CSE3026: Web Application Development Basic HTML

Scott Uk-Jin Lee

2.1: Basic HTML

Hypertext Markup Language (HTML)

- describes the *content* and structure of information on a web page
  - not the same as the *presentation* (appearance on screen)
- surrounds text content with opening and closing **tags**
- each tag's name is called an **element**
  - syntax: <*element*> *content* </*element*>
  - example: `<p>This is a paragraph</p>`
- most whitespace is insignificant in HTML
  - ignored or collapsed to a single space
- we will use a newer version called HTML5

Structure of an HTML page

```
<!DOCTYPE html>
<html>
    <head>
        information about the page
    </head>

    <body>
        page contents
    </body>
</html>
```

- the **header** describes the page and the **body** contains the page's contents
- an HTML page is saved into a file ending with extension `.html`
- `DOCTYPE` tag tells browser to interpret our page's code as HTML5, the latest/greatest version of the language

Page title: <title>

describes the title of the web page

```
<title>Chapter 2: HTML Basics</title>
```

- placed within the head of the page
- displayed in the web browser's title bar and when bookmarking the page

Paragraph: <p>

paragraphs of text (block)

```
<p>You're not your job.
You're not how much money you have in the bank.
You're not the car you drive.  You're not the contents
of your wallet. You're not your        khakis.  You're
   the all-singing, all-dancing crap of the world.</p>
```

You're not your job. You're not how much money you have in the bank. You're not the car you drive. You're not the contents of your wallet. You're not your khakis. You're the all-singing, all-dancing crap of the world.

- placed within the body of the page
- more paragraph examples

Headings: <h1>, <h2>, ..., <h6>

headings to separate major areas of the page (block)

```
<h1>University of Whoville</h1>
<h2>Department of Computer Science</h2>
<h3>Sponsored by Micro$oft</h3>
```

<h1>University of Whoville</h1>
<h2>Department of Computer Science</h2>
<h3>Sponsored by Micro$oft</h3>

- More heading examples

Horizontal rule: <hr>

a horizontal line to visually separate sections of a page (block)

```
<p>First paragraph</p>
<hr />
<p>Second paragraph</p>
```

First paragraph
<hr />
Second paragraph

- should be immediately closed with />

More about HTML tags

- some tags can contain additional information called **attributes**
  - syntax: <*element attribute*="*value*" *attribute*="*value*"> *content* </*element*>
  - example: `<a href="page2.html">Next page</a>`
- some tags don't contain content; can be opened and closed in one tag
  - syntax: <*element attribute*="*value*" *attribute*="*value*" />
  - example: `<hr />`
  - example: `<img src="bunny.jpg" alt="pic from Easter" />`

Links: <a>

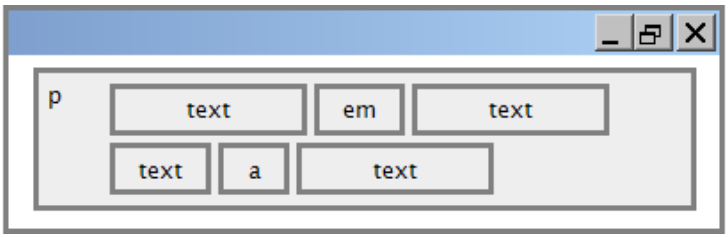
links, or "anchors", to other pages (inline)

```
<p>
  Search
  <a href="http://www.google.com/">Google</a> or our
  <a href="lectures.html">Lecture Notes</a>.
</p>
```

Search Google or our Lecture Notes.

- uses the `href` attribute to specify the destination URL
  - can be **absolute** (to another web site) or **relative** (to another page on this site)
- anchors are inline elements; must be placed in a block element such as `p` or `h1`

Block and inline elements



- **block** elements contain an entire large region of content
  - examples: paragraphs, lists, table cells
  - the browser places a margin of whitespace between block elements for separation
- **inline** elements affect a small amount of content
  - examples: bold text, code fragments, images
  - the browser allows many inline elements to appear on the same line
  - must be nested inside a block element

Images: <img>

inserts a graphical image into the page (inline)

```
<img src="images/gollum.jpg" alt="Gollum from LOTR" />
```

- the `src` attribute specifies the image URL
- HTML5 also requires an `alt` attribute describing the image

More about images

```
<a href="http://theonering.net/">
  <img src="images/gandalf.jpg" alt="Gandalf from LOTR"
   title="You shall not pass!" />
</a>
```

- if placed in an a anchor, the image becomes a link
- title attribute is an optional tooltip (on ANY element)

Line break:

forces a line break in the middle of a block element (inline)

```
<p>Teddy said it was a hat, <br /> So I put it on.</p>
<p>Now Daddy's sayin', <br /> Where the
heck's the toilet plunger gone?</p>
```

Teddy said it was a hat,
So I put it on.

Now Daddy's sayin',
Where the heck's the toilet plunger gone?

- Warning: Don't over-use br (guideline: >= 2 in a row is bad)
- br should not be used to separate paragraphs or used multiple times in a row to create spacing

Phrase elements: ,

em: emphasized text (usually rendered in italic)
strong: strongly emphasized text (usually rendered in bold)

```
<p>
  HTML is <em>really</em>,
  <strong>REALLY</strong> fun!
</p>
```

HTML is *really*, **REALLY** fun!

- as usual, the tags must be properly nested for a valid page

Nesting tags

```
<p>
  HTML is <em>really,
  <strong>REALLY</em> lots of</strong> fun!
</p>
```

- tags must be correctly nested
  - (a closing tag must match the most recently opened tag)
- the browser may render it correctly anyway, but it is invalid HTML
  - (how would we get the above effect in a valid way?)

Comments: <!-- ... -->

comments to document your HTML file or "comment out" text

```
<!-- My web page, by Suzy Student
     CSE3026, 2nd Semester 2025    -->
<p>CSE courses are <!-- NOT --> a lot of fun!</p>
```

CSE courses are a lot of fun!

- many web pages are not thoroughly commented (or at all)
- still useful at top of page and for disabling code
- comments cannot be nested and cannot contain a --

2.2: More HTML Elements

- 2.1: Basic HTML
- **2.2: More HTML Elements**
- 2.3: Web Standards

Unordered list: <ul>,

ul represents a bulleted list of items (block)
li represents a single item within the list (block)

```
<ul>
  <li>No shoes</li>
  <li>No shirt</li>
  <li>No problem!</li>
</ul>
```

- No shoes
- No shirt
- No problem!

More about unordered lists

- a list can contain other lists:

```
<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>
```

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

Ordered list:

ol represents a numbered list of items (block)

```
<p>RIAA business model:</p>
<ol>
  <li>Sue customers</li>
  <li>???</li>
  <li>Profit!</li>
</ol>
```

RIAA business model:

1. Sue customers
2. ???
3. Profit!

- we can make lists with letters or Roman numerals using CSS (later)

Definition list: <dl>, <dt>, <dd>

dl represents a list of definitions of terms (block)
dt represents each term, and dd its definition

```
<dl>
  <dt>newbie</dt> <dd>one who does not have mad skills</dd>
  <dt>own</dt> <dd>to soundly defeat
    (e.g. I owned that newbie!)</dd>
  <dt>frag</dt> <dd>a kill in a shooting game</dd>
</dl>
```

newbie
: one who does not have mad skills

own
: to soundly defeat (e.g. I owned that newbie!)

frag
: a kill in a shooting game

Quotations: <blockquote>

a lengthy quotation (block)

```
<p>As Lincoln said in his famous Gettysburg Address:</p>
<blockquote>
  <p>Fourscore and seven years ago, our fathers brought forth
    on this continent a new nation, conceived in liberty, and
    dedicated to the proposition that all men are created equal.</p>
</blockquote>
```

As Lincoln said in his famous Gettysburg Address:

> Fourscore and seven years ago, our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.

Inline quotations: <q>

a short quotation (inline)

```
<p>Quoth the Raven, <q>Nevermore.</q></p>
```

Quoth the Raven, “Nevermore.”

- Why not just write the following?
  `<p>Quoth the Raven, "Nevermore."</p>`

We don't use " marks for two reasons:

1. HTML shouldn't contain literal quotation mark characters; they should be written as `&quot;`
2. using `<q>` allows us to apply CSS styles to quotations (seen later)

HTML Character Entities

a way of representing any Unicode character within a web page

character(s)	entity
< >	< >
é è ñ	é è ñ
™ ©	™ ©
π δ Δ	π δ Δ
И	&#1048;
" &	" &

- More HTML entities
- How would you display the text & on a web page?

HTML-encoding text

```
&lt;p&gt;
  &lt;a href=&quot;http://google.com/search?q=marty&amp;ie=utf-8&quot;&gt;
    Search Google for Marty
  &lt;/a&gt;
&lt;/p&gt;
```

  Search Google for Marty

- To display the link text in a web page, its special characters must be encoded as shown above

Deletions and insertions: <del>, <ins>

content that should be considered deleted or added to the document (inline)

```
<p>
  <del>Final Exam</del> <ins>Midterm</ins> is on
  <del>Dec 10-15</del> <ins>Oct 10-11</ins>.
</p>
```

~~Final Exam~~ Midterm is on ~~Dec 10-15~~ Oct 10-11.

Abbreviations: <abbr>

an abbreviation, acronym, or slang term (inline)

```
<p>
  Safe divers always remember to check their
  <abbr title="Self-Contained Underwater Breathing Apparatus">SCUBA</abbr> gear.
</p>
```

Safe divers always remember to check their SCUBA gear.

Computer code: <code>

a short section of computer code (usually shown in a fixed-width font)

```
<p>
  The <code>ul</code> and <code>ol</code>
  tags make lists.
</p>
```

The ul and ol tags make lists.

Preformatted text: <pre>

a large section of pre-formatted text (block)

```
<pre>
        Steve Jobs speaks loudly
           reality distortion
         Apple fans bow down
</pre>
```

```
        Steve Jobs speaks loudly
           reality distortion
         Apple fans bow down
```

- displayed with exactly the whitespace / line breaks given in the text
- shown in a fixed-width font by default
- how would it look if we had instead enclosed it in code tags?

Web page metadata: <meta>

information about your page (for a browser, search engine, etc.)

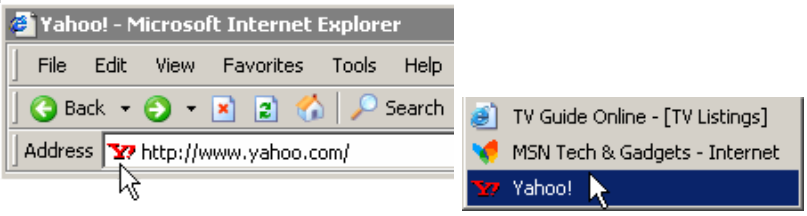
```
<meta charset="utf-8" />
<meta name="description"
    content="Authors' web site for Building Java Programs." />
<meta name="keywords" content="java, textbook" />
```

- placed **in the** `head` section of your HTML page
- `meta` tags often have both the `name` and `content` attributes
  - some `meta` tags use the `http-equiv` attribute instead of `name`

Favorites icon ("favicon")

```
<link href="filename" type="MIME type" rel="shortcut icon" />
```

```
<link href="yahoo.gif" type="image/gif" rel="shortcut icon" />
```



- the `link` tag, placed in the `head` section, attaches another file to the page
  - in this case, an icon to be placed in the browser title bar and bookmarks

HTML tables: <table>, <tr>, <td>

A 2D table of rows and columns of data (block element)

```
<table>
  <tr><td>1,1</td><td>1,2 okay</td></tr>
  <tr><td>2,1 real wide</td><td>2,2</td></tr>
</table>
```

1,1	1,2 okay
2,1 real wide	2,2

- `table` defines the overall table, `tr` each row, and `td` each cell's data
- tables are useful for displaying large row/column data sets
- NOTE: tables are sometimes used by novices for web page layout, but this is not proper semantic HTML and should be avoided

Table headers, captions: <th>, <caption>

```
<table>
  <caption>My important data</caption>
  <tr><th>Column 1</th><th>Column 2</th></tr>
  <tr><td>1,1</td><td>1,2 okay</td></tr>
  <tr><td>2,1 real wide</td><td>2,2</td></tr>
</table>
```

My important data

Column 1	Column 2
1,1	1,2 okay
2,1 real wide	2,2

- `th` cells in a row are considered headers; by default, they appear bold
- a `caption` at the start of the table labels its meaning

2.3: Web Standards

- 2.1: Basic HTML
- 2.2: More HTML Elements
- **2.3: Web Standards**

Web Standards

- It is important to write proper HTML code and follow proper syntax.
- Why use valid HTML and web standards?
  - more rigid and structured language
  - more interoperable across different web browsers
  - more likely that our pages will display correctly in the future
  - can be interchanged with other XML data: SVG (graphics), MathML, MusicML, etc.

W3C HTML Validator

```
<p>
  <a href="https://validator.w3.org/check/referer">
    <img src="images/w3c-html.png" alt="Validate" />
  </a>
</p>
```

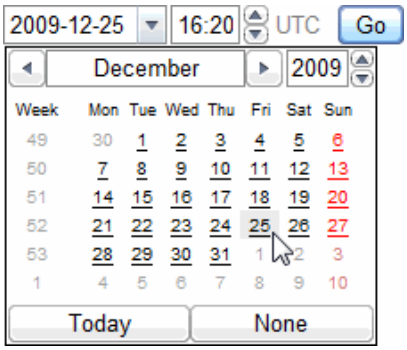
- validator.w3.org
- checks your HTML code to make sure it follows the official HTML syntax
- more picky than the browser, which may render bad HTML correctly

What is HTML 5?

- a new W3C standard version of the HTML markup language
- successor to HTML 4.01 and XHTML 1.1
- balance between too-loose-ness of HTML 4 and too-strict-ness of XHTML
- reduces the browser's need for plugins to display content, e.g. multimedia
- make web content more rich, semantically meaningful, descriptive, accessible
- more easily enable the web as an application platform (Web 2.0)

What's new in HTML 5?

- semantic **elements**: `nav`, `aside`, `header`, `footer`, `section`, `aside`, `article`
- **Forms** 2.0: sliders, search bars, color/number/email/url/date/time, placeholders, ...



- `audio` and `video` tags for embedding multimedia

- `canvas` tag for drawing 2D shapes in HTML/JS (like `DrawingPanel`) (ref 1, 2, 3)

What's new, continued

- semantic web / microdata **attributes**: `rel="..."`, `itemscope`, `itemtype`, `itemprop`

About Mark Pilgrim
Anytown PA - Developer advocate - Google, Inc.
Excerpt from the page will show up here.
diveintohtml5.org/examples/person-plus-microdata.html - Cached

- **accessibility** features ("ARIA")
- embedding of rich XML-like formats such as **SVG** vector graphics
- other stuff: offline apps, geolocation, cross-document messaging, MIME type registration, history management, ...

File formats

- some differ in features (animation, 5.1 stereo, transparency)
- many multimedia formats use **compression** to reduce file size
  - compression algorithms are also called **codecs** (list)
  - some compression algorithms are "lossless", others are "lossy"
- some formats are patented (unusable in free software)
- some formats are **encrypted** to protect information inside
  - Digital Rights Management (DRM) restricts what user can do with file
  - can be broken: DVD (DeCSS), HD DVD (09f911029d74e35bd84156c5635688c0)
- some formats are **streaming** (can play while downloading)

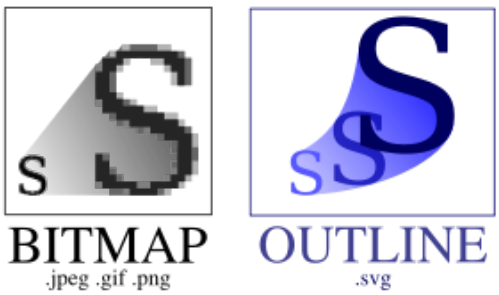
- comparisons of formats: audio/video containers

Image file formats

- JPEG : uses "lossy compression"; small file size; good for photos
- GIF : 256 colors; LZW run-length encoding lossless compression
  - allows **transparency** (can see behind parts of image)
  - possible to create animated GIFs
- PNG : free format created to avoid patent and color issues in GIF format; lossless compression, transparency
- others: TIFF, BMP
- image format comparisons: text, photo, PNG

Raster and vector graphics

- the image formats on the previous slide are **raster** or **bitmap** formats
  - they describe the pixels that should be drawn on the screen
- vector graphics formats such as SVG describe shapes/lines rather than pixels
  - advantage: infinite precision; good for zooming, printing
  - disadvantage: not supported on all platforms; computationally expensive



Audio file formats

- MP3 : uses lossy compression that eliminates inaudible sounds
- AAC : Apple's iTunes audio file format
- WMA / ASF: Microsoft Windows Media Audio format
- OGG : Linux hippie audio/video format
- RA / RM / RAM : Real Audio format
- other formats: WAV (MS), AU (Sun), AIFF / SND (Apple), FLAC
- Sequenced Music: MID, MOD

- comparison of formats

Video file formats

- MPEG : Motion Picture standard video format
  - DVDs are encoded using MPEG-2
  - HD DVDs are often compressed with MPEG-4 (H.264) codec
- MOV : Apple's QuickTime movie format
- WMV / ASF : Microsoft's Windows Media Video format
- AVI : classic Microsoft video format that can be encoded in many ways
- SWF / FLC : Macromedia Flash multimedia format
- RV : Real Video format

- comparisons of formats: 1, 2

Flash

- format for graphics, video, audio developed by Macromedia/Adobe
- widely used for many reasons:
  - supported in most major platforms/browsers
  - lightweight
  - can produce impressive interactive animated content
- downside: proprietary; editing software costs money (viewer is free)
- examples: Duck Hunt, Homestar Runner

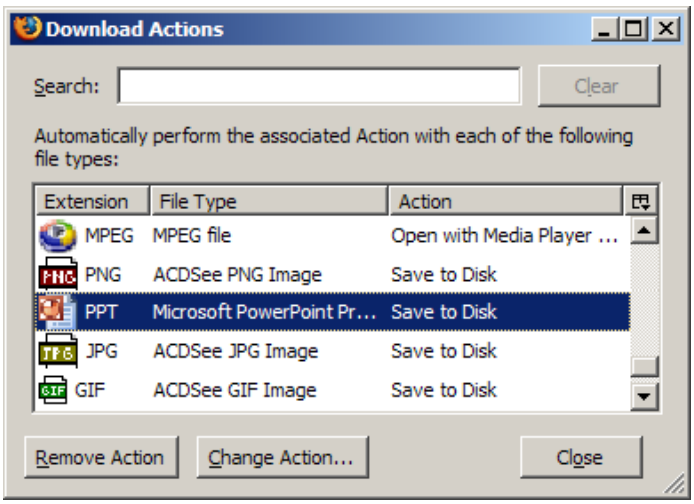
Linking to multimedia files

```
<a href="video.avi">My video</a>
```

- browser has a list of default applications to associate with each file type
- if it has an associated app, it will run it
- some file types are displayed within the browser using **plugins**
- if it doesn't know what to do, it will just download the file
- try it yourself: MPG, MOV, WMV, RM, SWF, WAV, MID

File types and browser plugins

- **plugin**: helper app launched within the browser to view certain file types
  - examples: Flash player, QuickTime, Windows Media Player, Acrobat Reader, Java
- about:plugins URL will show you list of plugins in Firefox
- enter preferences, then choose Content, File Types, Manage...
  - can change which app/plugin will be used to open particular file types



HTML 5 embedding a video

```
<video src="video.ogv" width="425" height="350"></video>
```

- yes, please
- it'll also be easy to manipulate (play/pause/stop/etc.) the video via the JS DOM
- one drawback: drama over codecs such as MPEG 4 / H.264, Ogg Theora / VP3, WebM, AVI

Embedded objects: <object>

```
<object data="video.avi" type="video/avi"></object>
```

- this is how you used to embed a video prior to HTML5
- attributes: archive, classid, codebase, codetype, data, declare, height, name, standby, type, usemap, width
- type attribute specifies file's MIME type

Parameters: <param>

```
<object data="duckhunt.mp4" width="100" height="50">
  <param name="autoplay" value="true" />
</object>
```

- indicates a parameter to be passed to the embedded object
- required name and value attributes tell the object what parameter this is and what value it should have

Embedding a YouTube video

```
<iframe width="width" height="height"
 src="videoURL">
</iframe>
```

<iframe width="425" height="350"
 src="http://www.youtube.com/embed/eKgPY1adc0A">
</iframe>