**CSS Declarations**

CSS Declarations are formed through properties, which are defined thru values. These blocks of declaration combined with selectors then altogether create rule sets (CSS Properties, n.d.).

\*Properties:

1. Shorthand properties - collection of values which allows identifying several elements. As stated in its definition on the CSS specification, shorthand properties identifies and groups together properties which targets or define the same object. This then makes developers to work on the style sheet more efficiently (Shorthand properties, 2018).

Example: border properties, margin properties

Note: In this code, the background color will be overridden by the Doge.png image.

background-color: red;

background: url(“Doge.png”);

Explanation: If you don’t set the values properly, they are sent back to their initial value

\*Additional Information:

CSS is capable of universally defining the value of each and every property in the style sheet by utilizing the universal shorthand property which is all.

* Universal Property Values:

1. inherit – inherits the value from its origin element, otherwise known as parent element.
2. initial – it defines the value of the property by the default value of the browser’s local style sheet.
3. unset – makes the value of the property to its original value.
4. revert – it reverts the value of the property back to its form of origin by basing it to the style sheet of the user. If the user’s style sheet is not made available, the default style sheet from the user-agent is utilized.

Source: (Shorthand properties, 2018)

1. Vendor-specific extensions - used by browser vendors as a prefix for the names of experimental or non-standard CSS. Only after the extensions are standardized will the prefix be removed and stated as is. Although this is the case, browser vendors are veering away from it in order to put user-controlled flags or preferences. It is denoted by the character ‘-’ followed by the identifier that the browser vendor used and anoder ‘-‘ then a name which identifies what the property will do (Roberts, 2014).

Example: -moz-opacity, -webkit-transition, -opera-animation

1. Custom properties - aka CSS variables (experimental). These are customized properties indicated by the characters (--) as the prefix.

Example:

1. fallback value - in case there is no defined variable, a default value or fallback value is utilized.

2. As a code:

:root{

--kulay: rgb(255, 0, 0);

}

#backgroundInColor{

Background-color: var(--kulay);

}

Source: (Custom properties, 2017)

For more information about CSS properties, click <a href = “https://www.quackit.com/css/properties/”>here</a>

\*Values:

1. Value Processing – processes are declared, cascaded, specified, computed, used, actual values.

**Processes:**

1. Declared Value – refers to any property which is defined and associated to a certain element.
2. Cascaded Value – a value produced by the process of cascading.
3. Specified Value - every element has one specified value. It is created by processing cascaded value to the processes involving defaulting values.
4. Computed Value - calculated value as a result from a given set of elements.
5. Used Value - actual value used in a specified condition. This is created by processing computed value through different calculations then comparing it to a set of accepted values.

**\*numbers** - integer or real number (scientific or decimal notation)

\***dimensions** - combination of numbers and unit ex. length, angle, time, frequency, resolution

Source: (Etemad & Atkins [Eds.], 2016)

1. Value Types

- keywords

\*CSS - wide keywords ex. initial, inherit, unset - acts as either inherit or initial, refers to specs.

\*Property-specific keywords

- numbers

\*Length

\*Angle

\*Ratio – uses percentage (%)

- Strings

- Labels – can have multiple string values concatenated with one another.

- Colors

Source: (CSS value types, n.d.)

**Dimensions of Value Types:**

1. Length units

\*font - em, ex, ca, rem

\*viewport - percentage: vw, vh, vmin, vmax

2 Types of Length Units

1.) Relative Lengths - the length is based on another property’s length.

Example: em, ex, ch, vmin, vmax, percent (%)

2.) Absolute Lengths - the length is fixed. It is defined by a number value and then a unit.

Example: cm, in, pt

Source: (CSS Units, n.d.)

1. Angle units - used in some gradient and transform durations

\*deg, grad, rad, turn

1. Duration or Time units - typically used in transitions and animations

\*s, ms

1. Frequency units - initially introduced in CSS2 for the now deprecated aural media type. It was reintroduced in CSS3 but is mostly unused.

\*Hz, kHz

1. Resolution units - used in controlling the resolution of images as well as media queries.

\*dpi (dots per inch), dpcm (dots per centimeter), dppx (dots per px)

\*numbers with a percent symbol (%) as a suffix

1. URL and URIs - url() function with an absolute or relative (to the stylesheet) URL.

\*colors i.e. color keywords

\*RGB hexadecimal notation - denoted by pound symbol (#)

\*RGB functions - example: rgb(255, 0, 0), rgba () - alpha

\*HSL functions -refers to the angles of the color wheel, example: hsl(0, 100%, 0%) where 0 refers to the angle in the color wheel, 100% for the saturation and 0% for lightness.

\*currentcolor, transparent

Additional information:

\* Color Theory - sets the basic understanding for the associating colors with contrast, complementation and vibrancy.

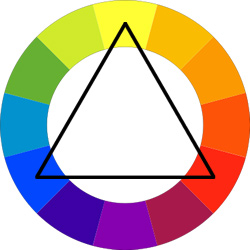
**Contrast** - sets the background and foreground apart by separating the colors used. **Give example for contrast**

**Complementation** - subjective way of relating colors with one another by choosing which combination of colors is visually appealing to the eye of the beholder.

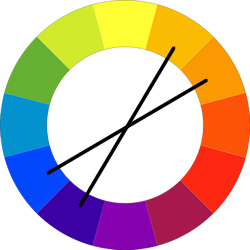
**Vibrancy** - refers to how the colors evoke emotions. This means that the brighter the color, the more energy it evokes while darker colors evokes less energy and more towards relaxing the viewer’s mind.

**Effective Color Scheme:**

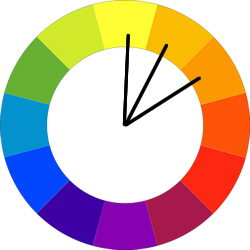
1. Triadic Color Scheme - colors which form an equilateral triangular shape in the color wheel compliments one another and have balanced vibrancy.



1. Split Complimentary Scheme - combination of colors opposite to one another are complimentary.



1. Analogous - combination of colors adjacent with one another.



Source: (Cannon, 2012)

**CSS Preprocessors, Frameworks, and Prolyfills**

* CSS Preprocessors – framework or library which generates CSS using a custom language syntax that typically features that don’t exist in pure CSS i.e variables, control flow, nesting, minutes.

Example:

**SASS** - abbreviation for Syntactically Awesome Style Sheets. This allows developers to utilize rules for nesting, variables but in order to have these features , Ruby must be installed. It is prefixed by the dollar symbol ($).

2 Syntax of Sass

1. Indented - otherwise known as Sass wherein it is identified with the file extension “.sass”. Instead of utilizing brackets, it makes use of indentation. It’s syntax is not completely supported by CSS
2. SCSS - abbreviation for Sassy CSS and identified by the file extension “.scss”. It’s syntax is completely supported by CSS.

Source: (Coron, 2017)

**LESS** - abbreviation for Leaner Style Sheets. Its syntax is very similar to CSS and prefixed by the at symbol (@) and defined by the file extension “.less”. In order to use it, Node.js must be available (Overview, n.d.).

**Stylus** - does not have a prefix and is defined by the file extension “.styl”. Its syntax is compatible with CSS with the additional option of utilizing colons, semi-colons and brackets optional (Croom, 2012).

Note: These are not compatible with most browsers. It will need some extensions in order to run it.

* CSS Frameworks – provides predefined CSS design functionality that can be reused, extended, or customized.

Example: Bootstrap, Bulma, Foundation

Source: (D., 2017)

* CSS Polyfills – provides features that developers expect browsers to provide natively (10 of the best polyfills, 2013).

Example: Modernizr, Selectivizr, html5shiv

References

10 of the best polyfills. (2013, February 13). Retrieved April 06, 2018, from https://www.creativebloq.com/html5/10-best-polyfills-2133015

Cannon, T. (2012, September 12). An Introduction to Color Theory for Web Designers. Retrieved April 06, 2018, from https://webdesign.tutsplus.com/articles/an-introduction-to-color-theory-for-web-designers--webdesign-1437

Coron, T. (2017, July 05). What is Sass? Retrieved April 06, 2018, from https://www.creativebloq.com/web-design/what-is-sass-111517618

Croom, J. (2012, March 09). Sass vs. LESS vs. Stylus: Preprocessor Shootout. Retrieved April 06, 2018, from https://code.tutsplus.com/tutorials/sass-vs-less-vs-stylus-preprocessor-shootout--net-24320

CSS Properties. (n.d.). Retrieved April 05, 2018, from http://www.htmldog.com/references/css/properties/

CSS Units. (n.d.). Retrieved April 05, 2018, from https://www.w3schools.com/cssref/css\_units.asp

CSS value types. (n.d.). Retrieved April 05, 2018, from http://docs.geoserver.org/latest/en/user/styling/css/valuetypes.html

Custom properties (--\*). (2017, December 19). Retrieved April 05, 2018, from https://developer.mozilla.org/en-US/docs/Web/CSS/--\*

Etemad, E. J., & Atkins, T., Jr. (Eds.). (2016, May 19). CSS Cascading and Inheritance Level 3. Retrieved April 05, 2018, from http://www.w3.org/TR/2016/CR-css-cascade-3-20160519/

Overview. (n.d.). Retrieved April 06, 2018, from http://lesscss.org/

Roberts, A. (2014, September 03). Vendor-specific Properties. Retrieved April 05, 2018, from https://www.sitepoint.com/vendor-specific-properties/

Shorthand properties. (2018, March 09). Retrieved April 05, 2018, from https://developer.mozilla.org/en-US/docs/Web/CSS/Shorthand\_properties