02 CSS Selectors and Layouts

CSS can be included:

- inline with the html tag on any element with a style= attribute
- within the document under a <style>element
- through an external document reference with a <link href=> element *(preferred / best practice)*

CSS Rules include:

- selector
- declaration block **{ }**

Types of Selectors *(I want to apply a styling rule to something)***:**

universal - ***** - matches to any element

element - *element-tag-name* - matches to elements with the tag specified

id - # - matches to elements with the id specified

class - **.** - matches to elements with the class specified

descendant combinator - selectors is that they can be combined to specify the elements you want from the page.

descendant selector - *space* - combines two selectors such that elements matched by the second selector are selected if they have an ancestor element matching the first selector

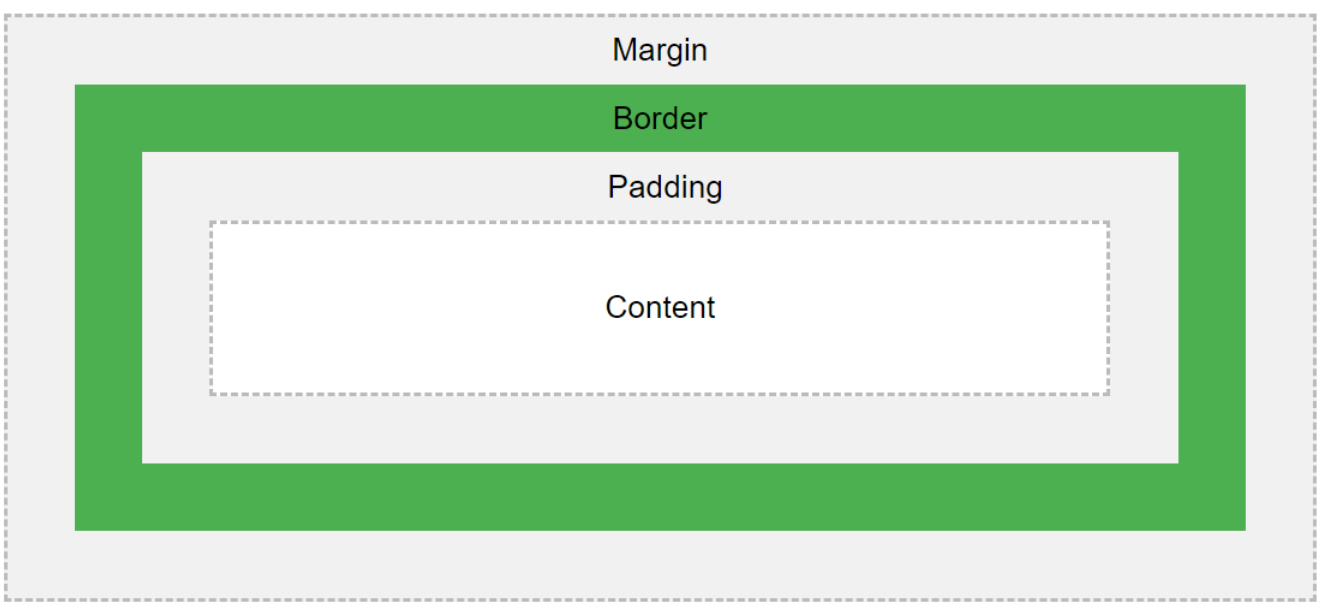
child selector - **>** - placed between two CSS selectors. It matches only those elements matched by the secondselectorthat are the direct children of elements matched by the first.

adjacent sibling selector - **+** - separates two selectorsand matches the second element only if it immediately follows the first element, and both are children of the same parent element .

general sibling selector - **~** - separates two selectors and matches the second element only if it follows the first element (though not necessarily immediately), and both are children of the same parent element.

pseudo-class - define a special state of an element of the DOM. But they don't point to an element by themselves . (e.g.a:hover)

The Box Model:



- Every element in web design is a rectangular box.
- The content, padding, border, and margin can be used to calculate the amount of space that an element takes up.
- Use the image above as an analogy to draw.
- Margin is the space outside an element. It does not affect the size of the box but affects other
- content that interacts with the box.
- Padding is the space inside an element.
- **The default width of a box isn't 100% but more so "whatever is left."**
- There are various circumstances where it is useful to set or not set a width.

Display Attribute

The display property specifies if and how an element should be displayed.

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

- A **block-level** element always **starts on a new line** and **takes up the full width available** — it stretches out to the left and right as far as it can.

  Examples of block level elements include:

  <h1>-<h6>
  <p>
  <form>
  <header>
  <footer>
  <section>

- An **inline** element **does not start on a new line** and **only takes up as much width as necessary.**

  Examples of inline elements include:

  <a>
  <img>

  This is the default for:

  <script>

  It **accepts margin and padding** while **ignoring height and width**.

- **inline-block** is very similar to inline, but it **allows you to set a width and height**.
- Use display:none to hide an element.

Positioning

The **normal flow** of a page is for elements to appear **left to right and top to bottom** based on the **order in which they appear** in the HTML document and the rules of block and inline display.

- **Static** position by default means the element conforms to **normal flow**.
- **Relative** position means relative to where it would otherwise be positioned in the normal flow.

  You can set the **top, right, bottom, and left** positioning attributes.

- **Absolute** position places the element **relative to the parent ancestor**—that is, the containing element — **exactly where you specify.**

  These **elements are removed from the flow** of the page.

  Setting both t**op and bottom, or both left and right** allows you to "stretch" an element's dimensions.

- **Fixed** position is **relative to the browser window and does not scroll** with the page.You can set the top, right, bottom, and left positioning attributes.

Floats

The float property specifies if the element should be taken from the normal flow and placed along the left or right side of the container. Text and inline elements wrap around it.

The float property can have several values:

- **none**: element does not float.
- **left**: element floats to the left of its container.
- **right**: element floats to the right of its container.
- **inherit**: element inherits the float direction of its parent.

Floated elements automatically display as block.

Floats should be avoided in modern web applications. In most cases, they were historically used to create multi-column layouts in older browsers.