CALIBRAINT TECHNOLOGIES

MILESTONE 4 - 05.09.2024

What and Why CSS
SASS and SCSS
Syntax
Basic Properties
Height, Width
Margins, Paddings, Border
Background
Text, Font, Image
Box Model, Display, Flex

What and Why CSS

- CSS stands for Cascading Style Sheets. It is the coding language that gives a website its look and layout.
- Along with HTML, CSS is fundamental to web design. Without it, websites would still be plain text on white backgrounds.
- CSS allowed several innovations to webpage layout, such as the ability to:
    1. Specify fonts other than the default for the browser
    2. Specify color and size of text and links
    3. Apply colors to backgrounds
    4. Contain webpage elements in boxes and float those boxes to specific positions on the page
- The “cascading” in CSS refers to the fact that styling rules “cascade” down from several sources. This means that CSS has an inherent hierarchy and styles of a higher precedence will overwrite rules of a lower precedence.

How does CSS works

- To understand the basics of how CSS works, you must first understand a little about modern HTML. Web developers lay out pages according to the "box model." A Web page is a series of boxes, each containing a discrete element. These boxes are nested, one inside another.
- For example, a page's header is a box, and it contains several smaller boxes comprising all the elements that make up a header: logo, navigation, social media buttons, shopping cart buttons, etc. Using CSS, a developer assigns styles to the "header" box. In this example, let's assume that the developer makes the text inside the header purple, Arial font and fifteen points high.
- Here's where the "cascading" part of cascading style sheets comes into play.

- The font styles applied to the header cascade down to all the elements contained inside the header. Elements containing text such as navigation, links or calls to action will all be purple, Arial and fifteen points high.

SASS and SCSS

- **SCSS (Sassy CSS)** is a syntax of SASS, offering a more CSS-like syntax. It includes features like variables, nesting, mixins, and inheritance while maintaining full compatibility with standard CSS. SCSS files use the .scss extension and enhance CSS with advanced functionalities for easier and more maintainable styling.

Features of SCSS

- **Variables:** Store reusable values like colors and fonts for consistent styling.
- **Nesting:** Nest CSS selectors in a hierarchical manner for better readability.
- **Mixins:** Create reusable chunks of styles, avoiding repetitive code.
- **Inheritance:** Use @extend to share styles between selectors, simplifying the codebase.
- **Partials and Importing:** Modularize CSS using @import, keeping styles organized and maintainable.

What is SASS?

- SASS (Syntactically Awesome Stylesheets) is a CSS preprocessor that extends CSS with features like variables, nested rules, mixins, and functions, allowing for more efficient and maintainable stylesheets. It compiles into regular CSS, providing enhanced functionality and organization for complex projects.

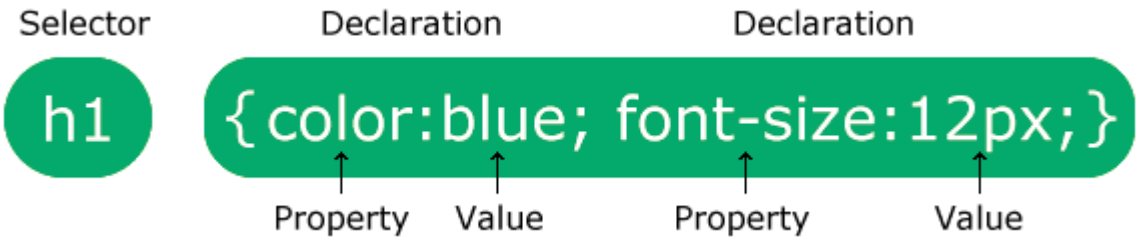
Features of SASS

- **Variables:** Store values like colors, fonts, or any CSS value, making styles reusable.
- **Nesting:** Nest CSS selectors in a way that follows the same visual hierarchy as your HTML.
- **Partials:** Use @import to include external Sass files, keeping styles modular and organized.
- **Mixins:** Define reusable chunks of code, allowing you to avoid repetitive styles.
- **Inheritance:** Use @extend to share a set of CSS properties from one selector to another, simplifying the style management.

Feature	SCSS	SASS
Syntax	CSS-like syntax with braces and semicolons	Indentation-based syntax without braces or semicolons
File Extension	`.scss`	`.sass`
Compatibility	Fully compatible with all versions of CSS	Requires a different syntax, not directly compatible with standard CSS
Flexibility	Familiar to those who know CSS	More concise and cleaner for some developers
Usage	Ideal for developers transitioning from CSS	Preferred by those who favor a streamlined syntax

Syntax

- A CSS rule consists of a selector and a declaration block.



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Basic Properties

- CSS has more than 200 properties some of the important properties are listed below
    - Text Properties
    - List Properties
    - Border Properties
    - Font Properties

TEXT PROPERTIES

Property	Description	Values
color	Sets the color of a text	RGB, hex, keyword
line-height	Sets the distance between lines	normal, *number*, *length*, *%*
letter-spacing	Increase or decrease the space between characters	normal, *length*
text-align	Aligns the text in an element	left, right, center, justify
text-decoration	Adds decoration to text	none, underline, overline, line-through
text-indent	Indents the first line of text in an element	*length*, *%*
text-transform	Controls the letters in an element	none, capitalize, uppercase, lowercase

LIST PROPERTIES

Property	Description	Values
list-style	Sets all the properties for a list in one declaration	*list-style-type*, *list-style-position*, *list-style-image*, inherit
list-style-image	Specifies an image as the list-item marker	URL, none, inherit
list-style-position	Specifies where to place the list-item marker	inside, outside, inherit
list-style-type	Specifies the type of list-item marker	none, disc, circle, square, decimal, decimal-leading-zero, armenian, georgian, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, lower-roman, upper-roman, inherit

BORDER PROPERTIES

Property	Description	Values
border	Sets all the border properties in one declaration	*border-width*, *border-style*, *border-color*
border-bottom	Sets all the bottom border properties in one declaration	*border-bottom-width*, *border-bottom-style*, *border-bottom-color*
border-bottom-color	Sets the color of the bottom border	*border-color*
border-bottom-style	Sets the style of the bottom border	*border-style*
border-bottom-width	Sets the width of the bottom border	*border-width*
border-color	Sets the color of the four borders	*color_name*, *hex_number*, *rgb_number*, transparent, inherit
border-left	Sets all the left border properties in one declaration	*border-left-width*, *border-left-style*, *border-left-color*
border-left-color	Sets the color of the left border	*border-color*
border-left-style	Sets the style of the left border	*border-style*
border-left-width	Sets the width of the left border	*border-width*
border-right	Sets all the right border properties in one declaration	*border-right-width*, *border-right-style*, *border-right-color*
border-right-color	Sets the color of the right border	*border-color*
border-right-style	Sets the style of the right border	*border-style*

FONT PROPERTIES

Property	Description	Values
font	Sets all the font properties in one declaration	*font-style*, *font-variant*, *font-weight*, *font-size/line-height*, *font-family*, caption, icon, menu, message-box, small-caption, status-bar, inherit
font-family	Specifies the font family for text	*family-name*, *generic-family*, inherit
font-size	Specifies the font size of text	xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, *length*, *%*, inherit
font-style	Specifies the font style for text	normal, italic, oblique, inherit
font-variant	Specifies whether or not a text should be displayed in a small-caps font	normal, small-caps, inherit
font-weight	Specifies the weight of a font	normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900, inherit **Careful, many of these are not supported!**

Height, Width

- The CSS height and width properties are used to set the height and width of an element.
- The CSS max-width property is used to set the maximum width of an element.
- The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.
- The height and width properties may have the following values:
    - auto - This is default. The browser calculates the height and width
    - length - Defines the height/width in px, cm, etc.
    - % - Defines the height/width in percent of the containing block
    - initial - Sets the height/width to its default value
    - inherit - The height/width will be inherited from its parent value
- The max-width property is used to set the maximum width of an element.
- The max-width can be specified in *length values*, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width).

Property	Description
height	Sets the height of an element
max-height	Sets the maximum height of an element
max-width	Sets the maximum width of an element
min-height	Sets the minimum height of an element
min-width	Sets the minimum width of an element
width	Sets the width of an element

Margin, Paddings, Border

- The CSS margin properties are used to create space around elements, outside of any defined borders.
- With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).
- CSS has properties for specifying the margin for each side of an element:
    - margin-top
    - margin-right
    - margin-bottom
    - margin-left
- All the margin properties can have the following values:
    - auto - the browser calculates the margin
    - *length* - specifies a margin in px, pt, cm, etc.
    - *%* - specifies a margin in % of the width of the containing element
    - inherit - specifies that the margin should be inherited from the parent element.

Property	Description
margin	A shorthand property for setting all the margin properties in one declaration
margin-bottom	Sets the bottom margin of an element
margin-left	Sets the left margin of an element
margin-right	Sets the right margin of an element
margin-top	Sets the top margin of an element

PADDING

- Padding properties
    - padding-top
    - padding-right
    - padding-bottom
    - Padding-left
- The padding property may be specified using one, two, three, or four values. Each value is a <length> or a <percentage>. Negative values are invalid.
    - When one value is specified, it applies the same padding to all four sides.
    - When two values are specified, the first padding applies to the top and bottom, the second to the left and right.
    - When three values are specified, the first padding applies to the top, the second to the right and left, the third to the bottom.
    - When four values are specified, the paddings apply to the top, right, bottom, and left in that order (clockwise).

- Paading applies to all elements, except table-row-group, table-header-group, table-footer-group, table-row, table-column-group and table-column. It also applies to ::first-letter and ::first-line.

Background

- The background property is a shorthand property for:
  - background-color
  - background-image
  - background-position
  - background-size
  - background-repeat
  - background-origin
  - background-clip
  - background-attachment

Value	Description
background-color	Specifies the background color to be used
background-image	Specifies ONE or MORE background images to be used
background-position	Specifies the position of the background images
background-size	Specifies the size of the background images
background-repeat	Specifies how to repeat the background images
background-origin	Specifies the positioning area of the background images
background-clip	Specifies the painting area of the background images
background-attachment	Specifies whether the background images are fixed or scrolls with the rest of the page

Text, Font, Image

TEXT COLOR

- The color property is used to set the color of the text. The color is specified by;
  - color name - like "red"
  - a HEX value - like "#ff0000"
  - an RGB value - like "rgb(255,0,0)"

TEXT ALIGNMENT

- Text-align - used to set the horizontal alignment of a text.A text can be left or right aligned, centered, or justified.
    - Eg : text-align: center;
- Text-align-last - property specifies how to align the last line of a text.
- Direction - change the text direction of an element
- Unicode-bidi - change the text direction of an element.
    - Eg : direction: rtl;
    - unicode-bidi: bidi-override;
- Vertical-align - sets the vertical alignment of an element.
    - Eg : vertical-align: baseline;
    - vertical-align: super;
    - vertical-align: sub;

TEXT DECORATION

- text-decoration-line - used to add a decoration line to text.
    - Eg : text-decoration-line: overline;
    - text-decoration-line: line-through;
    - text-decoration-line: underline;
    - text-decoration-line: overline underline;
- text-decoration-color - used to set the color of the decoration line.
    - Eg:text-decoration-line: overline;
    - text-decoration-color: red;
- text-decoration-style - used to set the style of the decoration line.
    - Eg: text-decoration-line: underline;
    - text-decoration-style: double;(solid,dotted,dashed)
- text-decoration-thickness
    - Eg:text-decoration-line: underline;
    - text-decoration-thickness: 5px;
- text-decoration

Property	Description
text-decoration	Sets all the text-decoration properties in one declaration
text-decoration-color	Specifies the color of the text-decoration
text-decoration-line	Specifies the kind of text decoration to be used (underline, overline, etc.)
text-decoration-style	Specifies the style of the text decoration (solid, dotted, etc.)
text-decoration-thickness	Specifies the thickness of the text decoration line

TEXT-TRANSFORM

- used to specify uppercase and lowercase letters in a text.
- Eg : text-transform: uppercase;

TEXT SHADOW

- adds shadow to text.
- Eg : text-shadow: 2px 2px;

FONT

- This property is a shorthand for the following CSS properties:
  - font-family
  - font-size
  - font-stretch
  - font-style
  - font-variant
  - font-weight
  - line-height
- it must include values for: <font-size> and <font-family>
- it may optionally include values for:
  - <font-style>
  - <font-variant>
  - <font-weight>
  - <font-stretch>
  - <line-height>
- font-style, font-variant and font-weight must precede font-size
- font-variant may only specify the values defined in CSS 2.1, that is normal and small-caps
- font-stretch may only be a single keyword value.
- line-height must immediately follow font-size, preceded by "/", like this: "16px/3"
- font-family must be the last value specified.

REFERENCE LINK : https://developer.mozilla.org/en-US/docs/Web/CSS/font

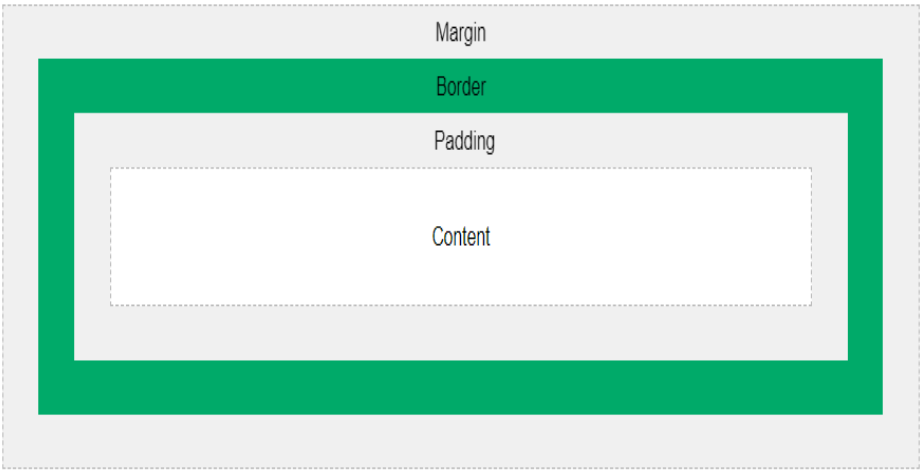
IMAGE

- The CSS images module defines the types of images that can be used (the <image> type, containing URLs, gradients and other types of images), how to resize them and how they, and other replaced content, interact with the different layout models.
- **Properties**
  - image-orientation
  - image-rendering
  - image-resolution
  - object-fit
  - object-position
- REFERENCE LINK https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_images

Box Model, Display, Flex

BOX MODEL

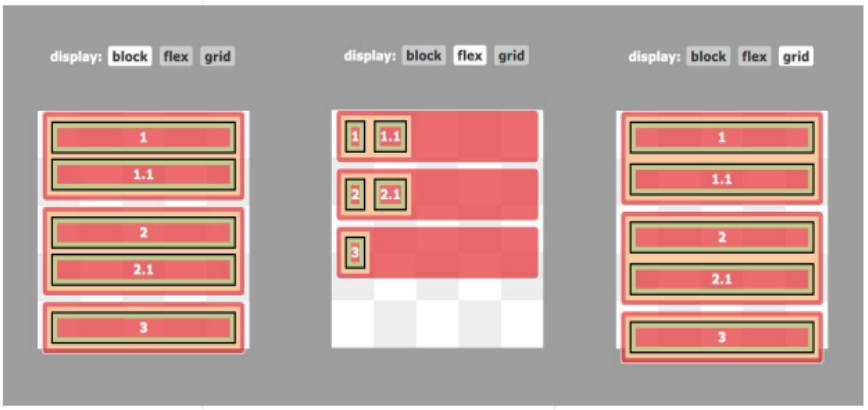
- The term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around every HTML element. It consists of: content, padding, borders and margins. The image below illustrates the box model:



- Content - The content of the box, where text and images appear
- Padding - Clears an area around the content. The padding is transparent
- Border - A border that goes around the padding and content
- Margin - Clears an area outside the border. The margin is transparent

DISPLAY

- The display CSS property sets whether an element is treated as a block or inline box and the layout used for its children, such as flow layout, grid or flex.
- The display property sets an element's inner and outer *display types*.
- The outer type sets an element's participation in flow layout.
- The inner type sets the layout of children.
- The formal syntax of display property is: Selector {display:} : Can be any valid display value. Can accept inline, block, inline-block, none, contents, flex, inline-flex, grid, inline-grid, etc.
- **1. Legacy Values:** These values can be classified into three parts. Block property: This makes an element full width and takes as much space available. It gives the element a minimum required height so whenever any variant of the block is used, the element will take full-width space and can no longer behave like an inline element. This breaks content before and after the block display. display: block; display: flex; display: grid;



- Inline properties: These are mostly used in text content where elements flow in line with other elements. This is a normal behavior of the text elements. Like span, strong, em, i, sup, sub, etc. Inline display type does not break content, instead, the content will be in the same line with other elements. Adding margin or padding to inline elements won't push other elements away.
    - display: inline;
    - display: inline-flex;
    - display: inline-grid;

- **2. Outside Values:** These display values work on the outer type of the element and behave as a flow layout.
    - display: block - It generates a block element box, when used in the normal flow, It generates line breaks both before and after the element.
    - display: inline - It generates one or more inline element boxes. They don't generate any line breaks anywhere. In normal flow, the next element will be on the same line if there is space.
- **3. Inside Values:** These display values work on the inner type of the element and work on the formatting of the context.
    - flow: it generates a normal flow layout using block and inline layouts. If its outer display type is inline or run-in, and it is used as a block or inline formatting context, it generates an inline box. Otherwise, it generates a block container box.
    - flow-root: It generates a block element box that creates a new BFC, defining where the formatting root lies.
    - table: These elements behave like HTML table elements.
    - flex: It behaves like a block element and formats its content according to the flexbox model.
    - grid: It behaves like a block element and formats its content according to the grid model.
    - ruby: It behaves like an inline element and formats its content according to the ruby formatting model.

- **4. List Item Values:** Display list-item generates a block box for the content and a separate list-item inline box. The single value of list-item behaves like a default list item. It can go along with list-style-type and list-style-position properties. Display list-item will add a default bullet to the element. This applies to the entire child and sub-child hierarchy.
- REFERENCE LINK - https://codecolorz.com/wp-content/uploads/notes/class-notes(How-to-use-display-property-in-css).pdf

FLEX

- The flexbox model provides for an efficient way to layout, align, and distribute space among elements within your document - even when the viewport and the size of your elements is unknown and/or dynamic.
- 1.The Flex-direction property controls the direction in which the flexitems are laid along the main axis. It may take any of four values. flex-direction: row || column || row-reverse || column-reverse;
- 2.The flex-wrap property can take on any of three values:flex-wrap: wrap || nowrap || wrap-reverse;
- 3.The flex-flow is a shorthand property which takes flex-direction and Flex-wrap values. flex-flow: row wrap; /*direction ‘row’ and yes, please wrap the items.*/
- 4.The justify-content property takes on any of the 5 values , justify-content: flex-start || flex-end || center || space-between || space-around
- 5.The align-items property is somewhat similar to the justifycontent property. Align-items can be set to any of these values: flex-start || flexend || center || stretch || baseline
- REFERENCE LINK - https://ohansemmanuel.github.io/assets/pdf/understanding-flexbox.pdf