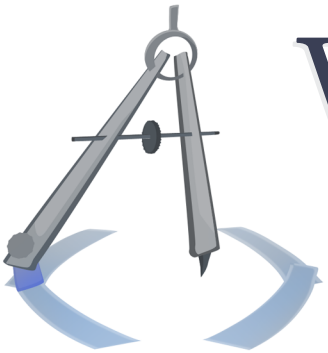


Website Architecture

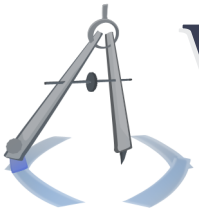
Lezione 1

HTML

An incomplete but intuitive walk thorugh HTML, especially as it is relevant to a site architect. More HTML features will be explored in future lectures as the HTML features correspond to our other technologies.

Michael Serritella

Summer 2010



Website Architecture

Lezione 1 | HTML

Intro to HTML

HTML is a way of easily "marking up" plain text using other plain text, so you can add formatting, like bold, italic, etc.

How do you mark up text? Surround it in a **tag**, like so[1]:

```HTML
<tagName>Hello</tagName>
```

```HTML
<bold>This would be bold</bold>
  <italic>(if HTML were a well-designed language)</italic>
```

Tags can also have **attribtues**, like this:

```HTML
<font family="sans-serif" size="4">This is large
  and <italic>without</italic> serif</font>
```

These were the initial types of motivating examples that influenced HTML's design. As HTML grew to provide more features, these same syntactic tools were used.

[1] None of this code actually works, but that's fine.

HTML can also describe heirarchical structure, since tags can be nested:

HTML

```
<someParentTag>

  <someChildTag>
    <italic>Italic (grand)child</italic>
  </someChildTag>

  <someChildTag>
    <bold>Bold (grand)child</bold>
  </someChildTag>

</someParentTag>
```

So, HTML can describe both the structure and appearance of a document (in addition to containing the document's content itself). In recent years, we realize that we want to divorce these two functions and specialize HTML to only describe structure (and content). Therefore, in this lesson, we won't focus on HTML's text formatting abilities.

Most common tags

Let us look at some example tags to broaden our scope:

- links (`<a>`)
- images (`<img>`)
- lists (`<ul>`, `<ol>`)
- tables (`<table>`, `<tr>`, `<td>`)
- headings (`<h1>`, `<h2>`, …)
- paragraphs (`<p>`)
- line breaks (`<br>`)

Links

To make an element - text or otherwise - a link to another page, we surround it in the `<a>` ("anchor") tag:

HTML

```
You're not really supposed to say
  <a href="../otherDirectory/otherPage.html">"Click Here"</a>
    anymore.
```


Title

Like other tags, it may have another helpful attribute: `title`.

HTML

```
Hey guess what
  <a href="otherPage.html"
     title="An Actually-Descriptive Title">click</a>
```

This will appear in a small box upon hovering over the link with your mouse.

Anchors

It is also possible to link to another area on the same page (e.g. jump down to a FAQ item). In order to do that, an empty-tag anchor has to be declared for the destination, which is given a name. Then other anchors can refer to it using a special syntax:

HTML

```
<a href="#FAQ12">I was wondering if..</a>
...
<a id="FAQ12"></a>No, you weren't. In the past progressive?
You were holding your chin for a while with great purpose? What.
```


And from another page:

```HTML
<a href="http://www.JustTheFAQs.com/FAQ3367.html#FAQ12">(see FAQ)</a>
```

Images

How to add an image? An image is sort of strange in that it is an element with only one tag, not an opening and closing tag. So (in earlier HTML), we just omit the end tag.

```HTML
My picture: <img src="images/Me.jpg">
```

An image may naturally have other attributes that we would like to encode in HTML:

```HTML
My picture: <img src="images/Me.jpg"
                 alt="A picture of me by a tree"
                 title="Those Summer Winds.."
                 width="160"
                 height="200">
```

The `alt` attribute is what is printed instead of the image("alternate

text") if it cannot be loaded or if the user is using a screen reader. In olden days, this was the mouse-over text.

Lists

Where would presentations be without lists?

- bam
- boom
- boring

This list is an *unordered* list - without item numbers - so we use `<ul>`.

HTML

```
<ul>
  <li>bam</li>
  <li>boom</li>
  <li>boring</li>
</ul>
```

Each *list item* (yeah) can contain any other HTML elements, practically speaking. The `<ul>`-to-`<li>` area is a dead zone.

Using `<ol>` instead of `<ul>` will automatically number the items; the code is otherwise the same. You can do other tricks such

as this, though[2]:

```HTML
<ol start="3">
  <li>three</li>
  <li>four</li>
  <li>five</li>
</ol>
```

Tables

Tables are inteneded for presentation of tabular data, but they are widely and pervasively abused for other purposes. They generally go like this:

[2]Deprecated

HTML

```
<table>
  <tr>
    <td>Student Name</td>
    <td>HTML Proficiency</td>
    <td>JavaScript Proficiency</td>
    <td>Passionately wants this opportunity for an elective</td>
  </tr>
  <tr>
    <td>John Smith</td>
    <td>2</td>
    <td>1</td>
    <td>Yes</td>
  </tr>
  <tr>
    <td>Maria McJones</td>
    <td>3</td>
    <td>0</td>
    <td>Yes</td>
  </tr>
</table>
```

The `<tr>` and `<td>` are effectively *table row* and *table cell*. In between are dead zones.

Tables actually allow for a lot of attributes that control the formatting of the contents of the table, including lots of width, height, and text-position controls. In the old days, these features were hijacked to build entire site layouts. This is ugly,

difficult to maintain, and breaks semantics, but you will unfortunately see it all the time.

For instance, check out this comprehensive example[3]:

[3]Of old-style code

HTML

```
<table width="80%" align="center">
  <tr>
    <td width="5%"></td> <!-- Padding column;
                              width spec. affects subsequent rows -->
    <td width="220"></td> <!-- ^^ > These are comments, btw -->
    <td></td> <!-- This column <= will have elastic width -->
  </tr>
  <tr>
    <td></td>
    <!-- I know, spanning is strange -->
    <td rowspan="3" align="center">
      <img src="users/Maria/ProfilePhoto.jpg"
           title="Maria McJones"
           border="1">
    </td>
    <td><font size="3" color="#550000"><b>Maria McJones</b></font></td>
  </tr>
  <tr>
    <td></td>
    <!-- Other one still spanning -->
    <td><a href="users/Maria/Profile.html">View Profile</a></td>
  </tr>
  <tr>
    <td></td>
    <!-- Other one still spanning (last time) -->
    <td><i>Likes:</i> Electives</td>
  </tr>
  <tr>
    <td></td>
    <td></td>
    <td></td>
  </tr>
</table>
```


It gets much worse than this, including the case of nested tables. Imagine dividing your desired layout into a grid of unit-size rectangles and then merging (spanning) rows and columns (`colspan`) as necessary until you have the grid you want. Or dividing them via nested tables.

This is how people used to make layouts. Some still do, but it is your duty as computer scientists to smite them. We will see far more intuitive (though, sadly, not always as effective) ways of designing layouts in later lessons.

Headings

The heading tags are pretty simple. Observe:

HTML

```
<h1>A large heading for the page</h1>
Some text..

  <h2>A slightly smaller heading, for a section</h2>
  Some less important text..

  <h3>Smaller..</h3>
```

Headings are a text-formatting feature, so you may think that

they would be deprecated. But they also describe the structure of the document, and their use adds to the semantic clarity of the code. For instance, search engines will know which parts of your page are main topics. So will people. So you should use 'em.

Paragraphs

Really, let's just amend the previous example:

HTML

```
<h1>A large heading for the page</h1>
<p>Some text..</p>

  <h2>A slightly smaller heading, for a section</h2>
  <p>Some less important text..</p>

  <h3>Smaller..</h3>
```

Paragraphs are very often seen with headings. They are useful for the same semantic reasons. They offer some slight formatting benefits, such as adding a line break after `</p>` and a generous amount of space until the next element. For semantics, you should still use 'em.

Line breaks

Now begins our discussion of whitespace in HTML. Whitespace is consumed and ignored in HTML in ways similar to other markup languages: Any amount of whitespace is considered to be one space (' '). If you want anything more than that, you have to specify. If the space exists within a tag, like some of the earlier examples with several attributes across several lines, then the space is inconsequential.

A line break is a way to explicitly write a newline. Like an image, it is a single tag.

HTML

```
Poetry<br>is<br>harder than<br>programming<br><i>lulz</i>
```

HTML document structure

HTML documents, on the whole, are expected to have a certain tag structure. There is a head and a body to the document. All of the previously-seen tags would go in the body. Use this example to get the gist:

HTML

```
<html>
  <head>
    <!-- This appears in the browser title bar and in bookmarks -->
    <title>My Dissertation</title>
    <!-- Search engines used to care about these until the world
         realized they were all BS -->
    <meta name="keywords"
          content="poetry Michio Kaku roulette secrets L@@K">
  </head>
  <body>
    Poetry<br>is<br>harder than<br>programming<br><i>lulz</i>
  <body>
</html>
```

This document structure is still not up-to-date with current standards, but the differences are minimal.

Frames

Sometimes, you want to display two HTML documents side by side, to aid in site navigation, or you want one to effectively contain the other one in a small display window. Sometimes, you are in 1998. It happens. But frames are still used, and you should know how they work.

Basic use

The actual code would be too convoluted and large to write here, but we'll describe the basic structure. There is a parent page called a *frameset*. It mainly contains the `<frameset>` tag and `<frame>` tags nested within it. Like `<img>`, the `<frame>` tags specify a URL for the source of the content. Then, each "framed" HTML document can live within a slice of the screen and may be completely ignorant of the frameset. It may be aware of the frameset in some way, however.

Link syntax

Frames can have IDs - names - within the frameset. They may be, for instance, "NavigationFrame" and "ContentFrame". Then a link within each frame may refer to them as follows:

HTML

```
<a href="Multimedia.html"
   alt="Multimedia"
   title="Multimedia Gallery (WMV, MOV, MKV)"
   target="ContentFrame"> <!-- <= boom -->
```

And the content frame may similarly affect the navigation frame.

The `target` attribute has some special possibilities. Here are its standard values, including an oddball:

_blank This opens the link in a new window. It isn't really related to frames.

_self Redundant.

_parent Opens the link in the parent frameset, recognizing the possibility of nested framesets.

_top Opens the link in the browser window, clearing all framesets.

Problems?

Frames are actually kinda deprecated. There are replacements (most notably `<iframe>` - *inline frame*) which obviate many of their problems. Most of their problems come from unintuitive behavior rather than truly broken technology, but they are significant. Let's look at some of those problems:

Scrolling don't work What. The page cannot be scrolled as a whole; each frame must be scrolled individually. In effect, a navigation frame never goes away and always consumes screen space. As we will see, this fixed-size encroachment into the user's space may rely on overbroad assumptions about what is appropriate for all users. Frames can be set to allow resizing and scrolling, but most framed layouts don't do this, and as long as you have frames, it might not make sense to offer these options.

Printing don't work What. When you click 'Print', you may only get the currently-focused frame. This is likely very annoying.

Bookmarking don't work What. When links are clicked within framed pages, the URL in the browser bar generally doesn't change. So if the user navigates to a new page in the aforementioned NavigationFrame and then clicks 'Add Bookmark', the bookmark will save the frameset URL.

Link sharing don't work What. Same thing, man. Same thing.

Back button don't work What. Your 'Back'/'Forward' navigation buttons will usually walk through the most recently traversed links, no matter which frame. You may click the button 8 times and then skip around among 3 or 4 frames. This is simply confusing and may lead to a schism of expectations between the site designer and the user. This may have dire consequences in some rare situations, e.g. "PRESS SUBMIT ONLY ONCE; DO NOT GO BACK; YOU WILL BE CHARGED." It depends on your site architecture as to whether this becomes a problem or whether the user will be tempted to do this, but be wary.

Refresh button don't work What. The 'Refresh' button on the browser refreshes the frameset and then all frames within it. This is almost never what the user wants, and it's almost never what you want.

Special characters in HTML

So.. How do you write less-than and greater-than in HTML? Your computer-science instincts would tell you that you need some kind of escape sequence. In HTML, you would write such characters like this:

HTML

```
Is 5 &lt; 7? Yes.
Is 5 &gt; 7? No.
```

There are many other special characters which follow the same escape sequence. There is a huge list somewhere. Most of them are for extended ASCII and Unicode characters, which don't exactly fit into HTML's initial goal of for-plaintext-by-plaintext markup. Some have names, such as "lt" and "gt", and all have numeric identifiers. For instance, the © symbol can be written like this:

HTML

```
&copy; 2010 Not You
&#169; 2010 Not You
```

The numbers don't correspond to ASCII values (they are Uni-

code), so you should look them up.

There is one strangely common character which must be escaped when in HTML. That is the ampersand ("&"):

```HTML
Wang &amp; Chang
```

It's because you must escape the escape!

A particularly interesting character is the *non-breaking space*, written ** **. This whitespace is not consumed by HTML's parser, and thus you can actually put several nbsp's in a row and expect several spaces to appear on screen.