

Art 198

tutorial-01 – Getting Started: Basic HTML Concepts and Dreamweaver Interface

From *A Dao of Web Design*

by John Allsopp

<http://www.alistapart.com/articles/dao/>

The web is still a new medium. Like many new mediums (think television), it borrows from existing medium, print. Some of what it borrows makes sense, but just like the early TV shows used voice-overs, left over from radio, the web emerges from hundreds of years of printing and thousands of years of writing.

It's time to throw out the rituals of the printed page, and to engage the medium of the web and its own nature.

The fact we can control a paper page is really a limitation of that medium. You can think – we can fix the size of text – or you can think – the size of text is unalterable. You can think – the dimensions of a page can be controlled – or – the dimensions of a page can't be altered. These are simply facts of the medium.

And they aren't necessarily good facts, especially for the reader. If the reader's eyesight isn't that of a well sighted person, chances are the choice the designer made is too small to comfortably read without some kind of magnification. If the reader is in a confined space, a train to work, an airplane, the broadsheet newspaper is too large. And there is little the reader can do about this.

The control which designers know in the print medium, and often desire in the web medium, is simply a function of the limitation of the printed page. We should embrace the fact that the web doesn't have the same constraints, and design for this flexibility. But first, we must "accept the ebb and flow of things".

It is the nature of digital content available on the Internet to be flexible and accessible to all comers. This inherent nature will inform our design choices, choosing flexible layouts, universal fonts and color choices, with valid, semantically marked up html.

I. BASIC WEB CONCEPTS: HTML, XHTML AND CSS

1. If not already on the computer, download Firefox Web Developer Toolbar – very helpful to deconstruct web pages and see how they are made, as well as troubleshoot your own designs.

<http://www.mozilla.com/en-US/firefox/>

<https://addons.mozilla.org/firefox/60/>

What is an HTML File?

- *HTML stands for Hyper Text Markup Language*
- *An HTML file is a text file containing small markup tags*
- *The markup tags tell the Web browser how to display the page*
- *An HTML file must have an htm or html file extension (html is modern way)*
- *An HTML file can be created using a simple text editor*
- *The most current version of hypertext markup language is eXtensible*

Hypertext Markup Language or XHTML

XHTML is not very different from the HTML 4.01 standard. So, our complete HTML 4.01 reference cheat sheet can help you with that. In addition, you should start NOW to write your HTML code in lowercase letters, and NEVER skip ending tags (like </p>).

- *XHTML elements must be properly nested*
- *XHTML elements must always be closed*
- *XHTML elements must be in lowercase*
- *XHTML documents must have one root element*

These are the basic elements of a web page.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html>
  <head>
    <title>Title of page</title>
  </head>
  <body>
    This is my first homepage.
  </body>
</html>
```

Part 1: Locate and copy tutorial files in the lab to your workstation

1. Login as **student** and password **artlab**
2. Locate the shortcut for Student Files
3. Locate the files in folder Art198 for **week1>tutorial-01**
 - Copy to your Documents, and close the network connection (eject the network connection on your desktop)
 - **DO NOT WORK IN THE FOLDER ON THE SERVER!**
4. Add your initials to the week1 folder you saved in your Documents, so that you can identify your files in the lab (**week1-*yourinitials***).

Part 2: Open TextWrangler (Mac) or Notepad (PC) and the Firefox Browser

1. Windows: Programs>Utilites>Notepad or Mac: Applications>TextWrangler (TextWrangler is available as Freeware from BareBones <http://www.barebones.com/products/TextWrangler/>)
2. Create a new document. Type the html basic elements (DOCTYPE, html, head, title,body) as shown the box above and save as **mypage-yourinitials.html** in **tutorial-01/basicHTML/** folder.
3. Connect to the **Internet** using the **Firefox** browser, which is a modern **standards** compliant **browser** (a software application that visually presents web pages from a server to the client's computer over the Internet).
4. In Firefox, **File>Open File** and browse to **Documents/tutorial-01/basicHTML/mypage-yourinitials.html** and view the results in your browser. Now view the source with Firefox, **View>Page Source**. You see the code that you just typed.
5. I placed a copy of a similar page on my public Web site before class, so it is published on the World Wide Web. In Firefox go to <http://www.patriciarees.com/art198/week1/tutorial-01/mypage.html> or click on the @ icon in your tutorial folder called @mypage.html.webloc

Note: The Internet is the worldwide, publicly accessible network of interconnected computer networks that transmit data by packet switching using the standard Internet Protocol (IP). It is a "network of networks" that consists of millions of smaller domestic, academic, business, and government networks, which together carry various information and services, such as electronic mail, online chat, file transfer, and the interlinked Web pages and other documents of the World Wide Web.

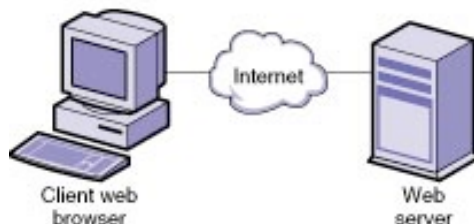
Note: The World Wide Web Consortium (W3C), along with other groups and standards bodies, has established technologies for creating and interpreting web-based content. These technologies, which we call "web standards," are carefully designed to deliver the greatest benefits to the greatest number of web users while ensuring the long-term viability of any document published on the Web.

- [The Web Standards Project](#)

client browser → internet → web server →



client browser ← internet ← web server ←



Note: Web browsers communicate (make requests) with web servers primarily using HTTP (hypertext transfer protocol). HTTP allows web browsers to send requests for information to web servers and the servers to deliver web page content to your computer's browser.

Examples of modern browsers:

MSIE 6+ (MSIE 7 - much better)
Firefox 2
Safari 2 (Mac) and 3 (Mac and PC)

Camino 1 (Mac)
Netscape 8
Opera 8+

Browser support chart -

http://www.westciv.com/style_master/academy/browser_support/index.html

Part 4: Explore a web page in the browser using the Firefox browser

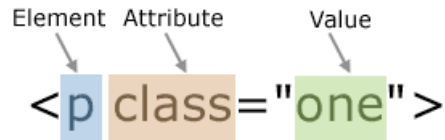
1. In the address bar of the Firefox browser, type the **URL** of this online document to see the online version of our sampler page.
<http://www.patriciarees.com/art198/week1/tutorial-01/sampler.html> or click on the icon named
@ www.patriciarees.com/art198/week1/tutorial-01/sampler.html in your tutorial folder. In Firefox “print” the browser

Note: URL (uniform resource locator is the means by which web pages are located; is treated as an address, beginning with http: for HTTP access. Many browsers also support a variety of other URL types and their corresponding protocols, such as ftp: for FTP (file transfer protocol), rtsp: for RTSP (real-time streaming protocol), and https: for HTTPS (an SSL encrypted version of HTTP).

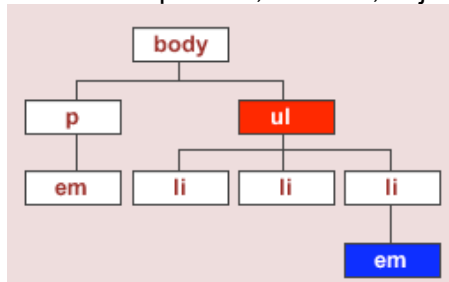
The file format for a web page is usually HTML (hyper-text markup language) and is identified in the HTTP protocol using a MIME content type. Most browsers natively support a variety of formats in addition to HTML, such as the JPEG, PNG and GIF image formats, and can be extended to support more through the use of plugins. The combination of HTTP content type and URL protocol specification allows web page designers to embed images, animations, video, sound, and streaming media into a web page, or to make them accessible through the web page. - [wikipedia](#)

1. Select FireFox toolbar **View Source** (or **Right Click>View Source**) and “print” the source code as PDF. Using Firefox Place the PDF “Source of: <http://www.patriciarees.com/art198/week1/tutorial-01/sampler.html>” displaying the source “structure” alongside the browser window displaying the “presentation”
2. Using Adobe Acrobat **Tools> Commenting and Markup** (Show Comment and Markup Toolbar) and the PDF source page, mark and identify the parts or tags on a typical XHTML webpage, as follows:
 - `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
 - Document Type Definition (DTD) has three parts
 - a. Type of XHTML
 - b. Language
 - c. URL for the definition of the type document structure such as XHTML Transitional
 - `<html xmlns="http://www.w3.org/1999/xhtml">` - XHTML Namespace or the start of the web page; also includes a URL
 - `<head>` - Includes all header information such as keywords, description, and language and identifies the unique page to the browser, server, and search engines and is not visible in the browser or presentation window except for the title
 - `<title>` - Title of web page and what appears in the title bar of the browser and is used by default as the name of a saved bookmark or favorite.

- `<body>` - Visible content of the web page that is displayed in the browser window
3. Find the closing tags for each: `</title>` `</head>` `</body>` `</html>`
 4. Compare the title in the title bar to see if it corresponds to the `<title>` tag information in the **source code**.
 5. HTML syntax
 - An html element can have an attribute and value
 - The surrounding characters are called angle brackets
 - HTML tags normally come in pairs like `<p>` and `</p>`
 - The first tag in a pair is the start tag, the second tag is the end tag
 - Everything except an attribute's value must be in lower case.
 - Attributes always have quotes surrounding the value.



- HTML tree: parents, children, adjacent siblings and descendants



6. Find the following content in the **browser display**:
 - The Main heading
 - A Sub-heading
 - Some Body text
 - Bulleted and/or numerical lists.
7. In the **source view or printed source** find the content building blocks of html and compare to the browser view:
 - Headings in the web document- `<h1>` `<h2>` `<h3>`
 - Paragraphs in the body of the web document- `<p>`
 - Lists – unordered `` and definition `<dl><dt>` `<dd>`
 - Strong and emphasized text - `` ``
 - Hyperlinks:
 - a. Anchors with `href` attribute for references to another web page
``
 - b. Anchor with `name` attribute to go to content on the same page
``
8. In the Firefox browser, now view the same page with visual presentation styles applied to the html structure. Click on the link “[Page with styles applied](#)” just below the first three heading examples. View the source (right click, View Page Source).
9. Within the opening and closing tags for the `<head>` tag, locate the `<style>` tag containing the CSS presentation rules for the look of our content.

*Note: The **CSS (Cascading Style Sheet)** is a style guide that separates the presentation or “look” from the structure or “mark up” of the page.*

10. Return to the browser view of the styled page. Using the Web Developer toolbar, select **CSS>View** and **CSS>Disable>All styles**, Notice how the semantic structure provided by the html elements stays when the styles are off.

Part 5: Using proper html elements to create the semantic structure of our sampler Web page

1. You will be using the sampler-text.txt file to add the html tag elements to create your webpage. Open the sampler-text.txt document in your tutorial-01 file, in TextWrangler or Notepad and add the html tags to create the web page sampler.html. In your Firefox browser, you can open sampler.html and Right click to “view source” for your guide.
2. To start re-save your file as **sampler-yourinitials.html**.
3. The first line is the doctype declaration. Add the beginning and closing angle brackets.
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
4. Begin the web contents with the html tag and include the provided attribute for the namespace url,
`<html xmlns="http://www.w3.org/1999/xhtml">`
and add the closing tag `</html>` at the bottom of the page. **Save**.
5. Next comes the `<head>` tag, which contains the title and meta elements as children.
6. The title tag
`<title>My Name HTML Basics</title>`
is displayed in the browser "chrome", and is the default name used when someone bookmarks or saves the page as a favorite.
7. The meta tag
`<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />`
The meta element is an example of a self closing tag. It has no ending tag, but include the forward slash / before it's opposite angle bracket. This element tells the server that this page is text html content and that it is using Unicode character set UTF-8 to determine the characters. Unicode UTF-8 is the most reliable method to render characters on the web that may include international characters and special symbols.
8. Now we begin the content, opening and closing the body element `<body>`. Preview your document so far in the browser with Firefox. Either go to **Firefox and File>Open and browse to your Documents/tutorial-01/sampler/sampler-yourinitials.html** file or in the **Finder**, drag the sampler.html file over the Fireworks application icon in the dock and let go. If you have coded properly you should see a web page, with text that pretty much runs together. Notice that even though the TextWrangler document shows an empty lines between paragraph, and white space, etc. the browser ignores this.
9. Next we want to start structuring the content, by adding several levels of headings `<h1>` `<h2>` `<h3>`. Headings help to organize the document, as in an outline. Headings should always be used in order, not skipping a level, but can be repeated and nested. Headings are block elements. **Save**.
10. Paragraphs `<p>` are the meat and potatoes of our content. Paragraphs are block elements. The first paragraph also has a soft return to separate the note text to a new line. Forcing part of a paragraph to a new line is done with a break element, which is also a self closing tag. `
`. **Save**.
11. We will skip the images and links for now. (If you are behind, open **Documents/tutorial-01/sampler/sampler-html/sampler-html-mid.html** and save with your initials)

12. A horizontal rule `<hr />` is another self-closing tag. It can be used to separate content into sections. We've used two of them in the sampler structure.
13. There are three types of lists. Ordered lists `` imply things in the list are done in a certain order or steps, Unordered lists `` are for a list of items with logical progression or steps. Ordered and unordered lists contain list items ``. Definition lists `<dl>` have terms `<dt>` and descriptions `<dd>`, such as a glossary. Add the tags to the unordered list, ordered list and definition list text. **Save.**
14. Tables are for organizing tabular data, information that is 2 dimensional, read by rows and columns. Tables `<table>` have table headers `<th>` that identify the label for each row or column and table data `<td>`, the information in each cell. Add the tags for the table, rows and data cells. **Save.**
15. The address element `<address>` provides the contact information for the author or owner of the web content. Add the tags for address element. **Save.**
16. Hyperlinks are made of an anchor element `<a>` with an attribute href that specifies the web location ``. Add the anchor tags for the links to the world wide web for Adobe (absolute link). Add the anchor tags for the links to mypage.html in the tutorial folder (relative link). **Save.**
17. A named anchor `` can also indicate a link to the same page or a specific point in another page. Add the anchor tags to the named anchor on the same page. **Save.**
18. An image `` is inserted by indicating the source file location and name. ``. Valid syntax requires alternate text for any page elements that are not text. The alt attribute should briefly describe what is the image's purpose or what it adds to the content of the page. Although not required, adding the image height and width, helps the browser render the page with images faster. Add the tags for the image source reference, including the alt attribute text description. **Save.**
19. xhtml will cause some symbols to render improperly in some browsers. To make sure that the copyright symbol will be rendered properly across all browsers, we will use the Unicode equivalent `©`. Unicode equivalents are usually in the format `&#xxx;`
20. This concludes the html content. When we look at the page in the browser now, the content, while not elegant, is clearly set out and very readable. In Firefox **File>Open File>tutorial-01>sampler>sampler-yourinitials.html.**
21. The underlying structure given by the html elements make this page readable by a variety of user agents or devices, such as browsers, assistive technology such as screen readers or even mobile phones and PDAs. These browsers (clients) usually apply styles to these html structure elements and provide a visual format. select **CSS>Disable>Browser default styles.** Notice that the Firefox browser provided it's own default styles to the html elements.

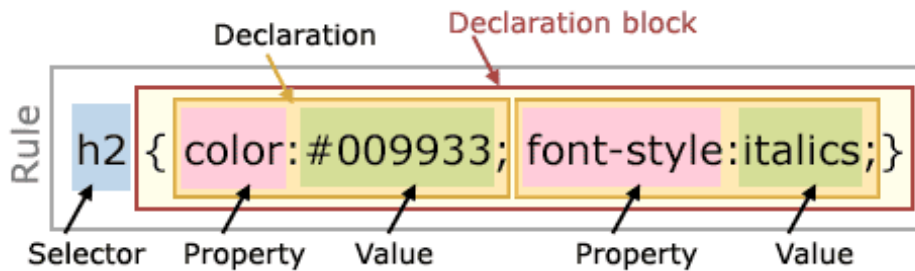
Part 6: Adding css styles (the look and feel, called “presentation layer”) to the html content structure

1. The next step is to apply the presentation look and feel to the html elements. We use something called CSS (Cascading Style Sheets) to do this.

Golden rules for adding font styles, spacing, etc. to your page:

- Never use extra `<p>`, `
` or spaces to format your text.
- Code spacing: extra white space in code is never displayed on the browser. There may be space around your text in the code, but the browser ignores it. The number of lines in a paragraph depends on the size of your browser window. If you resize the browser window, the number of lines in the paragraph will change.
- Use heading tags only for headings. Don't use them just to make something bigger or bolder. Use styles for that.

- For this exercise we will add the styles into the head of this page to make it easier to view and check our progress, but ultimately we will pull the styles out of the html page into a separate style sheet that the page will reference.
- The easiest way to style our page is to define how we want our html elements to look, like our paragraphs `<p>` and headings `<h1>`.



- Open the `css.txt` file in the `tutorial-01>sampler` folder so that we can copy and paste the code easily. If you didn't finish your sampler file, open the `sampler-css-start.html` in the `tutorial-01>sampler` folder and save it with your initials.
- Styles belong in the `<head>` of our html. Just below the `<meta>` tag, insert an opening and closing tag for the style element. `<style type="text/css">` and `</style>`
- First we will set the basic font for our page using the `body` element selector. Copy and paste the `body` rule which sets the color, size and font family for the text. To remove the browser default margins we set the margins on `body` element to zero. Copy and paste the next four margin properties and values (without the opening and closing comment tags `/* */`) below the `color:` rule and before the closing curly `}` bracket. Finally change the background color by pasting that property and value into the body rule in the same manner. **Save and preview in the browser.**
- Now we are going to change the way two of our html elements appear. We will add bolded font and margin to the definition term element `<dt>`. Copy and paste the css rule for `dt`. Notice that both placed in our markup where we used the `<dt>` element are now styled.
- Now we are going to change the way the table head row looks. Copy and paste the css rule for `th`.
- For the last part of modifying the page we will need to create new custom styles for some areas of our page where we don't actually have an html element that we can control. We want the note to stand out in red font. Copy and paste the custom class `.note` to your styles. Now we have the style as a class, but there is nothing in our html structure that relates to this custom class. In our other examples, the styles immediately affected the content because we actually changed the way the element would look. For our note, it is just part of the paragraph. We need to isolate it. We will use the inline element `span` to wrap our inline content that we want to look different. Add the span opening and closing tags `` and `` around the text beginning with "Note:" in the first paragraph. Now we can apply the style to just this text. In the opening span tag add the attribute `class="note"` to end up with this: `Note: Inline images must be in a block element like a paragraph or division.`
Add the custom class `note` to the first paragraph note with a span tag as well.
- And we want our footer to have a different color font, light blue background, some margin and be centered. Copy and paste the custom class `.siteinfo` into your `<style>` tag in the `<head>`.
- The footer containing our site info (My Name, email and copyright) is made up of two elements, the address and a paragraph element. In order to treat this area of content as one block or "division" of the page with a different background, we will wrap it in the block element `div` to select this content. Add the opening div tag `<div>` before the

<address> tag and the closing div tag </div> after the closing </p> tag. Now we can add the class attribute for **.siteinfo** to this and our styles will be applied to this "division" block of the page content. **<div class="siteinfo">. Save.**

Part 7: Moving the css to an external file.

1. In an ordinary Website, there are usually many pages and they have a common look and feel. Instead of repeating all the css style code in each page and then having to keep them all updated if we want to change something, we link each page to an external cascading style sheet.
2. Open a new document in TextWrangler and save it with the name **base-01-yourinitials.css** and save it in the css folder.
3. Back in our **sampler-css-yourinitials.html** document (open **sampler-css-external-start.html** if you didn't finish), save it with new name **sampler-css-external-yourinitials.html**.
4. We are going to copy all of the styles in between the **<style>** tags and paste them into the **base-01-yourinitials.css** document. **Save.**
5. Return to your sampler and delete the remaining style tags. In their place we add a link to the css style sheet file. Remove the html comment tags **<!--** and **-->** around the text **<link href="css/base_01.css" rel="stylesheet" type="text/css" />**
6. View the **sampler-css-external-yourinitials.html** in Firefox **File>Open File** to see that the styles are still applied.

Part 8: General Web Resources - become familiar with basic web xhtml structure and semantics

1. The World Wide Web Consortium (W3C) develops specifications, guidelines, software, and tools to lead the Web to its full potential. It is a great resource for any web designer or developer.
 - <http://www.w3.org>
 - <http://www.w3.org/TR/html401/index/elements.html>
 - <http://www.w3.org/TR/html401/index/attributes.html>
 - <http://www.w3.org/TR/html401/index/list.html>
2. HTML Cheat Sheet: designed to provide a quick reference for HTML elements
 - <http://www.ilovejackdaniels.com/cheat-sheets/html-cheat-sheet/>
3. W3 Schools - find all the Web-building tutorials you need, from basic HTML and XHTML to advanced XML, SQL, Database, Multimedia and WAP.
 - <http://www.w3schools.com/>
4. *Alistapart* – an ongoing blog with great articles and resources, including this one for web newbies:
 - <http://www.alistapart.com/articles/alaprimer2>
5. Web Style Guide, 2nd edition
By Patrick Lynch and Sarah Horton covers all the basic elements of creating a Web site, with emphasis on structure and how to design for overall style and professional appeal and usefulness and has become a standard.
 - <http://webstyleguide.com/>
6. *NYPL Online Style Guide* explains the markup and design requirements for all Branch Libraries web projects, along with various standards and best practices and has become a standard.
 - <http://www.nypl.org/styleguide/>
7. If you are ready to go beyond the basics, read the *html-semantics.pdf* found in week one.