CSS- Frameworks have a better ,faster and more effective way to build responsive websites and web application.

CSS Framework working ?

* It provides a basic structure includes grid , interactive UI , patterns , web typography , tooltips ,buttons, form elements ,icons.
* To build a website / web application faster
* To validate the design hypothesis

1. Bootstrap -The most widely used free and open source CSS framework
   * Powerful responsive design
   * Built-in libraries of resource
   * Low learning curve
   * Quickly build prototypes
2. Foundation
   * Creative responsive design
   * Powerful email framework
   * Online webinar training support
   * Easy to sutomize
3. Pure – lightweignt CSS framework
4. Bulma – Opensource CSS framework based on Flexbox
5. Semantic UI
6. UI Kit
7. Materialize CSS
8. Milligram
9. Skeleton
10. Tailwind CSS
11. Spectre
12. Base
13. Picnic CSS
14. Mustard UI
15. Dead Simple Grid

CSS allows formatting to be separate from content and defines style instructions through **style rules**. Each rule has a selector and a set of property/value pairs. Multiple property/value pairs are separated by a ;,ex. selector { style-property1: value; style-property2: value }.

* The **selector** indicates what the rules applies to.
* The **properties** are the style characteristics that are being modified.
* The **value** is the new value for the property.

In the example rule below, all the h1 tags are centered on the page and the color red:

Graphical user interface, text

Description automatically generated

Diagram

Description automatically generated

**!important**

!important at the end of any property/value pair indicates that it is a prioritized styling rule.

Shape

Description automatically generated

**Style Rule Placements**

There are 3 ways to include CSS styling:

1. Inline (at the tag)
2. Internal (within the <style> tag) (less important than inline)
3. External (in a separate file)

The *closer* a rule is to the selected element, the *stronger* the precedence.

**Inline**

A picture containing logo

Description automatically generated

**Internal**

Text

Description automatically generated

**External**

By convention, style.css is usually used. There is no style tag used. You can include an external style sheet by using the following:



**4 Types of Selectors**

There are 4 types of selectors:

1. **HTML tag** — applies to any instance of the tag on the page, ex. h1
2. **Style class**— starts with a . and applies for class="x", ex. .my-class
3. **Id**— starts with a # and applies for id="x", ex. #my-id
4. **Pseudo-element**— modifies another selector using a :, ex. li:first-child

**Selector Combinations**

* a:hover — rule applies when the mouse is hovered over a link
* a,b — rule applies to all a’s and b’s (note that there is no space after the comma)
* a b— rule applies when b is contained within a, ex. h1 em applies for <h1><em>Here it is</em></h1>
* tag[attr=value] — rule applies to the specified tag only when the attribute is set to that specific value
* tag.x — rule applies for tag when its class="x" , ex. <h1 class="x">Blah</h1>

**Colors**

Colors can be used for text color(color), backgrounds(background-color), borders(border-color, ex. 2px solid #123456). CSS uses hex, rgb, or rgba colors.

* **Hexadecimal color**— begin with # and leverage RGB(red-green-blue). The syntax is #rrggbb. Each section ranges from 00 to FF, ex. red is FF0000.
* **rgb() color**— ex. rgb(12,120,255)
* **rgba() color**— adds transparency, ex. rgba(12,120,255,.4)

**Font & Text Properties**

* font-family
* font-size
* font-style (ex. italic)
* color
* text-align
* text-transform (case)
* text-decoration
* line-height

**HTML Blocks and the Box Model**

Blocks like <p>, <div>, <span> are often used to facilitate styling. Other than line spacing, there is no default spacing.

**The Box Model**

* **Border** — around an item
* **Padding** — the space between the content and thee boundary of the content
* **Margin** — the space between items

Diagram

Description automatically generated with medium confidence

**Postions**

* **Absolute** — relative to the first parent element with position set
* **Fixed** — relative to the page; may also use “z-index” property
* **Relative** — where it would normally go on a page; may need to set position to relative in order to make position absolute work for a child element

A screenshot of a computer

Description automatically generated with medium confidence

**Tricks**

* **Margin or padding**— exact specific values can be set for top, left, bottom, right, ex. padding: 3px, padding: 3px 5px, margin: 2px 3px 4px 5px
* box-sizing:border-box — forces thee height/width to be a specified size
* margin: 0 auto — centers a block
* position: absolute paired with left:0; right:0; — centers a block
* position: absolute paired with top:0; bottom:0;— full column height