Box Model:

Wraps around every HTML element

Content > Padding > Border > Margin

Content: Text & Images in box

Padding: Clear area around content, Transparent

Border: Border goes around padding and content

Margin: Clears area outside of border, transparent

When you set width and height of an element, you’re setting width and height of the content area

  320px (width of content area)  
+ 20px (left padding + right padding)  
+ 10px (left border + right border)  
**= 350px (total width)**  
  
  50px (height of content area)  
+ 20px (top padding + bottom padding)  
+ 10px (top border + bottom border)  
**= 80px (total height)**

Block boxes or inline boxes: type refers to how box behaves in relation to page flow and in relation toother boxes

Inner display type and outer display type

Outer Display Type: Block:

Box will break onto new line, width and height properties are respected, padding, margin, and border will cause other elements to be pushed away from box, if width is not specified, box will extend in inline direction to fill space available in container

H1 and p use block as outer display type by default

Outer Display Type: Inline:

Box will not break onto new line, width and height properties will not apply, top and bottom padding, margins, and borders will apply but not cause other inline boxes to move away from the box, left and right padding, margins, and borders will apply and cause other inline boxes to move away from box

A, span, em, and strong elements use inline by default

Inner display types: dictates how elements inside box are laid out

Block and inline layouts are defaults

Display: flex can still be used with display:block

Common CSS Shorthand: most common are background, border, font, padding, and margin, allows you to set values of multiple other css properties simultaneously

Shorthand properties will overwrite previous properties

Shorthand properties try not to force a specific order for the values of the properties they replace; works well when these properties use values of different types, but not when several properties can have identical values

Shorthand cases to remember: properties related to the edges of a box and properties related to the corners of a box

Edge Properties

1. Value syntax: border-width: 1em;
2. Value syntax: border-width: 1em 2em; 1st value = vertical, 2nd = horizontal
3. Value syntax: border-width: 1em 2em 3em; 1st value represents top edge, 2nd represents horizontal (left and right) 3rd represents bottom
4. Value Syntax: border-width: 1em 2em 3em 4em; 1: top, 2: right, 3: bottom, 4: left

Corner Properties

1-value Syntax: border-radius:1em; single value represents all corners

2-value Syntax: border-radius: 1em 2em; 1st = top left and bottom right, 2nd = top right and bottom left

3-values Syntax: border-radius: 1em 2em 3em; first represents top left corner, 2nd = top right and bottom left, 3rd = bottom right

4-value Syntax: border-radius: 1em 2em 3em 4em; top left > top right >bottom right > bottom left

CSS Syntax Structure:

Selector Declaration Declaration

H1 { color: blue; font-size:12px;}

Property: Value

CSS Pseudo-Classes:

Keyword added to selector that specifies a special state of selected elements (‘:hover’, etc.), consists of ‘:’ followed by pseudo-class name, also contains a pair of parentheses to define arguments ( :dir() ). Element pseudo-class is attached to is called anchor element

Combinators:

Combines other selectors in a way that gives them a useful relationship to each other and the location of content in the document

Descendant (represented by single space so long as they haven an ancestor element matching the first selector) : body article p {}

Child combinator (uses ‘>’, matches only those elements matched by second selector that are the direct children of first selector: article > p {}

Next-Sibling combinator (uses ‘+’, matches only elements matched by second selector that are the next sibling elements of the first selector): p + img

Subsequent-Sibling combinator (uses ‘~’ to select all siblings of element, even those that aren’t adjacent): p ~ img

Specificity:

Algorithm used by browsers to determine the css declaration that is most relevant to an element

Pseudo-Elements:

A keyword added to a selector that allows you to styles a specific part of the selected element: selector : : pseudo-element {property: value}

CSS Grid and Flex Structures:

Grid layout is a two-dimensional layout system with rows and columns, useful for more complex and organized layouts, must use display:grid on element

CSS Flexbox is a one dimensional layout. Useful for allocating and aligning space among items in a grid container. Flex works with different kinds of display devices and screen sizes and is easier to design and build responsive pages, must use display: flex property

Consider using grid layout when:

You have a complex design to work with and want to add gaps over block elements

Consider using flexbox when:

You have a small design to work with a few rows and columns, need to align an element, and don’t know how your content will look on the page and you want everything to fit in