

▪ HTML vs. CSS

- HTML **structures** your content, but always gives it the same appearance
- CSS changes the **appearance** of your page and gives it *STYLE!*
- HTML is **semantic** (all meaning), CSS is **presentational**
- Together, HTML and CSS can create *beautiful* and functional websites
- When you don't include CSS with your HTML, the browser renders the HTML in its **default style** (e.g., different browsers render headers, links, etc. differently); it doesn't mean there's no CSS, it's just the default for the browser
- Updating **deprecated HTML**: if you learned to style with HTML, you'll need to unlearn those practices
- <http://www.webplatform.org/>
- <http://docs.webplatform.org/wiki/css/tutorials>

▪ Working with CSS directly in your HTML document

- Use `<STYLE>` tag or `STYLE=` attribute (with `<BODY>` tag) for overall page styling
- Use `STYLE=` attribute with various inline and block tags, including `` and `<DIV>`, for specific element styling

▪ CSS syntax and declarations

- **Rules** are the base unit of CSS
- Rules begin with a **selector**, which tells the browser which elements to apply the style to (more on selectors in a bit)
- Follow with one or more **declarations**
- Declarations are essentially the CSS equivalent to HTML attributes and values combined; they tell your browser how to style what you have selected with the selector
- Declarations are always contained inside `{ }`
- Declarations include **property** and **value**, with a **COLON** between property and value
- Declarations always end with a **SEMI-COLON**
- Use a **SEMI-COLON** to **separate** multiple declarations, as well
- Use a **HYPHEN** in between words in multiple-word properties
- You can often **collapse multiple declarations** into one property with multiple values (e.g., `BORDER: DASHED 1PX GREEN`)
- Make your CSS readable and include selector and open bracket on first line, declarations each on their own line, and close bracket on last line

▪ CSS selectors and pseudo-selectors

- **Elements** (no marking, just element name) apply to any instance of that element
- **Attributes** (element, then attribute and, optionally, value in `[]`) apply to any instance of that element with that attribute and, optionally, value
 - **ASTERISK** after attribute and before value indicates that the value must contain

- **CARAT** after attribute and before value indicates that the value must start with
 - **DOLLAR SIGN** after attribute and before value indicates says that the value must end with
 - **Classes** (marked with a **PERIOD** and the class name) apply across multiple elements, can be more specifically defined; must include corresponding **CLASS=** attribute in HTML
 - **IDs** (marked with a **POUND SIGN** and the ID Name) apply to a single element, are the most specific; must include corresponding **ID=** attribute in HTML
 - An **ASTERISK** serves as a wildcard (called a **universal selector**)
 - On its own, an asterisk selects everything on the page
 - Multiple asterisks will let you set generational descendants (more on that in a bit)
 - You can **chain** an element selector with a class or ID (e.g., **P.BLUE**)
 - You can specify **descendant selectors** (**ANCESTOR SELECTOR**, followed by a **SPACE**, then **DESCENDANT SELECTOR**)
 - If you want a more specific version of the same thing, you can specify direct **parent-child relationships** (**PARENT SELECTOR**, followed by a **GREATER-THAN SYMBOL**, then **CHILD SELECTOR**)
 - <http://meyerweb.com/eric/articles/webrev/200006b.html>
 - You can select **adjacent sibling elements** by including a **PLUS SIGN**
 - You can also specify multiple selectors (separated with a **COMMA**) to create a **group selector**
 - **Pseudo-selectors** come after an existing selector and are preceded by a **COLON** (no space between selector and pseudo-selector)
 - Pseudo-selectors (including **pseudo-classes** and **pseudo-elements**) allow you to select page elements that can't be selected in any other way
 - Common pseudo-selectors include **:HOVER** **:ACTIVE** **:FOCUS** **:NTH-CHILD()** **:FIRST/LAST-CHILD** **:LINK** **:VISITED** **:CHECKED** **:BEFORE** **:AFTER** **:NOT()** **::FIRST-LETTER** **::FIRST-LINE** **::SELECTION**
 - <http://net.tutsplus.com/tutorials/html-css-techniques/the-30-css-selectors-you-must-memorize/>
 - <http://www.tutorialfeed.org/2009/04/10-great-css-selectors-you-must-know.html>
- **CSS: Cascading Style Sheets**
 - The browser reads a style sheet from **top to bottom**, replacing previous declarations with identical selectors with later ones
 - This is important for you, since you'll often need to set a **default style** and **override** specific elements with a different style
 - In general, the more **specific** a selector, the more likely the browser is to render the rule
 - There is a **hierarchy of selectors**: element selectors, classes, IDs
 - The presentation will generally default to the **most specific selector**
 - Use **!IMPORTANT** after value and before **SEMI-COLON** to override other selectors
 - Some properties are **inherited** by children of parent elements

▪ **Setting up an external style sheet**

- Based on the principle of separation of **content** and **presentation**
- Just start writing CSS; there's none of the document structuring that happens in HTML
- I do encourage you to use a CSS **reset**: <http://www.cssreset.com/>
- Setting up a link tag in your HTML document: `<LINK REL=STYLESHEET TYPE=TEXT/CSS HREF=MAIN.CSS />`
- Comments: `/* COMMENT HERE. */`

▪ **Positioning with CSS**

- How should you use CSS to position page elements?
- Think of each block element in your HTML as a **box**
- Each box consists of four different **shells**: content, padding, border, and margin
 - **Content** is your text/image/video content
 - **Padding** is the space between the edge of the box and your content
 - **Border** is the demarcation around the edge of the box
 - **Margin** is the space outside the box
- Positioning declarations allow you to move elements around the page
 - **FLOAT** allows you to move an element (e.g., `` or `<DIV>`) to the right or left and wrap text around it
 - When a **FLOAT** collides with another, they end up next to each other
 - **CLEAR** allows you to clear all floats to the left, right, or both
 - **DISPLAY** allows you to change an element from block to inline or vice-versa
 - By default, the position of all elements is **static**, but you can change the position to **relative**, **fixed**, or **absolute**
 - **POSITION:RELATIVE** allows you to move an element (to the top, bottom, right, and/or left) around while leaving other elements alone
 - With **POSITION:RELATIVE**, use the opposite value of what you would normally think
 - **POSITION:FIXED** allows you to lock an item on the page
 - **POSITION:ABSOLUTE** allows you to position a child element relative to a (non-static) parent element, but can be tricky, so you should mostly avoid it
 - **Z-INDEX** property allows you to specify stack order (which element will be in front, which in back)
 - http://www.w3schools.com/Css/css_positioning.asp

▪ **Working with color in CSS**

- **Hexidecimal** colors: first two digits are for **red**, second for **green**, third for **blue**
- Each pair represents a value (0%-100%)
 - Divide traditional RGB value (0-255) by 16, convert to 0-9, A-F
 - First number is the **quotient**
 - Second number is **remainder**, so first number is more important
- You can also use **named** colors
- http://www.w3schools.com/cssref/css_colornames.asp

- http://www.w3schools.com/html/html_colors.asp
- http://www.rapidtables.com/web/color/RGB_Color.htm
- <http://colorschemedesigner.com/>

▪ Vendor extensions to CSS

- All the major browsers allow vendor-specific **extensions**
- Extensions allow you to create **dynamic effects**, including animations and transitions
- For Firefox, all extensions start with **-MOZ-**
- For Chrome and Safari, extensions start with **-WEBKIT-**
- For IE, extensions start with **-MS-**
- For Opera, extensions start with **-O-**
- If you write an extension, include its equivalents in all browsers (if available)
- Because extensions can be tricky (and aren't universal), you should primarily use them when the effect isn't **necessary** or **essential** for your site
- <http://2012.beercamp.com/>
- <http://www.tutorialchip.com/css/css-browser-extensions-vendor-specific-properties/>
- https://developer.mozilla.org/en-US/docs/CSS/CSS_Reference/Mozilla_Extensions
- <http://blogs.msdn.com/b/ie/archive/2008/09/08/microsoft-css-vendor-extensions.aspx>
- <http://peter.sh/experiments/vendor-prefixed-css-property-overview/>

▪ Some common CSS properties

- Font-weight, font-family, font-size, font-style, text-decoration, text-transform
- Color, background-color, background-image, background-repeat
- Letter-spacing, line-height
- Border, padding, margin
- Float, clear, display
- Text-indent, text-align
- Height, width
- List-style-type
- -moz-, -webkit-, -o-