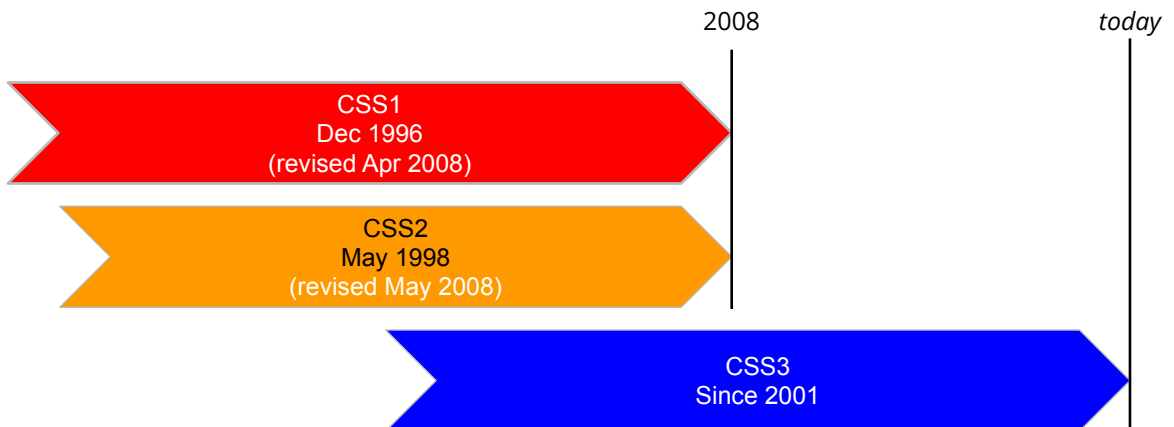


Styles in CSS

(Simplified) History of CSS



CSS Topics

- Why Learn CSS?
- Theming
 - Box Model
 - Spacing
 - Positioning
 - Dimension
 - Background, Border, Color
 - Font & Text
- CSS Selectors
- Layout (in a separate slide)

3

Applying CSS to HTML

```
<html>      Option 1: Internal
<head>
  <style>
    p {
      border: 2px solid red;
    }
  </style>
</head>
<body>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
</body>
</html>
```

```
<body>      Option 2: Inline (Not recommended)
<!-- inline style -->
<p style="border: 2px solid red">.....</p>
</body>
```

```
<html>      Option 3: Separate CSS USE THIS
<head>
  <link rel="stylesheet"
        href="mystyles.css">
</head>
<body>
  <p>Hello world</p>
</body>
</html>
```

```
/* in mystyles.css */
p {
  border: 2px solid red;
}
```

4

How to define styles?

- Styles are defined using a set of **rules**
- Each rule
 - begins with a selector to select the element(s) onto which the rule is applied
 - Specifies a group of properties to apply to the element(s)

```
selectorA { Rule 1  
  property1: value;  
  property2: value;  
}
```

```
selectorB { Rule 2  
  property1: value;  
  property2: value;  
}
```

Complete list of CSS properties:

<https://www.w3.org/Style/CSS/all-properties.en.html>

5

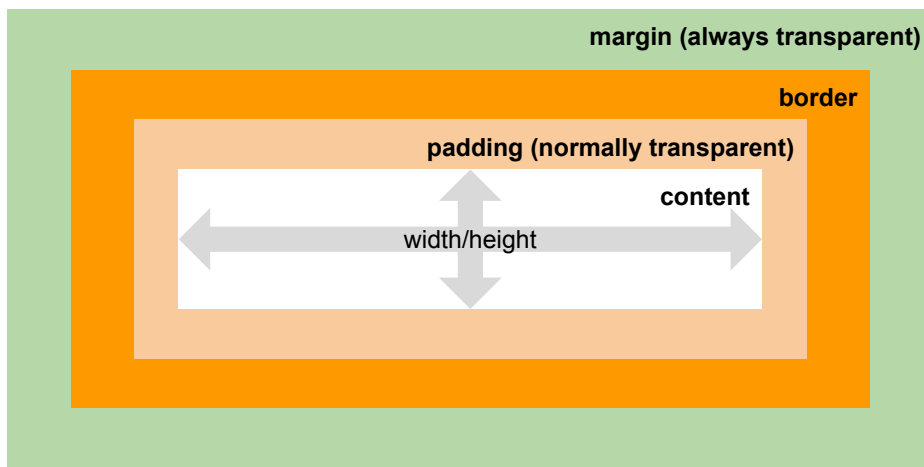
Too many CSS properties to memorize

Use VSCode *suggested completion* to help you find what you are looking for!



6

CSS Box Model



background-color paints the content, padding, and **border**

7

Web Dev Tools Demo
Elements, Inspector, Box Details

8

CSS Box Model: Padding (inside the border)

```
span {  
  padding: 4px;  
  border: 12px solid green;  
  background: beige;  
}
```

Sample Text

```
span {  
  padding: 16px;  
  border: 12px solid green;  
  background: beige;  
}
```

Sample Text

```
<span>Sample Text</span>
```

9

CSS Box Model: Margin (outside the border)

```
span {  
  margin-right: 2px;  
  border: 6px solid green;  
  background: beige;  
}
```

Sample Text

```
span {  
  margin-right: 8px;  
  border: 6px solid green;  
  background: beige;  
}
```

Sample Text

```
<span>Sample</span> Text
```

10

CSS Colors

12

Named Colors (140 standard names)

AntiqueWhite	BlueViolet	BurlyWood	CadetBlue
Coral	Crimson	DarkBlue	DarkGoldenRod
DarkGreen	DarkOrange	DarkRed	DarkSeaGreen
DodgerBlue	ForestGreen	Fuchsia	Gold
HotPink	IndianRed	Khaki	Lavender
LawnGreen	LightBlue	LightSalmon	LightSteelBlue
MediumAquaMarine	NavajoWhite	Olive	OliveDrab
Orange	OrangeRed	PaleGreen	YellowGreen

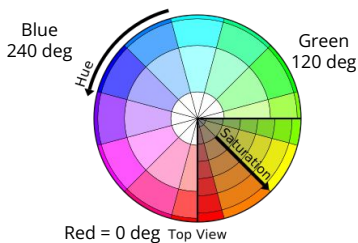
13

CSS Colors

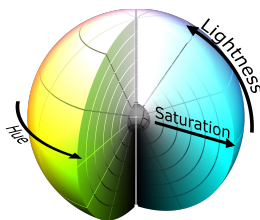
- RGB (0-255 per color) & Alpha Transparency
 - `rgb(155, 138, 73)`
 - `rgba(155, 138, 73, 0.6)`
- Hex String (00-FF per color)
 - `#C55` or `#FCA9`
 - `#9B8A49` or `#9B8A493F`
- HSL
 - `hsl(20, 85%, 30%)` or `hsla(20,85%,35%,0.7)`
 - *Benefit: easy to generate shades of a particular color (in code)*

Using these options if you attempt to generate color from code

HSL Colorspace



Red = 0 deg Top View

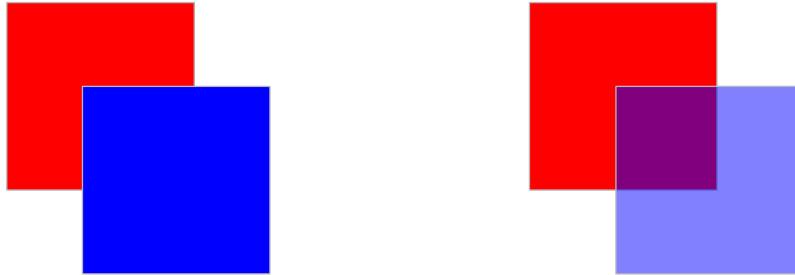


[HSL Color Picker \(CodePen\)](#)

[YUI HSL Color Picker](#)

	Description	Range of values
Hue	Color Tone	Red:0, Green:120, Blue:240
Saturation	How much "ink" in your paint	0%: no ink, 100%: max ink
Lightness	How much light available when you are viewing the color	0%: no light 100%: infinite amount of light

Color Transparency



	Opaque Blue	50% Transparent Blue
RGB	<code>rgb(0, 0, 255, 1.0)</code>	<code>rgb(0, 0, 255, 0.5)</code>
Hex String	<code>#0000FFFF</code>	<code>#0000FF7F</code>
HSL	<code>hsl(240, 100%, 50%, 1.0)</code>	<code>hsl(240, 100%, 50%, 0.5)</code>

16

HSL Practical Use: shade of color tones



`hsl(40,100%, 20%)`



`hsl(40,100%,30%)`



`hsl(40,100%, 40%)`



`hsl(40,100%, 60%)`



`hsl(40,100%, 80%)`

17

Applying CSS to HTML

HTML	CSS	Scope of Application
<pre><!-- by unique id --> <xyz id="ticket">some content</xyz></pre>	<pre>#ticket { padding-left: 2em; }</pre>	Only to one element
<pre><!-- by tag name --> <xyz>some content</xyz></pre>	<pre>xyz { font-weight: bold; }</pre>	All <code><xyz></code> tags in the document
<pre><!-- by class name --> <xyz class="weekend">some content</xyz></pre>	<pre>.weekend { border: 2px solid brown; }</pre>	All <code>.weekend</code> class in the document
<pre><!-- by other attributes --> <xyz anyattr="somevalue">some content</xyz></pre>	<pre>[anyattr] { background-color: white } [anyattr=somevalue] { font-weight: bold; }</pre>	All tags with this attribute in the document

18

CSS Specificity

19

CSS Selector Specificity

```
<div id="top" class="warn">
  Sample
</div>
```

```
// In mystyles.css
div {
  background: red;
}

#top {
  background: green;
}

.warn {
  background: yellow;
}
```

[Which selector wins?](#)

[Specificity Calculator](#)
(Higher score wins)

20

Example: Using CSS Selectors

```
<html>
<head>
<link rel="stylesheet" href="mystyles.css">
</head>
<body>
<span lang="en">Hello World<span>
<p>I am learning
<span id="abbrev">CSS</span></p>
</body>
</html>
```

```
mystyles.css
span {
  border-color: red
}
```

Hello World
I am learning CSS

```
mystyles.css
[lang] {
  border-color: red
}
```

Hello World
I am learning CSS

```
mystyles.css
#abbrev {
  border-color: red
}
```

Hello World
I am learning CSS

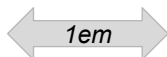
```
mystyles.css
[lang=de] {
  border-color: red
}
```

Hello World
I am learning CSS

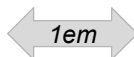
21

Font Size: 1 em

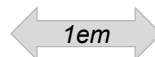
Font: Syncopate



Font: Roboto



Font: Droid Serif



1em: the width of uppercase M in the **current** font (traditional interpretation)

1em: the width of the **current** font (modern typography interpretation)

1 em: relative to the nearest parent's font

1 rem: relative to the root font

Use "em" for setting spacing around your text

Box/Element Positioning

	Description	Use top, bottom, left, right
position: static	Default	No
position: relative	Apply adjustment from its own default position	Yes: adjustment amount
position: absolute	Apply adjustment relative to nearest positioned parent	Yes: adjustment amount
position: fixed	Absolute position within the browser viewport	Yes

Relative Positioning: adjusted from own default position

```
<!-- HTML -->
<p>Text in <span>default</span> position</p>
<p>Text in <span class="sample">adjusted</span> position</p>
```

Shifted 5 pixels (down) from the default top, 10 pixels (left) from the default right

```
/* CSS */
.sample {
  color: red;
  position: relative
  top: 5px;
  right: 10px;
}
```

Text in default position
Text in **adjusted** position

[Demo on JSFiddle](#)

27

Absolute Positioning: adjusted from nearest parent

```
<!-- HTML -->
<p>Text in <span>default</span> position</p>
<p>Text in <span class="sample">adjusted</span> position</p>
```

```
/* CSS */
p {
  position: relative;
  border: 1px solid blue;
}
.sample {
  color: red;
  position: absolute;
  top: 5px;
  right: 10px;
}
```

- (Nearest) parent must use position other than static
- Shifted 5 pixels (down) from the parent top, 10 pixels (left) from the parent right

Text in default position

Text in position

adjusted

[Demo on JSFiddle](#)

28




CSS Cheat sheet(s)

<https://htmlcheatsheet.com/css/>

<https://devhints.io/css>




29



Advance Topics: CSS Selectors

Objective: selectively apply styles to specific part(s) of the DOM tree



30

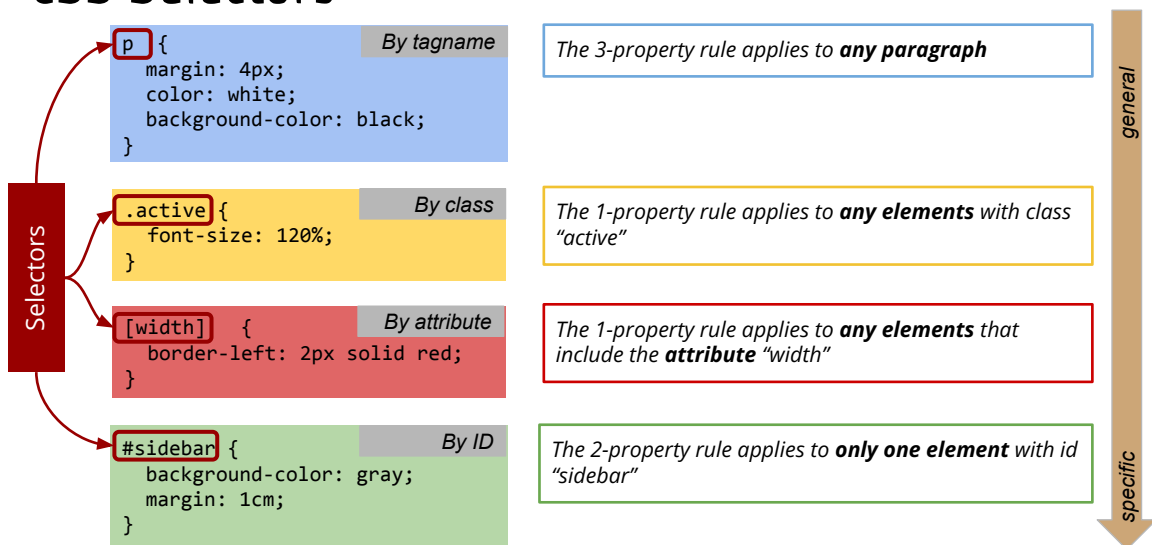
Being *selective* / (more) *specific*

The company will hire

- A student
- A GVSU student
- A GVSU student graduated before 2015
- A GVSU student graduated before 2015 with GPA at least 3.2
- A GVSU marketing student graduated with GPA at least 3.2 before 2015
- ... *and so on*

31

CSS Selectors



32

CSS “selectors” / “filters”

Various options to select *portions(s) of the DOM tree*. Select by:

- ID, tag name, CSS class (or combination of them)
- Attribute (with or without its value)
- Parent/Child relationship in the DOM tree, such as
 - All immediate children of ____
 - Any descendants of ____
 - The last grandchild of ____
 - and so on...
- Sibling relationship in the DOM tree
- *Permutations of all the above selectors*

33

A CSS selector targets DOM elements
(*it does NOT target text nodes*)

34

Selector Permutations: tag & class

```
/* in CSS */  
li.fruit {  
  color: red  
}
```

Apply only to list items with class `.fruit`

1. Strawberry
2. Raspberry Pi
3. Halle Berry

```
<!-- in HTML -->  
<ol>  
  <li class="fruit">Strawberry</li>  
  <li class="device">RaspBerry Pi</li>  
  <li>Halle Berry</li>  
</ol>
```

```
/* in CSS */  
li .fruit {  
  color: red  
}
```

Beware of *SPACE*.
This rule applies to **descendants** of ``
NOT `` themselves.

1. Strawberry
2. Raspberry Pi
3. Halle Berry

35

Selector Permutations: tag & attribute

```
/* in CSS */  
li[class] {  
  color: red  
}
```

Apply only to list items with class attribute set, regardless of its value

```
<!-- in HTML -->  
<ol>  
  <li class="fruit">Strawberry</li>  
  <li class="device">RaspBerry Pi</li>  
  <li>Halle Berry</li>  
</ol>
```

1. Strawberry
2. Raspberry Pi
3. Halle Berry

36

Selector Permutations: tag & attr & attr-value

```
/* in CSS */  
li[class*=t] {  
  color: red  
}
```

Apply only to list items with class attribute value containing "t"

```
<!-- in HTML -->  
<ol>  
  <li class="fruit">Strawberry</li>  
  <li class="device">RaspBerry Pi</li>  
  <li class="actor">Halle Berry</li>  
</ol>
```

1. Strawberry
2. Raspberry Pi
3. Halle Berry

37

Use Cases

Apply the particular styles **only to**:

- Odd rows (or even rows) of a table
- Paragraph immediately after heading level 2
- Input fields for password
- Empty list items
- Bold text inside the last row of a table
-

38

Selectors: by attribute(s)

- Objective: select elements with a particular attribute
- Selectors
 - [attr] ⇒ select elements that have attribute attr (regardless of its value)
 - [attr=val] ⇒ select elements whose attribute attr is set to “val”
 - [attr~=val] ⇒ select elements whose attribute attr **contains** “val” (whole word)
 - [attr*=val] ⇒ select elements whose attribute attr **contains** “val” (partial word)
 - [attr|=val] ⇒ select elements whose attribute attr **starts with** “val” (whole word)
 - [attr^=val] ⇒ select elements whose attribute attr **starts with** “val” (partial word)
 - [attr\$=val] ⇒ select elements whose attribute attr **ends with** “val” (partial word)
 - [attr1=val1][attr2*=val2] ⇒ use multiple attributes (**logical and**)

40

Selector by relative placement in DOM tree

41

Descendant/Younger Sibling Selectors

Types	Selector	Apply Rules to
Immediate children	<code>div > p { ...rules... }</code>	paragraphs which are an immediate children of div
Any descendant	<code>div p { ...rules... }</code>	paragraphs which are a descendant of a div (immediate children included)
Immediate (younger) sibling	<code>div + p { ...rules... }</code>	one paragraph (immediate younger sibling of a div)
Any younger sibling	<code>div ~ p { ...rules... }</code>	paragraphs which are younger sibling of a div (immediate sibling included)

Examples

42

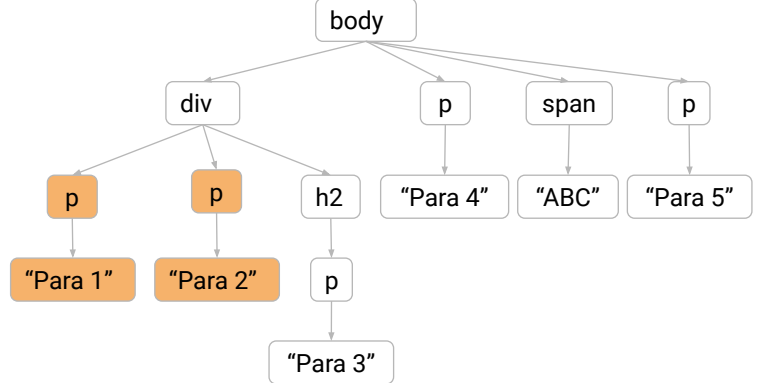
Target Element in *Complex Selectors*

Selector	Target Element
<code>div p</code>	Paragraph <p>
<code>h2 > p table</code>	Table <table>
<code>h2 ~ p table</code>	Table <table>
<code>#top div p .delay</code>	Elements with class .delay

43

Child (Immediate Descendant) Selector

```
<body>
  <div>
    <p>Para 1</p>
    <p>Para 2</p>
    <h2><p>Para 3</p></h2>
  </div>
  <p>Para 4</p>
  <span>ABC</span>
  <p>Para 5</p>
</body>
```

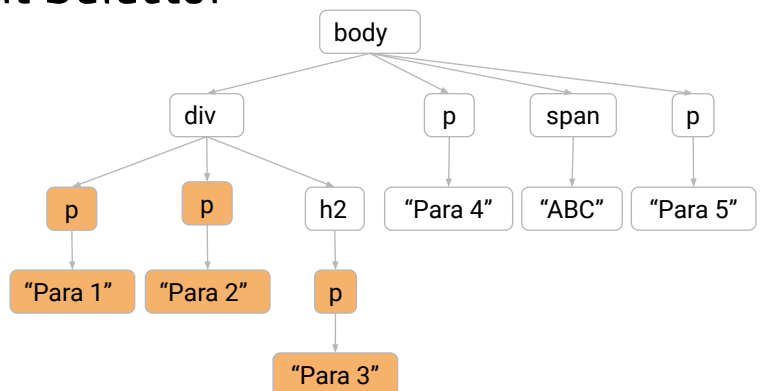


```
/* apply to paragraphs which are an immediate child of a div */
div > p {
  background-color: orange;
}
```

44

(Deeper) Descendant Selector

```
<body>
  <div>
    <p>Para 1</p>
    <p>Para 2</p>
    <h2><p>Para 3</p></h2>
  </div>
  <p>Para 4</p>
  <span>ABC</span>
  <p>Para 5</p>
</body>
```

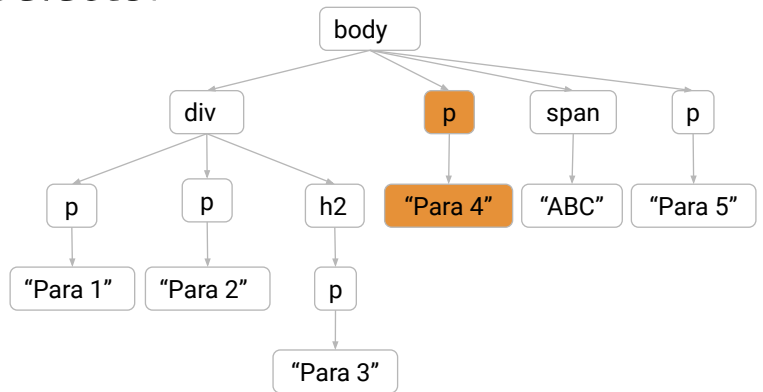


```
/* apply to paragraphs which are a descendant of a div */
div p {
  background-color: orange;
}
```

45

Immediate Sibling Selector

```
<body>
  <div>
    <p>Para 1</p>
    <p>Para 2</p>
    <h2><p>Para 3</p></h2>
  </div>
  <p>Para 4</p>
  <span>ABC</span>
  <p>Para 5</p>
</body>
```

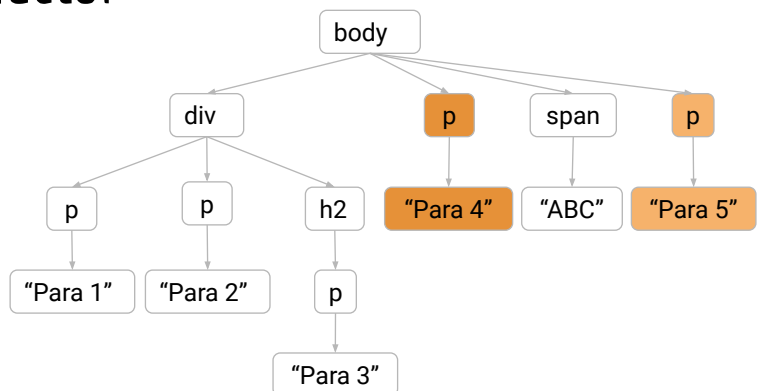


```
/* apply to paragraphs which are an immediate sibling following a div */
div + p {
  background-color: orange;
}
```

46

General Siblings Selector

```
<body>
  <div>
    <p>Para 1</p>
    <p>Para 2</p>
    <h2><p>Para 3</p></h2>
  </div>
  <p>Para 4</p>
  <span>ABC</span>
  <p>Para 5</p>
</body>
```



```
/* apply to paragraphs which are a (younger) sibling of a div */
div ~ p {
  background-color: orange;
}
```

47

Chain of descendant/siblings

- When the selector has multiple “groups” of pattern the CSS rules apply to the rightmost group

Selector & Rule(s)	Rule(s) applied to
<code>p li span { font-size: 125% }</code>	<code></code>
<code>div p ~ ol img { border: 2px dashed blue; }</code>	<code></code>
<code>div + p > ol.discount table { }</code>	<code><table></code>

48

Selector Modifiers :pseudo-classes

- Links (:link, :visited, :hover, :active)
- Input (:checked, :disabled, :enabled, :focus, :in-range, :out-of-range, :invalid, :valid, :optional, :required, :read-only, :read-write)
- Child order (:first-child, :last-child, :nth-child, :nth-last-child, :only-child)
- Of-Type order (:first-of-type, :last-of-type, :nth-of-type, :nth-last-of-type, :only-of-type)
- [Online reference](#) (look for “Pseudo-classes” on the left)

49

:first-child vs. :first-of-type

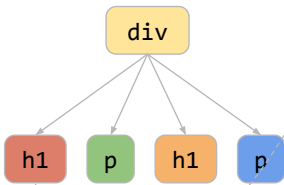
```
<div>  
  <h1>First Heading</h1>  
  <p>One paragraph</p>  
  
  <h1>Second Heading</h1>  
  <p>A bit longer paragraph</p>  
</div>
```

The **first** "daughter" in a family may be the **third** "child"

```
div p:first-child {  
  /* no matching element */  
}
```

```
div p:first-of-type {  
  /* applies only to "One paragraph" */  
}
```

```
div h1:first-child {  
  /* applies only to "First Heading" */  
}
```

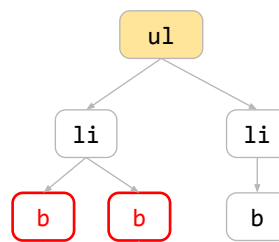


50

:first-child

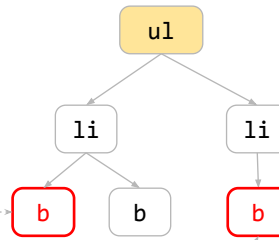
```
<ul>  
  <li>Test <b>one</b> and <b>two</b></li>  
  <li>Another <b>text</b></li>  
</ul>
```

```
li:first-child b {  
  color: red;  
}
```



```
<ul>  
  <li>Test <b>one</b> and <b>two</b></li>  
  <li>Another <b>text</b></li>  
</ul>
```

```
li b:first-child {  
  color: red;  
}
```



51

CSS3 :nth-child()

- :nth-child(4): select child #4
- :nth-child(odd): select children #1, #3, #5, ...
- :nth-child(even): select children #2, #4, #6, ...
- :nth-child(3n+1): select children #1, #4, #7, #10, ...

CSS Pseudo Classes Example

```
/* stylesheet: white on green */  
h1.active {  
  color: white;  
  background-color: green;  
}  
  
p.active:hover { font-weight: bold }
```

```
<!-- HTML doc -->  
  
<h1>First Heading</h1>  
<p class="active">This text is  
terse.</p>  
  
<h1 class="active">Second Heading</h1>  
<p>This text is slightly longer than  
the previous one.</p>
```

bold when mouse is over text

First Heading

This text is terse.

Second Heading

This text is slightly longer than the previous one.

rendered on browser

Pseudo Classes

- Practical use: **select only elements in a particular "state"**
- Link states (:link, :visited, :hover, :active)
- Input states (:checked, :disabled, :empty, :enabled, :focus)
- Positional (:first-child, :last-child, :nth-child(), :nth-last-child())

Pseudo Elements

- Practical use: **select only certain part of an element**
- Selectors ("::" in CSS3, ":" in CSS[1 | 2])
 - ::after
 - ::before
 - ::first-letter
 - ::first-line
 - ::selection

References: [Many more](#)

[Mozilla Dev Network](#)

55

Interactive CSS Selectors at
<https://www.w3schools.com/cssref/tryssel.asp>

57

Quiz: What is the effect of the CSS Styles?

```
<!-- HTML -->
<div>
  <div>Parent</div>
  <ol>
    <li>Child 1</li>
    <li>Child 2</li>
    <li>Child 3</li>
  </ol>
</div>
```

```
/* CSS */
div > ol {
  display: none;
}

div:hover ol {
  display: block;
}
```

See it live on [JSBin](#)

58

Quiz: What is the effect of the CSS Styles?

```
<!-- HTML -->
<div id="xyz">
  Before
  <span>Mid</div>
  After
</div>
```

```
/* CSS */
div > span {
  display: none;
}

div:hover > span {
  display: inline;
}

#xyz {
  background: limegreen;
  display: inline;
  padding: 8px;
}
```

See it live on [JSBin](#)

59

Media Query: @media

```
p {  
  font-size: 110%;  
}  
  
.box {  
  background: red;  
}  
  
@media (max-width: 600px) {  
  p {  
    font-size: 85%;  
  }  
  
  .box {  
    background: blue;  
  }  
}
```

Styles that apply when browser canvas width > 600px

Styles that apply when browser canvas width ≤ 600px

Other query params: (min|max)-height, *-resolution, *-aspect-ratio, *-color, and many more

62

Extensions to (standard) CSS

64

SCSS/SASS

- SCSS: Sassy CSS
- SASS: Syntactically Awesome Style Sheets
- Must be transpiled into .CSS
- Enhancements
 - Variables
 - Block Nesting
 - Functions (user-defined and builtin)
 - Mixins

65

SCSS to CSS Compiler



```
<html>
  <head>
    <link rel="stylesheet" href="mystyle.css" >
  </head>
  <body>
  </body>
</html>
```

66

SCSS Variables

```
/* SCSS */
$themeColor: hsl(60, 40%, 70%);

p {
  color: $themeColor;
  padding: 4px;
}

#top {
  border: 2px solid $themeColor;
}
```



```
/* pure CSS */
p {
  color: hsl(60, 40%, 70%);
  padding: 4px;
}

#top {
  border: 2px solid hsl(60, 40%, 70%);
}
```

67

SCSS Nesting

```
/* SCSS */
p {
  border-left: 2px solid red;

  > b {
    color: blue;
    font-size: 125%;
  }

  span {
    text-transform: uppercase;
  }
}
```



```
/* Pure CSS
p {
  border-left: 2px solid red;
}

p > b {
  border-left: 2px solid red;
  color: blue;
  font-size: 125%;
}

p span {
  border-left: 2px solid red;
  text-transform: uppercase;
}
```

68

SCSS @import

```
/* apptthemes.scss */  
$themeColor: hsl(60, 40%, 70%);
```

```
/* styles.scss */  
@import './apptthemes';  
  
p {  
  color: $themeColor;  
  padding: 4px;  
}  
  
#top {  
  border: 2px solid $themeColor;  
}
```



```
/* pure CSS */  
p {  
  color: hsl(60, 40%, 70%);  
  padding: 4px;  
}  
  
#top {  
  border: 2px solid hsl(60, 40%, 70%);  
}
```

69

SCSS Arithmetic Operators

```
/* apptthemes.scss */  
$themeColor: hsl(60, 40%, 70%);  
$gap: 8px;
```

```
/* styles.scss */  
@import './apptthemes';  
  
p {  
  color: $themeColor;  
  padding: $gap * 2;  
}  
  
#top {  
  border: ($gap / 4) solid $themeColor;  
}
```



```
/* pure CSS */  
p {  
  color: hsl(60, 40%, 70%);  
  padding: 16px;  
}  
  
#top {  
  border: 2px solid hsl(60, 40%, 70%);  
}
```

70

SCSS @extend class

```
/* SCSS */  
  
.big {  
  font-size: 200%;  
  font-family: Arial;  
}  
  
.bigRed {  
  @extend .big;  
  
  color: red;  
}
```



```
/* pure CSS */  
  
.big, .bigRed {  
  font-size: 200%;  
  font-family: Arial;  
}  
  
.bigRed {  
  color: red;  
}
```

71

SCSS @mixin and @include class

```
/* SCSS */  
  
@mixin big-arial {  
  font-size: 200%;  
  font-family: Arial;  
}  
  
.bigRed {  
  @include big-arial;  
  color: red;  
}  
  
p.big {  
  @include big-arial;  
  border-left: 1px solid black;  
}
```



```
/* pure CSS */  
  
.bigRed {  
  font-size: 200%;  
  font-family: Arial;  
  color: red;  
}  
  
p.big {  
  font-size: 200%;  
  font-family: Arial;  
  border-left: 1px solid black;  
}
```

72

SCSS @mixin with parameter(s)

```
/* SCSS */
@mixin big-arial($font-size) {
  font-size: $font-size;
  font-family: Arial;
}

.bigRed {
  @include big-arial(150%);
  color: red;
}

p.big {
  @include big-arial(125%);
  border-left: 1px solid black;
}
```



```
/* pure CSS */

.bigRed {
  font-size: 150%;
  font-family: Arial;
  color: red;
}

p.big {
  font-size: 125%;
  font-family: Arial;
  border-left: 1px solid black;
}
```

73

SCSS Color Functions

`$baseColor: hsl(340, 60%, 75%)`

<code>lighten(\$baseColor, 10%)</code>	<code>hsl(340, 60%, 85%)</code>
<code>darken(\$baseColor, 10%)</code>	<code>hsl(340, 60%, 65%)</code>
<code>saturate(\$baseColor, 25%)</code>	<code>hsl(340, 85%, 75%)</code>
<code>desaturate(\$baseColor, 25%)</code>	<code>hsl(340, 35%, 75%)</code>
<code>adjust-hue(\$baseColor, 30deg)</code>	<code>hsl(10, 60%, 75%)</code>
<code>complement(\$baseColor)</code>	<code>hsl(160, 60%, 75%) // adjust-hue 180deg</code>

74

SCSS vs. SASS

```
/* SCSS */  
$themeColor: hsl(60, 40%, 70%);  
  
p {  
  color: $themeColor;  
  padding: 4px;  
}  
  
#top {  
  border: 2px solid $themeColor;  
}
```



```
/* SASS */  
$themeColor: hsl(60, 40%, 70%);  
  
p  
  color: $themeColor;  
  padding: 4px;  
  
#top  
  border: 2px solid $themeColor;
```

SCSS Cheatsheet