

# CS 2316 Data Manipulation for Engineers

## HTML

Christopher Simpkins

`chris.simpkins@gatech.edu`

# Hello, HTML!

```
<!doctype html>
<html>
  <head>
    <title>Hello, World!</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

- `html`, `head`, and `body` elements required
- Like XML, opening and closing tags are enclosed in angle brackets
- Everything between an opening a closing tag is the *content* of the element

Follow along by downloading [hello.html](#) and opening it in your browser.

# HTML Document Structure

```
<!doctype html>
<html>
  <head>
    <title>Hello, World!</title>
  </head>
  <body>
    <h1>Level One Heading (H1)</h1>
    <p>This is paragraph text under the first heading.</p>
    <h2>Level Two Heading (H2)</h2>
    <p>Chris Simpkins, Director General of the Royal British Legion</p>
    
    <p>Got this picture from <a href="http://goo.gl/uoaBmP">this
page</a>.</p>
  </body>
</html>
```

- The `head` element contains metadata, like title, JavaScript and CSS file references, and so on (we won't bother with these in this course)
- The `body` element contains the stuff that's rendered to the screen by the browser

# Headings

Headings to multiple levels specified with h1, h2, h3, ...

```
<h1>Level One Heading (H1)</h1>  
<p>This is paragraph text under the first heading.</p>  
<h2>Level Two Heading (H2)</h2>  
...
```

Use headings to organize web page content just as you would in a printed document.

# Paragraphs and Images

Paragraph text should be enclosed in a `p` element:

```
<p>This is paragraph text.</p>
```

Images are loaded with an `img` element:

```

```

We can also load images locally:

```

```

And specify their size

```

```

Try these out by modifying [hello.html](#).

# Anchors

Anchor elements, `a`, serve many purposes. A few examples:

- Name part of a page that can be referenced:

```
<a name="top" />
```

- Create an internal hyperlink to a named part of a page:

```
<p>Back to <a href="#top">top</a></p>
```

- Create an external hyperlink that loads an external web page:

```
<p>Got this picture from <a href="http://goo.gl/uoabmP">this  
page</a>.</p>
```

- Create a `mailto` link:

```
<p>Send email to <a href="mailto:bob@aol.com">Bob</a></p>
```

Each of these is demonstrated in [hello.html](#).

# Uniform Resource Locators (URLs)

Consider the followign URL:

```
http://www.gatech.edu/
```

- `http://` means "use the http protocol"
- `www.gatech.edu` is the server on which the resource is located
- The trailing slash after `www.gatech.edu` means "load the index, or default page" (which may be generated by server program or may be a file such as `index.html`)

You can also load files from your local filesystem, as we've been doing with [hello.html](#):

```
file:///Users/chris/work/vcs/github/data-python/code/web/hello.html
```

and use other network protocols, like File Transfer Protocol (FTP):

```
ftp://releases.ubuntu.com/releases/14.04
```

# URL Shortening

Notice the URLs in [hello.html](#) that look like:

```
http://goo.gl/uoaBmP
```

This is a shortened URL. The full address that this redirects to is:

```
http://www.lboro.ac.uk/service/publicity/news-releases/2011/157_Hon-Degs-w
```

Shortened URLs are more convenient to work with in code, and essential for services like [Twitter](#). Create your own shortened URLs at <http://goo.gl> (or any of the many other URL shortening services).



# Lists

Unordered lists are created with `ul` elements:

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
```

Ordered lists are created with `ol` elements:

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
</ol>
```

Notice that in both cases list items are enclosed in `li` elements.

# Tables

Tables are created with `table` elements:

```
<table>
  <tr>
    <th>Col1</th><th>Col2</th><th>Col3</th>
  </tr>
  <tr>
    <td>1, 1</td><td>1, 2</td><td>1, 3</td>
  </tr>
  <tr>
    <td>2, 1</td><td>2, 2</td><td>2, 3</td>
  </tr>
</table>
```

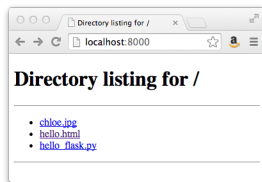
- Table rows are enclosed in `tr` elements
- Table cells within rows are created with `td` elements, or `th` elements for the header row

# Static Web Sites

We've been loading HTML files directly in the browser. Python provides a simple static web server. In the directory containing your [hello.html](#) file, launch Python's simple built-in web server with:

```
$ python3 -m http.server
```

Now go to your browser and load <http://localhost:8000/>. You should see something like this:



If you had an `index.html` file, it would be rendered instead. Create an `index.html` file in this directory and hit "refresh" on your browser.

# Beyond HTML

We've only scratched the surface of HTML web pages

- What you've learned here is all you need for this course, and for basic web page creation
- Modern web pages also incorporate JavaScript and Cascading Style Sheets (CSS)
- Learn more at [Mozilla Developer Network's HTML Documentation](#)