positive integer or percentage value.
- The padding between cell walls and the content they surround is controlled by the **cellpadding** attribute, which is also often set to **0** in tables used for layout.

# Cellpadding and Cellspacing Example

```
<table border="1" cellspacing="15" cellpadding="40">
```

```
  <tr>
```

```
    <td>Element 1</td>
```

```
    <td>Element 2</td>
```

```
    <td>Element 3</td>
```

```
  </tr>
```

```
</table>
```

```
<br /><br />
```

```
<table border="1" cellspacing="0" cellpadding="0">
```

```
  <tr>
```

```
    <td>Element 1</td>
```

```
    <td>Element 2</td>
```

```
    <td>Element 3</td>
```

```
  </tr>
```

```
</table>
```

# Lists

- HTML supports several mechanisms for specifying lists of information.
- In HTML there are 3 different types of lists, all of which may be nested.
- If used they should be present in the <BODY> of an HTML document.
- Common HTML list elements are:
  - <UL> : unordered list; *bullets*
  - <OL> : ordered list; *numbers*
  - <DL> : definition list; *dictionary*

# <OL>...</OL>

- The Ordered List element is used to present a numbered list of items, sorted by sequence or order of importance and is typically rendered as a numbered list.
- An ordered list must begin with the <OL> element. Place the <LI> (list item) tag between your opening <OL> and closing </OL> tags to create list items.
- Example:

```
<h4 align="center">Goals</h4>
```

```
<ol>
```

```
  <li>Find a Job</li>
```

```
  <li>Get Money</li>
```

```
  <li>Move Out</li>
```

```
</ol>
```

## <OL>...</OL> Cont..

- The OL element has **TYPE attribute** which allows authors to specify whether the list items should be marked with.
- It has following values:
  - <OL TYPE=a> Small letters. For example a, b, c ...
  - < OL TYPE =A> Capital letters. For example A, B, C ...
  - < OL TYPE =i> Small roman numerals. For example i, ii, iii ...
  - < OL TYPE =I> Large roman numerals. For example I, II, III ...
  - < OL TYPE =1> The default numbers. For example 1, 2, 3 ...



## <OL>...</OL> Cont..

- The OL element has **START attribute** which allows authors to start at values other than 1.
- Thus START=5 would display either an 'E', 'e', 'V', 'v', or '5' based on the TYPE attribute.

- Example

```
<h4 align="center">Goals</h4>
```

```
<ol start="4" >
```

```
  <li>Buy Food</li>
```

```
  <li>Enroll in College</li>
```

```
  <li>Get a Degree</li>
```

```
</ol>
```

`<UL>...</UL>`

- The Unordered List element is used to present a list of items which is typically separated by white space and/or marked by bullets.
- The bullet itself comes in three flavors: squares, discs, and circles.
- The default bullet displayed by most web browsers is the traditional full disc.
- An unordered list must begin with the `<UL>` element, which is immediately followed by a `<LI>` (list item) element.

## <UL>...</UL> Cont..

- The **TYPE attribute** can be used in the <UL> element to specify bullet type
  - <UL TYPE= disc>
  - < UL TYPE = circle>
  - < UL TYPE = square >
- Example:  
Try it by yourself.....

## < DL>...</DL>

- Definition lists are typically rendered by browsers, with the definition term <DT> flush left in the display window with the definition data <DD> rendered in a separate paragraph, indented after the definition term.
  - <DL> - defines the start of the list
  - <DT> - definition term
  - <DD> - defining definition
- Single occurrences of a <DT> element without a subsequent <DD> element are allowed and have the same significance as if the <DD> element had been present with no text.
- The opening list element must be <DL> and must be immediately followed by the first term (<DT>).

## < DL>...</DL> Cont..

Code	Result
<pre>&lt;dl&gt;   &lt;dt&gt;&lt;b&gt;Fromage&lt;/b&gt;&lt;/dt&gt;   &lt;dd&gt;French word for cheese.&lt;/dd&gt;   &lt;dt&gt;&lt;b&gt;Voiture&lt;/b&gt;&lt;/dt&gt;   &lt;dd&gt;French word for car.&lt;/dd&gt; &lt;/dl&gt;</pre>	<p><b>Fromage</b> French word for cheese.</p> <p><b>Voiture</b> French word for car.</p>

# Try IT...

- Example 1

- Coffee
- Tea
  - Black tea
  - Green tea
- Milk

- Example 2

- Coffee
- Tea
  - Black tea
  - Green tea
    - China
    - Africa
- Milk

# HTML Iframes

- An iframe is used to display a web page within a web page.
- `<iframe src="URL"></iframe>`
- The URL points to the location of the separate page.
- The **height** and **width** attributes are used to specify the height and width of the iframe.

`<iframe src="http://www.google.com" width="200" height="200"></iframe>`

- The frameborder attribute specifies whether or not to display a border around the iframe. Set the attribute value to "0" to remove the border

# Forms

- HTML Forms are used to select different kinds of user input and to pass data to a server.
- A form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more.
- A form can also contain select lists, textarea, fieldset, legend, and label elements.
- A form will take input from the viewer and depending on your needs; you may store that data into a file, place an order, gather user statistics, register the person to your web forum, or maybe subscribe them to your weekly newsletter.



# Forms Cont...

- The following elements are used to create forms:
  - `<FORM>...</FORM>` : A form within a document
  - `<input ...>...</input>` : One input field
  - `<OPTION>` : One option within a Select element
  - `<SELECT>...<SELECT>` : A selection from a finite set of options
  - `<TEXTAREA ...>...</TEXTAREA>` : A multi line input field

# <FORM>...</FORM>

- The <FORM> element is used to delimit a data input form.
- There can be several forms in a single document, but the <FORM> element **cannot be nested**.

*<FORM ACTION="\_URL\_" METHOD="GET|POST" EncTYPE="MIME type">*

- **Form attributes**
- **The ACTION** attribute is a URL specifying the location to which the contents of the form data fields are submitted to elicit a response. The way data is submitted varies with the access protocol of the URL to which the form data is sent and with the values of the METHOD and EncTYPE attributes.
- **The METHOD** attribute specifies a method of accessing the URL specified in the ACTION attribute. Generally, the method will be either GET or POST.
  - The **GET method** is ideal for form submission where the use of the form data does not require external processing.
  - The **POST method** should be used where the form is used to provide information for example, that updates a database.
- **The EncTYPE** attribute specifies the media type used to encode the form data. The default EncTYPE is the MIME type application/x-www-form-urlencoded.

# <input>

- The <input> element represents a field whose contents may be edited or activated by the user.
- Attributes of the <input> element are:
  - **The NAME** attribute assigns a name to the given field so that author may reference it later. The NAME attribute is required for most input types and is normally used to provide a unique identifier for a field, or for a logically related group of fields.
  - **The CHECKED** attribute can be used with a TYPE= CHECKBOX or TYPE=RADIO setting, this indicates that the checkbox or radio button is selected.
  - **The MAXLENGTH** attribute is used with TYPE=TEXT setting, this indicates the maximum number of characters that can be entered into a text field. This can be greater than specified by the SIZE attribute, in which case the field will scroll appropriately. The default number of characters is unlimited.

# Attributes of the <input> element

- Attributes of the <input> element are:
  - **The SIZE** attribute specifies the size or precision of the field according to its type. For example, to specify a field with a visible width of 24 characters:  
`<input type="text" SIZE="24" />`
  - **The TYPE** attribute determines what kind of input field it will be. Defaults value is free text. Several types of fields can be defined with the type attribute:
    - TEXT
    - PASSWORD
    - RADIO
    - CHECKBOX
    - BUTTON
    - HIDDEN
    - RESET

# TYPE =TEXT

- `<input type="text" />` defines a one-line input field that a user can enter text into:

```
                <form>
    First name: <input type="text" name="firstname" /><br />
    Last name: <input type="text" name="lastname" />
    </form>
```

- The TEXT field should be used in conjunction with the SIZE and MAXLENGTH attributes to set the maximum amount of text that can be entered.
- Explicit VALUE and NAME attributes are also required.

# TYPE = PASSWORD

- `<input type="password" />` defines a password field. PASSWORD is the same as the TEXT attribute, except that text is not displayed as it is entered.

`<form>`

    Password: `<input type="password" name="pwd" />`

`</form>`

# TYPE = RADIO

- `<input type="radio" />` defines a RADIO button.
- RADIO button is used for attributes that accept a single value from a set of alternatives.
- Only the selected radio button in the group generates a name/value pair in the submitted data. Radio buttons require an explicit VALUE and NAME attribute.
- VALUE - specifies what will be sent if the user chooses this radio button. Only one value will be sent for a given group of radio buttons (see *name* for more information).
- NAME - defines which set of radio buttons that it is a part of.  
`<input type="radio" name="gender" value="male" /> Male<br />`  
`<input type="radio" name="gender" value="female" /> Female`
- Checked is an optional attribute and can be used to specify which options are selected for initial form display.

# TYPE = CHECKBOX

- `<input type="checkbox" />` defines a checkbox.
- Checkboxes let a user select **ONE or MORE** options of a limited number of choices.
- The check box's NAME and VALUE attributes behave the same as a radio button. Also CHECKED is an optional attribute.

```
<input type="checkbox" name="vehicle1" value="Bike" /> I have  
a  
bike<br />  
<input type="checkbox" name="vehicle2" value="Car" /> I have  
a car
```



# TYPE = BUTTON

- This can be used to embed buttons directly into HTML documents that add functionality when used in conjunction with Script.
- The NAME attribute is used to give the button a unique name, which can be used to set its function in the script.
- The VALUE attribute specifies the text that is displayed on the button in the document

# TYPE = SUBMIT

- `<input type="submit" />` defines a SUBMIT button.
- A SUBMIT button is used to send form data to a server.
- The data is sent to the page specified in the form's action attribute.
- The file defined in the action attribute usually does something with the received input.
- The NAME attribute is used to give the button a unique name.
- The VALUE attribute specifies the text that is displayed on the button in the document.

# TYPE = HIDDEN

- With this input type, no field is presented to the user, but the content of the field is sent with the submitted form.
- This value may be used to transmit state information about client/server interaction.

```
<input type="hidden" name="HiddenField" value="100" />
```

# TYPE = RESET

- RESET is a button that when pressed resets the form's fields to their specified initial values.
- The label to be displayed on the button may be specified just as for the SUBMIT button.

# HTML Input Types Example

[http://www.w3schools.com/html/html\\_form\\_input\\_types.asp](http://www.w3schools.com/html/html_form_input_types.asp)

# <OPTION>

- The <OPTION> element can only occur within a <SELECT> element.
- <SELECT> is the list itself and each <OPTION> is an available choice for the user.
- <OPTION> can take following attributes:
  - ***The SELECTED** attribute indicates that this option is initially selected.*
  - ***The VALUE** attribute indicates the value to be returned if this option is chosen. The returned value defaults to the contents of the <OPTION> element.*
- The contents of the <OPTION> element are presented to the user to represent the option. It is used as a returned value if the VALUE attribute is not present.

# <SELECT ...>...</SELECT>

- The <SELECT> element allows the user to choose one of a set of alternatives described by textual labels.
- Every alternative is represented by the <OPTION> element.
- Attributes used with the <SELECT> are listed in the following sections.
  - **The MULTIPLE** attribute is needed when users are allowed to make several selections, for example <SELECT MULTIPLE>.
  - **The NAME** attribute specifies the name that will be submitted as a name/value pair.
  - **The SIZE** attribute specifies the number of visible items. If this is greater than one, then the resulting form control will be a list.
- The SELECT element is typically rendered as a **pull down** or **pop-up list**.

# <SELECT ...>...</SELECT>

```
<form method="post" action="mailto:youremail@email.com">
```

```
  Education?
```

```
    <select name="degree">
```

```
      <option>Choose One</option>
```

```
      <option>Some High School</option>
```

```
      <option>High School Degree</option>
```

```
      <option>Some College</option>
```

```
      <option>Bachelor's Degree</option>
```

```
      <option>Doctorate</option>
```

```
    </select>
```

```
    <input type="submit" value="Email Yourself">
```

```
</form>
```



# <TEXTAREA>...</TEXTAREA>

- The <TEXTAREA> element lets users enter more than one line of text.
- Forums and the like use text areas to post what you type onto their site using scripts. For this form, the text area is used as a way to write comments to somebody.
- When submitting a form, lines in a TEXTAREA should be **terminated using CR/LF**.
- **ROWS** and **COLUMNS** need to be specified as attributes to the <TEXTAREA> tag.
- **ROWS** are roughly 12pixels high, the same as in word programs.
- The value of the **COLUMNS** reflects how many characters wide the text area will be.

# <TEXTAREA>...</TEXTAREA>

- Another attribute to be aware of is the **WRAP**. Wrap has 3 values.
- **WRAP=**
  - OFF : The default setting. Wrapping doesn't happen. One ongoing line. Lines are sent exactly as typed.
  - VIRTUAL : The display word-wraps, but long lines are sent as one line without new-lines. That is, the viewer will see the words wrapping as they type their comments, but when the page is submitted to you, the web host, the document sent will not have wrapping words.
  - PHYSICAL : The display word-wraps, and the text is transmitted at all wrap points. The text will appear both to you, the web host, and the viewer including any page breaks and additional spaces that may be inputted. The words come as they are.

# HTML Multimedia

- Multimedia on the web is sound, music, videos, and animations.
- Modern web browsers have support for many multimedia formats.
- **HTML Helpers (Plug-ins)**
  - A helper application is a small computer program that extends the standard functionality of the browser.
  - Helper applications are also called plug-ins.
  - Plug-ins are often used by browsers to play audio and video.
  - Examples of well-known plug-ins are Adobe Flash Player and QuickTime.
  - Plug-ins can be added to Web pages through the `<object>` tag or the `<embed>` tag.

# HTML Audio/ Videos

- Sounds can be played in HTML by many different methods.
  - **Using The <embed> Element**
    - The <embed> tag defines a container for external (non-HTML) content.
    - <embed height="50" width="100" src="abc.mp3">
  - **Using the <object> Element**
    - The <object> tag can also define a container for external (non-HTML) content
    - <object height="50" width="100" data="abc.mp3"></object>
- **<object>** tag is for Internet Explorer, while the **<embed>** tag is for Netscape and related to it browsers using Netscape plugin to display a flash movie

# <embed> element attributes

- **align** - Determines how to align the object. It takes either *center*, *left* or *right*.
- **autostart** - Indicates if the media should start automatically. Netscape default is true, Internet Explorer is false.
- **loop** - Specifies if the sound should be played continuously (set loop to true), a certain number of times (a positive value) or not at all (false). This is supported by Netscape only.
- **playcount** - Specifies the number of times to play the sound. This is alternat option for *loop* if you are usiong IE.
- **hidden** - Defines if the object shows on the page. A false value means no and true means yes.
- **height** - Height of the object in pixels or en.
- **width** - Width of the object in pixels or en.
- **pluginspage** - Specifies the URL to get the plugin software.
- **name** - A name used to reference the object.
- **src** - URL of the object to be embedded. This can be any recognizable by the user's browser. It could be .mid, .wav, .mp3, .avi and so on).
- **volume** - Controls volume of the sound. Can be from 0 (off) to 100 (full volume). This attribute is supported by Netscape only.