CSE417: WEB ENGINEERING

Daffodil International University

You Will Be Able To

- ✔ Know basics of static web page
- ✔ Use HTML elements

“To acquire knowledge, one must study; but to acquire wisdom, one must observe.”
― **Marilyn vos Savant**

Contents

- HTML Elements
  - Structure
  - Paragraphs
  - Headings
  - Text
  - Lists
  - Hyperlinks
  - Images
  - Tables

Tags vs. Elements

- HTML specifies a set of *tags* that identify structure and content type
  - tags are enclosed in `<  >`

    `<img src="image.gif" />` specifies an image

  - most tags come in pairs, marking a beginning and ending

    `<title>` and `</title>` enclose the title of a page

- an HTML *element* is an object enclosed by a pair of tags

  `<title>My Home Page</title>` is a TITLE element

  `<b>This text appears bold.</b>` is a BOLD element

  `<p>Part of this text is <b>bold</b>.</p>`
  is a PARAGRAPH element that contains a BOLD element

HTML document is a collection of elements (text/media with context)

Structural Elements

- an HTML document has two main structural elements
  - HEAD contains setup information for the browser & the Web page

    e.g., the title for the browser window, style definitions, JavaScript code, …
  - BODY contains the actual content to be displayed in the Web page

```
<html>
 <!-- Version information --
 -- File: page01.html  --
 -- Author: DIU--
 -- Creation: 15.08.09 --
 -- Description: introductory page  --
 -- Copyright: DIU--
 -->
 <head>
  <title>My first HTML document</title>
 </head>
 <body>
  Hello world!
 </body>
</html>
```

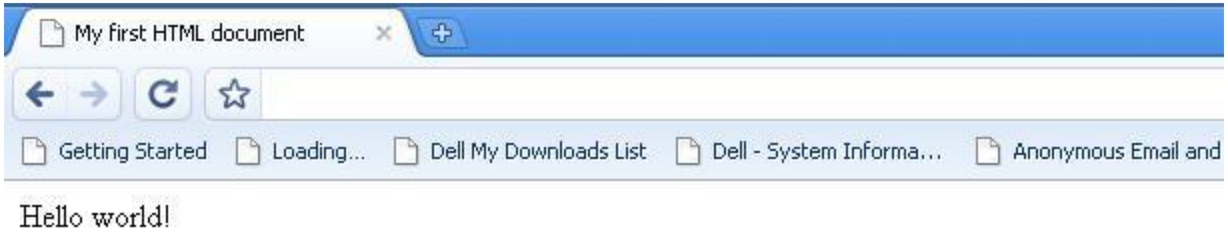
HTML documents begin and end with `<html>` and `</html>` tags

Comments appear between `<!--` and `-->`

HEAD section enclosed between `<head>` and `</head>`

BODY section enclosed between `<body>` and `</body>`

`* Find more info on HTML docs!`



Hello world!

Text Layout(Pragraph)

```
<html>
<head>
  <title>Text Layout</title>
</head>

<body>
  <p>
  This is a paragraph of text <br/>
  made up of two lines.
  </p>

  <p>
  This is another paragraph with a
  &nbsp; GAP &nbsp; between
  some of the words.
  </p>

  <p>
  &nbsp;&nbsp; This paragraph is<br/>
  indented on the first line <br/>
  but not on subsequent lines.
  </p>
</body>

</html>
```

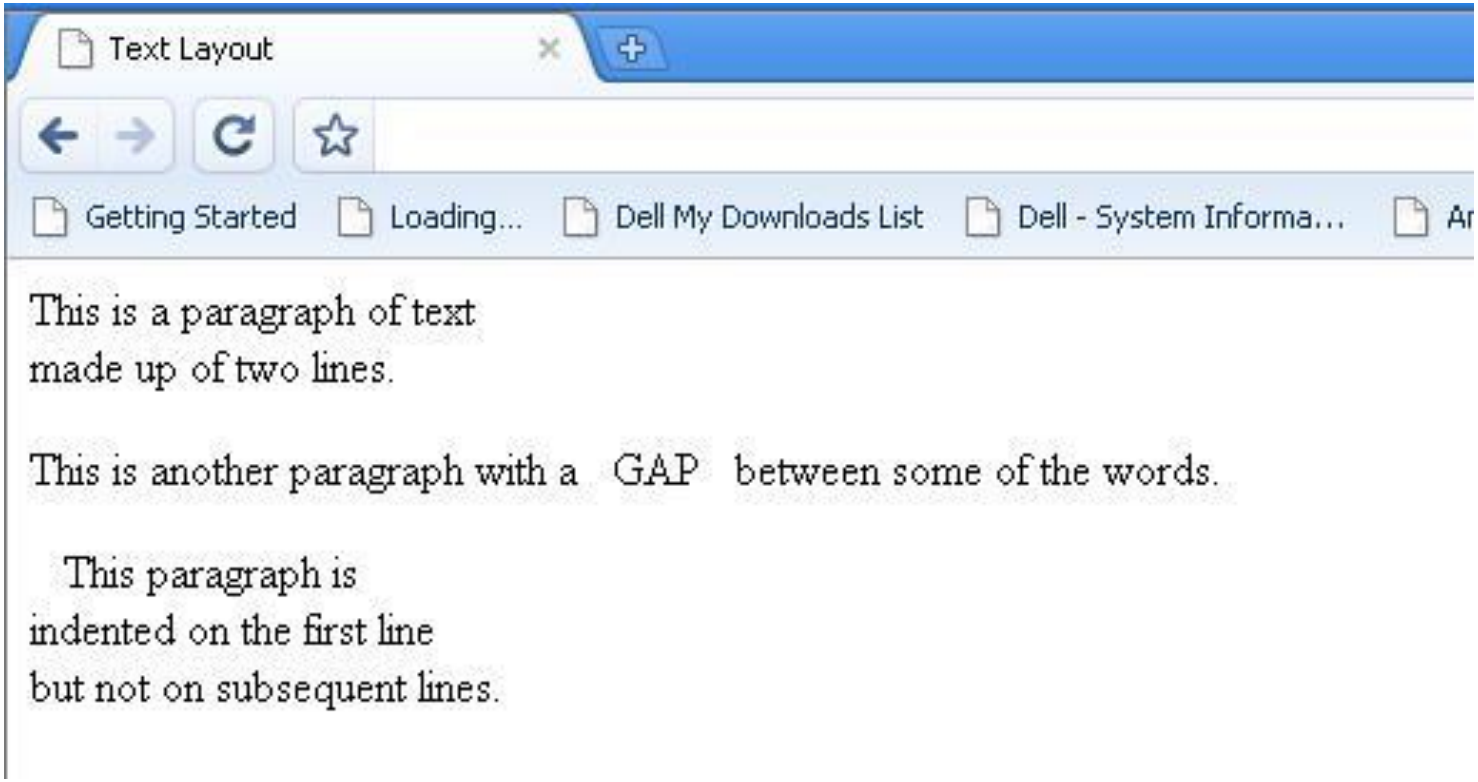
for the most part, layout of the text must be left to the browser

- every sequence of whitespace is interpreted as a single space
- browser automatically wraps the text to fit the window size

can override some text layout

- can cause a line break using the `<br/>` tag (no closing tag)
- can specify a new paragraph (starts on a new line, preceded by a blank line) using `<p>…</p>`
- can force a space character using the symbol for a non-breaking space: `&nbsp;`

Text Layout…



This is a paragraph of text
made up of two lines.

This is another paragraph with a   GAP   between some of the words.

This paragraph is
indented on the first line
but not on subsequent lines.

Headings: Separating Blocks of Text

```
<html>
<head>
  <title>Blocks of Text</title>
</head>

<body>
  <h1>Major heading 1</h1>
  <p>
  Here is some text.
  </p>

  <h2>Subheading</h2>
  <p>
  Here is some subtext.
  </p>

  <hr/>

  <h1>Major heading 2</h1>
  <p>
  Here is some more text.
  </p>
</body>

</html>
```

can specify headings for paragraphs or blocks of text

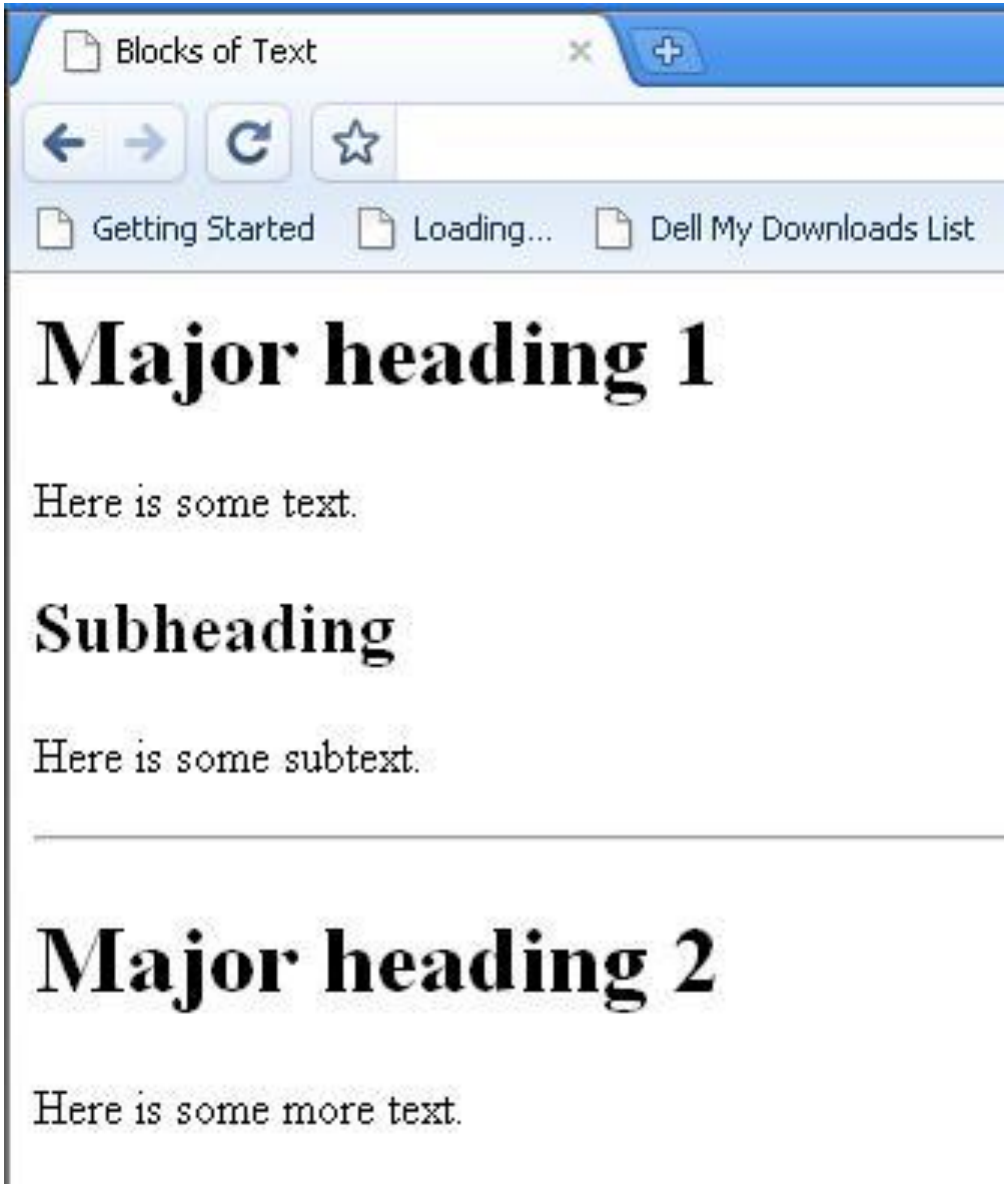
- `<h1>…</h1>` tags produce a large, bold heading
- `<h2>…</h2>` tags produce a slightly smaller heading

  . . .
- `<h6>…</h6>` tags produce a tiny heading

can insert a horizontal rule to divide sections

- `<hr/>` draws line across window

Headings: Separating Blocks of Text



Major heading 1

Here is some text.

Subheading

Here is some subtext.

Major heading 2

Here is some more text.

The Basic Web page – A Worked Example

```
<html>
<head>
      <title> Bill Smiggins Inc. </title>
   </head>
   <body>
      <h1>Bill Smiggins Inc.</h1>
      <h2>About our Company...</h2>
      <p>This Web site provides clients, customers,
         interested parties and our staff with all of
         the information that they could want on
         our products, services, success and failures.
      </p>
      <hr/>
      <h3> Products </h3>
      <p> We are probably the largest
      supplier of custom widgets, thingummybobs, and bits
      and pieces in North America. </p>
      <hr/>
   </body>
</html>
```

Text Appearance

```
<html>
<head>
  <title>Text Variations and Escape
Sequences</title>
 </head>
 <body>
  <h1>Text Variations</h1>
  <p>We can use <b>simple</b> tags to
     <i>change</i> the appearance of
     <strong>text</strong> within
     <tt>Web pages</tt>.
     Even super<sup>script</sup>
     and sub<sub>scripts</sub> are
      <em>supported</em>.</p>

   <h1>Text Escape Sequences</h1>
  <p>
    &amp;  &lt;  &gt;  &quot;  &copy;
  </p>
 <h1>Preformatted text</h1>
        <pre>
     BRAC University
     Department of Computer Science
     CSE391: Programming For Internet :)
</pre>
 </body>
</html>
```

can specify styles for fonts

- <b>… </b> specify bold
- <i>… </i> specify italics
- <tt>… </tt> specify typewriter-like (fixed-width) font

- <big>… </big> increase the size of the font
- <small>… </small> decrease the size of the font
- <em>…</em> put emphasis
- <strong>…</strong> put even more emphasis

- <sub>… </sub> specify a subscript
- <sup>… </sup> a superscript

- <pre>…</pre> include ready-formatted text

- &amp; &al; &gt; &quot; &copy; escape characters used in HTML control

* *Find more info on text tags!*

Lists

```
<html>
<head>  <title>(Sort of) Simple Lists</title>
  <style type=“text/css”>
  .my_li:before {content: counter(list) “: “;
                 counter-increment: list;  }
  </style>  </head>
<body>

<ul style="list-style-type: square;">
 <li> ... first list item... </li>
 <li> ... second list item... ...  </li>
</ul>
<dl>   <dt> Dweeb </dt>
    <dd> young excitable person who may
  mature into a <em>Nerd</em>  </dd>
 <dt> Hacker </dt>
      <dd> a clever programmer  </dd>
 <dt> Nerd </dt>  <dd> technically bright but
        socially inept person  </dd>
</dl>
 <ol style="list-style-type: none;
           counter-reset: list 29;" >
  <li class=“my_li”>Makes first item number
30.</li>
  <li class=“my_li”>Next item continues to number
31.</li>
</ol>

</body>
</html>
```

there are 3 different types of list elements

- <ol>…</ol> specifies an ordered list (using numbers or letters to label each list item)
  <li> identifies each list item

  can set type of ordering, start index

- <ul>…</ul> specifies unordered list (using a bullet for each)
  <li> identifies each list item

- <dl>…</dl> specifies a definition list <dt> identifies each term
  <dd> identifies its definition

* We will learn more about the “style” attributes soon enough.

Hyperlinks

```
<html>

<head>
  <title>Hyperlinks</title>
</head>

<body>
  <p>
  <a href="http://www.google.com">
    Google University</a>
  <br/>
  <a href="page22.html" target="_blank">
   Open page22 in a new window </a>
  </p>
</body>

</html>
```

perhaps the most important HTML element is the hyperlink, or ANCHOR

- `<a href="`*`URL`*`">…</a>`

  where URL is the Web address of the page to be displayed when the user clicks on the link

  if the page is accessed over the Web, must start with `http://`

  if not there, the browser will assume it is the name of a local file

- `<a href="`*`URL`*`" target="_blank">…</a>`

  causes the page to be loaded in a new Window

** Find more info on attribute TARGET*

Hyperlinks (cont.)

```
<html>
<head>
  <title>Internal Links in a
Page</title>
</head>

<body>
  <p>
  [ <a href="#HTML">HTML</a>  |
    <a href="#HTTP">HTTP</a>  |
    <a href="#IP">IP</a>  |
    <a href="#TCP">TCP</a>  ]
  </p>
  <p>
  Computer acronyms:
  <dl>
   <a name="HTML"></a><dt>HTML</dt>
   <dd>HyperText Markup Language
   <a name="HTTP"></a><dt>HTTP</dt>
   <dd>HyperText Transfer Protocol…</dd>
   <a name="IP"></a><dt>IP</dt>
   <dd>Internet Protocol…</dd>
   <a name="TCP"></a><dt>TCP</dt>
   <dd>Transfer Control Protocol…</dd>
  </dl>
  </p>

</body>
</html>
```

for long documents, you can even have links to other locations in that document

- `<a name="`*`ident`*`">…</a>`

  where ident is a variable for identifying this location

- `<a href="#`*`ident`*`">…</a>`

  will then jump to that location within the file

- `<a href="URL#`*`ident`*`">…</a>`

  can jump into the middle of another file just as easily

Images

can include images using `IMG`

- by default, browsers can display GIF and JPEG files
- other image formats may require plug-in applications for display

```
<img src="URL"|"name" height="n" width="n" alt="text" title=
"text" />
```

again, if file is to be accessed over the Web, must start with http:// (if not, will assume local file)

* Find more info on `<img/>`

```
<html>
<head>
  <title>Images</title>
</head>
<body>
<img src="w3javascript.png" alt="JS" title="JavaScript" />
  <p>W3Schools JavaScript Tutorials</p>
</body>
</html>
```

Tables

- tables are common tools for arranging complex layout on a Web page
  - a table divides contents into rows and columns
  - by default, column entries are left-justified, so provide for alignment

```
<html>
<head>
  <title>Tables</title>
</head>
<body>
<h2>A Simple Table</h2>
  <table>
   <tr>
     <td> Left Column </td>
     <td> Right Column </td>
   </tr>
   <tr>
     <td> Some data </td>
     <td> Some data </td>
   </tr>
 </table>
</body>
</html>
```

`<table>…</table>` specify a table element

`<tr>…</tr>` specify a row in the table

`<td>…</td>` specify table data (i.e., each column entry in the table)

Layout in a Table

```
<html>

<head>
  <title>Table Layout</title>
</head>

<body>
  <table style="border: 1px solid;">
     <tr style="text-align: center;">
     <td style="border: 1px solid;">
            Left<br/>Column</td>
     <td style="border: 1px solid;
         vertical-align: top;">
            Right Column</td>
    </tr>
    <tr>
      <td style="border: 1px solid;">
           Some data</td>
      <td style="border: 1px solid;">
           Some data</td>
    </tr>
  </table>
</body>
</html>
```

can have a border on tables using the “border” attribute

```
<table style= “border: 1px solid;” >
```

increasing the number makes the border thicker

can control the horizontal & vertical layout within cells

```
<td style= “text-align:center”>

<td style= “vertical-align:
   bottom”>
```

can apply layout to an entire row

```
<tr style=“text-align: center”>

<tr style=“vertical-align: top”>
```

We will explore this more with Cascading Style Sheets.

Table Width

```
<html>
<head>
  <title>Table Width</title>
</head>

<body>
<table style="width: 100%;">
    <tr>
      <td>left-most </td>
      <td style="text-align: right;">
                right-most</td>
    </tr>
</table>
</body>
</html>
```

by default, the table is sized to fit the data

can override & specify the width of a table relative to the page

```
<table style=“width: 60%”>
```

Other Table Attributes

```
<html>
<head>
  <title>Table Formatting</title>

 <style type="text/css" media="screen">
     table { border: 1px solid; padding: 1px;}
     th, td {  border: 1px solid; padding: 10px;
               text-align: center; }
  </style>
</head>

<body>
  <table>
    <tr>
      <th>HEAD1</th> <th>HEAD2</th>
<th>HEAD3</th>
    </tr>
    <tr>
      <td>one</td> <td>two</td> <td>three</td>
    </tr>
    <tr>
      <td rowspan="2"> four </td>
      <td colspan="2"> five </td>
    </tr>
    <tr>
      <td> six </td> <td> seven </td>
    </tr>
</table>
</body>
</html>
```

can control the space between cells & margins within cells

This is the “padding” attribute in the table and th,td style sheet declarations (more on this with Cascading Style Sheets).

can add headings

`<th>` is similar to `<td>` but displays heading centered in bold

can have data that spans more than one column

`<td colspan=“2”>`

similarly, can span more than one row

`<td rowspan=“2”>`

(This example uses CSS style sheet commands in the page <header>.)

Frames

- frames provide the ability to split the screen into independent parts

Frames are going out of fashion, partly because they interact poorly with web search engines (i.e. search engines cannot generally access the data stored in the inset frame objects).

Frames can also “break” the regular behaviour of browsers, most notably the “Back” button on the browser can behave in unexpected ways.

Because of these drawbacks to frames, I will not be discussing them in this course.

If you wish to design websites using frames, I would encourage you to use the XHTML XFrames specifications (see the W3C website for more details), but this specification isn’t fully supported by all browsers at this time.

Exercise

- Why do we need protocols?
- Write two applications of HTTP.
- **READINGS**
  - M Schafer: Ch. 4, 5, 6, 7, 8
  - https://www.w3schools.com/html

Acknowledgement

- This module is designed and created with the help from following sources-
  - https://cgi.csc.liv.ac.uk/~ullrich/COMP519/
  - http://www.csc.liv.ac.uk/~martin/teaching/comp519/