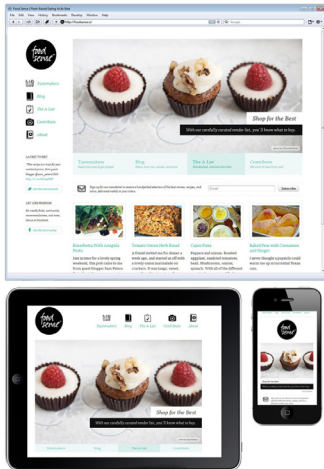


# “The best way to build web pages is with web-standards-compliant html and css.”

—Patrick Griffiths, *The Best Practice Guide to XHTML and CSS*

ARTV 330-001: Interaction Design for Visual Communication I / NKU / Professor Julie Mader-Meersman

Vocabulary Sheet / Standards-based Web Design Basics



FoodSense.com design  
by Ryan Essmaker

## ONLINE RESOURCES

[csszengarden.com](http://csszengarden.com)  
[mediaqueri.es](http://mediaqueri.es)  
[w3.org](http://w3.org)  
[css-tricks.com](http://css-tricks.com)

## BEGINNING GENERALITIES

**web site** — a set of linked documents and assets with shared attributes, such as related topics, a similar design, or a shared purpose accessible on the internet

**directory** — a container, such as a folder, used to organize other folders and files. In the web, the structure of the directory (i.e., the specific locations of folders and the files within them) dictates the pathways of access for files to find one another

**URL** (uniform resource locator) — web address

**WYSIWIG** — what you see is what you get\* (\*sort of, at least much of the time, especially in browsers)

**color mode** = RGB, hexadecimal (hex) values

## STATIC SITES

**Static sites** have content that is fixed in the HTML source code.

**HTML, XHTML, HTML5** (Hypertext Markup Language, .htm, .html extensions) — a language used to generate and structure content on the world wide web

**CSS** (Cascading Style Sheets, .css extension) — a language used to format / present / style HTML content, and control the positioning of various page elements

### Web standards / advantages of web standards:

- cross-compatibility — web standards are not specific to one browser or another
- forward compatibility — browsers and devices headed in direction of web standards
- centralized control of presentation — separation of content from presentation makes site-wide presentation changes very easy
- device independence — not only useful for computer screens
- search engine optimization — improves findability for users
- lightweight pages — faster pageload times and lower bandwidth
- accessibility — considerate of users with disabilities
- employer and client expectations — a highly desired skill

👉 **Big message: HTML = content, CSS = presentation of that content**

## DYNAMIC SITES

**Dynamic sites** have data/database-driven content. Content in the site can change because it's pulled from a server or database(s).

### SAMPLE DYNAMIC PAGE FILE FORMATS:

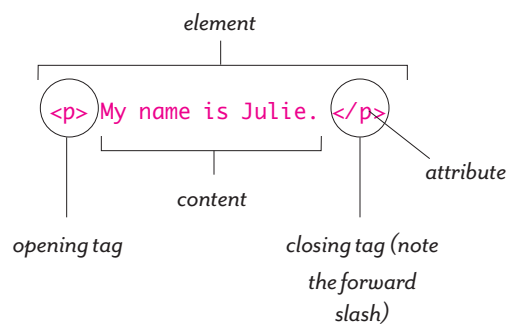
CFML (ColdFusion Markup Language); ASPX (ASP.NET files, .aspx extension); PHP (Hypertext Preprocessor files, .php extension)



“HTML&CSS: Design and Build Web Sites,” by Jon Duckett

### HTML SYNTAX

HTML syntax describes the rules that define the methods for creating valid html code, documents or markup fragments.



**elements** — components of an html document that define content and its hierarchical structure;

instances of elements are identified by descriptive markup, usually a start-tag and end-tag

**block elements** — elements used to structure something other than just a single line of content

(examples: paragraphs, headings, forms and lists)

**inline elements** — elements within a line (examples: links, images, emphasized text, form fields)

**nested elements** — elements inside of other elements

**tags** — define the start and close of elements (they use the <> and </> signs to contain attributes)

**attributes** — a hierarchical/categorical setting for an element (examples are p, h1, h2, href, etc.)

### RULES FOR TAGS AND ATTRIBUTES

- All tags and attributes must be in lowercase:

`<p> </p>` is okay; `<P> </P>` is not.

- All tags must close:

Use a forward slash (/) inside of the greater-than and less-than tag brackets to close a tag.

- All elements must nest properly (nesting elements are referred to as parent and child elements):

`<p>This is a <i>good</i> example of tag nesting.</p>` = This is a *good* example of tag nesting.

`<p>This is a <i>bad example of tag nesting.</p></i>` = This is a *bad example of tag nesting.* </i>

- Some attribute values must be in quotation marks (as in the case of defining links):

`<a href="http://www.html5dog.com"> HTML Dog </a>`

### BASIC STRUCTURE OF AN HTML DOCUMENT

A standard HTML document must have the **head**, **title**, and **body** structural elements in order to be valid. When you make a new dreamweaver HTML file, it generates these items and element tags for you:

**document type declaration** — tells your HTML file how it will talk to the browsers

**opening html element** — this defines everything within the page content as an HTML file and contains the following two required block elements:

**head** — gives general information *about* the content of the html document (but isn't content itself) and must include a **title** element that creates a label for that page that will appear as the title of any browser window and will show up as the title for that page when it gets bookmarked.

**body** — this is where all of the viewable, audible or otherwise experienced content goes

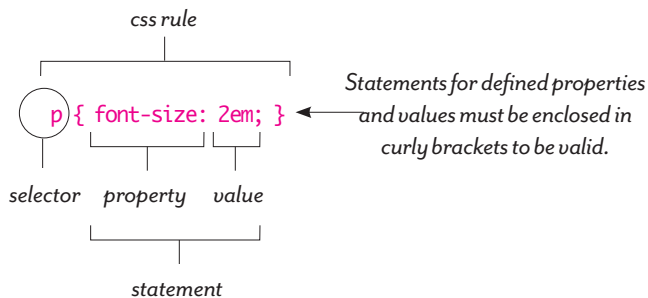
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Untitled Document</title>
</head>
<body>
<!-- A whole load of content -->
</body>
</html>
```

**comment** — invisible comments addressed to people interested in understanding the code can be placed right within the html of a file, in this format: `<!-- Here's a comment -->`

**div** — a generalist block tag that groups together a chunk of HTML; it can be used for layout purposes

### CSS SYNTAX

Linked to HTML, but completely different in its programming language, CSS syntax is a collection of rules made up of selectors, properties and values that all define the visual appearance of its HTML.



**css rule** — a combination of settings, including a statement of designations for selectors, properties and values that determine the visual presentation for specific tags and HTML instances (attributes)

**selector** — defines the part of the HTML (the attribute) to which to apply the declaring style statement

**html selector** — specifies an HTML element to which the declarations get applied

**id selector\*** — (#) specifies an HTML element *with a corresponding id* to which declarations get applied; can be used only once on a page; can be used across multiple pages in a site

**class selector\*** — (.) specifies an HTML element *with a corresponding class* to which declarations get applied; can use many times on a single HTML page; can also use across multiple pages in a site

**property** — defines a specific presentational aspect for the targeted element you want to set (must end with a colon); examples range from colors to font sizes to line heights to background images

**value** — defines what the property should be set to (must end with a semicolon)

**units** — measurements made by either numbers, percentages or lengths

**color** — defined by either hex values (#fff or #0000ff), RGB (0, 0, 255), or a name (i.e., "blue")

**Three levels of style sheets:** browser (3), author (2), user (1)

**Applying css to html documents:** CSS is useless on its own; must be attached to HTML. Apply 3 ways:

**inline** — quick-fix method, styles can occur straight in the body of the HTML doc; *discouraged*.

**embedded** — where CSS rules are defined and live in the head of the HTML doc; good for styles applying to only one page in a site.

**external** — the cleanest, purest, best way of attaching CSS to a site with many pages having the same over all styling; the CSS in this case exists as a totally separate file from the HTML docs, and can be attached to the HTML files various ways.

\*TIP: *id and class selectors allow for a lot of customization. Names that you give them should be content-based, not style-based.*