HTML5

Three Parts of a Web Page

- **HTML: provides *content* and *structure***
  - What do I mean by "content" and "structure"?
  - See for example Wikipedia
- CSS: provides formatting/presentation
- JavaScript/jQuery: provides behavior/interaction

HMTL

- it's highly recommended that you keep the three parts separated
- resource: www.w3schools.com/html/

A Very Simple HTML Page

```html
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>The Window/Tab Title</title>
    </head>
    <body>
        <h1>Level-1 Heading</h1>
        <p>I am a lonely paragraph.</p>
    </body>
</html>
```

How does this look in the browser? Let's see...

Document Type

- `<!doctype html>`
- it *must* be the first line of the document
- it tells the browser what version of HTML the file uses
- (`html` actually specifies HTML5)

Tags

- HTML tags almost always appear in pairs
- a *start tag* comes before the content: `<h1>`
- an *end tag* comes after the content: `</h1>`
- example:

```
<h1>Fishing for Dollars</h1>
```

Elements

- both tags, plus the content, are called an *HTML element*
- elements can be *nested* within other elements
- when you nest elements, be careful where you place the end tags:
  - the right way:

    ```
    <p>The concert was <em>awesome</em>.</p>
    ```

  - the WRONG way:

    ```
    <p>The concert was <em>awesome.</p></em>
    ```

Attributes

- the start tag can contain *attributes*
- attributes help *describe* the element but are not part of the content
- attributes are written with the format: `name="value"`
- example:

```
<html lang="en">
```

The <html> Element

- all elements are nested in the `<html>` element
- only `<!doctype>` is not placed between the `<html>` and `</html>` tags; it *must* come first

The <head> Element

- the `<head>` element contains *meta data* that is not content, so it is not displayed in the document
- some of the tags it can contain:
  - `<title>` is displayed in the title bar or a tab
  - `<style>` to include CSS styles on the page
  - `<script>` to include JavaScript on the page

The <body> Element

- all document content is placed in this element
- similarly, the content will be visible in the browser

HTML Comments

- HTML comments are ignored by the browser
- they are used to
    - temporarily hide some HTML

```
<p>My name is Al.</p>
<!--
<p>I live in Barcelona.</p>
-->
```

    - write a note/reminder to be read by a human

```
<!-- "Al" is a placeholder for the real name. -->
<p>My name is Al.</p>
```

A Very Simple Page (Revisited)

```html
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>The Window/Tab Title</title>
    </head>
    <body>
        <h1>Level-1 Heading</h1>
        <p>I am a lonely paragraph.</p>
    </body>
</html>
```

(Blank lines and indenting are for readability only.)

Some HTML Structure Tags

Headings

- there are six different levels of heading: `<h1>` (level 1) to `<h6>` (level 6)
- a level-1 heading can be used as the title for the entire document
- use smaller headings (with higher numbers) to break the document into sections and subsections

Paragraphs

- Surround each paragraph with a `<p>` tag

Structure Tag Example

```
<body>
    <h1>All About Fish</h1>
    <p>(Talk about all fish.)</p>

        <h2>Freshwater Fish</h2>
        <p>(Introduce freshwater fish.)</p>

            <h3>Behavior</h3>
            <p>(Discuss freshwater fish behavior.)</p>

            <h3>Feeding Habits</h3>
            <p>(Discuss freshwater fish feeding habits.)</p>

        <h2>Saltwater Fish</h2>
        <p>(Introduce saltwater fish.)</p>
```

Exercise H1

1. load `H0-starter.html`, and save it as `H1-structure.html`
2. define a level-1 heading, two level-2 headings, and a level-3 heading
3. place a paragraph under each of the headings
   - you can use lipsum.com to create some filler text for your paragraphs
4. add a note to yourself (as a comment) above the `<h3>` tag to "double-check this section later"
5. view the file in your browser

Validating Your HTML

- an *HTML validator* checks your HTML for mistakes... (but *not* content mistakes!)
- try it: validator.w3.org

Inspecting Your HTML

- modern browsers let you "inspect" your HTML
- (this functionality is useful for checking CSS and JavaScript too)

Whitespace

- *whitespace* characters are: space, tab, return/newline
- the browser compresses multiple whitespaces into one
- it also wraps the text as best it can to fill the line
- therefore you can indent your HTML to make it more readable (to you)

This HTML...

```
<p>Here is
one                         sentence.</p>
```

...will be displayed in the browser like this

```
Here is one sentence.
```

"Formatting" Tags

Tag	Description
`<strong>`	give text strong semantic importance
`<b>`	give text bold appearance (not recommended)
`<em>`	give text semantic emphasis
`<i>`	give text italic appearance (not recommended)

Semantic Tags

- some HTML tags still exist that only affect appearance; we want to avoid these
- *semantic tags* not only look different to human eyes; they also add meaning to the text, which is useful for programs (like search engines) that try to determine the important parts of our web pages
- therefore you should use semantic tags (like `<em>`) to give meaning to emphasized content
- if you simply want to italicize text, that's considered presentation, and it's the job of CSS (coming up later)

Exercise H2

1. make a copy of your file from the previous exercise, `H1-structure.html`, and call it `H2-semantic.html`
2. add semantic emphasis to the first few words of the first and third paragraphs
3. insert a few blank lines in the middle of the second paragraph
4. save the file and view your changes in the browser
5. validate your file at validator.w3.org

Links

- links are the "magic" we use to link from one page to another
- in HTML-speak they are called *anchors* (that explains the <a> tag)
- here is a simple example of an *internal link* (within the same site):

```
Find out more on the <a href="about.html">About page</a>.
```

- and it looks like this:
  Find out more on the About page.

Links: Other Attributes

Attribute	Description
`target`	the browser window/tab where the link will be opened
`title`	provide a tooltip when the link is hovered over

```html
<!-- Open the link in a new window/tab -->
View our
<a href="terms.html" target="_blank">terms</a>
of service.

<!-- Show the file type and size as a tooltip -->
<a href="report.pdf" title="PDF, 2.1MB">Download</a>
the report.
```

Linking to Pages in Other Directories

```html
<!-- Link to a page in the same directory as the current page -->
<a href="contact.html">Contact Us</a>

<!-- Link to a page in the 'stuff' subdirectory -->
<a href="stuff/contact.html">Contact Us</a>

<!-- Link to a page in the parent directory -->
<a href="../contact.html">Contact Us</a>
```

External Links

- links to other web sites require the entire URL, *including* `http://`

```
Visit the
<a href="http://store.apple.com">Apple Store</a>
to check out the new iMacs.

Find out more about Butifarra on the
<a href="http://en.wikipedia.org/wiki/Butifarra">Wikipedia page</a>.
```

Link Fragments/Bookmarks

A *fragment* or *bookmark* lets us link to somewhere besides the top of the page, and to scroll there. We implement it in two steps:

1. give an ID to the spot we want to be able to jump to (using the `id` attribute)

   ```
   <h3 id="fresh-behavior">Behavior</h3>
   ```

2. specify that location in our link (on the same page in this example)

   ```
   See <a href="#fresh-behavior">Section Behavior</a>
   ```

Link Fragments/Bookmarks (More)

A common use of a fragment is to return to the top of a long document.

```
<body id="top">

<!--
... Lots
and lots
of other stuff here ...
-->

<a href="#top">Go to top</a>
```

External Link Fragments/Bookmarks

A fragment we want to jump to can also exist on another page. Do these two steps:

1. define the location of the fragment in file `fish.html`

```
<h3 id="fresh-behavior">Behavior</h3>
```

2. specify that location in our link in a different file (in the same folder)

```
Find out more in
<a href="fish.html#fresh-behavior">Section Behavior</a>.
```

Exercise H3: Links

1. make a copy of your file `H1-structure.html`, and call it `H3-links.html`
2. insert a new paragraph at the top, just below the level-1 heading briefly describing two of your favorite web sites, and providing links to each of them
3. as the slideshow demonstrated, add a link to the bottom of your file, with a fragment/bookmark, that will scroll back up to the top of the file
4. add a sentence at the end of the first paragraph that links to file `H1-structure.html`
5. validate your file

Lists

- there are three kinds of lists we can use to structure our data
- an *unordered list* uses bullet points for each item, and it is used when the order of the items is not important
- an *ordered list* uses a number for each item, and it is used when the order *does* matter
- a *definition list* is like a glossary, that allows us to list *terms* and *definitions*

Lists

With *unordered lists*, the order of the items is not important.

HTML	Display
`<ul>` `<li>blue</li>` `<li>red</li>` `<li>green</li>` `</ul>`	• blue • red • green

```
<ul>
    <li>blue</li>
    <li>red</li>
    <li>green</li>
</ul>
```

- blue
- red
- green

Lists

With *ordered lists*, the order of the items *is* important.

HTML	Display

```
<ol>
    <li>preheat oven</li>
    <li>mix ingredients</li>
    <li>bake</li>
</ol>
```

1. preheat oven
2. mix ingredients
3. bake

Definition List

HTML

```
<dl>
    <dt>web browser</dt>
        <dd>a program to view web pages</dd>
    <dt>file browser</dt>
        <dd>a program to view files and folders</dd>
</dl>
```

Display

web browser

    a program to view web pages

file browser

    a program to view files and folders

Nesting Lists

We can also nest lists within a list item.

HTML

```
<ol>
    <li>preheat oven</li>
    <li>mix ingredients
        <ul>
            <li>bacon</li>
            <li>lemon</li>
            <li>chocolate</li>
        </ul>
    </li>
    <li>bake</li>
</ol>
```

Display

1. preheat oven
2. mix ingredients
    - bacon
    - lemon
    - chocolate
3. bake

Exercise H4: Lists

1. load `H0-starter.html`, and save it as `H4-lists.html`
2. add a level-1 heading "A World of Lists"
3. create an ordered list with your four favorite bands
4. create an unordered list with four cities you have visited
5. create a definition list with three "hip" words and their meanings
6. create a level-2 heading for each list that describes what is in the list
7. as always, validate your page

Images

- the `<img>` tag is used for displaying images
- it is considered a "self-closing" or "empty" tag because there is no end tag

```
<img src="frog.jpg" width="100" height="200" alt="my pet">
```

Attribute	Description
`src`	the filename of the image
`width`	the image width (in pixels)
`height`	the image height (in pixels)
`alt`	alternate text, to be shown in place of image

Images

- accepted image formats are GIF, JPEG and PNG
- although the width/height are not required, it is *highly* recommended to supply them
- the `alt` attribute should give a description of what the image is
- be sure you respect the intellectual property rights of the image's owner!

Optimizing Images

- images can be very large files and take a long time to download
- therefore you should always *optimize* the images you use
  - crop them to remove parts that aren't important
  - save them in "web-optimized" format (i.e. smaller filesize)

Character Entities

Use *character entities* to specify special HTML characters. If you don't, you may confuse the web browser.

Entity	Character	Description
&quot;	"	double quote
&amp;	&	ampersand
&lt;	<	left angle bracket (less than)
&gt;	>	right angle bracket (greater than)

More Character Entities

Also use them for characters not found on your keyboard.

Entity	Character	Description
&uuml;	ü	'u' with umlaut
&oacute;	ó	'o' with acute accent
&copy;	©	copyright symbol

See some more, located here.

Character Entity Examples

This HTML...

```
&iquest;Porqu&eacute;? RudolfSoft&trade; &Aring;lesund
```

...will be displayed in the browser like this

```
¿Porqué? RudolfSoft™ Ålesund
```

This HTML...

```
Emphasis tag use: It is &lt;em&gt;really&lt;/em&gt; important.
```

...will be displayed in the browser like this

```
Emphasis tag use: It is <em>really</em> important.
```

Exercise H5: Images and Entities

1. make a copy of your file `H1-structure.html`, and call it `H5-img-ent.html`
2. load this image into your browser, and drag it from the browser to your exercise folder
3. some browsers will show the image dimensions; if not, do a "get info" on the file to determine its width and height
4. add the image after the first paragraph in the file, and supply all four image attributes
5. add a new paragraph after the image, and use character entities to show the syntax of the <a> tag
6. add a sentence that uses some other character entities from the entities page, either to write some foreign words with accents, or trademark, or currency symbols... you are allowed to use your imagination ;-)
7. validate your page

Semantic Elements

- *semantic elements* tell other programs what the different parts of your document are
- the rules for their use is "flexible"
- more info on the w3schools.com Semantic Elements page

Element	Description
`article`	self-contained piece(s) of content, typically with title(s)
`aside`	content related to the article (like related links)
`nav`	a section with navigational links
`main`	the main content of the `<body>`
`header`	the title, teaser and author of an article or page
`footer`	the info at the bottom of every page

Semantic Element Example

```html
<!-- This example uses only some of the semantic elements. -->
<body>
    <header>
        <div id="site-name">Jim's Web Site</div>
        <nav>
            <ul>
                <li><a href="index.html">Home</a></li>
                <li><a href="about.html">About</a></li>
                <li><a href="contact.html">Contact</a></li>
            </ul>
        </nav>
    </header>
    <main>
        <article>
            <h1>This is the Document Title</h1>
            <p>And this is the (limited) document content.</p>
```

Exercise H6: Semantic Elements (Optional)

1. load `H6-semantic.html`, and save it in your exercise folder
2. add the following semantic tags to the document: `main`, `article`, `header`, `footer`, `nav`
3. each blog entry summary (identified by a level-2 heading) should be wrapped in an article tag
4. the entry summaries and the level-1 heading constitute the main content
5. validate your page