HTML5 and CSS3

Web Coding for Designers

Lesson 1: Semantic HTML

Michael Slater, CEO
Lisa Irwin, Sr. Developer
course-support@webvanta.com
888.670.6793

www.webvanta.com

Welcome!

- Four sessions
  - 1: Semantic HTML
  - 2: Styling with CSS
  - 3: Putting it All Together
  - 4: The Cool New Stuff
- Homework after each lesson
- Discussion group for questions

CSS3

Questions

- Ask questions via chat during the live course
  - Second instructor standing by
- Join the discussion group
  - **http://groups.google.com/group/webvanta-html5-css3**
  - **webvanta-html5-css3@googlegroups.com**
- Email us
  - course-support@webvanta.com
  - *Please use the discussion group* for non-private questions

Michael Slater
Cofounder & CEO

Lisa Irwin
Web Developer

Goals for This Course

- Demystify HTML and CSS
- Provide simple approach to creating good code
  - Clear away the cruft and superstition
- Give a solid foundation in HTML markup
- Provide a well-grounded introduction to CSS
- Establish a strong foundation for either
  - Coding your own sites entirely (after more learning...)
  - Designing sites visually and working efficiently with developers who build them for you

Testing Your Code

- The only way to learn is through lots of practice
- You want to be in complete control, and be able to iterate quickly
  - Use a simple text editor
    - ***Not*** Dreamweaver or other visual editor
  - View files locally using your browser
  - Use Firebug or Chrome Dev Tools to inspect and modify your pages

Code Editors

- Use an editor that provides HTML syntax highlighting (makes it easier to spot mistakes)
- Autocomplete or menu-driven markup *might* be helpful
- Good Mac solutions
  - TextMate, BBEdit, Coda, Espresso, Sublime Text
- Good Windows solutions
  - Sublime Text, Ultraedit, HTML-Kit, EditPlus, TextPad

```
<html lang="en">
  <head>
  </head>
  <body>
    <h1>Headline</h1>
    <p>Body text</p>
  </body>
</html>
```

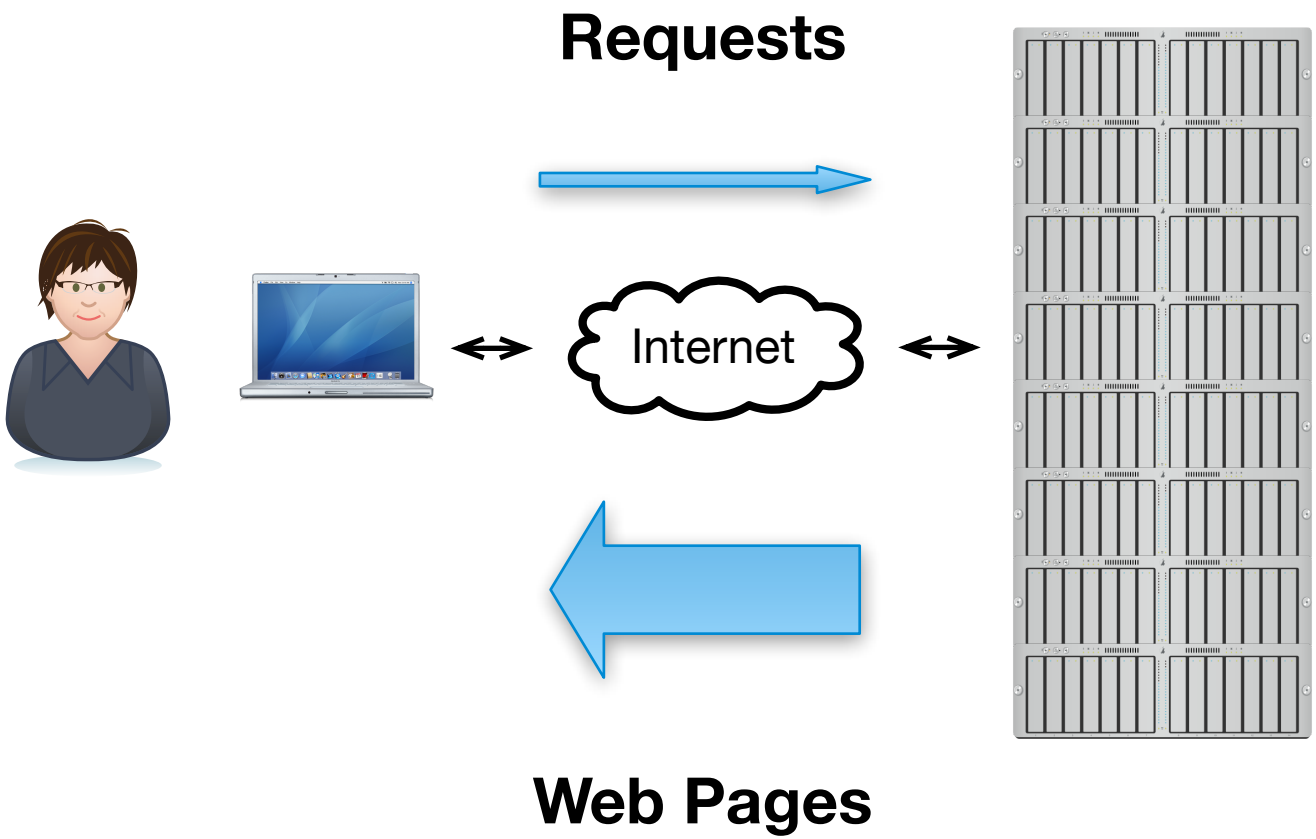
Week 1: Semantic HTML

- **A brief history of HTML**, and why you can ignore most of the junk in the code that you see in common code examples
- **The small set of HTML tags** that you need to know, for headings, paragraphs, lists, images, and forms
- **A simple, effective approach** to marking up content

Our Approach This Lesson

- Move quickly through a lot of material
  - Some will be review for many of you
- Cover the major topics
  - Not possible to cover all the details in a course of this length
- Show only the simplest, most up-to-date, standards-based approach
- Establish a strong foundation for ongoing learning
- Eliminate mystery and magic

How the Web Works

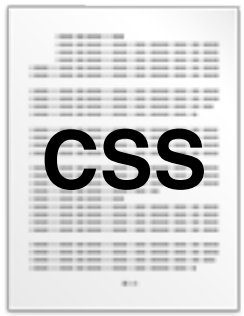


Web Page Components

Text and Markup

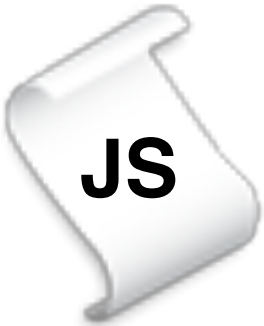
+

Styling



+

Dynamic Behavior



+

Graphics & Photos

HTML

- **H**yper-**T**ext **M**arkup **L**anguage
- The primary native "language" of the browser
  - Markup, not programming
  - CSS, JavaScript, images all brought in through HTML document

```
<div>
    <h1>My Wonderful Website</h2>
    <h2>The Web's Best Source for:</h2>
    <ul>
        <li>Thing one</li>
        <li>Thing two</li>
    </ul>
    <p>View <a href="/specials">specials</a>.</p>
</div>
```

- Originally created to present pure text content, without regard to device or formatting
- Extended with images and molded into a presentation-oriented interactive medium

CSS

- **C**ascading **S**tyle **S**heets
- The **styling** language for the web
  - Takes presentation info out of HTML
- Hooks into HTML through structure of document
- Straightforward for styling of text, creating simple outlines
- More complex for positioning and multicolumn layouts

```
h1 {
        font-size: 36px;
        font-weight: bold;
        font-family: Helvetica, Arial;
}
p {
        font-size: 14px;
}
.wrapper {
        width: 100px;
        margin: 20px auto;
        padding: 30px;
}
```

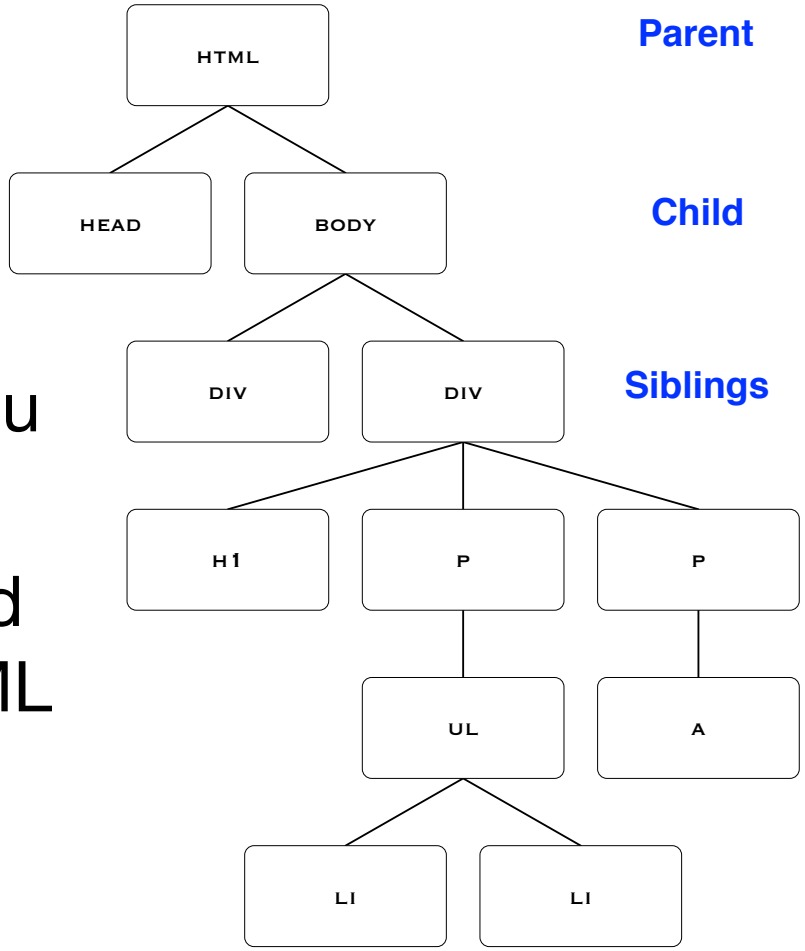
JavaScript

- Programming language understood by (virtually) all browsers
- Key to enabling interactivity
  - Required any time a page changes without requiring a new page to be loaded from server
- Empowers animations
  - The "page" can now include instructions for how to change itself
- Allows new info to be fetched from server and used to modify existing page (Ajax)

```
<script>
    $(document).ready(function(){
      $("#action_button").click(function(event){
          $("#old_thing").hide();
          $("#new_thing").show();
      });
    });
</script>
```

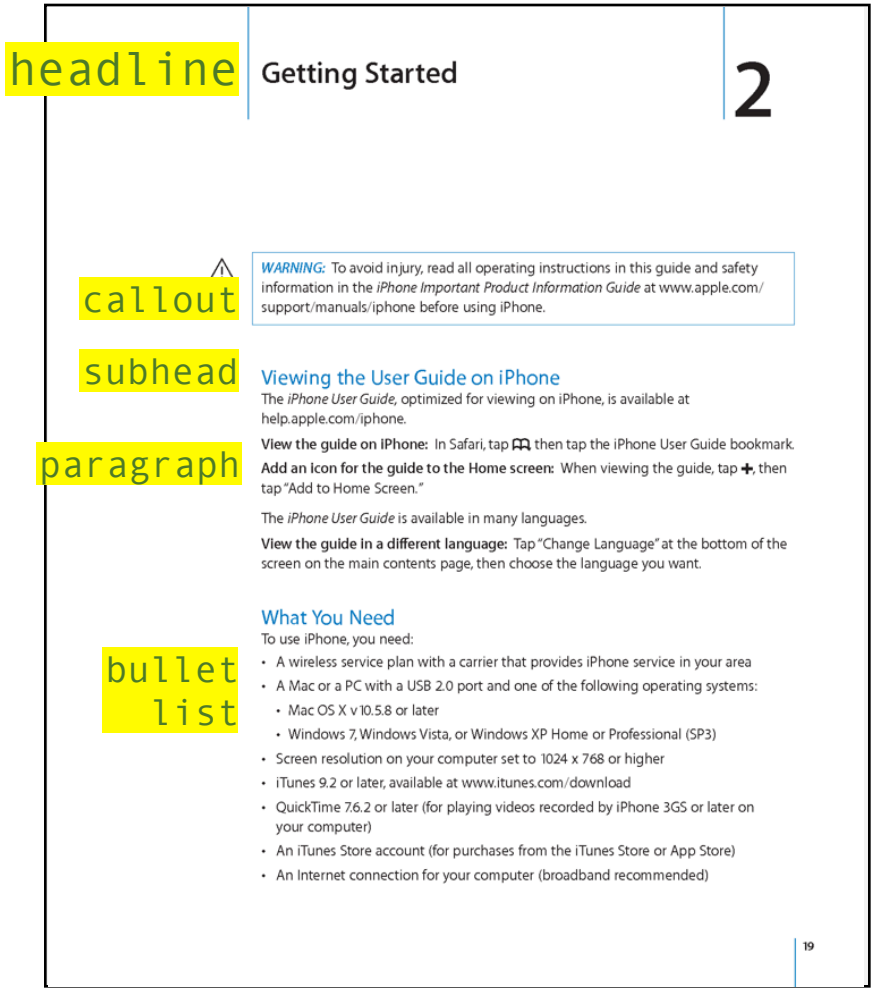
Document Object Model (DOM)

- Abstract representation of an HTML document as a hierarchical structure
- Created implicitly by how you write the HTML document
- Easily referenced in CSS and JavaScript to hook into HTML content
  - The glue that binds all the pieces together



The Essence of Markup

- Identifying the parts of a document
  - Headlines
  - Paragraphs
  - Lists
  - Sections
- Turns a natural-language document into one that makes structural sense to a machine



HTML Versions and Flavors

- Created in 1990-91 by Sir Tim Berners Lee at CERN for sharing research documents
- HTML 2.0, 1995
  - Web starts to take off
- HTML 4.1, 1999
  - Most of today's web
- HTML 5, 2010-2020
  - The near-term future
- XHTML 1.0
  - A dead-end branch, but coding style remains relevant

World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aim
large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, inc
the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
Pointers to the world's online information, subjects , W3 servers, etc.
Help
on the browser you are using
Software Products
A list of W3 project components and their current state. (e.g. Line Mode ,X11 Vio
Mail robot , Library)
Technical
Details of protocols, formats, program internals etc
Bibliography
Paper documentation on W3 and references.
People
A list of some people involved in the project.
History
A summary of the history of the project.
How can I help ?
If you would like to support the web..
Getting code
Getting the code by anonymous FTP , etc.

Browsers and Standards

- W3C sets standards
  - Often based on practices already in wide use (“paving the cowpaths”)
- Browser makers make their own choices
  - Implement parts of standards in process
  - Add proprietary extensions
  - Sometimes make mistakes
- "Web standards" in practice:
  - The least-common denominator of capabilities one can reasonably depend upon

Browser Support

- Displaying exactly the same everyone is not achievable — or desirable
- Write off old browsers
  - No more IE6. Hope to drop IE7 soon.
  - Most Firefox, Chrome, and Safari users keep up with rapid updates to browsers
- Progressive enhancement / graceful degradation
- Need to support browsers without Javascript
  - Robots and screen-readers

Basic Tag Structure

```
<tagname>contents of the element</tagname>

<tagname attribute="value">contents</tagname>
```

Can have any number of attribute/value pairs

```
<tag-that-is-self-closing />
```

Space is optional

Structural Tags

`<div>` – for block-level elements

`<span>` – for inline elements

- New in HTML5 (partial list)

```
<section>
<article>
<nav>
<header>
<footer>
<aside>
```

Basic Text Elements

- Headings

  `<h1>, <h2>, <h3>, ...`

- Paragraphs

  `<p>`

- Unordered (bullet) lists

  `<ul>`

- Ordered (numbered) lists

  `<ol>`

- List items

  `<li>`

A Minimal HTML Document

```
<html>
  <head>
  </head>
  <body>
    <h1>Headline</h1>
    <p>Body text</p>
  </body>
</html>
```

Semantic Markup

- Use elements that match the structural meaning of the content (***not*** how it is presented)
  - Section
  - Article
  - Headline
  - Subhead
  - Paragraph
  - List
  - Table
- Can add additional semantics through microformats and HTML5 microdata (we'll touch on this in Session 4)

Why Bother with Semantics?

- Makes content easier to maintain
- Better for accessibility
  - Use by visually impaired people through screen-readers and other devices
- Gives more information to search engines
  - Better search results, improving over time
- Makes information usable by other websites and programs

IDs and Classes

- Attributes you can add to any HTML element
- Provide names that CSS and JavaScript can use to reference specific elements, or classes of elements

- **IDs** should be unique on a page and allow a single item to to targeted
- **Classes** are used for a "class" of items that (generally) are all treated the same

Using IDs and Classes

```
<div id="headlines">
  <h1>My Wonderful Website</h1>
  <h2>The Web's Best Source for Whatever</h2>
</div>

<div class="bodycopy">
  <h2>My Story About Nothing</h2>
  <p>This is the first paragraph.</p>
  <p>And this is another paragraph.</p>
</div>

<div class="bodycopy">
  <h2>Another Story About Nothing</h2>
  <p>This is the first paragraph.</p>
  <p>This is one last paragraph.</p>
</div>
```

Links

- The "Hyper" in HTML
- Turns text or image into path to another page

```
<a href="http://www.site.com">Click me!</a>

<a href="http://www.site.com"
target="_blank">Open in new window or tab</a>
```

The Wonderful URL

- **U**niform **R**esource **L**ocator
- A key piece of the web's success
  - Reference any file on any computer with a simple string!

`http://`	Protocol
`http://www.mysite.com`	Domain
`http://www.mysite.com/grouping`	Folder
`http://www.mysite.com/grouping/item.type`	File

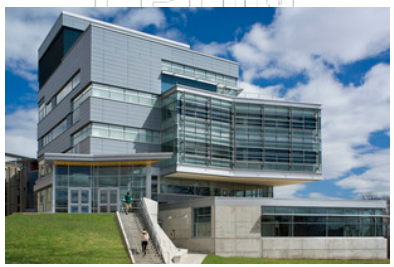
  - Folder/file can be just an abstract reference
  - Type not required for HTML pages

Relative and Absolute References

- Absolute
  - http://www.mysite.com/mypage
  - Always refers to specific item on a specific server, no matter where the reference occurs
- Relative
  - /mypage is relative to the current ***site***
    - Behavior may be different when viewing files locally
  - Assumes same protocol and domain if none specified
  - mypage is relative to the current ***page***

Images


- Visual elements displayed as bit-maps as part of a page
- May be graphics or photos
- Can be part of content or background (via styles)
- Referenced by filename



```
<img src="/images/mypicture.jpg" />
```

Alternate Text

- Used by search engines to determine what the photo is of
- Read by screenreaders for visually impaired users
- Required by standards, but browsers are forgiving

```
<img src="/images/picture.jpg" alt="My Cat" />
```

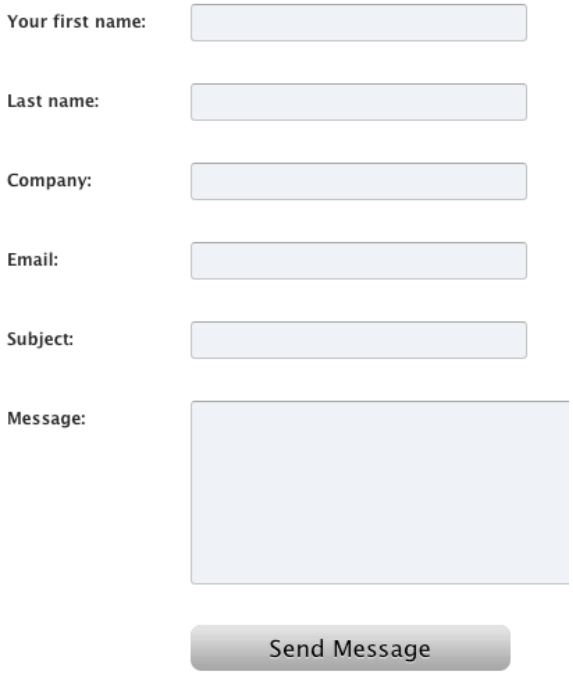
Tables

- Tables can be good, even in a web standards world!
  - Just use them for tabular information, ***not for layout***
  - Browser chooses column widths, given constraints
  - Lots of styling and markup options

```
<table>
  <tr><th>Col 1 Head</th><th>Col 2 Head</th></tr>
  <tr><td>item 2</td><td>item 2</td></tr>
  <tr><td>item 3</td><td>item 3</td></tr>
</table>
```

Forms

- A collection of elements for user input
- Information collected is sent to server when "submit" action occurs
- HTML doesn't specify what happens to the form data, other than to what URL it should be sent
  - Server side software must process the info and generate email, store in a database, or whatever
  - Data can be sent via "post" (not visible in URL) or "get" (regular URL with parameters, e.g. "/form-action?name=John&state=CA")



A Minimal Form

Last Name [ ] Submit Form

```
<form action="/server-process" method="post">
  <label for="last-name">Last Name</label>
  <input type="text" name="last-name" id="last-name" />
  <input type="submit" value="Submit Form" />
</form>
```

Many Form Elements

- Text field
- Text area
- Checkbox
- Radio button
- Select (list)
- New elements in HTML5
  - Number, Email, Date, Range, URL, Color, ... and more
- Often enhanced with JavaScript

And Yet More Elements

- Definition Lists (dl, dt, dd)
- Quotes (blockquote)
- Button (button)
- Image maps (map)
- Horiz. rule (hr) and Break (br)
- Iframe (a page within a page)
- Object (for software elements, such as Flash)
- Canvas

The Head

And now, back to the start of the document

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Title of the Page</title>
    <link href="/mystyles.css" rel="stylesheet" />
    <script src="/myjavascript.js" />
  </head>
  <body>
    (HTML body goes here)
  </body>
</html>
```

Doctypes

- Doctype must be first thing in the HTML file, but it serves only two purposes
  - Keeps the browser from going into "quirks mode"
  - Tells a validator what syntax to check the document against
- You're probably used to seeing things like:

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

- Forget it! Use HTML5

This is all you ever need: `<!DOCTYPE html>`

Metadata

- Provides information about the page that is not, for the most part, directly displayed
- Basic metadata

```
<title>Title of the Page</title>
    <!-- shown in browser tab, critical for SEO -->
<meta name="description" content="The web's best page about kittens" />
    <!-- appears in search result pages -->
<meta name="keywords" ...>
    <!-- irrelevant due to search spam long ago -->
```

Loading CSS and JavaScript

- CSS should always be loaded in the head
- JavaScript should generally be loaded at the end of the body, but may need to be in the head
  - Join our JavaScript/jQuery class!
- Load separate files, or embed in the head section:

```
<style>
  h1 { font-weight:bold; }
</style>

<script>
  (JavaScript code here)
</script>
```

More Head Goodies

```
<!-- specify the character set -->
<meta charset="utf-8">


<!-- small icon to show in browser tab -->
<link rel="icon" href="/favicon.ico" />


<!-- source for associated RSS feed -->
<link rel="alternate" type="application/rss
+xml" title="RSS Feed" href="/feed.xml" />
```

Some Things You've Not Seen
and should continue to avoid

- Presentational tags
  - <font>, <i>, <b>
- Styling attributes
  - height=, width=, size=, color=, border=
- Inline JavaScript event handlers
  - onload=, onclick=
- There's (almost) no reason to use any of this today

Homework

1. **Take any text page** from a magazine or book, or a short piece that you have written.
2. **Create a minimal HTML page** with that content using a simple code editor.
3. **View it in your web browser**.
4. **Repeat** until you get what you expect!
5. Try again with a different sort of document. Search the web or use our reference pages when you aren't sure what element to use.

Help Spread the Word!

- It's not too late for your friends and colleagues to join the course
  - www.webvanta.com/courses
  - Click the share links on each of the course pages
- Help spread the word so we can develop more courses
  - Share via your favorite social media networks
  - Email a friend or a mailing list
  - Write something on your blog
- Thanks!