CIS 192

Intro to Website Development

Module 5

Advanced Selectors

More Selectors

— — —

- **Universal selector**
  - The ***universal selector***, specified using an asterisk character (*), matches *all* elements in the webpage. The universal selector is implied when an element name is not specified.
- **Multiple selector**
  - The ***multiple selector***, specified using a comma (,) to separate selectors, matches all listed elements to apply a style rule.
- **Sibling selectors**
  - ***Sibling elements*** are elements that share the same parent element.

Selector	Example	Example description
element element	div p	Selects all \<p\> elements inside \<div\> elements
element>element	div > p	Selects all \<p\> elements where the parent is a \<div\> element
element+element	div + p	Selects the first \<p\> element that are placed immediately after \<div\> elements
element1~element2	p ~ ul	Selects every \<ul\> element that are preceded by a \<p\> element

More advanced selectors

- **:nth-child(n)** – Selects elements based on their position within a parent element's list of child elements. Useful for styling specific elements in a repeating pattern
- **:nth-of-type(n)** – Similar to :nth-child(n), but it only counts elements of the specified type.
- **:not(selector)** – Selects all elements that do not match the given selector. Useful for excluding specific elements from styling.
- **:first-child** – Selects elements that are the first child of their parent element. Useful for styling the first element in a container differently.
- **:last-child** – Selects elements that are the last child of their parent element. Useful for styling the last element in a container differently.
- **:first-of-type** – Selects the first element of its type among sibling elements. Useful for styling the first occurrence of a specific type of element.
- **:last-of-type** – Selects the last element of its type among sibling elements. Useful for styling the last occurrence of a specific type of element.
- **:empty** – Selects elements that have no children. Useful for styling empty elements, such as empty divs or paragraphs.
- **:only-child** – Selects elements that are the only child of their parent element. Useful for styling elements that are the sole child within a container.
- **:hover** – Selects elements when they are being hovered over by the mouse pointer. Useful for creating hover effects.
- **:focus** – Selects elements when they gain focus, such as when clicked on or tabbed to. Useful for styling form elements or interactive elements.
- **:checked** – Selects checkbox or radio button input elements that are checked. Useful for styling checked input elements, such as for custom checkboxes or radio buttons.
- **:enabled** – Selects form elements that are enabled (not disabled). Useful for styling enabled form elements differently from disabled ones.
- **:disabled** – Selects form elements that are disabled. Useful for styling disabled form elements differently from enabled ones.

Pseudo Element

```
.topic-games::before {
  content: '🎮 ';
}

.topic-weather::before {
  content: '🌥 ';
}

.topic-hot::before {
  content: url('../../media/examples/f:
  margin-right: 6px;
}
```

🎮 Game News: A new inFamous is not planned

🌥 Weather for Today: Heat, violent storms and twisters

🔥 Trending Article: Must-watch videos of the week

--- --- ---

- The ***pseudo-element selector***, specified with two colon characters (::) followed by a pseudo-element, matches parts of elements.
- They allow styles to apply to the first line or first letter of text of an element or to text that is selected by the user, or allow additional content to be inserted before or after an element.

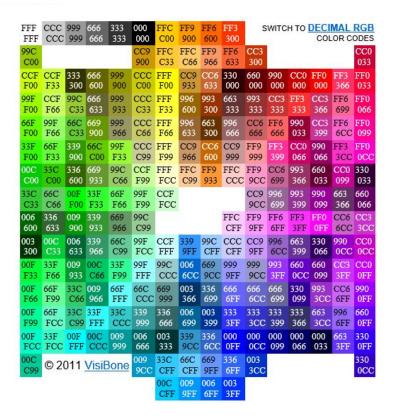| Pseudo-element	Explanation	CSS example
::*after*	Add content after the matched element.	`li::after { content: "<" }`
::*before*	Add content before the matched element.	`li::before { content: "***" }`
::*first-line*	Match the first line of text in a block element.	`p::first-line { color: red }`
::*first-letter*	Match the first letter of text in a block element.	`p::first-letter { font-size:200% }`
::*selection*	Matches the text selected by user.	`::selection { background: yellow }`

Common Properties

Color

Color property

The **color** CSS property changes the text color to a specified color value. A color value can be specified in several ways:

- CSS defines 140 color names. Ex: white, blue, black, gray, forestgreen, magenta.
- An **RGB color value** specifies a color using the **rgb(red, green, blue)** function by indicating the red, green, and blue intensities. Each intensity for red, green, and blue is between 0 and 255, where 0 is the lowest intensity and 255 is the highest.
  Ex: rgb(0, 0, 0) is black, rgb(0, 0, 255) is blue, rgb(255, 255, 0) is yellow, and rgb(255, 255, 255) is white.
- A **hexadecimal color** specifies a color using the #RRGGBB format by indicating the red, green, and blue intensities. Each intensity for red, green, and blue is between 00 and FF hexadecimal numbers, where 00 is the lowest intensity and FF is the highest.
  Ex: #000000 is black, #0000FF is blue, #FFFF00 is yellow, and #FFFFFF is white.
- An **HSL color value** specifies a color using the **hsl(hue, saturation, lightness)** function by indicating the hue, saturation, and lightness values. The hue value ranges between 0 and 360, and the saturation and lightness values range between 0% and 100%. Ex: hsl(0, 0%, 0%) is black, hsl(120, 100%, 50%) is green, and hsl(0, 100%, 25%) is dark red.

RGB – Hexadecimal Code

RGB = Red Green Blue

Hex Codes = #xxxxxx
- Values 0-9 and a-f
  - 0 = zero and f = 255 (max)
- #000000 = black
- #ffffff = white

Use lowercase letters in all hex codes
There is no letter "o" in any hex code so if you see 0, it is always a zero.

https://www.w3schools.com/colors/colors_picker.asp

Background

Background properties

Every element in a webpage has a set of background properties. Common background properties include:

- The ***background-color*** property specifies the background color.
- The ***background-image*** property specifies a background image.
- The ***background*** property is shorthand for setting several of the element's background properties at the same time.

Floating

Float property

CSS properties float and clear control how text flows around other HTML elements.

The *float* property specifies whether the element will float to the right or left of the element's parent container, allowing text to flow around the element. Values for the float property include:

- **left** - Element floats to parent container's left side
- **right** - Element floats to parent container's right side
- **none** - Element does not float (default value)

The *clear* property can stop elements from floating. Values for the clear property include:

- **both** - No elements allowed to float
- **left** - No element allowed to float on parent container's left side
- **right** - No element allowed to float on parent container's right side
- **none** - Elements allowed to float (default value)

Font/Text Properties

Basic Units of Size and Length

pt, mm, in, cm, and pc

- These are "absolute" units. This means the same in CSS as it does elsewhere if, and only if, the output device has a high enough resolution
- Low-end devices and printers will not render the units correctly
- The same size on one device will not be appropriate on another

px

- Not related to any current settings
- Is not an absolute size
- Identified as the smallest dot that can be displayed
- Designed to display with sharp edges
  - Think about a photo you enlarge that becomes blurry

em

- Relative size
- Adjusts according to default font size
- May be different for each element within a document

rem

- Relative size
- font size of the root element of the document
- Constant throughout the document

Font-Properties

———

Many CSS properties control the font properties for displaying text. CSS font properties include:

- The **font-family** property specifies the font family, such as "Times New Roman" or serif.
- The **font-size** property changes the font size, such as 120%, small, or 12px.
- The **font-weight** property specifies the font weight, such as normal or bold.
- The **font-style** property changes the text style, such as normal, italic, or oblique.
- The **font-variant** property specifies the variant of the text, such as normal or small-caps.
- The **font** property is shorthand for setting several font properties at the same time. Ex: font: italic 12pt Georgia, serif;

Box Model

Box Model

- HTML elements take up space on the webpage. The **box model** describes the size of each element as a series of nested boxes. The box model is important to understand when considering design and layout.
  - *Content*: The innermost box contains the content of the element, such as text and images.
  - *Padding*: The padding box contains the content box and adds a transparent area around the content.
  - *Border*: The border box contains the padded content and adds an optionally colored area around the padding.
  - *Margin*: The margin box contains all three boxes and adds a transparent area around the border.