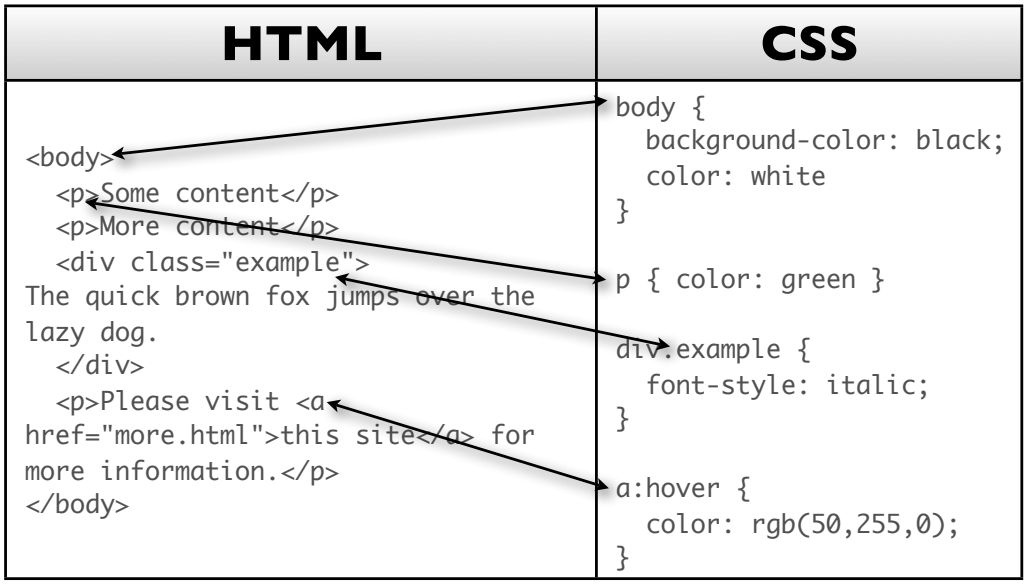
The Tags That Bind

- We have mentioned, many times now, how XHTML, CSS, and JavaScript files can coordinate and connect to form an overall Web application while separating the functional roles of content, presentation, and interaction
- We now go into more detail on the key role that XHTML tags play in this cycle

Tag	What It Defines	Key Components
html	The overall document	head, body
head	Document setup	title, meta, link, script
body	Displayed document content	content tags: div, span, a, h, p, hr, ul, ol, li, table…and many more
title	Document title	title text
meta	Document property, characteristic, or setting	assorted: Google "html meta dictionary" for reference lists
link	Related ("linked") files, particularly CSS	*rel*, *href*, *type* attributes
script	JavaScript code to execute	*src* attribute; script code
div	Distinct document block	*class* attribute
span	Inline document block	*class* attribute
a	Anchor element (the ubiquitous link)	*href* attribute; link body
img	Inline image	*src*, *width*, *height* attributes

From Model to View



- "C" stands for "cascade" — styles in enclosing tags affect tags within unless otherwise specified
- The *class* attribute essentially creates subcategories of a particular tag (e.g. different kinds of *div*, different kinds of *p*)
- A colon (:) instead of a period (.) after a style defines a *pseudo-class* or *pseudo-element* — typically a transient state for an HTML element, such as *link*, *hover*, *visited*, and *active* for the <a> (anchor/link) tag

CSS Properties

- Range of acceptable properties depends on the tag for which you are defining a style — exhaustive lists can be found on reference sites like *htmlhelp.com*
- Properties consist of a name and a value
- Range of acceptable property *values* depends on the specific property that you are setting

Property	Value	Example Values
font-family	generic or specific family name	generic: *serif*, *sans-serif* specific: "Arial"
font-size	absolute or relative size	absolutes: *small*, *medium*, 12pt relatives: *larger*, –50%
color, background-color	color keyword or value; *transparent*	keyword: *black*, *blue*, *gray*, *green* value: #005500, rgb(0,85,0) use hex for #, decimal for *rgb*
text-align	horizontal text alignment	*left*, *right*, *center*, *justify*
vertical-align	vertical text or image alignment	*text*, *middle*, *bottom*, +25%
margin, padding	space around blocks	margin-left: 24ex padding-top: 2em margin: 8em padding-left: +5%

CSS "Field Notes"

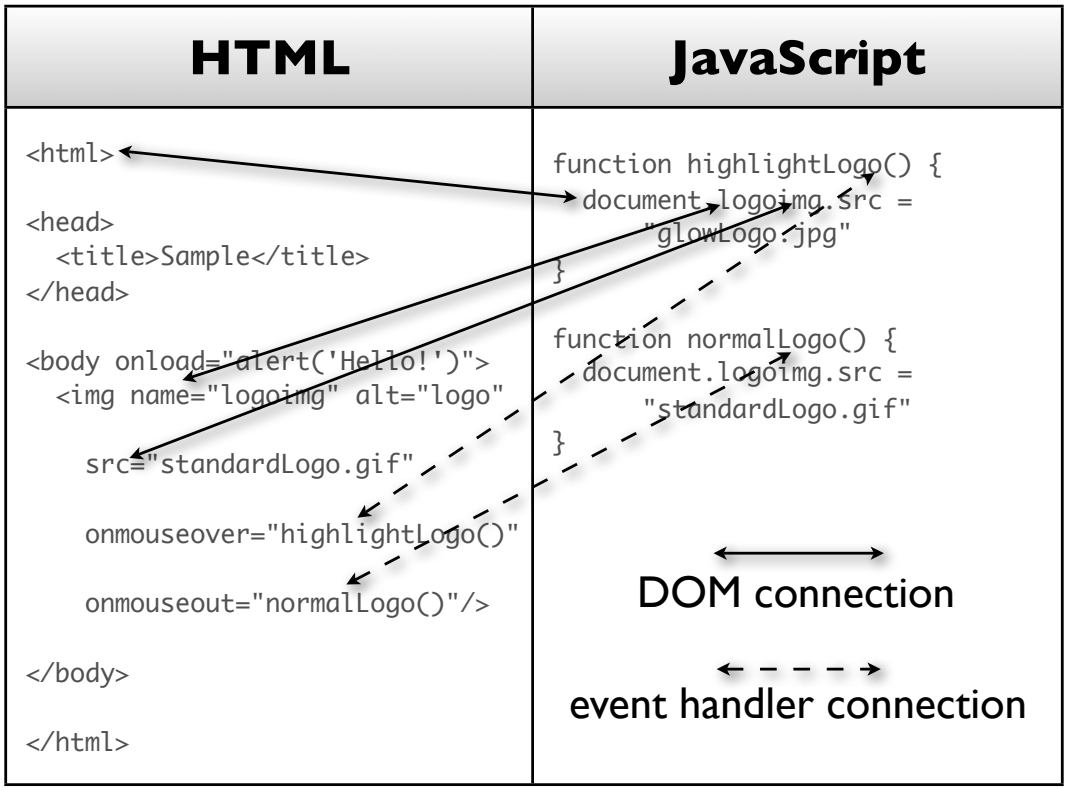
- Some style properties (*background-image*, *list-style-image*) accept images in the form of URL expressions (*url(image-file.jpg)*) — note how these image files are *relative* to the location of the CSS file
  - Recommended practice: locate these images within the same folder as the CSS file (whether at the top level or in subfolders), *not* along side your image assets
  - Makes sense if you think about it — after all, these images are part of the *presentation*, not the content

- As much as possible, use *em* and *ex* for sizes, not *px* or *pt* — *em* and *ex* are based on the *current font size*, so that text areas will scale along with the browser font
- MS IE does CSS notoriously badly — style for standards compliance first, then add IE workarounds as separate files (the `<!--[if IE]>...<![endif]-->` backdoor helps here)
- Use the *Firefox Web Developer Extension* — *http://chrispederick.com/work/webdeveloper*

From Model to Controller

Two mechanisms connect HTML to and from JavaScript:

- JavaScript "sees" the HTML through the *document object model (DOM)* — based on HTML tags and any *name* attributes defined in those tags
- HTML "contacts" JavaScript through *event handlers* — specified through tag attributes



JavaScript/DOM Tips

- Top-level DOM objects (i.e., start here) are *navigator*, *window*, and *document*
- Subsequent properties vary depending on the object — search the Web for "javascript DOM reference" or the like
- Event handlers are expressed as attributes in XHTML tags — content should be small bits of JavaScript, typically function calls

- The most popular events include *load*, *unload*, *mouseover*, *mouseout*, *click*, *focus*, *blur*, *change*, *submit*, *reset* — typically linked to XHTML by adding "on" to the event name and using that as the attribute (e.g., *onfocus*, *onmouseover*)

- As with DOM, full reference is available on the Web — "javascript event handler reference" or something similar will do the trick when searching