Notes

* A document is a collection of information that is structured using a markup language.
* Cascading Style Sheets (CSS) is a language for specifying how documents are presented to users.
* CSS is efficient for targeting specific element to apply the style.

|  |  |
| --- | --- |
| [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML) | for web pages |
| [XML](https://developer.mozilla.org/en-US/docs/XML) | for structured documents in general |
| [SVG](https://developer.mozilla.org/en-US/docs/Web/SVG) | for graphics |
| [XUL](https://developer.mozilla.org/en-US/docs/XUL) | for user interfaces in Mozilla |

CSS with HTML

* Alter elements through Javascript
* <div id="show\_more" style="display: none;>
* Additional text!
* </div>>
* <script>
* document.getElementById('show\_more').style.display = 'block';
* </script>

Importing a CSS file from within CSS

* @import "newstyles.css";

Why uses CSS

* Use CSS to define styles for your documents, including the design, layout and variations in display for different devices and screen sizes.

Advantages of External Style Sheet

* Helps avoid duplication
* Makes maintenance easier
* Allows you to make a site-wide change in one place

HTML & CSS

* HTML to describe the content of the document, not its style
* CSS to specify the document's style, not its content

**How CSS works**

* When a browser displays a document, it must combine the document's content with its style information
* It processes the document in two stages

First Stage

* The browser converts the markup language and the CSS into the DOM (Document Object Model).
* The DOM represents the document in the computer's memory. It combines the document's content with its style.

Second Stage

* The browser displays the contents of the DOM.

DOM

* A DOM has a tree-like structure. Each element, attribute and run of text in the markup language becomes a node in the tree structure.
* The nodes are defined by their relationship to other DOM nodes.
* Some elements are parents of child nodes, and child nodes have siblings.
* Understanding the DOM helps you design, debug and maintain your CSS, because the DOM is where your CSS and the document's content meet up.

CSS Selectors

* Selectors are patterns used to select the element(s) you want to style.

Class selectors

* Multiple elements in a document can have the same class value.
* Type a full stop (period) before the class name when you use it in a selector.
* Use the class attribute in an element to assign the element to a named class.
* Example of class Selectors:

.key {

color: green;

}

ID selectors

* Use the id attribute in an element to assign an ID to the element.
* The ID name must be unique in the document.
* In your stylesheet, type a number sign (hash) before the ID when you use it in a selector.
* Example of id selectors

#principal {

font-weight: bolder;

}

Attribute Selectors

[disabled]

* Selects all elements with a "disabled" attribute.

[type='button']

* Selects elements with a "button" type.

[class~=key]

* Selects elements with the class "key" (but not e.g. "keyed", "monkey", "buckeye"). Functionally equivalent to .key.

[lang|=es]

* Selects elements specified as Spanish. This includes "es" and "es-MX" but not "eu-ES" (which is Basque).

[lang|=es]

* Selects elements specified as Spanish. This includes "es" and "es-MX" but not "eu-ES" (which is Basque).

a[href^="https://"]

* Selects secure links.

img[src$=".png"]

* Indirectly selects PNG images; any images that are PNGs but whose URL doesn't end in ".png" (such as when there's a query string) won't be selected.

Pseudo-classes selectors

* A CSS pseudo-class is a keyword added to selectors that specifies a special state of the element to be selected. For example :hover will apply a style when the user hovers over the element specified by the selector.
* Example of Pseudo-class selectors

selector:pseudo-class {

property: value;

}

* List of Pseudo-classes

:link

:visited

:active

:hover

:focus

:first-child

:last-child

:nth-child

:nth-last-child

:nth-of-type

:first-of-type

:last-of-type

:empty

:target

:checked

:enabled

:disabled