Hypertext Markup Language
HTML
Chapter 2

Supporting Material for “Using Google App Engine” - O’Reilly and Associates
www.appenginelearn.com
Internet

HTML
CSS
JavaScript
AJAX

HTTP Request
GET
POST

Python Templates
Data Store
memcache
The big picture...

A web server produces HTML which is handed to a browser which needs to lay it out in a blink of an eye and have it pixel perfect as god as a print brochure.
HTML

- A way of marking up text to indicate that some text is different than other text

- We “tag” portions of the text to communicate meaning

<h1>Google App Engine: About</h1>
<p>Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.</p>
Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.
Learning from “View Source”
Evolution of HTML
The Web is a Young Technology

- Invented in early 1990’s
- Popular in 1994
- Robert Cailliau - coFounder of the World-Wide-Web

http://www.dr-chuck.com/media.php?id=70
HTML has evolved a *lot* over the years - as computers and networks have gotten faster.
An Iterative Process

- Designers want to do something
- They figure out how to do it with current generation browsers
- We figure out requirements - but HTML gets uglier and quirkier - then the HTML starts to break
- We develop new standards that handle new requirements in an elegant manner
- New Browsers arrive in the marketplace with the new standard
History of HTML / CSS

- HTML 1.0 - 1993 - The Good Old Days - life was simple
- HTML 2.0 - 1995 - Some interesting layout features - abused
- CSS 1 - 1996
- HTML 3.2 - 1997
- HTML 4.0 - 1997 - Layout moving toward CSS
- CSS Level 2 - 1998
- HTML 4.01 - 1999 - What we use today

HTML has evolved a *lot* over the years - as computers and networks have gotten faster.
The Good Old Days

In the good old days you wrote HTML and browsers displayed it - since we wrote it by hand - and modems were slow - it was never too long and never too complex. The browser was never the rate limiting factor.

Writing HTML was like using a simple, weak word processor. The tags acted as formatting commands to the browser.
The Ugly Middle Ages

- Web Designers designed to browser capabilities - down to particular minor versions of browsers.

- Extensive testing was needed on lots of browsers

- Designers used tables, nested tables, and chopped up graphics to gain control of the look and feel or web pages to produce a “print-like” experience.

- HTML was UGLY, Hard to develop, and brittle - what looked superb on one browser - often was broken on another browser - even a later release of the same browser.
The Modern Era

- HTML is clean and simple
- There is no presentation in HTML - font, color, spacing, etc etc
- No use of tables except for tabular data
- CSS controls all layout, and look and feel
- Still a bit challenging - but converging
<table>
<thead>
<tr>
<th>Year Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1990-1994</td>
<td>HTML was simple and pages looked pretty ugly.</td>
</tr>
<tr>
<td>1995-1999</td>
<td>HTML became more complex and each browser was different.</td>
</tr>
<tr>
<td>2000-2005</td>
<td>Browsers slowly supported CSS to varying levels. HTML was still ugly to support multiple browsers.</td>
</tr>
<tr>
<td>2005-2008</td>
<td>New browsers supported CSS. Old browsers were still pretty pervasive but diminishing.</td>
</tr>
<tr>
<td>2009+</td>
<td>The last “pre-CSS” browser (IE5) is &lt; 0.1%</td>
</tr>
</tbody>
</table>

http://www.w3schools.com/browsers/browsers_stats.asp
What does this mean for us?

- Don’t bother with the intermediate steps :)
- Either keep it simple - or do it well - simple does work
- If you want a professional site use all of the best practices
  - Presentation in CSS + Semantic markup in HTML
  - [http://validator.w3.org/](http://validator.w3.org/)
  - [http://jigsaw.w3.org/css-validator/](http://jigsaw.w3.org/css-validator/)
Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.
Tags have a beginning and end...

<h1>Google App Engine: About</h1>
<p>Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.</p>
### HTML Tag Basics

Tags “mark up” the HTML document. The tags are read and interpreted by the browser - but now shown.

A self closing tag does not need a corresponding end tag.
What about <‘s in HTML?

<h1>HTML: Special Characters</h1>
<p>Special characters are indicated by the & character. We can use this to display &lt; and &gt;</p>

<table>
<thead>
<tr>
<th>Entity</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
</tr>
<tr>
<td>&amp;</td>
<td>&amp;</td>
</tr>
<tr>
<td>£</td>
<td>£</td>
</tr>
<tr>
<td>©</td>
<td>©</td>
</tr>
<tr>
<td>™</td>
<td>™</td>
</tr>
</tbody>
</table>
HTML Links

• One of the key things about HTML is making a set of pages and making “hypertext” links amongst those pages.

• Links are what make the “web” a “web” - it is a web of interlinked documents.

• The interlinked nature of the web leads to the “knowledge” the search engines like Google appear to have.
A link is a “hot spot” on the page. It can be text or an image. Often it is visually marked to make it easier to “notice” so as to encourage users to click!

“a” is short for “anchor” and “href” is short for “hypertext reference”
A link is a “hot spot” on the page. It can be text or an image. Often it is visually marked to make it easier to “notice” so as to encourage users to click!

“a” is short for “anchor” and “href” is short for “hypertext reference”
Absolute Reference

<a href="http://www.dr-chuck.com/page2.htm">Second Page</a>

Start tag
Where to go if clicked.
Clickable Text
End tag

Relative Reference

<a href="page1.htm">First Page</a>

Clickable Text
Absolute .vs. Relative

Hypertext references can be a full URL - and refer to some other page on anywhere on the Internet.

Or the reference can be a file name that is assumed to be in the same folder as the current document (relative reference).
Navigation With Anchor Tags

• Sometimes we want to make a menu that provides our users consistent navigation across page.
Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.
Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.
Multiple Files

- We can put multiple files in the same directory and then use them in relative links.

```bash
csev$ ls -l
-rw-r--r-- 1 csev staff 618 Dec 18 22:56 index.htm
-rw-r--r-- 1 csev staff 883 Dec 18 22:57 sites.htm
-rw-r--r-- 1 csev staff 679 Dec 18 22:57 topics.htm
csev$
```
<h1><a href="index.htm">AppEngineLearn</a></h1>
<ul>
  <li><a href="sites.htm">Sites</a></li>
  <li><a href="topics.htm">Topics</a></li>
</ul>

Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.
We create the appearance of a fixed “menu” by including the navigation at the same place in each page.
Special File Names

• When a URL points to a directory in your web server, it looks for a file with a special name:
  • index.html, index.htm, index.php, default.htm, etc..

• While there is a convention, the “default file” is configurable - so nothing is “sure”

• Usually index.htm or index.html is a safe bet

• This only works when viewing through a web server - when viewing from disk, you must view the file.
<body>
  <!-- Make sure to style the h1 -->
  <h1>App Engine: Topics</h1>
  <ul>
    <li>Python Basics</li>
    <li>Python Functions</li>
    <li>Python Objects</li>
    <!-- Leave these two out for a while
    <li>Hello World</li>
    <li>The WebApp Framework</li>
    -->
    <li>Using Templates</li>
  </ul>
</body>
Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.
Images

All information is communicated through the attributes of the img tag.

```
<img src="appengine.jpg"
     width="142" height="109"
     alt="Google App Engine Logo"
     style="float:right" />
```

Put the image on the right and wrap text around it. Which image to display

Optional - makes display quicker. Will resize to fit.

Show this when hovering, images are off, or for screen readers.
Images

<h1>
<img src="appengine.jpg" width="142" height="109"
     alt="Google App Engine Logo"
     style="float:right"/>
Google App Engine: About</h1>
<p>
Welcome to the site dedicated to learning the Google Application Engine.
We hope you find www.appenginelearn.com useful.
</p>

In this case, the file appengine.jpg needs to be in the same directory as the file index.html.
Google App Engine: About

Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.
Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.
HTML Document Structure
Well-Formed HTML Documents

- Browsers need to know what contract you are agreeing to (doctype)
- We need a section for meta-data about the HTML document (head)
- And then the displayable content of the HTML document (body)
Outline of an HTML Document

<!DOCTYPE >  Contract about the HTML style used the page.
<html>  Metadata about the page.
<head>  Displayable content of the page.
  meta-data....
</head>
<body>
  Page content....
</body>
</html>
Outline of an HTML Document

<!DOCTYPE >
<html>
<head>
    meta-data....
</head>
<body>
    Page content....
</body>
</html>

Contract about the HTML style used the page.

Metadata about the page.

Displayable content of the page.
A Simple but Modern Page

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Learning the Google App Engine</title>
</head>
<body>
<h1>Google App Engine: About</h1>
<p>
Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.
</p>
</body>
</html>
Browsers: Lost in Transition

- Between 1995 and 2003 browsers were different - often on purpose to attempt to make their market share “sticky”
- At some point we knew there needed to be a future where all browsers did *exactly* the same thing based on CSS and HTML standards
- How to get from “here” to “there” - no one could “win”
How do we let browsers continue their evil ways while transitioning to standards?
Web developers and browsers are expected to be “perfect.” Responsibility goes both ways.
Designing for the Browsers out there...

<table>
<thead>
<tr>
<th>2009</th>
<th>IE7</th>
<th>IE6</th>
<th>IE8</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
</tr>
</thead>
<tbody>
<tr>
<td>August</td>
<td>15.1%</td>
<td>13.6%</td>
<td>10.6%</td>
<td>47.4%</td>
<td>7.0%</td>
<td>3.3%</td>
<td>2.1%</td>
</tr>
<tr>
<td>July</td>
<td>15.9%</td>
<td>14.4%</td>
<td>9.1%</td>
<td>47.6%</td>
<td>6.5%</td>
<td>3.3%</td>
<td>2.1%</td>
</tr>
</tbody>
</table>

www.w3schools.com/browsers/browsers_stats.asp
You get to make a choice...

• (a) Do you want to be lazy and sloppy and hope that your pages look good across multiple browsers and multiple versions of the same browsers

• (b) Do you want to be professional and have your pages to render identically across all browsers?

• If you choose (b) - you take on some additional responsibility.
DOCTYPE: Indicating your Choice

- Browsers look at the first line of your HTML file to see if you have agreed to comply with HTML standards

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Learning the Google App Engine</title>
</head>
<body>
  <h1>Google App Engine: About</h1>
</body>
</html>
```
Which DOCTYPE?

<!doctype HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<!doctype HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<!doctype HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

Which DOCTYPE?

• Easy to start a fight - just Google “what is the best doctype”

• Loose - legacy

• Transitional - Mix of Developers

• Strict - New Project which can enforce rules from the start

• http://www.webmasterworld.com/html/3061635.htm (and others)
Well-Formed HTML Documents

- Browsers need to know what contract you are agreeing to (doctype)
- We need a section for meta-data about the HTML document
- And then the displayable content of the HTML document
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
 "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>
Learning the Google App Engine</title>
</head>
<body>
<h1>
Google App Engine: About</h1>
<p>
Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.
</p>
</body>
</html>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Learning the Google App Engine</title>
</head>
<body>
<h1>Google App Engine: About</h1>
<p>Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.</p>
</body>
</html>
Checking Validity

• We can check to see if a page “meets the rules” - does it live up to its doctype?

• http://validator.w3.org

• Enter, HTML, upload a file, or enter a web-accessible URL
Note: When the validator fails, only look at the first message and fix that - and then re-run. One simple error will often cascade into 30-40 “The validator is lost and confused errors”.

Success
Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Adobe PageMill 3.0 Win">
<TITLE>dr-chuck.com</TITLE>
</HEAD>
<BODY BGCOLOR="#000000" LINK="#AAAAAA" VLINK="#AAAAAA" ALINK="#AAAAAA">
<table Border=0>
<tr>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>

<title>University of Michigan</title>
</head>
....
Summary

- HTML has gone through many changes and evolutions
  - It started clean and simple - then got ugly and nasty - now we are back to a clean and simple approach
- HTML Markup needs to focus on meaning - not formatting
- Formatting is handled using CSS - Cascading Style Sheets