

# L3P1: Notes

Wednesday, 13 March, 2019 3:12 PM

## Objective

1. History of CSS
  2. Style sheets
  3. Precedence & inheritance
  4. Color
  5. Contextual selector
  6. Attribute selectors
- Define list styles  
Work with margins and padding space  
Use pseudo-classes and pseudo-elements  
Insert page content with CSS

## Introducing CSS

1. Determine appearance of page
2. Latest version: CSS3
3. CSS3 built upon several modules

## Types of style sheets

1. **Browser styles/user agent styles**
  - a. Built into browser
2. **User-defined styles**
  - a. Defined by user using browser
3. **External styles**
  - a. In CSS file linked to page
4. **Embedded styles**
  - a. Added to HEAD of HTML document
5. **Inline styles**
  - a. Added as element attributes

## General syntax

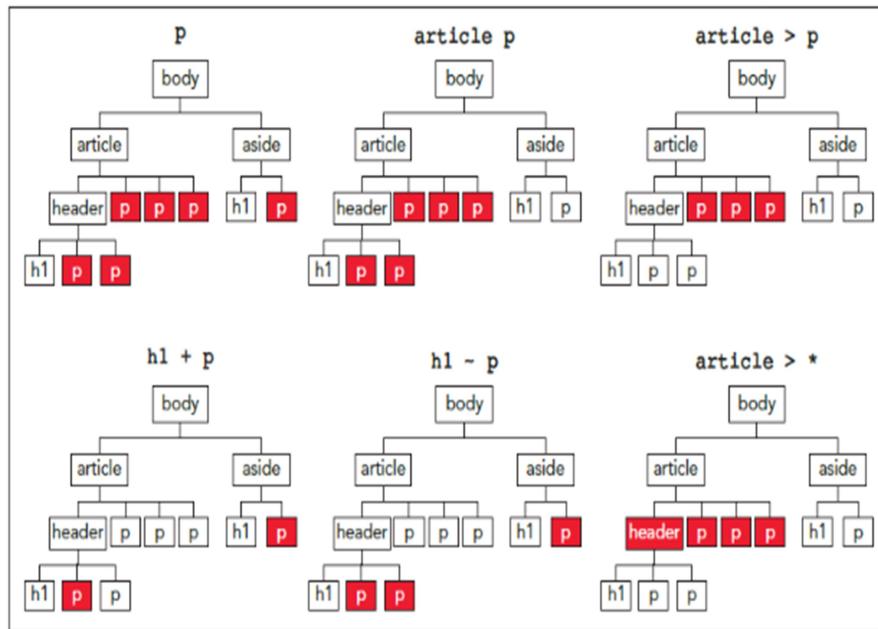
1. Selector {
  - a. Property1: value1;
2. ...}

## Style rules

Figure 2-12 Contextual selectors

Selector	Description
*	Matches any element
elem	Matches the element <i>elem</i> located anywhere in the document
elem1, elem2, ...	Matches any of the elements <i>elem1</i> , <i>elem2</i> , etc.
parent descendant	Matches the <i>descendant</i> element that is nested within the <i>parent</i> element at some level
parent > child	Matches the <i>child</i> element that is a child of the <i>parent</i> element
elem1 + elem2	Matches <i>elem2</i> that is immediately preceded by the sibling element <i>elem1</i>
elem1 ~ elem2	Matches <i>elem2</i> that follows the sibling element <i>elem1</i>

Figure 2-13 Contextual selector patterns



Selector	Selects	Example	Selects
<code>elem#id</code>	Element <code>elem</code> with the ID value <code>id</code>	<code>h1#intro</code>	The <code>h1</code> heading with the id <code>intro</code>
<code>#id</code>	Any element with the ID value <code>id</code>	<code>#intro</code>	Any element with the id <code>intro</code>
<code>elem.class</code>	All <code>elem</code> elements with the <code>class</code> attribute value <code>class</code>	<code>p.main</code>	All paragraphs belonging to the <code>main</code> class
<code>.class</code>	All elements with the class value <code>class</code>	<code>.main</code>	All elements belonging to the <code>main</code> class
<code>elem[att]</code>	All <code>elem</code> elements containing the <code>att</code> attribute	<code>a[href]</code>	All hypertext elements containing the <code>href</code> attribute
<code>elem[att="text"]</code>	All <code>elem</code> elements whose <code>att</code> attribute equals <code>text</code>	<code>a[href="top.html"]</code>	All hypertext elements whose <code>href</code> attribute equals <code>top.html</code>
<code>elem[att~="text"]</code>	All <code>elem</code> elements whose <code>att</code> attribute contains the word <code>text</code>	<code>a[rel~="glossary"]</code>	All hypertext elements whose <code>rel</code> attribute contains the word <code>glossary</code>
<code>elem[att = "text"]</code>	All <code>elem</code> elements whose <code>att</code> attribute value is a hyphen-separated list of words beginning with <code>text</code>	<code>p[id = "first"]</code>	All paragraphs whose <code>id</code> attribute starts with the word <code>first</code> in a hyphen-separated list of words
<code>elem[att^="text"]</code>	All <code>elem</code> elements whose <code>att</code> attribute begins with <code>text</code> [CSS3]	<code>a[rel^="prev"]</code>	All hypertext elements whose <code>rel</code> attribute begins with <code>prev</code>
<code>elem[att\$="text"]</code>	All <code>elem</code> elements whose <code>att</code> attribute ends with <code>text</code> [CSS3]	<code>a[href\$="org"]</code>	All hypertext elements whose <code>href</code> attribute ends with <code>org</code>
<code>elem[att*="text"]</code>	All <code>elem</code> elements whose <code>att</code> attribute contains the value <code>text</code> [CSS3]	<code>a[href*="faq"]</code>	All hypertext elements whose <code>href</code> attribute contains the text string <code>faq</code>

Note that some of the attribute selectors listed in Figure 2-15 were first introduced in CSS3 and, thus, might not be supported in older browsers.

### 3. Browser extensions

- a. Extended library of style properties in browser

### 4. Vendor prefix

- a. Indicate browser vendor that created and supports style property

Selector	Selects	Example	Selects
<code>elem#id</code>	Element <code>elem</code> with the ID value <code>id</code>	<code>h1#intro</code>	The <code>h1</code> heading with the id <code>intro</code>
<code>#id</code>	Any element with the ID value <code>id</code>	<code>#intro</code>	Any element with the id <code>intro</code>
<code>elem.class</code>	All <code>elem</code> elements with the <code>class</code> attribute value <code>class</code>	<code>p.main</code>	All paragraphs belonging to the <code>main</code> class
<code>.class</code>	All elements with the class value <code>class</code>	<code>.main</code>	All elements belonging to the <code>main</code> class
<code>elem[att]</code>	All <code>elem</code> elements containing the <code>att</code> attribute	<code>a[href]</code>	All hypertext elements containing the <code>href</code> attribute
<code>elem[att="text"]</code>	All <code>elem</code> elements whose <code>att</code> attribute equals <code>text</code>	<code>a[href="top.html"]</code>	All hypertext elements whose <code>href</code> attribute equals <code>top.html</code>
<code>elem[att~="text"]</code>	All <code>elem</code> elements whose <code>att</code> attribute contains the word <code>text</code>	<code>a[rel~="glossary"]</code>	All hypertext elements whose <code>rel</code> attribute contains the word <code>glossary</code>
<code>elem[att = "text"]</code>	All <code>elem</code> elements whose <code>att</code> attribute value is a hyphen-separated list of words beginning with <code>text</code>	<code>p[id = "first"]</code>	All paragraphs whose <code>id</code> attribute starts with the word <code>first</code> in a hyphen-separated list of words
<code>elem[att^="text"]</code>	All <code>elem</code> elements whose <code>att</code> attribute begins with <code>text</code> <small>[CSS3]</small>	<code>a[rel^="prev"]</code>	All hypertext elements whose <code>rel</code> attribute begins with <code>prev</code>
<code>elem[att\$="text"]</code>	All <code>elem</code> elements whose <code>att</code> attribute ends with <code>text</code> <small>[CSS3]</small>	<code>a[href\$="org"]</code>	All hypertext elements whose <code>href</code> attribute ends with <code>org</code>
<code>elem[att*="text"]</code>	All <code>elem</code> elements whose <code>att</code> attribute contains the value <code>text</code> <small>[CSS3]</small>	<code>a[href*="faq"]</code>	All hypertext elements whose <code>href</code> attribute contains the text string <code>faq</code>

Note that some of the attribute selectors listed in Figure 2-15 were first introduced in CSS3 and, thus, might not be supported in older browsers.

## Embedded Style Sheets

1. `<style> style rules </style>`

## Inline styles

1. `<element style="property1:value;">`

## Style specificity & precedence

1. Specific > general
2. Specificity: issue when 2 or more styles same specific
3. If 2 equal specific & importance, last defined precedence

## Style inheritance

1. Properties passed from parent to children

## Browser Dev Tools

1. Allow to view H

## 2. HTML code

### Writing Style Comments

1. `@charset "utf-8"`
  - a. Defines character encoding used in style sheet
2. `@import: CSS at-rule to import content of style sheet file`
  - a. Ex: `@import url(url)`
  - b. Ex:

```
<style type="text/css">
@import url('ExternalStyle.css');
</style>
```
3. `/* */` comments

### Working with color in CSS

1. Color values - color is given by an exact numeric representation
2. RGB Triplet - Intensity of primary colors in CSS
  - a. `rgb(red, green, blue)`
3. Hexadecimal - Base 16 numbering system

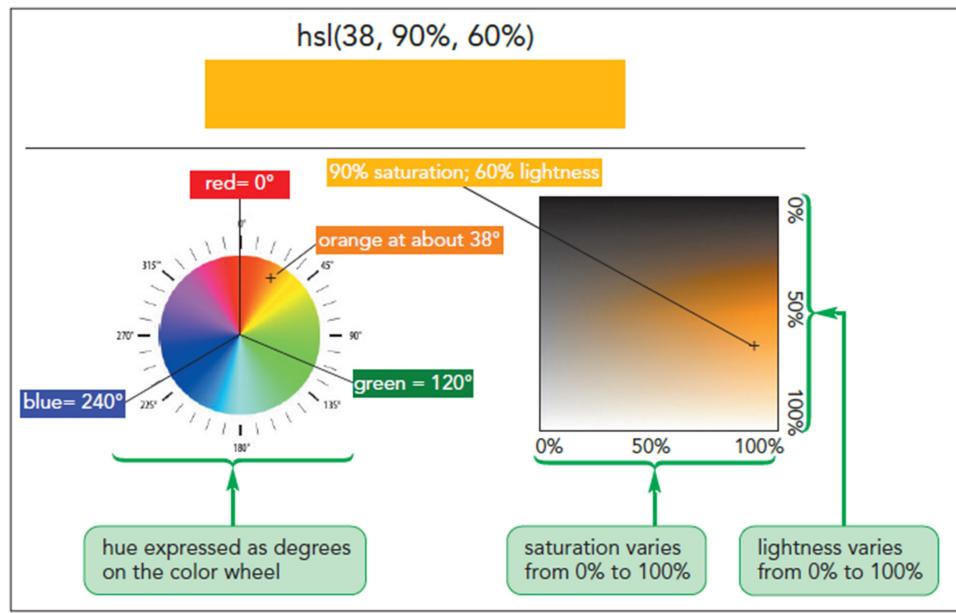
```
#p1 {background-color: #ff0000;} /* red */
a. #p2 {background-color: #00ff00;} /* green */
#p3 {background-color: #0000ff;} /* blue */
```

### HSL Color values

- a. **Hue** -Tint
- b. **Saturation** - Intensity
- c. **Lightness** - Brightness

Figure 2-9 Defining the color orange under the HSL color model

1.



© 2016 Cengage Learning

### Semi-opaque colors

1. **Opacity:** Define solidity of a color

*rgba(red, green, blue, opacity)*  
*hsla(hue, saturation, lightness, opacity)*  
where opacity sets the opacity of the color ranging from 0 (completely transparent) up to 1.0 (completely opaque)

## Setting Text & BG color

### 1. Text

- a. **color: color;**
- b. **Background-color: color;**

## Working with fonts

### 1. Typography

- a. Art of designing appearance of characters

### 2. Fonts

- a. Define style & appearance of each character
- b. *font-family: fonts;*
- c. *Fonts* = comma-separated list, **AKA font stack**
- d. **Specific font**
  - i. Identified by name, based on font definition file
- e. **Generic font**
  - i. Describe general appearance
  - ii. Ex: serif, sans-serif, monospace, cursive, fantasy

- **serif**—a typeface in which a small ornamentation appears at the tail end of each character
- **sans-serif**—a typeface without any serif ornamentation
- **monospace**—a typeface in which each character has the same width; often used to display programming code
- **cursive**—a typeface that mimics handwriting with highly stylized elements and flourishes; best used in small doses for decorative page elements
- f. • **fantasy**—a highly ornamental typeface used for page decoration; should never be used as body text

Because you have no control over which font definition file the browser will choose for a generic font, the common practice is to list specific fonts first, in order of preference, and end the font stack with a generic font. If the browser cannot find any of the specific fonts listed, it uses a generic font of its own choosing. For example, the style

```
font-family: 'Arial Black', Gadget, sans-serif;
```

- g. **Web font** - Definition font supplied in external file
  - i. Web safe fonts limit font choices
  - ii. Add the *@font-face* rule, it is then included in font stack
- h. The general syntax to include *@font-face* is

```
@font-face {  
    font-family: name;  
    src: url ('url1') format ('text1'),  
         url ('url2') format ('text2')...;  
    descriptor1: value1;  
    descriptor2: value2;... }
```

- i. Ex:

- ```

@font-face {
    font-family: Quicksand;
    i.      src: url('Quicksand-Regular.woff') format('woff'),
             url('Quicksand-Regular.ttf') format('truetype');
}
j. Name = name of font
k. URL = location
l. Text = optional description
m. Descriptor: value; optional style properties

```

## Setting font size

1. **Font-size:** size
  - a. **Absolute units**
  - b. **Relative units**
2. **Ems an rem**s
  - a. **Em** = scale with relative to parent font size
  - b. **rem** = scale with relative to the default font at the root directory

## Using Viewport units

1. Express length as percentage of width and height of browser
2. Vw, vh, vmin, vmax

## Sizing keywords

1. Xx-small etc

## Controlling spacing & kerning

1. **Kerning:** Space between character
  - a. letter-spacing: value;
2. **Tracking:** space between words
  - a. word-spacing: value;
3. **Leading:** Space between lines
  - a. line-height: size;
4. **Text spacing (indent)**
  - a. text-indent: size;

## Font styles

1. font-style: type;
2. Font-weight: weight; < bold formatting

|                  |                                                       |                                              |
|------------------|-------------------------------------------------------|----------------------------------------------|
| Decoration       | <i>Text-decoration: type;</i>                         | Type=none, underline, overline, line-through |
| Transform        | <i>text-transform: type;</i>                          | Type=capitalize, uppercase, lowercase, none  |
| Font variant     | <i>font-variant: type;</i>                            | Type=normal/small-caps                       |
| Horizontal align | <i>text-align: alignment;</i>                         |                                              |
| Vertical align   | <i>Vertical-align: alignment;</i>                     |                                              |
| Combining        | <i>Font: style variant weight size/height family;</i> |                                              |

## Formatting Lists

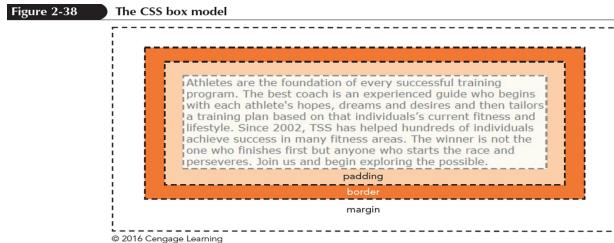
1. **List marker** - default browser style symbol
  - a. *list-style-type: type;*
  - b. Using pictures: *list-style-image: url(url);*

## Setting List Marker Position

1. *List-style-position: position;*

## Working with margins/paddings

2. **Box model**



3.

4. **Padding: size;**

5. **Margin: size;**

6. **Border-width: size;**

## Pseudo-class & pseudo-elements

1. Classifies element based on current status/position/use in document
2. **element: pseudo-class**
3. **Structural pseudoclass:** clasify based on location in structure
4. CSS selector
  - a. **Element::pseudo-element**

## Generating content with CSS

1. **element::before {content: text;}**
2. **element::after {content: text;}**

## Displaying attribute values

1. **content: attr(attribute);**
2. where attribute is an attribute of the selected element

## Inserting Quotation Marks

1. Blockquote & q elements
2. Content: **open-quote;**
3. Content: **close-quote;**

Week 5 2 o'clock, C1-C3 (Part 1 & Part 2)