Introduction to HTML
OBJECTIVES

In this unit, you will learn:

- Characteristics of HTML5 vs. older HTML standards and XHTML.
- How to write HTML to create web pages:
  - Controlling document structure
  - Making paragraphs and lists
  - Including images in a document
  - Inserting hyperlinks for navigation
  - Including meta information for search engines to use
• Hyper-Text Markup Language: foundation of the World-Wide Web

• Design goals:
  – Platform independence: pages can be viewed using a variety of different computers and browsers.
  – Universality: servers can store information in their own data formats, but convert it to HTML for access by browsers.
  – Convenient linking from one page to another (hypertext).
  – HTML conveys the structure of the document, not its precise appearance, allowing for varying display capabilities.

• Openness (not proprietary) was key to the adoption of HTML and growth of the Web
HTML versions

• HTML versions 1 through 4 added functionality
  – But too simple for demands placed on it
  – “Wild West” situation: incompatibility galore

• XHTML: more precise and predictable
  – Documents are valid XML
  – But too strict for most developers
  – Never got much traction

• HTML5: simple, clean
  – Separation of content and style
  – Designed to integrate CSS, JavaScript, etc.
HTML5

- Not yet fully supported by browsers
  - We will follow HTML5 standards but not try bleeding edge stuff
- Backwards compatible with older HTML
- For more information:
  - Standard posted at w3c.org
  - Tutorials and references at w3schools.com
Basic form of HTML document

- Plain text (no binary codes)
  - Can be edited with any text editor (but don’t use Word or Wordpad!)
- Two categories of material:
  - Content: what the user sees when visiting the page
  - Meta-information: structure, layout, etc.
- Meta-information is distinguished from content by using tags.
  - A tag is a tag-name enclosed in angle brackets. Tags usually come in pairs: an opening tag and a closing tag: the same tag-name preceded by a slash:

  <tag-name>Content affected by tag</tag-name>
Nesting of tags

• Regions enclosed by tags must *nest*, not *overlap*.

Yes:
\(<\texttt{tag1}>\text{Some text } \texttt{tag2}>\text{more text } \texttt{/tag2}>\text{and more.} \texttt{/tag1}>\)

No:
\(<\texttt{tag1}>\text{Some text } \texttt{tag2}>\text{more text } \texttt{/tag1}>\text{and more.} \texttt{/tag2}>\)
Rules about tags

• Not all tags require closing tags
  – Tags that enclose a region, closing tag is implied.  
    **Example:** `<p>` (paragraph) tag: closing `</p>` implied by new paragraph.
    • Closing tag for these is mandatory in XHTML
  – Tags that mark a location, not a region never have a closing tag.
    **Example:** `<br>` (line break): never use `</br>`.
    Optionally use **self-closing flag**: `<br />`
    • Self-closing flag is mandatory in XHTML
Tag names

• Tag names are case-insensitive in HTML

Example: `<div> <Div> <DIV>` all equivalent
  - XHTML mandates lowercase
  - Corresponding Document Object Model (DOM) element names are lowercase
  - Best to use lowercase

• Unknown tags are ignored.
  - Allows new tags to be introduced into later HTML standards without causing older browsers to barf.
  - But also means be careful to spell tag names correctly!
Attributes in tags

- Some tags accept attributes that provide needed information or change default properties of the tag
- Attributes are included inside the angle brackets of opening tag (never in closing tag)

Example:

```html
<img src="mypic.jpg" alt="picture of me" />
```
Attributes

• Attribute values may be numbers or text strings
• Quotes around values may be optional
  – Safest always to use them
  – No “smart quotes” — only "straight quotes"
Tag for document type

• Was not in HTML4
  – Introduced in XHTML
  – Very complicated syntax: nobody could type it from memory. Example:
    ```html
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
    ```

• Simplified in HTML5
  – `<!DOCTYPE html>`
  – Always first tag of document. No closing tag.
Tags for document structure

- All HTML documents have this overall structure:

```html
<!DOCTYPE html>
<html>
  <head>
    Meta-information
  </head>
  <body>
    Page content
  </body>
</html>
```
Tags for document head

<title>Document Title</title> - specifies title that appears in title bar of browser window (not in content area).

Optional, but should always be included to assist user in navigating browser history list.

<meta name="keywords" content="word, phrase" /> - specifies keywords to assist search engines in indexing the page.

Other uses of meta tag exist as well.

Example 1
Tags for document body

• There are many tags for content area.
  – Structure
  – Formatting
  – Forms
  – Images
  – Links
  – Etc.
Heading tags

- Define headings at 6 levels of prominence:
  
  `<h1>Main heading</h1>`
  `<h2>Sub-heading</h2>`
  ...
  `<h6>Minor heading</h6>`

- Browser will choose effects such as size and boldface depending on display capabilities
Organizational tags

- Paragraphs and blocks of text
  `<p>`Paragraph of text.`</p>`
  `<br />` Line break. Use where extra space of `p` tag is not desirable, e.g. in an address.
  `<blockquote>`Lengthy quotation`</blockquote>`
  `<pre>`Pre-formatted text`</pre>`
  preserves line breaks and blank spaces as typed, using a monospace (tty) font.
- Except for `<pre>`, browser ignores whitespace in HTML when rendering document
Lists

• List types:
  – Ordered (numbered) list
    <ol>
      <li>first item</li>
      <li>second item</li>
    </ol>
  – Unordered (bulleted) list
    <ul>
      <li>item</li>
      <li>another item</li>
    </ul>
Definition lists

• Head of each item is word or phrase, item content is description

```html
<dl>
  <dt>Head word</dt>
  <dd>Description</dd>
  <dt>Next head</dt>
  <dd>Next description</dd>
</dl>
```

Example 3
Tags for style

• Use CSS for most styling, but these are handy

• Explicit styles
  
  <b>Boldfaced text</b>
  <i>Italicized text</i>
  <u>Underlined text</u>
  <tt>Typewriter-font text</tt>  removed in HTML5

• Logical styles
  
  <strong>Prominent text</strong> - usually bold
  <em>Emphasized text</em> - usually italics
  <cite>Cited text</cite> - usually italicized
  <code>Computer code</code> - usually in typewriter font

Example 4
The `<font>` tag

- In the early days of web design, this tag was introduced to allow specifying font family, typeface, weight, color.
- This font is not supported in HTML5.
  - But you may see it on some web pages
- Use style attribute or CSS instead.
Comments

- Enclosed inside `<!-- -->`
- Nothing appears in rendered page
- Use to document the HTML
Special characters

• To display characters with meaning to HTML (such as `< >`) or characters not available on keyboard, use character entity references of form `&code;`

  &lt; - less-than sign <
  &gt; - greater-than sign >
  &amp; - ampersand &
  &copy; - copyright symbol ©
  &Theta; - uppercase Θ
  &theta; - lowercase θ

• Many more are defined: see w3c.org for list

Example 5
Hypertext links

• These are what make the Web a web
• Two components: link text, which is what the user sees, and link target, which is where the browser goes when link text is clicked
• Form:
  \[
  \text{\lt a href=":target-URL\"\gt link text\lt/\text{a}\gt}
  \]

  The a stands for “anchor” which is another use this tag had in earlier versions of HTML
Link target

• The link target is in the form of a URL: Uniform Resource Locator

• A URL has 3 components, not all of which need to be supplied
  – A protocol
  – An internet address
  – A file path

Example:
http://www.fordham.edu/current/index.html
**URL protocol**

- The protocol portion of the URL specifies the way the web browser will interact with the server to obtain the resource. Commonly used protocols include:
  - `http` – Hypertext Transport Protocol
  - `https` – Secure HTTP
  - `ftp` – File Transfer Protocol, sometimes used for downloads
  - `file` – For URLs that are files on the machine where browser is running
  - `mailto` – Email address, invokes mail agent
URL address

• The internet address can be either a name, e.g. www.fordham.edu, or an IP address, e.g. 150.108.13.113

• If omitted, the address is the same as for the page containing the link
  – Best to omit for linking to other pages on server
  – Then whole site can easily be ported
URL file path

• The file path specifies the chain of directories to where the file is located, and the file itself.
  – Separate components of path by / not \n  – If file name is omitted, it defaults to a server-specific value such as index.html or index.php

Example:
http://www.myplace.com/shopping/fruit/
Lacks a filename, so it may be equivalent to
http://www.myplace.com/shopping/fruit/index.html
Relative URLs

- If address portion of URL is omitted, then file path can be *absolute* or *relative*
- Absolute path begins with /, relative path does not
- Absolute path is from top of document space
- Relative path is relative to directory containing the page

```html
<a href="/somepage.html">Absolute link</a>

<a href="somepage.html">Relative link</a>
```
Relative URLs

- A relative URL can reference other directories
- Use `../` to indicate parent directory

Link referencing a file in a subdirectory of the current page's directory

```html
<a href="assets/somepage.html">Relative link</a>
```

Link referencing a file in a sibling directory of the current page's directory

```html
<a href="../assets/somepage.html">Relative link</a>
```
Images

- Use `<img>` tag to embed an image in page
- Attributes:
  - `src` – URL giving location of image
  - `alt` – text description (used by screen readers)
  - `height, width` – size of image in pixels

*Example:*

```html
<img src="myface.jpg" height="256" width="300"
alt="Picture of me" /> 
```
Image height and width

- Can be used to resize image in page, but you should not do that
  - Their purpose is to reserve space for image while page is being laid out: set them to actual image size
  - If different size image is desired, right-size the image externally using an image editor such as GIMP