Survey of CSS

Relevant Links

- Semantic Code[1]
- devdocs[2]
- List of CSS properties[3]
- Learn HTML&CSS[4]
- MDN's links for learning CSS[5]
- "Can I use" site[6]
- Details on CSS Layout[7]
- Free online books[8]

Notes

CSS stands for *Cascading Style Sheets*.

CSS is used to control the *appearance* of the web page.

Here are some key notions, before we look at some examples:

rules A rule consists of a "selector" and series of property-value specifications. It indicates that those elements that match the selector should have those properties set to those values.

selectors Selectors[9] "target" certain elements in the page. For instance a selector could say: "I want to look at the `li` elements that have a class of `todo` and are contained inside an element with and id of `main`." This would be expressed as `#main li.todo`.

There are a great many types of things that can be targeted via selectors (both in CSS and in Javascript as we will see when we discuss jQuery). For instance:

- The first paragraph within a div.
- Every second table row.
- Anchor links whose corresponding urls end in ".pdf".
- A paragraph that immediately follows a paragraph with class "red".
- Any list item with class "selected" that is within an unordered list that is somewhere within the element with the id of "heading".

[1] https://boagworld.com/dev/semantic-code-what-why-how/
[2] http://devdocs.io/
[3] https://developer.mozilla.org/en-US/docs/Web/CSS/Reference
[4] http://learn.shayhowe.com/html-css/
[5] https://developer.mozilla.org/en-US/learn/css
[6] http://caniuse.com/
[7] https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout
[8] https://github.com/vhf/free-programming-books/blob/master/free-programming-books.md#html--css
[9] https://developer.mozilla.org/en-US/docs/Web/CSS/Reference#Selectors

properties and values CSS properties specify the appearance of elements. There is a long list of properties[10], and each property accepts a specific set of values. Each property-value pair must end with a semicolon.

specificity Each selector has a *specificity*[11] value, indicating how "precisely" it is pinpointing the elements. For instance, targetting a specific id is more "specific" than targetting a specific class, which is more "specific" than targetting a specific HTML tag.

If two rules try to apply the same property to an element, the rule with higher specificity wins. If they have equal specificity, the rule that appears later in the file wins.

Browsers also provide their own rules. In general user-provided rules take precedence.

inheritance Many of the properties are *inherited*[12] from a parent node to its children. For example specifying the font family in a parent node enforces that setting to all its children as well. This behavior lends cascading style sheets their name.

box model The box model[13] specifies how different properties specifying dimensions of elements should behave. It consists of *content width*, *padding*, *border* and *margins*.

- Each element has a certain width (and height), either enforced directly or through its parent or simply expanding to contain its contents.
- The element is further extended via padding specifications on its four sides. The extra space thus created is still part of the element (e.g. you can click on it if the element is clickable).
- The border, if any, goes around the padded area.
- The margins specify spacing that this element should have from its neighbors. If both an element and its neighbor has margin settings, only one of these margins will apply (whichever is larger).

layout mode At any given time, depending on the element and its settings, the browser is in one of four main layout modes[14]: *block*, *inline*, *table* and *positioned*.

- Block mode treats the element as a block with a width and height. Unless otherwise specified, block elements are positioned below each other. Floating can alter that behavior.
- Inline mode treats the element as if it was text. It is placed in line with the text surrounding it.
- Table mode is used when laying out tables.

[10] https://developer.mozilla.org/en-US/docs/Web/CSS/Reference
[11] https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity
[12] https://developer.mozilla.org/en-US/docs/Web/CSS/inheritance
[13] https://developer.mozilla.org/en-US/docs/Web/CSS/box_model
[14] https://developer.mozilla.org/en-US/docs/Web/CSS/Layout_mode

- The positioned mode allows other forms of placing of elements, that deviate from the normal flow. It is related to the position property setting, whose values can be static (the normal behavior of elements), relative (positioning the element relative to its normal position), fixed (positioning of the element relative to the page) and absolute (positioning an element relative to its closest relatively positioned ancestor). See this page[15].

floats Floats[16] are a difficult concept to work with at first. A "floating" element is taken out of the normal flow and placed along the left or right side of the page, while other elements wrap around it.

Floats are especially important for creating the illusion of columns. For example we can create two columns by floating the first one to the left, giving it a fixed width, and having the second one with a left margin equaling that width.

flexbox and grid Flexbox[17] and Grid[18] layouts are the modern ways to structure a web-page's look and feel.

There are too many properties to list here, but we will list the **various selectors** we can use to target elements:

aTag An HTML tag when used as a selector targets all elements in the page with that tag.
#anId Targets only the element with that specific id.
.aClass Targets all elements with this class.
sel1sel2 Putting selectors right next to each other indicates that they should all apply to an element for it to be targetted. For example div#main targets only a div element with an id of main, and nothing else.
sel1 sel2 A space between two selectors indicates that the second selector should target descendants of elements matched by the first selector. For instance .todo p targets paragraph elements only when they appear within an element with class "todo".
sel1, sel2 A comma between two selectors indicates that this rule should apply if either selector applies.
sel1 > sel2 Targets elements that match the second selector, that are children (immediate descendants) of elements that match the first selector.
sel1 ~ sel2 Targets elements that match the second selector, that are siblings of elements that match the first selector.
sel1 + sel2 Targets elements that match the second selector, that are adjacent siblings (immediately follow) of elements that match the first selector.

[15] http://learnlayout.com/position.html
[16] https://developer.mozilla.org/en-US/docs/Web/CSS/float
[17] https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox
[18] https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Grids

[attr=value] Attribute selectors target elements with a specific value for a specific attribute. There are a number of variants[19] depending on the relation between the specified value and the element's value.

::pseudo There are a few pseudo-elements[20] that can be targetted this way, for instance the first line in a paragraph or the first letter in a paragraph.

:pseudo These are pseudo-classes[21] and represent specific behavior of the targetted elements. For instance, you can use :hover to indicate something that should happen only when the element is hovered over by the mouse, you can use :visited to highlight links that have already been followed, and so on.

For an example of how a CSS stylesheet might look like, have a look at the CSS file[22] used for our class notes.

[19] https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors
[20] https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements
[21] https://developer.mozilla.org/en-US/docs/Web/CSS/pseudo-classes
[22] https://github.com/skiadas/skiadas.github.io/blob/master/css/course.css