

## Contents

<b>1</b>	<b>More on Security (0:00–24:00)</b>	<b>2</b>
<b>2</b>	<b>Web Development and HTML (24:00–110:00)</b>	<b>3</b>
2.1	Tags (24:00–42:00) . . . . .	3
2.2	Attributes and Color Values (42:00–55:00) . . . . .	5
2.3	Formatting and Hyperlinks (55:00–60:00) . . . . .	6
2.4	Embedding Video (60:00–70:00) . . . . .	7
2.5	More on Formatting and Links (70:00–102:00) . . . . .	8
2.6	Going Live (102:00–107:00) . . . . .	14
2.7	Images (107:00–108:00) . . . . .	14
2.8	Firebug (108:00–110:00) . . . . .	15

## 1 More on Security (0:00–24:00)

- The headline of today's Metro reads "Tackling Crime Online in Massachusetts." Reading further, we find that Boston was deemed the second riskiest city online. Apparently they are conflating the prevalence of wireless connectivity with the risk of online identity theft. The article highlights the sentencing of a cyber criminal from a few years ago who was sniffing unencrypted or weakly encrypted wireless traffic outside a TJ MAXX. It goes on to suggest that public wireless spots are "not the place to do a lot of shopping." The article concludes with a few tips: 1. ensure websites are secure prior to submitting personal information 2. do your homework to ensure the business or website is legitimate 3. monitor and review credit card statements.
- Reactions to the article:
  - Ultimately, the onus is on the user to protect his credit card and other private information.
  - Pushing back a little, however, it's the site's responsibility to use a minimum level of security like SSL in order to prevent information from being intercepted.
  - Sending in a paper check might be an alternative to using your credit card online, but David argues that checks are probably more insecure considering they have your home address, bank account number, routing number, and sometimes even your social security number printed on them.
  - Why are many public WAPs unencrypted? Logistically, it would be unreasonable to require users to find out the wireless network's password from some authority, especially in very large establishments like airports.
  - The answer, of course, is to only visit encrypted websites when using public wifi. If you log on to a banking website that uses SSL for *all* its pages, it doesn't matter if you're using the relatively insecure medium of public wifi to do so. Of course, if the site uses SSL only for the login page, then a savvy adversary could still hijack your session once you're logged in. Sites like Amazon combat this by asking you to login a second time right before you checkout. As always, security is about raising the bar.
- The source of the survey for riskiest online cities is Norton, a division of Symantec. They rank the top 5 riskiest cities online as follows: Seattle, Boston, Washington D.C., San Francisco, and Raleigh.
- Online credit card theft is probably not as big of a concern as the compromise of an entire database of credit card numbers. There's more money to be made in the latter endeavor and credit card companies these days are generally very good about forgiving charges that you claim are fraudulent.

- A few years ago, there was a company that advertised identify theft protection by publicizing the CEO's social security number and claiming that even he was safe. His identity actually ended up being stolen, which goes to show that keeping certain information private is perhaps the most important security measure one can take. Consider that even taking your computer for a "free tune-up" at a large retail store like Staples isn't really a good idea because it involves granting complete access to your computer to someone you can't necessarily trust. Anyway, most of these free tune-ups, especially on a brand new computer, probably won't have any affect at all on performance.
- There's a sign next to the check-in desk at the hotel David stays at in New York that reads: "BEWARE IDENTITY THEFT WHILE USING WIRELESS." This message is really just to cover them in the event of an incident and so, not surprisingly, the take-away message isn't exactly clear. Should you not use the internet at all? Should you only visit certain sites?
- Question: what can you do to protect yourself when you take your computer in to get repaired? David actually did so recently and rested easy knowing that all of his files were encrypted on his hard drive using File Vault. In order to fix the trackpad, then, he didn't even need to give them access to his hard drive via his username and password. You can implement similar security on a PC using software like PGP and Full Disc Encryption. You could also remove the hard drive entirely or wipe it if you've backed up your data. Of course, if your hard drive is what's being repaired, you'll be forced to share your files with a stranger.

## 2 Web Development and HTML (24:00–110:00)

### 2.1 Tags (24:00–42:00)

- Years ago, David spent an hour in a computer lab with a friend who gave him a crash course in HTML. At the end of the session, he had made his first (albeit hideous) web page. This speaks to the relative ease with which you can dive into web development.
- HTML stands for hypertext mark-up language, meaning it describes how to display the content of the page. To be clear, you can simply write a message like `hi!` and it will be displayed to the user. If you want to make that message bold, however, you'll need to add a tag to it. In the end, all HTML files are nothing but plain text. To make your message display as bold, then, you need to tell the computer to do so using only plain text. In this case, we would write: `<b>hi!</b>`, where the `<b>` is an open tag and the `</b>` is a close tag. A forward slash always indicates a close tag. The open tag tells the computer to start making text bold and the close tag tells the computer to stop making text bold.

- Let's begin by opening the source code for the course website. A number of tags, including `<li>` for list item and `<a>` for anchor or hyperlink. The anchor tags have the form `<a href="...">` where the ... represents the site the hyperlink is pointing to.
- How does this HTML actually get transferred from the server to our computers? Recall that HTTP is the protocol for sending and receiving this HTML data. Don't get HTML and HTTP confused! HTML is the actual content that is transferred via HTTP.
- When you make a request for a website, the server knows which directory to retrieve HTML files from based on the URL that you send in the HTTP headers. If the header reads `GET / HTTP 1.1`, the `/` refers to the default file in the home directory, which is usually `index.html`. There are other default files, including `index.htm`, `index.php` and `index.aspx`. `.htm` is basically interchangeable with `.html`, whereas `.php` and `.aspx` represent other programming languages that actually serve up dynamic content on a page. Each time you visit that page, then, something different might be displayed. Even with these languages, what is sent to the user is an HTML file. In the case of PHP or ASP, the contents of this HTML file may change from time to time.
- What makes the internet fun an interesting is that you don't need any special tools to get started with web development. Once you've learned a few basic rules, you can begin observing how other websites work and teaching yourself more and more. To begin, let's open a single text editor like TextEdit on Mac or Notepad on Windows and write the following:

```
<html>
  <head>

  </head>
  <body>

  </body>
</html>
```

As you can see, we're doing a bit of forward thinking by closing any tags we open so that we don't forget later on. The first tag `<html>` denotes the beginning of the HTML document. Then we specify two parts, the `head` and the `body`, which every website consists of. Note that for readability's sake, we are indenting and spacing out these tags, but the computer will actually understand our code even if there is no whitespace at all between tags.

```
<html>
  <head>
```

```
<title>My First Webpage</title>
</head>
<body>
  Welcome to my home on the Web!
</body>
</html>
```

There are only a few tags that are allowed inside the **head**, one of them being **title**, which is required. In the **body**, we put our actual content which is just a silly message for now.

- If we save this file as **index.html**, we can then access it in our browser simply by dragging the file onto the browser window or possibly by double-clicking. When we do so, we see that the **title** we chose is visible in the tab and at the top of the browser.
- Question: HTML is not a programming language, per se, because it doesn't allow you to execute functions or run commands. The difference will become more clear when we dive into a real programming language in a few weeks.
- Question: it doesn't matter that the open and close tags for **head** and **body** are on different lines whereas the open and close tags for **title** are on the same line. They will be interpreted the same by the computer.

## 2.2 Attributes and Color Values (42:00–55:00)

- Let's get a little more fancy by adding some attributes to our mark-up:

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body bgcolor="green">
    Welcome to my home on the Web!
  </body>
</html>
```

Here, **bgcolor** is an *attribute* of the **body**. The general form for specifying an attribute is **attribute="value"**. You can put spaces around the equals sign if you prefer, though the computer will interpret it the same either way and convention is to have no spaces. You can also use single quotes or double quotes around the value so long as you are consistent throughout the document. Now if we reload the webpage, we see that the background is green!

- Question: the range of values for **bgcolor** is somewhat restricted to well-known colors that you can spell out (red or blue, for example). If you want

to specify other colors, you'll need to give their hexadecimal representation. Recall that we discussed bit depth in regard to colors. Typically, we use 8 bits to represent each of red, green, and blue, so we can write 0 through 255 for each. Hexadecimal is a base system which allows us to succinctly represent large numbers. Whereas in decimal, each digit can only be 0 through 9 for a total of 10 values, in hexadecimal, each digit can be 0 through 9 or A through F, for a total of 16 values (hence the name hexadecimal). If we string two hexadecimal values together, we have 16 possible values for the first and 16 possible values for the second for a total of  $16 * 16 = 256$  values for the both.

Thus, to represent a color in which each of red, green, and blue can take one of 256 values, we need two hexadecimal places for each color for a total of six hexadecimal places. Playing around in Photoshop, we see if we assign the value 255 to red and 0 to each of blue and green, we get a pure red color that can be represented in hexadecimal as 0xFF0000.<sup>1</sup> If we write #0272B2 as the value for `bgcolor`, we'll get a blend of green and blue. It's important to include the # symbol when writing hexadecimal color values in HTML. Google around and you'll find color wheels that display the exact shade you're looking for along with its hexadecimal representation. There's also a Mac application called Color Wheel that allows you to hover over any color on your screen and see its hexadecimal representation. Know that 0x000000 is black and 0xFFFFFFFF is white!

### 2.3 Formatting and Hyperlinks (55:00–60:00)

- Let's try adding a second line of text to our webpage:

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body bgcolor="#FFFFFF">
    Welcome to my home on the Web!

    Goodbye!
  </body>
</html>
```

When we open this page in our browser, however, we see that “Goodbye!” is displayed right next to the sentence beginning with “Welcome.” This is because the computer ignores all whitespace in our HTML document except the very first space.

- If we want to separate these sentences onto two different lines, we need to add the `<br />` line break tag:

---

<sup>1</sup>The 0x is just a convention to denote that the numbers that follow are in hexadecimal.

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body bgcolor="#FFFFFF">
    Welcome to my home on the Web!
    <br />
    <br />
    <br />
    Goodbye!
  </body>
</html>
```

- Next we'll add an anchor tag and link our visitors to Disney's website (our favorite site, naturally):

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body bgcolor="#FFFFFF">
    Welcome to my home on the Web!
    <br />
    <br />
    <br />
    <a href="http://www.disney.com/">Goodbye!</a>
  </body>
</html>
```

Even though our webpage is not live on the internet yet, the link to Disney still works!

## 2.4 Embedding Video (60:00–70:00)

- As you might have guessed, when you watch a video in a web browser, it's not just HTML that is being used to make it play. In fact, it's a technology called Flash made by Adobe. Apple has come under fire lately because devices like the iPhone and iPad which are becoming extremely popular don't support Flash. Competing with Flash will be HTML5, a new version of HTML which will support embedded video without the need for an additional plugin. YouTube actually has a [beta version of their website](#) that uses only HTML5 to play videos. The catch is that because HTML5 is relatively new, it's not supported by many browsers, including Internet Explorer.
- Also, as a side note, technically what we've been coding so far is XHTML, which is basically a flavor of HTML.

## 2.5 More on Formatting and Links (70:00–102:00)

- A common theme in web development is that there are many different ways to do the same thing. As a proof of concept, let's replace the three line break tags from earlier with a single `<p>` paragraph tag:

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body bgcolor="#FFFFFF">
    <p>
      Welcome to my home on the Web!
    </p>
    <p>
      <a href="http://www.disney.com/">Goodbye!</a>
    </p>
  </body>
</html>
```

We can see that there's not quite as much whitespace between lines, but the effect is generally the same as before.

- Although we'll spare you from running through all possible HTML tags,—there are so many that it would be very, very boring and not all that informative—we'll take a look at just a few more, including the `<h1>` tag. The default behavior for this tag is to make text large and bold. What's interesting and somewhat annoying, however, is that this tag will not be rendered the same across all browsers. Long ago, Mozilla and Microsoft made different judgment calls as to how big `h1` text should be. Frankly, Microsoft tends to be the most frustrating of all the major browser vendors because they refuse to adhere to the standards which all the others have adopted.
- One more trick we'll show you is a `mailto` link:

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body bgcolor="#FFFFFF">
    <p>
      Welcome to my home on the Web!
    </p>
    <p>
      <a href="mailto:malan@post.harvard.edu">Goodbye!</a>
    </p>
```

```
</body>
</html>
```

This link will appear like any other hyperlink but when clicked, it will open your default mail program with the e-mail address automatically filled in. The risk with this kind of tag is that your e-mail address is now displayed for the public to see. There are many spammers out there who write programs to crawl the web and harvest e-mail addresses just so they can send junk mail.

- If we want to link to a page that we ourselves have created, we can shorten the `href` value like so:

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body bgcolor="#FFFFFF">
    <p>
      Welcome to my home on the Web!
    </p>
    <p>
      <a href="two.html">Goodbye!</a>
    </p>
  </body>
</html>
```

Now when we click on the link, the browser will look for a file named `two.html` in the same directory that we're currently in. Even if our website is not live on the internet, this page will be served up so long as it exists in the same directory. We could create `two.html` so that it links back to our original page like so:

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body bgcolor="#FFFFFF">
    <p>
      Page 2!!
    </p>
    <p>
      <a href="index.html">Go back to page 1</a>
    </p>
  </body>
</html>
```

- We can even make the link's text the same as its value:

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body bgcolor="#FFFFFF">
    <p>
      Page 2!!
    </p>
    <p>
      <a href="http://computerscience1.net">http://computerscience1.net</a>
    </p>
  </body>
</html>
```

Recall that this has security implications. We can change the link text to appear as though it links to the course website when in fact it links to Disney's website:

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body bgcolor="#FFFFFF">
    <p>
      Page 2!!
    </p>
    <p>
      <a href="http://www.disney.com">http://computerscience1.net</a>
    </p>
  </body>
</html>
```

- We can create lists of items using the <ul> and <li> tags:

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body>
    <ul>
      <li>Goonies</li>
      <li>Godfather</li>
      <li>Godfather 2</li>
    </ul>
```

```
</body>
</html>
```

- Change the `<ul>` to `<ol>` and the bullets will be replaced by numbers in sequence:

```
<html>
<head>
  <title>My First Webpage</title>
</head>
<body>
  <ol>
    <li>Goonies</li>
    <li>Godfather</li>
    <li>Godfather 2</li>
  </ol>
</body>
</html>
```

- Tables are also very easy to create:

```
<html>
<head>
  <title>My First Webpage</title>
</head>
<body>
  <table border="1">
    <tr>
      <td>1</td>
      <td>2</td>
      <td>3</td>
    </tr>
    <tr>
      <td>4</td>
      <td>5</td>
      <td>6</td>
    </tr>
    <tr>
      <td>7</td>
      <td>8</td>
      <td>9</td>
    </tr>
  </table>
</body>
</html>
```

Here we have our own little keypad. `tr` stands for “table row” and `td` stands for “table data,” otherwise known as a cell. The `border` attribute ensures that each cell will be contained within its own box, like an Excel spreadsheet that might be familiar to you.

- We can have links within our table cells, as well:

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body>
    <ol>
      <li><a href="http://www.imdb.com/title/tt0089218/">Goonies</li>
      <li>Godfather</li>
      <li>Godfather 2</li>
    </ol>
  </body>
</html>
```

Notice here that we failed to close the `a` tag properly, so the other list items also appear to be hyperlinks. If ever a whole section of your webpage becomes a hyperlink, it’s probably because you failed to close an anchor tag. Different browsers will often render mistakes like this differently simply because there’s no standardized behavior.

- If we add a `style` attribute, we can begin changing the color and size of our link text:

```
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body>
    <ol>
      <li><a style="color: #FF0000; font-size: 96pt;"
        href="http://www.imdb.com/title/tt0089218/">Goonies</li>
      <li>Godfather</li>
      <li>Godfather 2</li>
    </ol>
  </body>
</html>
```

The value of the `style` attribute is actually a different language called CSS, of cascading stylesheets. Within the `style` attribute, there are many different CSS properties, each of which can take on a specified value. CSS property values are specified after a colon and properties are separated by semicolons.

- If we want to change the styling of *all* of our links without adding a **style** attribute for each, we can write some CSS within the **head** tag:

```
<html>
  <head>
    <title>My First Webpage</title>
    <style type="text/css">
      <!--

      a {
        color: #FF0000;
        font-size: 96pt;
      }

      -->
    </style>
  </head>
  <body>
    <ol>
      <li><a href="http://www.imdb.com/title/tt0089218/">Goonies</li>
      <li>Godfather</li>
      <li>Godfather 2</li>
    </ol>
  </body>
</html>
```

Now, we have the same CSS properties and values being designated, but because they're in a block within a **style** tag, they apply to all **a** tags. The **<!--** and **-->** are tags that represent an HTML comment. Anything that appears between these two tags will not be printed out to the browser window. Using the **style** tag, we can also change the background color of the whole page:

```
<html>
  <head>
    <title>My First Webpage</title>
    <style type="text/css">
      <!--

      body {
        background-color: #0000FF;
      }

      a {
        color: #FF0000;
        font-size: 96pt;
      }

      -->
    </style>
  </head>
  <body>
    <ol>
      <li><a href="http://www.imdb.com/title/tt0089218/">Goonies</li>
      <li>Godfather</li>
      <li>Godfather 2</li>
    </ol>
  </body>
</html>
```

```
    }  
  
    -->  
    </style>  
</head>  
<body>  
    <ol>  
        <li><a href="http://www.imdb.com/title/tt0089218/">Goonies</li>  
        <li>Godfather</li>  
        <li>Godfather 2</li>  
    </ol>  
</body>  
</html>
```

CSS provides a way for us to separate the content from the style of a webpage. The word *cascading* refers to the fact that if you assign a value to a property twice, the lower assignment trumps the ones above.

- Using a WYSIWYG (what you see is what you get) editor like Dreamweaver is compelling because it allows you to do some advanced styling very easily, but it's also very error-prone and will often leave you with a mess of CSS code that you have to wade through if you want to fix some small cross-browser bug.

## 2.6 Going Live (102:00–107:00)

- Using a program like Cyberduck or SecureFX which support SFTP (secure file transfer protocol), we can upload our HTML files to an FAS account which provides free web hosting. Once we've changed the permissions on the file, meaning we allow everyone to have read access to `index.html`, we can view it live on the web just as we've been viewing it locally on our file system.
- Of course, if you're making changes locally to a file, you must upload it to the web server before you'll be able to view those changes live on the internet. You might also have to forcibly refresh the page in case your browser has cached a copy of the old page.

## 2.7 Images (107:00–108:00)

- One tag we haven't looked at yet is the `img` tag, which allows you to display images on your website:

```

```

Like the `br` tag, the `img` tag is *empty*, meaning you open and close it with the same tag.

## **2.8 Firebug (108:00–110:00)**

- One last recommendation we'll make is that you install Firefox at least for web development purposes. Firefox has a plugin called Firebug which is fantastically useful. Once installed, it will display a bug icon in the bottom right of the browser window which, when clicked, will pop up an interface that allows you to examine the HTML of the page you're currently viewing in a very convenient hierarchical, expandable format.