Unless otherwise noted, the content of this course material is licensed under a Creative Commons Attribution 3.0 License. 
http://creativecommons.org/licenses/by/3.0/.

Copyright © 2009, Charles Severance.

You assume all responsibility for use and potential liability associated with any use of the material. Material contains copyrighted content, used in accordance with U.S. law. Copyright holders of content included in this material should contact open.michigan@umich.edu with any questions, corrections, or clarifications regarding the use of content. The Regents of the University of Michigan do not license the use of third party content posted to this site unless such a license is specifically granted in connection with particular content. Users of content are responsible for their compliance with applicable law. Mention of specific products in this material solely represents the opinion of the speaker and does not represent an endorsement by the University of Michigan. For more information about how to cite these materials visit http://michigan.educommons.net/about/terms-of-use.

Any medical information in this material is intended to inform and educate and is not a tool for self-diagnosis or a replacement for medical evaluation, advice, diagnosis or treatment by a healthcare professional. You should speak to your physician or make an appointment to be seen if you have questions or concerns about this information or your medical condition. Viewer discretion is advised: Material may contain medical images that may be disturbing to some viewers.
CSS

Cascading Style Sheets

Charles Severance

www.dr-chuck.com

http://en.wikipedia.org/wiki/Cascading_Style_Sheets
In the modern era of web design we represent content and meaning in HTML and formatting and layout in CSS.
HTML has evolved a *lot* over the years - as computers and networks have gotten faster.

Source: www.yahoo.com
Everyone needs to know some HTML and some CSS and some programming - but to be truly skilled at a professional level requires deep understanding and specialization.
Transforming the look and feel of a page using a CSS style sheet.

```css
body {
  font-family: arial, san-serif;
}

a, a:link {
  color: #0000cc;
}
...
```
Applying Basic Styles
The Browser has “default styling” for all tags.

<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>

<h2>Google App Engine: About</h2>

<p>Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.</p>
We will apply CSS to the tags in the document.

With no changes to the HTML.
Lots of CSS properties to play with

background-color, border-width, border-color, margin-top, padding, font-family, top, left, right, float, font-size, background-image, text-align, text-decoration, font-style, font-weight, vertical-align, visibility, overflow, ....

We can set these properties on any HTML tag in a document.
# CSS Cheat Sheet

## Shorthand
- `background`<br>- `border`<br>- `border-bottom`<br>- `border-left`<br>- `border-right`<br>- `border-top`<br>- `font`<br>- `height`<br>- `margin`<br>- `padding`<br>

## Comments
```html
/* Comment */
```

## Pseudo Selectors

## Media Types
- `all`<br>- `braille`<br>- `embossed`<br>- `handheld`<br>- `print`<br>- `projection`<br>- `screen`<br>- `speech`<br>- `tty`<br>- `tv`<br>

## Units
- Length ½ em<br>- `pt`<br>

## Syntax

**Syntax**
- Syntax: `selector {property: value;}`
- External Style Sheet:
  ```html
  <link rel="stylesheet" type="text/css" href="style.css" />
  ```
- Internal Style:
  ```html
  @import "css/style.css";
  ```
- Inline Style:
  ```html
  <style type="text/css">
  .selector {property: value;}
  </style>
  ```
- Media Style:
  ```html
  @media only screen and (max-width: 600px) {
  .selector {property: value;}
  }
  ```

## Box Model

**Box Model**
- `height`<br>- `width`<br>- `margin-top`<br>- `margin-right`<br>- `margin-bottom`<br>- `margin-left`<br>- `padding-top`<br>- `padding-right`<br>- `padding-bottom`<br>- `padding-left`<br>

## General

**General**
- `class`: String preceded by a period
- `id`: String preceded by a hash mark
- `div`: Formats structure or block of text
- `span`: Inline formatting
- `color`: Foreground color
- `cursor`: Appearance of the cursor
- `display`: block; none; list-item; none
- `overflow`: How content overflowing its box is handled
- `visibility`: visible; hidden; scroll; auto

## Border

**Border**
- `border-width`: Width of the border
- `border-style`: dashed; dotted; double; groove; inset; outset; ridge; solid; none
- `border-color`: Color of the border

## Position

**Position**
- `clear`: Any floating elements around the element?
- `float`: Floats to a specified side
- `left`: The left position of an element
- `top`: The top position of an element
- `position`: static, relative, absolute
- `z-index`: Element above or below overlapping elements?

## Font

**Font**
- `font-style`: italic; normal
- `font-variant`: normal; small-caps
- `font-weight`: bold; normal; lighter; bolder; integer (100-900)
- `font-size`: Size of the font
- `font-family`: Specific font(s) to be used

## Text

**Text**
- `letter-spacing`: Space between letters
- `line-height`: Vertical distance between baselines

## Background

**Background**
- `background-color`: Background color
- `background-image`: Background image
- `background-position`: Background position

---

Anatomy of a CSS Rule

**Selector** - which part of the document does this rule apply to

**Property** - which aspect of CSS are we changing

**Value** - What are we setting the property to.

```css
body {
  font-family: arial, sans-serif;
  font-size: 100%;
}
```
Multiple tags with same styling

h1, h2, h3 {
  color: yellow;
  background-color: black;
}

Making a noticeable background color is a fun way to debug / identify blocks.
Three ways to add style rules

• Inline Style - Add style information to a tag
• Embedded Style - Add style information to the document at the beginning
• External Style Sheet - Put all of your style in an external file
• Preferred - because two people can work independently
Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Learning the Google App Engine</title>
<style type="text/css">
    body {
        font-family: arial, sans-serif;
    }
</style>
</head>
<body>
<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>
</body>
</html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Learning the Google App Engine</title>
<link type="text/css" rel="stylesheet" href="glike.css">
</head>
<body>
<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>

External Style Sheets

---

glike.css:

body {
  font-family: arial, sans-serif;
}
<head>
<title>Learning the Google App Engine</title>
<link type="text/css" rel="stylesheet" href="glike.css"/>
</head>

We put the CSS file in the same directory so the link works.
Fonts

• Default fonts are ugly and they have Serifs - which make them harder to read on a screen

• So the first thing I usually want to do is override the font in my document

• And I want to do this everywhere.
Fonts

Most Favourite

body {
  font-family: "Trebuchet MS", Helvetica, Arial, sans-serif;
  font-size: x-large;
}

Least Favourite

Fallback fonts: serif, sans-serif, monospace, cursive and fantasy.
Font Factors

- font-size: xx-small, x-small, small, medium, large, x-large, xx-large
- font-weight: bold or normal
- font-style: normal or italic
- text-decoration: none, underline, overline, or line-through
Color Names

- W3C has listed 16 color names that will validate with an HTML validator.
- The color names are: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.

http://www.w3schools.com/html/html_colors.asp
Colors by the number...

Three Numbers, Red, Green, and Blue - each from 00 - FF (Hexidecimal)

#ffffff = white
#000000 = black
#ff0000 = red
#00ff00 = green
#0000ff = blue

Source: http://www.w3schools.com/css/css_colornames.asp
Default Styling for Links

Post-Click:
Hello there my name is Chuck.
Go ahead and click on here.

Downright Ugly!

Source: www.yahoo.com
Styling Links

link - before a visit
visited - after it has been visited
hover - when your mouse is over it but you have not clicked
active - you have clicked it and you have not yet seen the new page

Browser default styling for links is downright ugly!
CSS Tags and Attributes

- As CSS was introduced they introduced two new tags that are pretty much there to serve as handles for styling
  - `<div>` - A block tag (breaks justification)
  - `<span>` - An inline tag that does not break justification
- There are two attributes with special meaning to CSS
  - `id=` - Marks a unique block within the document for styling (use only once)
  - `class=` - Marks a non-unique tag within the document for styling (multi-use)
div as Container

The id attribute on the tag allows us to uniquely mark a div in a document. The id tag is also useful for screen readers.

“div” stands for “division” as it allows us to divide our page into parts or sections and then do something different with each “section”.
Styling a block with “id”

Everything within block

```html
#footer {
  font-style: italic;
  font-family: Times, serif;
}
```

Paragraphs within block

```html
#footer p {
  font-style: italic;
  font-family: Times, serif;
}
```

```html
<div id="footer">
  <p>Please send any comments to csev@umich.edu</p>
</div>
```

id= identifies a *particular* block - only one in a document
Nested divs

<div id="outer">
  <div id="nested1">
    <p>A paragraph inside the first nested div.</p>
  </div>
  <div id="nested2">
    <p>A paragraph inside the second nested div.</p>
  </div>
</div> <!-- End of the outer div -->

Adding divs give us a “handle” to apply styling (CSS) to a block of text.
Paragraphs and Divs

<p>This is a paragraph.</p>
<div>This looks like a paragraph, but it's actually a div.</div>
<p>This is another paragraph.</p>
<div>This is another div.</div>
Styling with class=

```css
.fun {
  color: #339999;
  font-family: Georgia, Times, serif;
  letter-spacing: 0.05em;
}
```

class can be used many times in a document.

```html
<p class="fun">A man walks into a bar; you would've thought he'd see it coming!</p>
<p>Have a nice day.</p>
<p class="fun">More fun stuff</p>
```
Span (Invisible tag)

<p><span class="fun">Bubble Under</span> is a group of diving enthusiasts based in the south-west UK who meet up for diving trips in the summer months when the weather is good and the bacon rolls are flowing. We arrange weekends away as small groups to cut the costs of accommodation and travel and to ensure that everyone gets a trustworthy dive buddy.</p>

Sometimes you want to style something smaller than a whole block - then use span. Do not use span if you are applying something to a whole block - just put your styling on the enclosing block tag.
When building HTML, we use id and class to add little “handles” in the HTML to make it so we can “style” areas of the document.

Pick div id’s to indicate meaning.

This course is a survey course covering a broad range of technology topics at a high level. The course is aimed at students with no prior technical skills other than the general use of a computer. Really!
# CSS CHEAT SHEET

## Shorthand

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Internal Style</th>
<th>External Style Sheet</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>background</code></td>
<td><code>selector {property: value;}</code></td>
<td><code>&lt;link rel=&quot;stylesheet&quot; type=&quot;text/css&quot; href=&quot;style.css&quot; /&gt;</code></td>
</tr>
<tr>
<td><code>border</code></td>
<td><code>selector {property: value;}</code></td>
<td><code>&lt;style type=&quot;text/css&quot;&gt;</code></td>
</tr>
<tr>
<td><code>border-top</code></td>
<td><code>selector {property: value;}</code></td>
<td><code>&lt;/style&gt;</code></td>
</tr>
<tr>
<td><code>border-right</code></td>
<td><code>selector {property: value;}</code></td>
<td><code>&lt;/style&gt;</code></td>
</tr>
<tr>
<td><code>border-bottom</code></td>
<td><code>selector {property: value;}</code></td>
<td><code>&lt;/style&gt;</code></td>
</tr>
<tr>
<td><code>border-left</code></td>
<td><code>selector {property: value;}</code></td>
<td><code>&lt;/style&gt;</code></td>
</tr>
</tbody>
</table>

## Comments

// Comment

## Pseudo Selectors

<table>
<thead>
<tr>
<th>Class</th>
<th>ID</th>
<th>div</th>
</tr>
</thead>
<tbody>
<tr>
<td>String preceded by a period</td>
<td>String preceded by a hash mark</td>
<td>Format structure or block of text</td>
</tr>
</tbody>
</table>

## Media Types

- all
- braille
- embossed
- handheld
- print
- projection
- speech
- tty
- tv

## Units

- Length
- 1em
- 3pt

## Syntax

<table>
<thead>
<tr>
<th>property</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>width</td>
<td></td>
</tr>
<tr>
<td>height</td>
<td></td>
</tr>
<tr>
<td>margin-top</td>
<td></td>
</tr>
<tr>
<td>margin-right</td>
<td></td>
</tr>
<tr>
<td>margin-bottom</td>
<td></td>
</tr>
<tr>
<td>margin-left</td>
<td></td>
</tr>
<tr>
<td>padding-top</td>
<td></td>
</tr>
<tr>
<td>padding-right</td>
<td></td>
</tr>
<tr>
<td>padding-bottom</td>
<td></td>
</tr>
<tr>
<td>padding-left</td>
<td></td>
</tr>
</tbody>
</table>

## Box Model

- Width
- Height
- Margin
- Border
- Padding

## General

<table>
<thead>
<tr>
<th>border-color</th>
<th>Width of the border</th>
</tr>
</thead>
<tbody>
<tr>
<td>border-style</td>
<td>dashed, dotted, double, groove, inset, outset, ridge, solid, none</td>
</tr>
<tr>
<td>border-width</td>
<td>Width of the border</td>
</tr>
</tbody>
</table>

## Position

<table>
<thead>
<tr>
<th>clear</th>
<th>Any floating elements around the element?</th>
</tr>
</thead>
<tbody>
<tr>
<td>both</td>
<td>left, right, none</td>
</tr>
</tbody>
</table>

## Font

<table>
<thead>
<tr>
<th>font-size</th>
<th>Size of the font</th>
</tr>
</thead>
<tbody>
<tr>
<td>font-family</td>
<td>Specific font(s) to be used</td>
</tr>
<tr>
<td>font-weight</td>
<td>normal, lighter, normal, bolder, bold, bolder, 100-900</td>
</tr>
<tr>
<td>font-style</td>
<td>italic, normal, small-caps</td>
</tr>
<tr>
<td>font-variant</td>
<td>normal, small-caps</td>
</tr>
</tbody>
</table>

## Text

<table>
<thead>
<tr>
<th>letter-spacing</th>
<th>Space between letters</th>
</tr>
</thead>
<tbody>
<tr>
<td>line-height</td>
<td>Vertical distance between baselines</td>
</tr>
</tbody>
</table>

## Background

| background-color | Background color |
| background-image | Background image |
| background-position | Background position |

A Running Example...
Transform from ugly to fancy with CSS
First: Just work with the tags and fix fonts / colors
<head>
<title>Learning the Google App Engine</title>
<link type="text/css" rel="stylesheet" href="glike.css"/>
</head>
Block Layout
Quick Advertisement - Firefox

• You pretty much need to use Firefox for serious website development

• Important plugins:
  • Web Developer - Chris Pedrick
  • FireBug - Joe Hewitt

http://addons.mozilla.org/
This site contains a bunch of stuff including software, televisions, miscellaneous material. This also includes a number of activities and stories.

Teaching:
- SI 182: Building Applications for Information Environments
- SI 539: Design of Complex Web Sites (Rails)
- SI 543: Introductory Programming (Java) (Login)

See also [RubyLearn](http://www.rubylearn.com) and my raw course podcasts.

Web/Multimedia sites:
- A Film About Brent and His ATV
- How to Install a hardwood floor in 4 minutes!
- Dr. Chuck goes motorcross racing (2007)
- Dr. Chuck goes stock car racing (2002)
- Audition Tape which we sent to TechTV which was rejected
- Making a Simple Cartoon Three short animations from Chuck and Brent
- Nuthin’ But Net A Television show
- More Multimedia Projects...

Software Tools:
- PHP Image software for v300 and Treo-600 SMS
- ClipBoard-2000 - Distance Education Software
- Sync-O-Matic 2000 - Distance Education Software
- Other software Tools

Projects:

Source: www.dr-chuck.com
Two kinds of elements

- **Inline** - affects how text looks
  - `strong`, `span`
- **Block** - Containers that can be laid out
  - Paragraphs, etc
- CSS can change a tag from inline to block

```css
#navigation li {
  display: inline;
}
```
Inline Elements

• Flowed with other text

• span, em, strong, cite, a

• Inline tags can be nested as long as they match

  • <span class="important">cite>Stuff</cite></span>

• Block can contain inline - but inline cannot contain block
Block Level Elements

• Starts on its own line - ends justification and starts a new block

• Can be a container for other elements

• h1 - h6, p, div, blockquote, ul, ol, form

• Blocks can contain other blocks

```
<div id="content">
<p>One</p>
<p>Two</p>
</div>
```
Now we will move things around, add background and borders, etc.
Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.

Mark our major blocks with divs and id attributes.
Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.

Next we will move these blocks around.
#header {
  background-color: #dde;
  border-top: 3px solid #36c;
  height: 100%;
  overflow:hidden;
}

#header h1 {
  font-size: 20px;
  float: left;
  vertical-align: middle;
}

#header li {
  font-size: 14px;
  display: inline;
}

#header ul {
  list-style: none;
  float:right;
  vertical-align: middle;
}
Now let's do some "tweaking"
CSS Box Model

• **height** and **width** properties size the block element

• **margin** properties define the space around the block element

• **border** properties define the borders around a block element

• **padding** properties define the space between the element border and the element content

• **background** properties allow you to control the background color of an element, set an image as the background, repeat a background image vertically or horizontally, and position an image on a page

http://reference.sitepoint.com/css/boxmodel
The Box Model

I am trapped in a glass case of emotion which is 100px high and 200px wide.
Border, padding, and margin are additive.
#header {
  background-color: #dde;
  border-top: 3px solid #36c;
  height: 100%;
  overflow: hidden;
  padding: 7px;
  margin-top: 5px;
}

#header h1 {
  font-size: 20px;
  float: left;
  vertical-align: middle;
  margin: 0;
  padding: 0 0 0 .3em;
}

#header li {
  font-size: 14px;
  display: inline;
  padding: .5em;
}

#header ul {
  list-style: none;
  text-align: right;
  float: right;
  vertical-align: middle;
  margin: 0;
  padding: 0;
}

top, right, bottom, left
<div id="header">
  <h1><a href="index.htm">AppEngineLearn</a></h1>
  <ul>
    <li><a href="sites.htm" class="selected">Sites</a></li>
    <li><a href="topics.htm">Topics</a></li>
  </ul>
</div>
Transforming the look and feel of a page using a CSS style sheet.
CSS Validation

- You can validate your CSS to make sure it has no syntax errors
- Browsers will generally quietly ignore bad CSS syntax
- http://jigsaw.w3.org/css-validator
- The validator can save you time and sanity

Source: W3C http://validator.w3.org/check
Zen Garden

- A social site where CSS designers show their “stuff”
- The HTML never changes - only the CSS
- Inspiration for us designers

Source: http://www.csszengarden.com/
Summary

- CSS Layout is its own art and science
- CSS Basics are well established and well supported in all modern browsers
- The box model is pretty straightforward - and allows nice design within the standards with reasonable effort levels.
- Site layout and markup is further evolving - mostly to make it increasingly possible to support desktop like experiences on the web.
- These innovations will naturally cause incompatibilities - which make things interesting and frustrating at times.