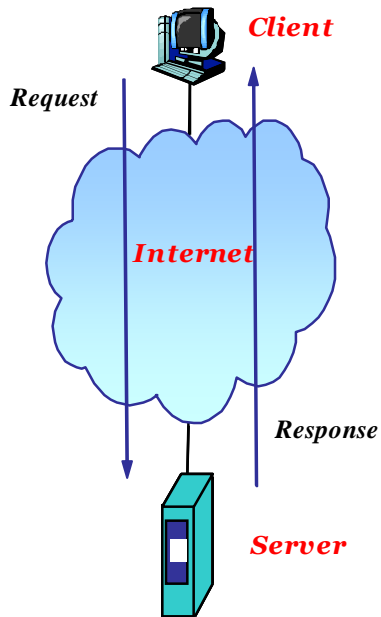


ICS 571: CLIENT-SERVER PROGRAMMING



Fall Semester 2016 (2016-1)

Overview of HTML

Dr. Nasir Al-Darwish

Computer Science Department

King Fahd University of Petroleum and Minerals

darwish@kfupm.edu.sa

Outline

- Basic HTML document structure and tag format
- HTML elements
- HTML Doctype declaration
- Text formatting
- Hyperlinks
- Lists, Images
- Block versus inline elements
- HTML Character Entities
- XHTML versus HTML
- HTML Colors
- HTML Tables
- Tags within the <head> element
- Styling with CSS

What is HTML?

- HTML is a language for composing web pages
 - ❑ HTML stands for *Hyper Text Markup Language*
 - ❑ HTML is a *markup language*; a markup language defines a set of **markup tags**
 - ❑ An *HTML document (page, file)* is a **plain text** file with HTML tags and other text; the tags define document content and structure
 - ❑ Tags can be nested which results in a **tree structure** with parent-child relationship
 - ❑ HTML documents are also called **web pages**; they are normally stored at **web servers**, fetched and displayed by **web browsers**.
 - ❑ HTML files (web pages) use the extension “.htm” or “.html”

HTML Tags

- HTML defines a set of markup tags (simply known as *HTML Tags*)
 - HTML tags are keywords (tag names) surrounded by **angle brackets** like `<body>`
 - HTML tags normally **come in pairs** like

`<body> ... </body>`

`<p> ... </p>`

- The general format is:

`<tagname attr1=value1 attr2=value2 >content</tagname>`

- The first tag in a pair is the **start tag**, the second tag is the **end tag**
- Start and end tags are also called **opening tags** and **closing tags**
- The end tag is same as the start tag, with a **forward slash** before the tag name
- **Note:** HTML tags are not case sensitive; however, by convention they are normally written in lower case letters.

HTML Elements

- An **HTML element** is the whole text from the start tag to the end tag, including the tags themselves.
- Example of an HTML Element, the `<p>` (*paragraph*) element:

```
<p>This is a paragraph.</p>
```
- The **element content** is everything between the start and the end tag.
- **Note:** A web browser only displays (renders) the content part of an HTML element.
- Most HTML elements have **attributes**; these appear inside the start tag.

Empty HTML Elements

- Some HTML elements (e.g., ``, `
`) have **empty content** and therefore do not use a closing tag.
- The `
` tag (specifies a line break) is an empty element without a closing tag
- **Tip:** In XHTML, all elements must be closed. Adding a slash inside the start tag, like `
`, is the proper way of closing empty elements in XHTML (and XML).

Nesting of HTML Elements

- Most HTML elements can be nested (can contain other HTML elements).
- **Proper Nesting Rule: *An inner element must be closed before closing the outer (enclosing) element.***

□ *Example of invalid nesting (it should be ` </i>`):*

`<p> <i> Look at this </i> </p>`

- An HTML document consists of nested HTML elements.
- *Example:*

```
<html>
<body>
<p>This is my first paragraph.</p>
</body>
</html>
```

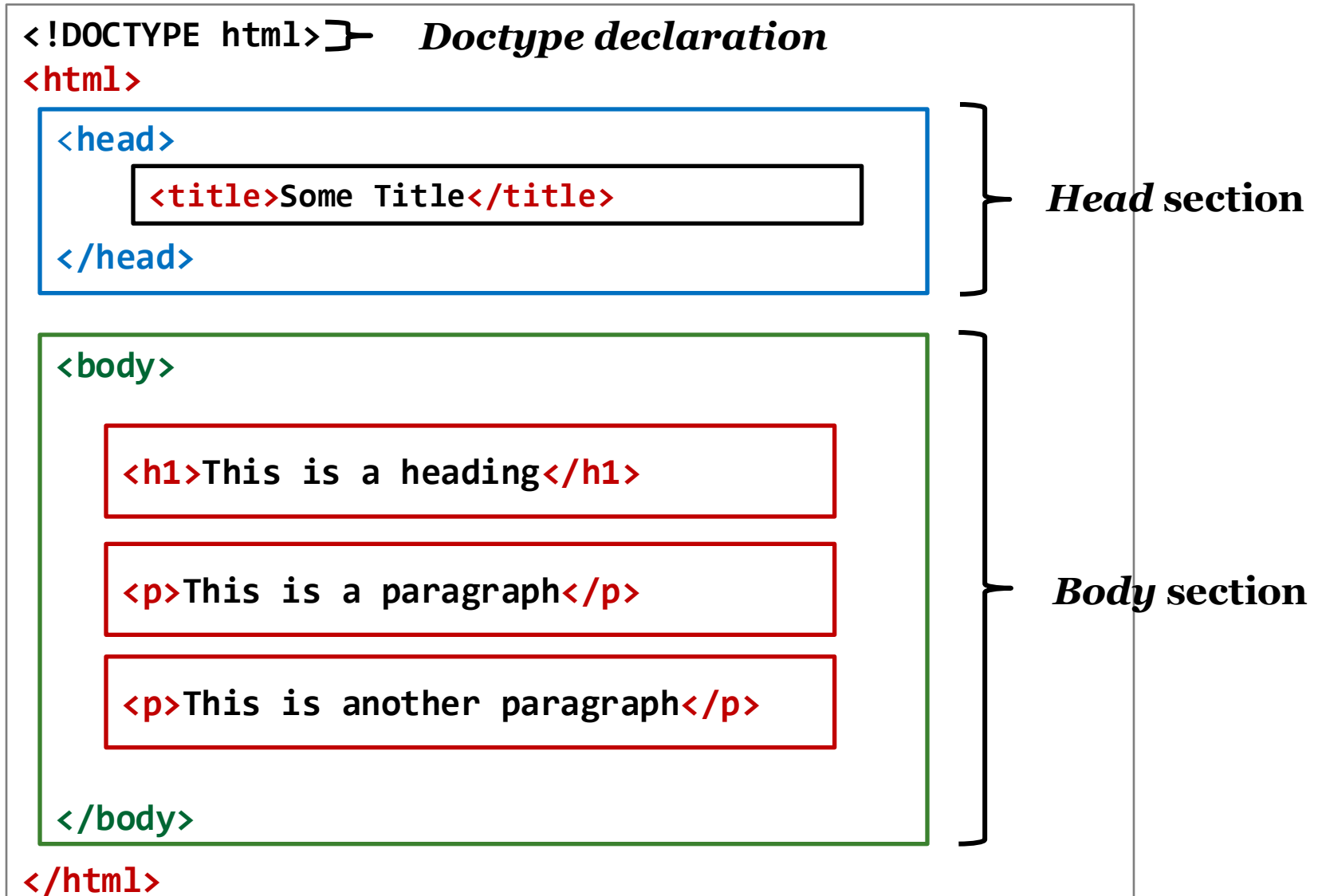
HTML Attributes

- HTML elements can have **attributes**
- Attributes provide additional information about an element
- Attributes are always specified in **the start tag**
- Attributes are specified as ***name=value*** pairs like:
attribute_name="value"
- Attribute names and attribute values are case-insensitive.
- Attribute values should always be enclosed in quotes using either double quotes or single quotes.

HTML Attributes (cont.)

- Examples of attributes common to many HTML elements (tags):
 - ❑ *Class*: Specifies one or more class names for an element (refers to a class in a style sheet)
 - ❑ *Id*: Specifies a unique id for an element
 - ❑ *Style*: Specifies an inline CSS style for an element
 - ❑ *Title*: Specifies extra information about an element (displayed as a tool tip)

HTML Page Structure



HTML Page Structure Explained

- `<!DOCTYPE html>` tells the browser that the document is an HTML version 5 document.
- The first tag in an HTML document is `<html>`. The corresponding end tag is `</html>`, which marks the end of the document.
- The text between the `<head>` (head) tag and the `</head>` (end head) tag is header information. Header information is not displayed in the browser window.
- The text between the `<title>` and `</title>` tags is the title of the document. *The title is displayed as the **caption** of the browser's window.*
- The content inside the `<body>` element will be displayed in your browser.

The `<!DOCTYPE>` Declaration

- The *DOCTYPE* (document type definition, *dtd*) declaration helps the browser to display a web page correctly.
- The *DOCTYPE* declaration tells the HTML version used by the html document and it must appear at the start of the document.
- Previous versions of HTML defined many doctypes, and choosing the right one could be tricky. Prior to HTML5, we normally use **XHTML Strict 1.0** (or **XHTML Transitional 1.0**) doctype:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- In HTML5, we only need to use the following doctype declaration:

```
<!DOCTYPE html>
```

Strict vs. Transitional HTML Markup

- There are two DOCTYPEs versions for XHTML 1.0 (also HTML 4.01) defined as the **Transitional** and **Strict**.
(http://www.w3schools.com/tags/ref_html_dtd.asp)
- The **Strict** versions of XHTML 1.0 and HTML 4.01 excludes many of HTML *presentation* elements and attributes that are more appropriately handled using CSS. The following is not allowed:
 - ❑ HTML Elements: `<center>`, ``, `<frameset>`, `<iframe>`
 - ❑ HTML Attributes: `bgcolor`, `background`, `align` & `border` (except for `<table>` and child elements of `<table>`)
- A **Transitional** DOCTYPE is more relaxing and meant to be used when you have a lot of legacy markup that cannot easily be converted to comply with a Strict DOCTYPE.
- Web developers should use the Strict DTD when possible, but may use the Transitional DTD when support for presentation elements or attributes is required.

HTML Versions

Version	Year
HTML	1991
HTML+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML 1.0	2000
HTML 5	2012
XHTML 5	2013

HTML5 template

```
<!DOCTYPE html>
<html lang="en"> <!-- document language -->
<head>
  <title>Some Title</title>
  <meta charset="utf-8"><!-- document character set -->
</head>
<body>
  ...
</body>
</html>
```

Uniform Resource Locator (URL)

- A URL is a fancy phrase for a document's address on the world-wide web.

- Uses the format:

`scheme://host:port/path/filename`

- The value for the scheme can be *ftp*, *http*, *https*, .. etc.
- A URL identifies a host on the Internet where the document is stored. A host can be specified using
 - ❑ The host's domain name, such as "w3schools.com",
 - ❑ Or the host's an Internet Protocol (IP) address: 192.68.20.50.

HTML Hyperlinks (Links)

- The HTML anchor element (i.e., the **<a>** tag) defines a hyperlink. It uses the form:

```
<a href="url" >Link text or <img> element</a>
```

- A hyperlink (or link for short) is *a word* (or *a phrase*), or *an image* that you can click on to jump to another web page (or another location within same page).
- The **href** attribute indicates the link's destination. The value for the href attribute is a URL (absolute or relative).
- When you move the cursor over a link in a web page, the cursor changes from an arrow into a little finger. Also, the status bar (of the browser's window) will reveal the destination of the link.

HTML Links: the ID Attribute

- The *ID* attribute can be used to create bookmarks (to aid navigation in a long web page); For this, we can assign an anchor element or any other HTML element an ID.
- Bookmarks are not displayed in any special way. They are invisible to the reader.
- *Example:*

An anchor with an id inside a web page:

```
<a id="tips" >Useful Tips Section</a>
```

Create a link to the "Useful Tips Section" inside the same page:

```
<a href="#tips" >Useful Tips</a>
```

Or, create a link inside the current page to the "Useful Tips Section" found in another page "HealthInfo.htm" :

```
<a href="http://www.w3schools.com/HealthInfo.htm#tips" >  
Useful Tips</a>
```

HTML Links: Target attribute, URL media types

- If the ***target*** attribute is set to "_blank", the target page will open in a new browser window, as in:

```
<a href="http://www.w3schools.com/" target="_blank" >Visit  
W3Schools!</a>
```

- The URL value for the href attribute in a link can reference content of varying media types as:

```
<a href="http://archive.org/download/fateha_403/fateha.mp3" >  
Surat Fateha</a>
```

HTML Headings

- HTML headings are defined with the `<h1>`, `<h2>`, ..., `<h6>` tags.
- `<h1>` is rendered as largest/boldest, `<h6>` is rendered as smallest.

- *Example:*

```
<h1>This is a level 1 heading</h1>
```

```
<h2>This is a level 2 heading</h2>
```

```
<h3>This is a level 3 heading</h3>
```

```
<h4>This is a level 4 heading</h4>
```

```
<h5>This is a level 5 heading</h5>
```

```
<h6>This is a level 6 heading</h6>
```

- Use HTML headings for headings only. Don't use headings to make text **BIG** or **bold**.
- Search engines use your headings to index the structure and content of your web pages.
- H1 headings should be used as main headings, followed by H2 headings, then the less important H3 headings, and so on.

HTML Paragraphs

- HTML documents are normally divided into paragraphs
- HTML paragraphs are defined using the `<p>` tag

- *Example:*

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

- By design, browsers automatically add an empty line before and after a paragraph.
- Browsers often display paragraphs correctly even if the closing tag is missing.

What happens if you omit a closing tag?

- A browser may not display the HTML correctly if you omit the closing tag.
- It is possible (but not guaranteed) that a browser will display the HTML correctly even if you omit the closing tag, as in the following example:

```
<p>This is a paragraph
```

```
<p>This is another paragraph
```

HTML Line Breaks

- Use the `
` tag if you want a line break (a new line) without starting a new paragraph:
- The `
` element is an empty HTML element and it does not use an end tag.
- *Example:*

HTML Source

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

<p>This is<br/>a para<br/>graph
with line breaks</p>

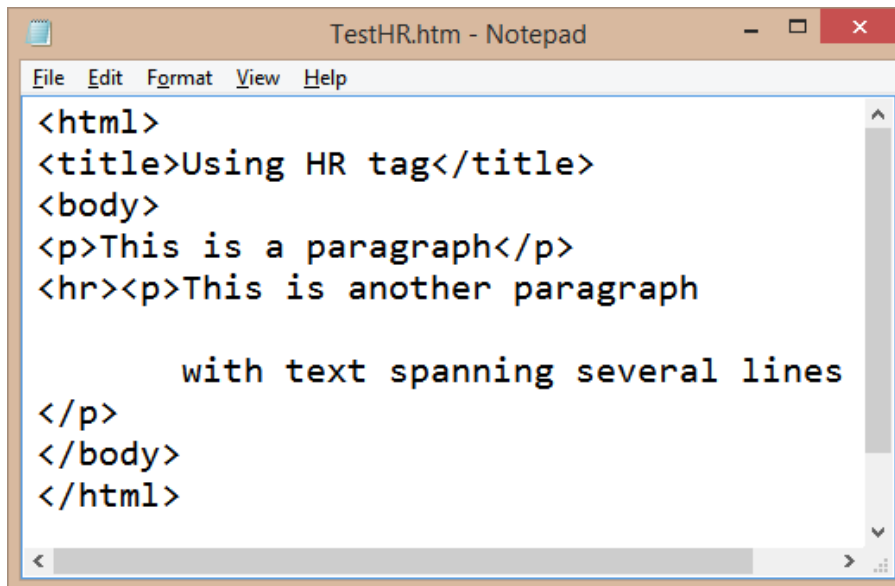
</body>
</html>
```

Output in Browser

This is
a para
graph with line breaks

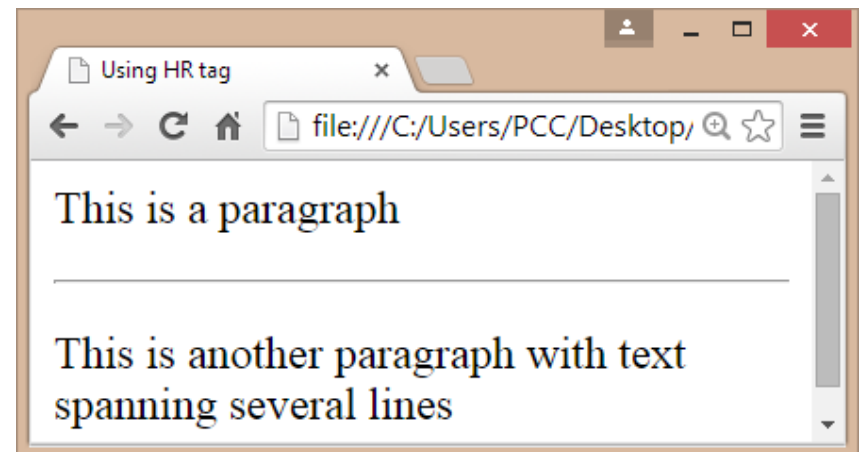
HTML Lines

- The **<hr>** (horizontal rule) tag renders a horizontal line. It is used to show a **visible line** as a way of separating content
- In the example, the image on the left shows the html source and the image on the right shows rendering by browser.
- Note how the explicit spaces and line breaks in html source are rendered. *They are ignored!*



```
<html>
<title>Using HR tag</title>
<body>
<p>This is a paragraph</p>
<hr><p>This is another paragraph

        with text spanning several lines
</p>
</body>
</html>
```



Non-breaking Space

- The browser automatically removes extra spaces and extra lines
 - If you write 10 spaces in your text, the browser will remove 9 of them
- Large or small screens, and resized windows will create different results
- To force the browser to keep spaces, you can do any of the following:
 - Use the * * character entity
 - Use the `<pre>... </pre>` (preformatted) tag; this honors spaces and line breaks

HTML Comments

- Comments can be inserted into the HTML code to make it more readable and understandable by developer.
- Comments are ignored by the browser and are not displayed.
- An HTML comment starts with `<!--` and ends with `-->`
- Example

```
<!-- This is a comment, the next paragraph is ignored by  
browser
```

```
<p>Hell </p> -->
```

HTML Text Formatting Tags

Tag	Description
	Defines bold text
	Defines emphasized text
<i>	Defines a part of text in an alternate voice or mood
<small>	Defines smaller text
	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
	Defines deleted text

- **Note:** It is better to avoid using presentational tags such as `` and `<i>`; instead use `` and `` tags.

HTML Text Formatting Example

```
<!DOCTYPE html>
<html>
<body>
<p><b>This text is bold</b></p>
<p><i>This text is italic</i></p>
<p>This is<sub> subscript</sub>
  and <sup>superscript</sup></p>
</body>
</html>
```

This text is bold
This text is italic
This is_{subscript} and^{superscript}

HTML Images

- HTML images are defined using the `` (image) tag.
- The image tag has no closing tag.
- The filename and the size of the image are specified as attributes.
- The ***src*** attribute specifies the *source URL* of the image.

- *Example:*

```

```

- The ***alt*** attribute specifies an alternate text for an image if the image cannot be displayed.

HTML Ordered (Numbered) Lists

- An ordered (numbered) list starts with the `` tag and ends with ``
- Each item within the list starts with the `` tag and ends with ``
- The browser displays the list items marked with as numbers (1,2, ..)
- *Example:*

```
<ol>
```

```
<li>tea</li>
```

```
<li>Coffee</li>
```

```
<li>Milk</li>
```

```
</ol>
```

1. tea
2. coffee
3. milk

HTML Unordered Lists

- An unordered list starts with the `` tag and ends with ``
- Each item within the list starts with the `` tag and ends with ``
- The browser displays the list items marked with bullets
- *Example:*

```
<ul>
```

```
  <li>tea</li>
```

```
  <li>Coffee</li>
```

```
  <li>Milk</li>
```

```
</ul>
```

- tea
- coffee
- milk

HTML Definition Lists

- A definition list is a list of items, with a description of each item.
- The `<dl>` tag defines a definition list. Inside a `<dl>`, the tags `<dt>` (definition term) and `<dd>` (definition data) can be used multiple times.
- *Example:*

```
<dl>  
  <dt>Coffee</dt>  
  <dd>- black hot drink</dd>  
  <dt>Milk</dt>  
  <dd>- white cold drink</dd>  
</dl>
```

Coffee

- black hot drink

Milk

- white cold drink

HTML Elements: block versus inline

- Most HTML elements are defined as either **block** elements or **inline** elements.
- Block elements are rendered with a line break before and after the element
 - *Examples: <h1>, <p>, , , <table>, <div>*
- Inline elements are normally displayed without being surrounded by empty lines
 - *Examples: , , , <a>, , *
- **Note:** A block element can contain both inline and block elements but an inline element can not contain block elements.
- **Question:** What is a typical example where an inline element appears inside another inline element?

The `<div>` Element

- HTML elements can be grouped using either `<div>` or `` tags.
- A `<div>` element acts as a container (box) for grouping other HTML elements
 - ❑ A `<div>` is a block element; that is, the browser will display a line break before and after it
 - ❑ By default, a `<div>` element uses a width (and is resized to) equal to the width of the browser's window; the div's height varies so as to accommodate the div's content
 - ❑ To see this, assign a background color as in:

```
Testing<div style="background-color:yellow" >text inside  
div</div>the div tag.
```
- When used together with CSS, a `<div>` element can be used to set style attributes to be shared (inherited) by all child elements.
- Using `<div>` elements is essential for creating flexible page layout.

The `` Element

- Unlike `<div>`, the `` element is an inline element; that is, it does not result in line breaks surrounding it
- A `` element acts as a container for text or other inline elements
- When used together with CSS, a `` element can be used to set style attributes to parts of the text
- The following is an example:

```
Testinginside span</span>the span tag.
```

The *<iframe>* Element

- An iframe is used to display a web page within a web page.
- The height and width attributes are used to specify the height and width of the iframe.
- The values for the height and width are specified in pixels by default, but they can also be in percent (like in the example "80%" of width of parent element)

```
<iframe src="demo.htm" height="200"  
width="80%" ></iframe>
```

The `<iframe>` Element (cont.)

- The *frameborder* attribute specifies whether to display a border around the iframe; set the attribute value to "0" to remove the border:

```
<iframe src="demo_iframe.htm" frameborder="0" ></iframe>
```

- An iframe can be used as the target for a link

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
```

```
<p> <a href="http://www.w3schools.com" target="iframe_a">  
  W3Schools</a>
```

```
</p>
```

HTML Entities

- Some characters are reserved in HTML.
- It is not possible to use the *less than* (<) or *greater than* (>) signs in your text, because the browser will mix them with tags.
 - For example, “`<p>if X < 10 then increment X</p>`” is reported as having a syntax error when using HTML validation tools
 - To display a less than sign, we must write: **<** or **<**
- To have the browser display reserved characters, we must use character entities in the HTML source code.

&entity_name;

OR

&#entity_number;

HTML Character Entities

Result	Description	Entity Name	Entity Number
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
1/2	fraction 1/2	½	½
1/4	fraction 1/4	¼	¼
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
§	section	§	§
©	copyright	©	©
®	registered trademark	®	®
™	trademark	™	™

Reference: <http://dev.w3.org/html5/html-author/charref>

HTML Colors

- HTML colors are defined using a hexadecimal or decimal notation for the combination of Red, Green, and Blue color values (RGB)
- The lowest value that can be given to one of the light sources is zero (in HEX: 00) , the highest value is 255 (in HEX: FF)

- *Examples:*

Black color: #000000 OR rgb(0,0,0)

Red Color: #FF0000 OR rgb(255,0,0)

HTML Color Names

- Color Names are supported by all modern browsers
- 147 color names are defined in HTML/CSS color specification
 - ❑ 16 basic colors: *aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow*
 - ❑ Light and dark versions of colors can also be specified: *lightblue, darkgreen, ... etc.*

XHTML

- Many pages on the internet contain "bad" HTML
 - No closing tag
 - Mixed upper/lower case for tag names
- XHTML is HTML that complies with XML rules
- XML is a markup language where documents must be marked up correctly and "well-formed"
- XHTML is a stricter and cleaner version of HTML 4.01
- XHTML stands for *EXtensible HyperText Markup Language*
- XHTML is supported by all major browsers

Differences Between XHTML and HTML

- Document Structure
 - ❑ XHTML DOCTYPE is **mandatory**
 - ❑ The XML namespace attribute in <html> is **mandatory**
 - ❑ <html>, <head>, <title>, and <body> is **mandatory**
- XHTML Elements
 - ❑ XHTML elements must be **properly nested**
 - ❑ XHTML elements must always be **closed**
 - ❑ XHTML elements must be in **lowercase**
 - ❑ XHTML documents must have **one root element**
- XHTML Attributes
 - ❑ Attribute names must be in **lower case**
 - ❑ Attribute values must be **quoted**
 - ❑ Attribute minimization is **forbidden**

The *<head>* Element

- The *<head>* element is a container for a specific subset of HTML elements. These elements normally correspond to nonvisible elements
- The following tags can be included inside the *<head>* section:
 - ❑ *<title>*
 - ❑ *<style>*
 - ❑ *<meta>*
 - ❑ *<link>*
 - ❑ *<script>*
 - ❑ *<noscript>*
 - ❑ *<base>*

The *<title>* Element

- The *<title>* tag defines the title of the document
 - The title is displayed as the caption of the browser's window
 - It provides a title for the page when it is added to favorites
 - It displays a title for the page in search-engine results

The `<base>` Element

- The `<base>` tag specifies a default address or a default target for all links on a page:

```
<head>
```

```
  <base href="http://www.w3schools.com/images/">
```

```
  <base target="_blank">
```

```
</head>
```

The `<link>` Element

- The `<link>` tag defines the relationship between a document and an external file (resource).
- The `<link>` tag is mostly used to link to an external style sheet:

```
<head>
```

```
  <link rel="stylesheet" type="text/css"  
        href="mystyle.css" >
```

```
</head>
```

The `<style>` Element

- The `<style>` tag is used to define style information for an HTML document.
- Inside the `<style>` element you specify how HTML elements should render in a browser:

```
<head>
```

```
  <style type="text/css">
```

```
    body {background-color:yellow; font-size:10pt; }
```

```
    p {color:blue; font-family:Tahoma; }
```

```
  </style>
```

```
</head>
```


The *<meta>* Element

- Metadata is data (information) about data
- The `<meta>` tag provides metadata about the HTML document
- Metadata will not be displayed on the page, but will be machine parsable.
- Meta elements are typically used to specify encoding character set, page description, keywords, author of the document, last modified
- The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

The `<meta>` Element (cont.)

- Define keywords for search engines:

```
<meta name="keywords" content="HTML, CSS, XML">
```

- Define a description of your web page:

```
<meta name="description"  
      content="Free Web tutorials on HTML and CSS">
```

- Define the author of a page:

```
<meta name="author" content="Nasir Darwish">
```

- Refresh page every 30 seconds:

```
<meta http-equiv="refresh" content="30">
```

The `<script>` Element

- Used to define a client-side script, such as a JavaScript
- *Example:*

```
<script>  
    document.write("Hello World!")  
</script>
```

- The `<script>` element either contains script code or point to an external script file using the ***src*** attribute
- Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

The *<noscript>* Element

- The *<noscript>* element provides an alternate content for users that have disabled scripts in their browser or have a browser that doesn't support client-side scripting.
- The *<noscript>* element can contain HTML elements that can appear inside the *<body>* element.
- The content inside the *<noscript>* element will only be displayed if scripts are not supported, or are disabled in the user's browser:

```
<script>
```

```
    document.write("Hello World!")
```

```
</script>
```

```
<noscript>
```

```
    Sorry, your browser does not support JavaScript!
```

```
</noscript>
```

HTML Tables

- An HTML table is defined using the **<table>** tag
- A table consists of a set of rows (each row uses a **<tr>** tag), and each row consists of column data cells (a column cell uses a **<td>** tag)
- A **<td>** tag can contain text, links, images, lists, forms, other tables, etc.
- The example shows an HTML table with three rows and two columns

```
<table border="1">
<tr>
  <td>row 1, cell 1</td>
  <td>row 1, cell 2</td>
  <td>row 1, cell 3</td>
</tr>
<tr>
  <td>row 2, cell 1</td>
  <td>row 2, cell 2</td>
  <td>row 2, cell 3</td>
</tr>
</table>
```

row 1, cell 1	row 1, cell 2	row 1, cell 3
row 2, cell 1	row 2, cell 2	row 2, cell 3

HTML Tables: Header Row

- A header row can be specified using `<th>` tags inside a `<tr>`
- Browsers normally display the text in a `<th>` element as bold and centered

```
<table border="1">  
  <tr>  
    <th>Course</th>  
    <th>Instructor</th>  
  </tr>  
  <tr>  
    <td>ICS 251</td>  
    <td>Darwish</td>  
  </tr>  
  <tr>  
    <td>ICS 353</td>  
    <td>Khateeb</td>  
  </tr>  
</table>
```

Course	Instructor
ICS 251	Darwish
ICS 353	Khateeb

HTML Forms

- HTML forms are used to pass data from the user to a web server .
- An HTML form can contain input elements like text fields, checkboxes, radio buttons, submit buttons, select droplists, textarea, fieldset, legend, and label elements

`<form>`

..

various input elements and submit/reset buttons

..

`</form>`

Styling with CSS

- CSS was introduced together with HTML 4, to provide a better way to style HTML elements.
- CSS can be added to HTML in the following ways:
 - *Inline*: using the ***style*** attribute in HTML elements
 - *Internal* (embedded): using the **<style>** element in the <head> section
 - *External*: using **a separate CSS file**
- Putting CSS data in separate CSS files is the preferred way for a site with many pages.

Inline Styles

- An inline style applies CSS styles to one single occurrence of an HTML element.
- An inline style is specified using the ***style*** attribute whose value can specify one or more CSS properties.
- *Examples:*
 - ❑ `<body style="background-color:yellow" >`
 - ❑ `<h1 style="font-family:verdana;font-size:16pt" >`
Some heading</h1>
 - ❑ `<p style="color:blue;margin-left:20px" >`
Some paragraph</p>

Internal Style Sheet

- An internal style sheet can be used if one single document has a unique style.
- Internal styles are defined using the `<style>` element which is normally placed in the `<head>` section.
- *Example:*

```
<head>
```

```
<style type="text/css">
```

```
    body { background-color:yellow;}
```

```
    p {color:blue;}
```

```
</style>
```

```
</head>
```

External Style Sheet

- An external style sheet is ideal when the same set of styles is applied to many pages.
- A page can use the styles defined in the external sheet if it *links* to the style sheet using the **<link>** tag.
- The **<link>** tag is normally placed inside the **<head>** section.
- *Example:*

```
<head>
```

```
<link href="mystyles.css" rel="stylesheet" type="text/css" >
```

```
</head>
```

Useful Resources

- HTML Basics (from “Web Programming Step by Step”), <http://www.webstepbook.com/supplements-2ed/slides/chapter02-html.shtml>
- W3 Schools – HTML Tutorial, <http://www.w3schools.com/html/>
- SitePoint – HTML Reference, <http://reference.sitepoint.com/html>
- W3C HTML Validation Service, <http://validator.w3.org/>