WEEK 2
DAY 2

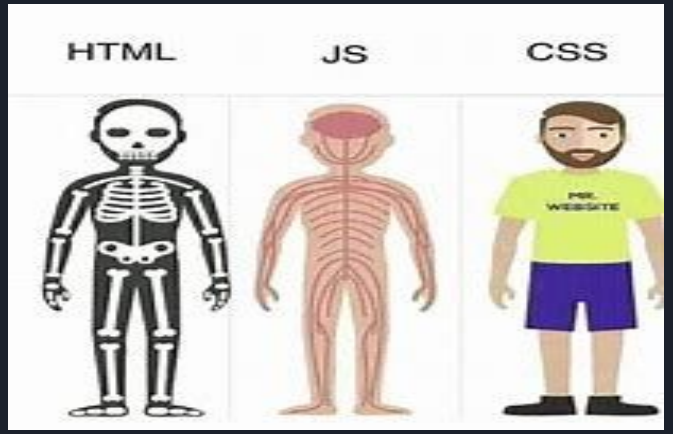
CASCADING STYLE SHEETS (CSS)

Introduction to CSS

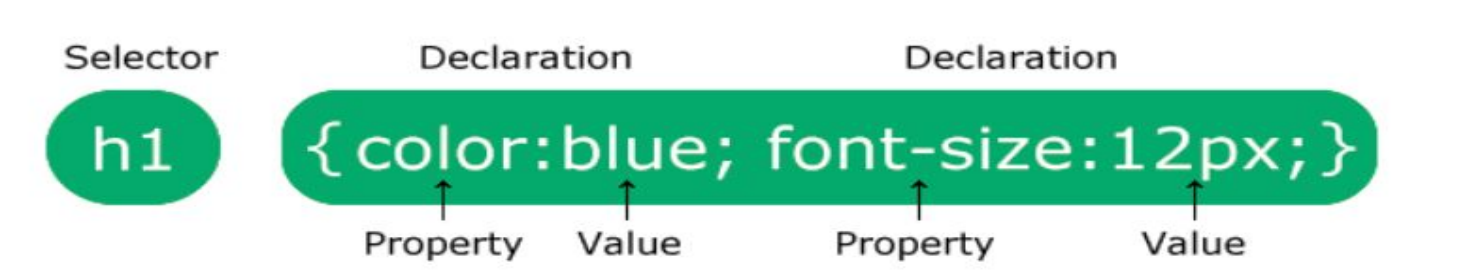
CSS is a stylesheet language we use to style an HTML document.

CSS describes how HTML elements should be displayed.

Along we styling benefit we get access to adding animations and make our page responsive for different devices using media queries.



Syntax and Semantics



Things to remember :

1. Selector(s)
2. Declaration(s) = Property : Value

Note: Don’t forget the curly braces

Syntax breakdown

- Selector(s)

  The selector points to the HTML element we want to style

- Declaration Block

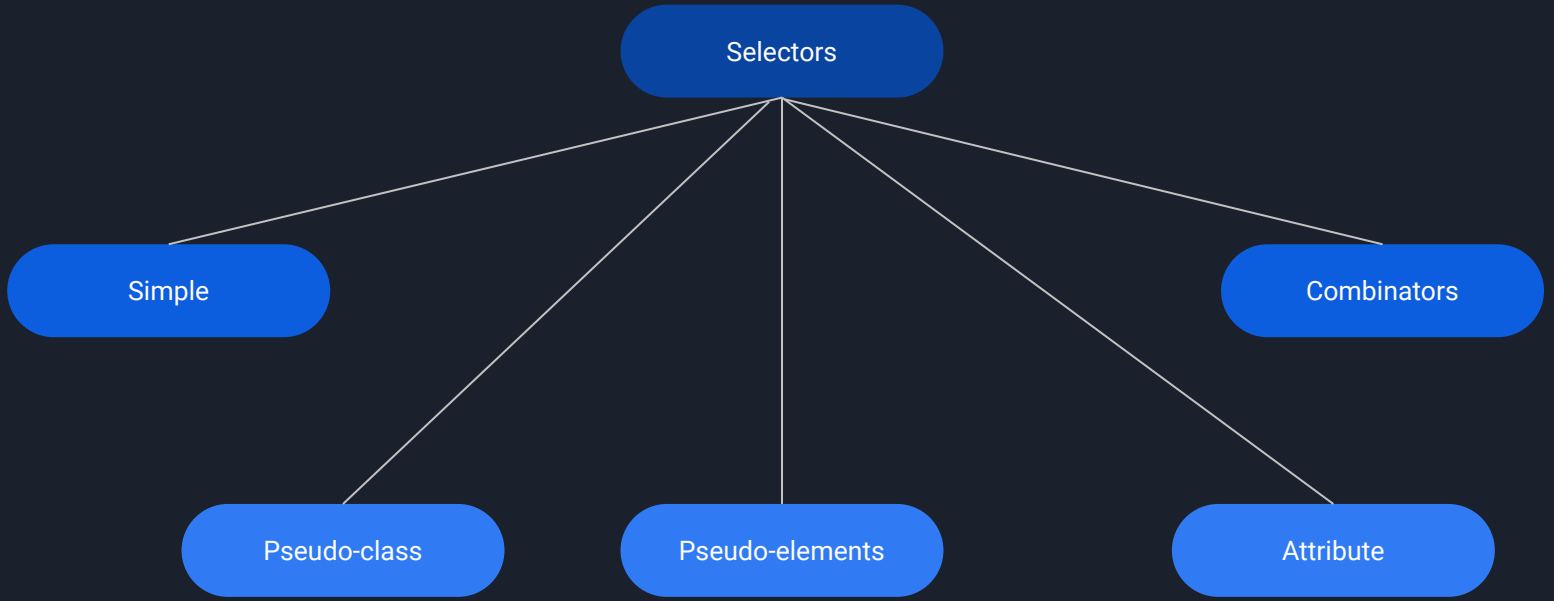
  The declaration block contains one or more declarations separated by semicolon (;)

- Declaration(s)

  Each declaration includes a CSS property name and a value, separated by a colon (:)

CSS Selectors

Selectors are used to find (or select) the HTML elements we want to style.



Types of Selectors

Simple Selectors

- Name
- id
- class

Ex: Name Selector

```
p {
  text-align: center;
  color: red;
}
```

Selectors

CSS id Selector

The id selector uses id attribute of an HTML element to select a specific element.

The id element is unique within a page, so the id selector is used to select one unique element !

```
#para1 {
  text-align: center;
  color: red;
}
```

Selectors

CSS class Selector

The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class , we use (.) period character, followed by the class name.

```
.center {
  text-align: center;
  color: red;
}
```

CSS Combinators Selector

A Combinator is something that explains relationship between the selectors

There are 4 different combinators in CSS :

- Descendant selector
- Child selector
- Adjacent sibling selector
- General sibling selector

CSS Combinators Selectors

Descendant Selector

The descendant selector matches all the elements that are descendants of a specified elements.

```
div p {
  background-color: yellow;
}
```

Child Selectors

This selects all the elements that are the children of a specified element (direct children).

```
div > p {
  background-color: yellow;
}
```

CSS Combinators Selectors

Adjacent Sibling Selector

This selector selects an element that is directly after specific element. Both elements should have same parent .

```
div + p {
  background-color: yellow;
}
```

General Sibling Selector

The general sibling selector selects all elements that are next siblings of a specified element.

```
div ~ p {
  background-color: yellow;
}
```

CSS Pseudo-classes

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

Syntax format

```
selector:pseudo-class {
  property: value;
}
```

Ex: anchor tag can hover, visited, active, unvisited state .

CSS Pseudo-elements

A CSS pseudo-element is used to style specified parts of an element.

Ex: Style the first letter, or first line, of an element

```
selector::pseudo-element {
  property: value;
}
```

Ex :

```
p::first-line
```

```
p::first-letter
```

```
h1::before
```

```
h1::after
```

CSS Attribute Selectors

It’s possible to style HTML elements that have a specific attributes or attribute values.

1. [attribute] Selector : used to select all the elements that have the specified attribute

```
[ class ] {
        color : red ;
 }
```

2. [attribute = ‘value’] Selector : This selector is used to select all the elements whose attribute has value exactly same as the specified value.

```
a[ href = ”https://google.com ”   ] {
        color : red ;
 }
```

CSS Attribute Selectors

3. CSS [attribute ~= “value”] Selector

This is used to select elements with an attribute value containing a specified word.

```
<img src="klematis.jpg" title="klematis flower" width="150" height="113">
<img src="img_flwr.gif" title="flower" width="224" height="162">
<img src="img_tree.gif" title="tree" width="200" height="358">
```

4. . CSS [attribute| = “value”] Selector

```
<h1 class="top-header">Welcome</h1>
<p class="top-text">Hello world!</p>
<p class="topcontent">Are you learning
CSS?</p>
```

Matches only either the whole word or value separated by hyphen(-)

CSS Attribute Selectors

5. CSS [attribute ^= “value”] Selector

Selects elements whose attribute value begins with specified value.

```
[class^="top"] {
  background: yellow;
}
```

```
<h1 class="top-header">Welcome</h1>
<p class="top-text">Hello world!</p>
<p class="topcontent">Are you learning CSS?</p>
```

6. CSS [attribute$=”value”] Selector

Selects elements whose attribute value ends with a specified value

```
[class$="test"] {
  background: yellow;
}
```

```
<div class="first_test">The first div element.</div>
<div class="second">The second div element.</div>
<div class="my-test">The third div element.</div>
<p class="mytest">This is some text in a paragraph.</p>
```

CSS Attribute Selector

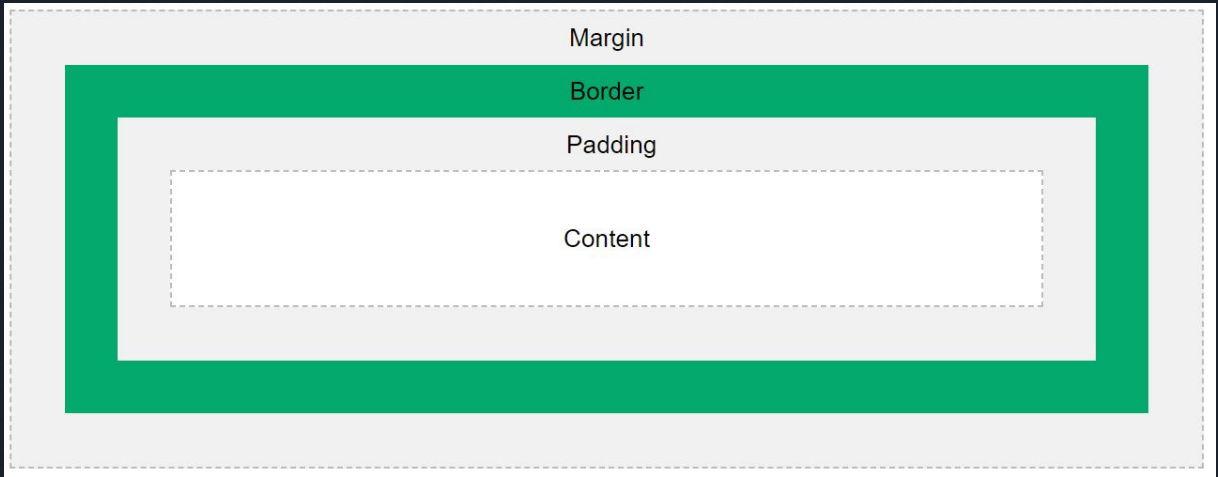
CSS [attribute*= “value”] Selector

This is used to select elements whose attribute value contains a specified value.

```
[class*="te"] {
  background: yellow;
}
```

```
<div class="first_test">The first div element.</div>
<div class="second">The second div element.</div>
<div class="my-test">The third div element.</div>
<p class="mytest">This is some text in a paragraph.</p>
```

CSS Box Model



The CSS box model is essentially a box that wraps around every HTML element.

It consists of : Content, Padding, Border, Margin

CSS Position

Elements in a page can be positioned in one of the 5 ways :

1. static

   HTML elements are positioned static by default . Cannot use L,R,T,B.

2. relative

   An element is positioned relative to its normal position. Here we can apply LRTB.

3. fixed

   An element is positioned relative to viewport, that mean, it always stays in same place even if page is scrolled. Ex: Navigation

CSS Position

4. absolute

   Element are positioned relative to the nearest positioned ancestor. If positioned element has no positioned ancestors, it uses document body.

   Positioned : means that element whose position is anything except static.

5. sticky

   An sticky element toggle between relative and fixed, depending upon scroll position.

   It remains in relative until certain offset is met and then becomes fixed.

CSS Flexbox

The flexbox layout model, makes it easier to design flexible responsive layout structures without use of float and positioning .

Before this layout model there were 4 layout modes :

1. Block, for sections in a webpage
2. Inline, for text
3. Table, for 2-D table data
4. Positioned, for explicit position if an element.

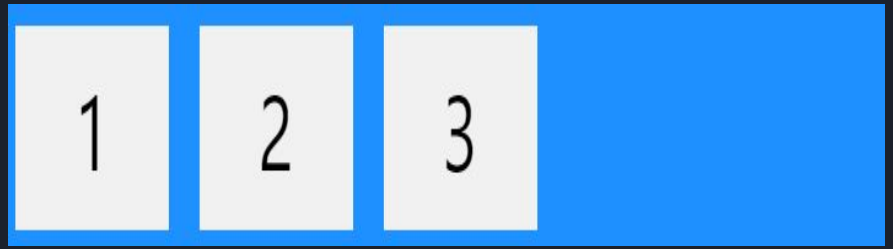
Flexbox

Flexbox is simply based on one container containing element.

Element that acts as container is known as : Flex container

Elements which are contained are called : Flex items

Output :



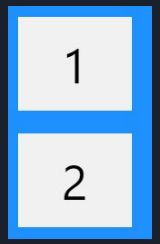

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

```
.flex-container {
  display: flex;
}
```

Flex container property

1. flex- direction (column | column-reverse | row | row-reverse)

   The items can be aligned vertically or horizontally. The default behaviour is horizontal alignment.

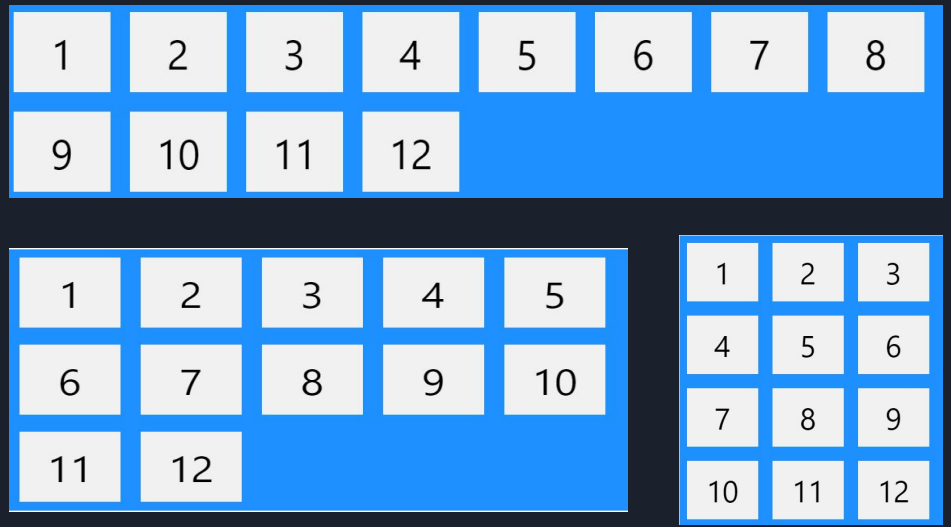



```
.flex-container{
    display: flex ;
    flex-direction: column;
}
```

Flex container property

2. Flex-wrap property (wrap | nowrap | wrap-reverse)

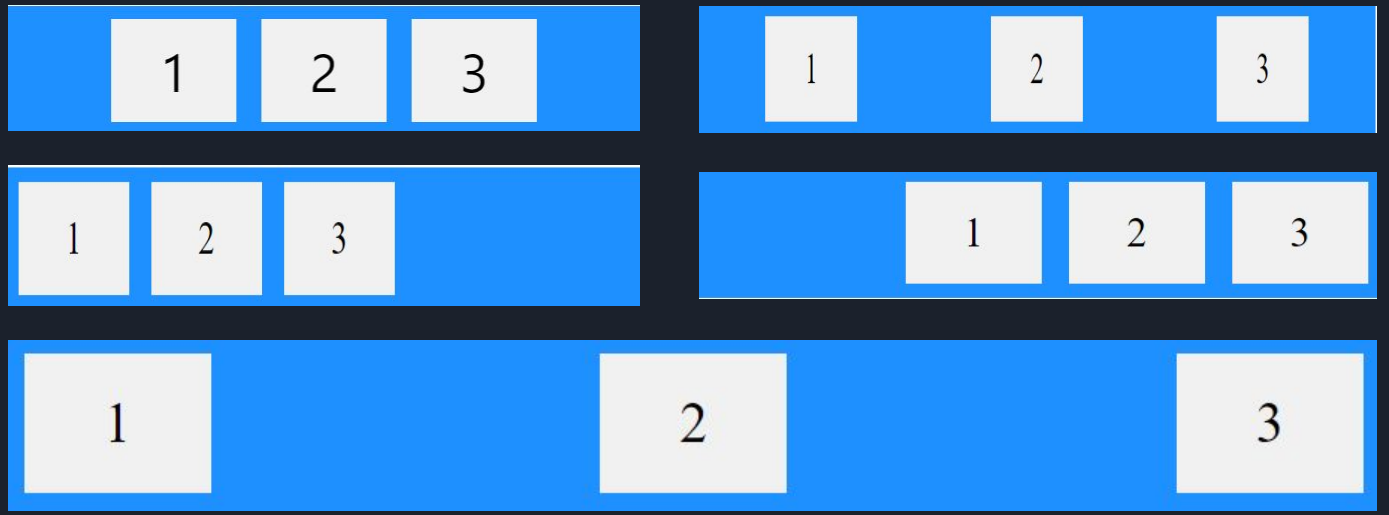
The flex-wrap property specifies whether the flex items should wrap or not



Flex-container property

3. justify-content Property (center | flex-start | flex-end | space-around | space-between)

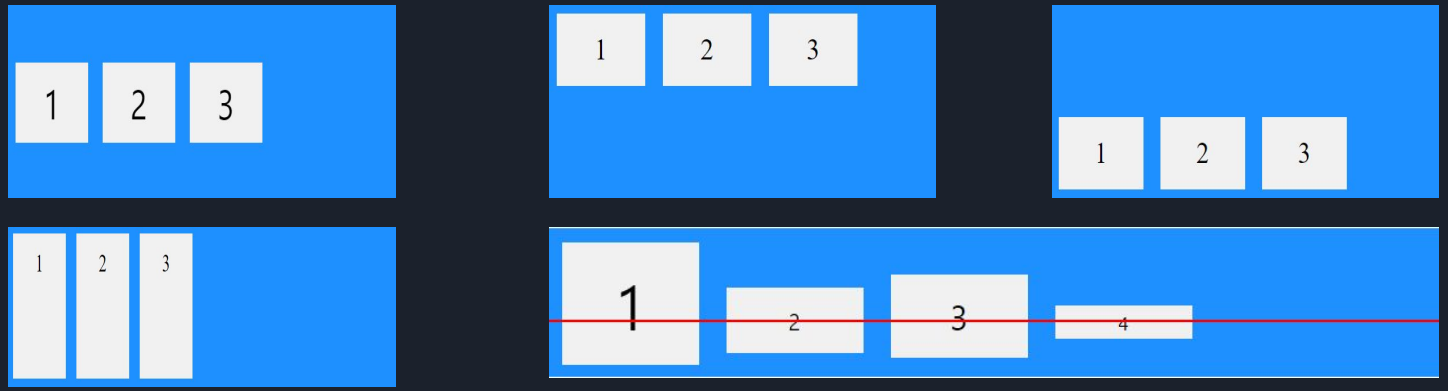
The justify-content property is used to align the flex-items. It's used to describe items or containers along the horizontal axis.



Flex container property

4. align-items property (center | flex-start | flex-end | stretch | baseline)

The align-items property is used to align the flex-items. It's used to describe items or containers along the vertical axis.



CSS Flex Items

Flex Items : These are the direct child elements of a flex container.

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
```

Properties

1. order
2. flex-grow
3. flex-shrink
4. flex-basis
5. flex
6. align-self

Properties of flex-items

1. order Property

   The order property is used to specify the order of the flex items.


```
<div class="flex-container">
  <div style="order: 3">1</div>
  <div style="order: 2">2</div>
  <div style="order: 4">3</div>
  <div style="order: 1">4</div>
</div>
```

2. flex-grow Property

   The flex-grow property is used to specify how much a flex item will grow relative to the rest of the flex-items.

   Default value: 0 (zero)


```
<div class="flex-container">
  <div style="flex-grow: 1">1</div>
  <div style="flex-grow: 1">2</div>
  <div style="flex-grow: 8">3</div>
</div>
```

Properties of flex-items

3. flex-shrink

The flex-shrink property specifies how much a flex item will shrink relative to the rest of the flex items.

Default value : 1


```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex-shrink: 0">3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
  <div>9</div>
  <div>10</div>
</div>
```

4. Flex-basis

The flex-basis property specifies the initial length of the flex item.

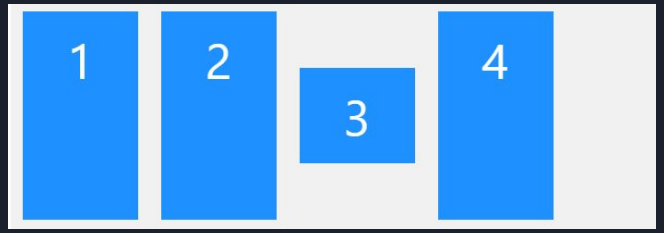
```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex-basis: 200px">3</div>
  <div>4</div>
</div>
```

Properties of flex-items

5. align-self property

   The align-self property specifies the alignment for the selected item inside the flexible container.

   Overrides the default alignment set by the container's align-items property.




```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="align-self: center">3</div>
  <div>4</div>
</div>
```

6. The flex property is a shorthand for the flex-grow,

   flex-shrink and flex-basis

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex: 0 0 200px">3</div>
  <div>4</div>
</div>
```

CSS Units

Length is expressed in different units.
Where we will be using length :

- Width
- Margin
- Padding
- Font Size

There are two types of length units: ‘ **absolute** ‘ and **‘relative’.**

Absolute values can be expressed in :

- cm
- mm
- in
- px
- pt
- pc

Unit	Description	
cm	centimeters	Try it
mm	millimeters	Try it
in	inches (1in = 96px = 2.54cm)	Try it
px *	pixels (1px = 1/96th of 1in)	Try it
pt	points (1pt = 1/72 of 1in)	Try it
pc	picas (1pc = 12 pt)	Try it

CSS Units

Relative Lengths

Relative length units specify a length relative to another length property.

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element