# Introducing CSS

## Types of style sheets

### A websites design

1. Is NOT usually the product of only one style sheet.
2. Is usually a combination of styles starting from the browser style sheet then superseded by the:
   1. Browser styles / user agent styles
      1. Styles built into the browser itself.
      2. Used if no other style sheets
   2. User-defined style sheet
      1. Styles defined by the user based on the settings she or he makes in configuring the browser.
      2. Example: altering display settings to make things bigger
      3. overrides browser styles
   3. External style sheets
      1. Styles that the website author creates and places within a CSS file and links to the page.
      2. Multiple docs can use the same style sheet
      3. overrides user-defined styles
   4. Embedded style sheets
      1. styles added to the head of an HTML doc
         1. Only apply to that HTML doc
         2. Not accessible to other docs in the website
      2. Overrides any styles I an external style sheet
   5. Inline styles
      1. Added as element attrubites within an HTML doc
         1. Apply to that element alone
      2. Should be avoided
         1. They violate the basic tenents of HTML
         2. HTML should only describe the content and structure of the doc
      3. Highest presidence

## Exploring Style Rules

### Style Rule

1. Defines the styles applied to an element or group of elements
2. Building block of CSS style sheet
3. Form:
   1. Selector{ Property1: value1; Property2: value2; …. }
   2. Selector
      1. identifies an element or group of elements
      2. Can be comma separated lists
   3. Property: value pairs
      1. Specify the style properties and their values applied to those elements
   4. Example
      1. h1 { color: red; text-align: center;}
      2. h1, h2 {color: red;}
4. CSS IGNORES WHITESPACE!!!

## Browser Extensions

### A browser’s own extended library of style properties.

#### sometimes the only way to support a particular css feature is through a browser extenstion

### Vendor Prefix

1. Identifies browser extentions
2. Indicates the browser vendor that created and supports the property
3. Vendor prefixes for browser extensions:
   1. Prefix (Rendering Engine) Browsers
   2. -khtml- (KHTML) Konqueror
   3. -moz- (Mozilla) Firefox, Camino
   4. -ms- (Trident) Internet Explorer
   5. -o- (Presto) Opera, Nintendo Wii browser
   6. -webkit- (Webkit) Android browser, Chrome, Safari
4. Example: To display content in separate columns
   1. article { -webkit-column-count: 3; -moz-column-count: 3; column-count: 3;}

### Broswers process style properties in the order theyre listed. Ignoreing properties they don’t recognize or support.

1. ALWAYS PUT MOST CURRENT SPECIFICATIONS LISTED LAST.

## Embedded Style Sheets

### Style rule structure is also used in:

1. Embedded style sheets
   1. Are inserted directly into HTML file as metadata by adding style element to doc head:
      1. <style> Style rules </style>
   2. Example:
      1. <style> h1 {color : red; text-align : center;} </style>
   3. THE STYLE LOADED LAST OVERRIDES THE STYLES DEFINED BEFORE
   4. Example: emdeded will override css file rules
      1. <link href=”tss\_styles.css” rel”stylesheet” />
      2. <style> style rules </style>

## Inline Styles

### Very last styles to be interpreted by the browser

### Style rule structure applied directly to specific elements using styling attribute:

1. <element style=”property1:value1; property2: value2; …”>
2. Content
3. </element>
   1. Property:value pairs
      1. Define the styles
      2. Directly applies to that element

### Example

1. <h1 style=”color: red; text-align: center;”>
2. Tri and Succeed Sports
3. </h1>

### APPLIES ONLY TO THAT ELEMENT

### RECOMMENDED TO ALWAYS USE EXTERNAL STYLE SHEETS INSTEAD OF INLINE OR EMBEDDED.

## Style Specficity and Presedence

### The more specific style rule overrides the more general style rule.

1. Only an issue when 2 or more styles conflict.
2. If two rules have equal specificity, then the one defined last has precedence .

## Style Inheritance

### Properties that are passed from a parent element to its children

### Rule sets an article blue, everthing within that article is blue.

### Style sheets and Style rules form a CASCADE of styles in Cascading Style Sheets

## Browser Developer Tools

### Most browsers include developer tools allowing the designer to view HTML code, CSS styles, and other parts of the webpage.

#### Makes it easier to locate the source of a style that has been applied to a speficic page element

### Each are different

### Are constantly being updated and improved

## !Important Style

### You can override the style cascade by marking a particular property with:

1. Property: value !important;
2. Example:
   1. h1 {color: orange !important;}
   2. sets the color of all h1 headings to orange
   3. marked as important
      1. it take presidence over any conflicting syltes found in other style sheets

### DO NOT USE generally

## Creating a style sheet

### Comments:

#### /\* comment \*/

Page html 95

## Defining the Character Encoding

### Define the character encoding used in every CSS document.

#### @charset “encoding”;

#### Encoding defines the character encoding used in the file:

#### @charset “utf-8”;

* 1. CSS at-rule

## Importing Style Sheets

### CSS at-rule

#### Rule used to send directives to the browser indicating how the contents of the CSS file should be interpreted and parsed.

* 1. @charset “encoding option”;
  2. @import url(the url):
     1. Used to import the contents of a style sheet
     2. Used to combine style rules from several sheets into a single file.
     3. Simplifies your html code by placing the decision on which style sheets to include and exclude in the CSS file rather than html.