Css intro

10 January 2022 21:34

CSS -casacading style sheet

Use for design purpose

- Css 3 is the current version
- It is a style sheet language which is used to describe the look and formatting of a document written in markup language.
- It is generally used with HTML to change the style of web pages and user interface
- Commenting in css :- /*----------*/
- .css is the extension

Link for icons

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">
```

Syntax

Selector Property Value

P { color : blue ; }

Selector is element/tag.

Ex:-

```
h1{
    color:red;
}
```

History of css

- CSS was first purposed by Hakon Wium Lie on oct 10 1994. At the time, Lie was working with tim.

How to add CSS

- Inline styles
- Head Styles/ Internal Styles

- External Styles

Inline Styles

Inline CSS is used to apply CSS on a single line or element
Here we use style attribute
Particular element

```
  <!-- Inline StyleSheet -->
    <h1 style="color: aqua; font-family:cursive;">Inline StyleSheet</h1>
```

Internal Style

We use style tag
Apply for particular/single document or page.
Use for particular page
Within html we write css which is <style> tag inside <Head> tag.
We will not create separate css file

```
 <!-- Internal StyleSheet -->
    <style>
h1{
    background-color: violet;
    color: black;
    font-size: 40px;
    text-align: center;
}
    </style>
</head>
<body>
    <!-- Internal StyleSheet -->
    <h1>Internal StyleSheet</h1>
```

External Style

We use link tag

```
<link rel="stylesheet" href="StyleSheet.css">
```

it is use to apply CSS on multiple pages or all pages. Here we write css code in a css file
Extension will be .css

```
  <!-- External StyleSheet -->
    <div> I am an External Stylesheet</div>
```

```css
/* CSS Property using External StyleSheet */
div {
  font-size: x-large;
  color: gold;
  background-color: chocolate;
  text-align: center;
}
```

Property Values :

- Url/string : A url or a string represents the location of the resources to be imported. The url may be relative or absolute.
- List-of-mediaqueries: The list of media queries condition the application of the css rules defined in the linked URL

import

10 January 2022 21:36

@import

The @import rule is used to import one style sheet into another style sheet.

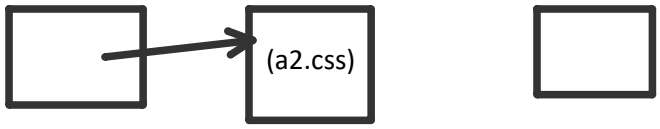
This rule also support media queries so that the user can import the media-departmnet style sheet.

The @import rule must be declared at the top of the document after any @charset declaration

Syntax

@import url|string list-of-mediaqueries;

@import url("./a2.css")



a1.html a1.css a2.css

When we want styling common for multiple at that time we use few common css for both file

1. We can link one html with one css
2. We can link 1 html and 2 css
    a. This can be done in 2 ways
        i. Using link
        ii. @import
            1) It should be written first

```
/* using import */
@import url("./app.css");
```

Link for icons

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
```

```
awesome/6.0.0-beta3/css/all.min.css">

Favicon
Font-awesome
```

Webkit

- Webkit refers to the browser rendering engine used for Apple's Safari and Google's Chrome browsers.
- It is also the CSS syntax used for CSS3 modules.
- It works with the web browser engine.
- It is the component of a web browser that is responsible for rendering the content.
- It is also reffered to as the web rendering engine.
- The CSS -webkit-appearance property enables web authors to change the appearance of HTML elements to resemble native User Interface (UI) controls.
- Webkit extensions contain the -webkit- prefix, which indicates that it belongs to the webkit open source framework.

Background and border

10 January 2022 21:39

Colors

1. Color name
2. Hexadecimal denoted by #
3. Rgb range(0-255)
4. Rgba //alpha
5. HSL- hue(0-360 degree) saturation (%)lightness(%)
6. HSLA - // alpha(opacity)

Background properties

1. Background-image
2. Background-repeat
   a. No-repeat

```
background-repeat: no-repeat;
```

   b. Repeat-x

```
background-repeat: repeat-x;
Repeat particular things in rowwise
```

   c. Repeat-y

```
background-repeat: repeat-y;
Repeat particular things in colwise
```

3. Background- position
   a. Left
   b. right
4. Background-attachment
   a. Scroll - image will scroll with content
   b. Fixed - only contents will scroll not body.
5. Background-size

Border
The css border properties are use to specify style, color,width ,radius ,etc

Border-style:
Border-color:
Border-width:
Border-radius:

Font and text

10 January 2022 21:43

Font

- CSS font property is used to control the look of text. By the use of CSS font property you can change the text size , color, style and more

These are some important font attributes

1. CSS Font color: This property is used to change the color of text(standalone attribute).
2. CSS font family : This property is used to change the face of font.
3. CSS Font size : This property is used to increase or decrease the size of font(px).
4. CSS Font style: This property is used to make the font bold ,italic or oblique.
5. CSS Font variant: This property creates a small-caps effect.
6. CSS Font weight: This property is used to increase or decrease the boldness and lightness of the font.
    a. Range of font weight is 400-900

Eg:

```
<h1>hi this is my new page</h1>
```

HI THIS IS MY NEW PAGE

```
h1 {
  color: aquamarine;
  font-family: "Times New Roman", Times, serif;
  font-size: 100px;
  font-style: italic;
  font-weight: 200;
  font-variant: small-caps;
  font-style: oblique;
}
```

Font Effects(&effects this we have to add in link)

```
@import url("https://fonts.googleapis.com/css?
family=Bakbak+One&family=Bebas+Neue&family=Corinthia:wght@700&family=Montserrat:wght@100
&family=Noto+Serif:ital,wght@1,700&family=Smooch&display=swap&effect=fire|neon|emboss|shadow-
multiple
```

```
<link href="https://fonts.googleapis.com/css?family=Bakbak+One&family=Corinthia:wght@700
&family=Montserrat:wght@100&display=swap&family=Noto+Serif:ital,wght@1,700
&family=Smooch&effect=fire|neon|emboss" rel="stylesheet">
```

- Fire
- Neon
- Emboss

- Shadow-multiple

Text formatting

1. Color
2. Text-alignment
    a. Right
    b. Left
    c. Center
    d. Justify: all the elements will be arrange in equal width both side
3. Text-transform
    a. Uppercase
    b. Lowercase
    c. Captilize: will convert only first letter to uppercase
4. Letter-spacing
5. Word-spacing
6. Text-direction
    a. rtl
    b. Ltr
7. Text-shadow
8. Text-decoration
    a. Underline
    b. Overline
    c. Line-through
    d. None

1. Text-overflow
    a. Ellipsis(… there are dots if we click we can see content, by default value is click)

```
<body>
   <h1>MERN stack stands for MongoDB,Express,React,Nodejs</h1>
 </body>

h1 {
  width: 300px;
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
  border: 2px solid gray;

}
```

    b. Clip

Text-outline

1. Line height

2. Text Indentation
    a. Text indentation property is used to indent the first line of paragraph

List-style:none will remove bullet points from unordered list and display only content.

Text effects

- CSS is the mechanism to adding style in the various web documents
- Text Effects allows us to apply different types of effect on text used in an HTML document.

Below are some of the properties in CSS that can be used to add effects to text:

1. Text-overflow
    a. Ellipsis
    b. clip
2. Word-wrap
    a. Break-word
3. Word-break
    -
    a. Keep-allgr

```
#break {
  word-break: keep-all;
  border: 2px solid black;
  color: red;
  width: 300px;
  resize: horizontal;
}
```

LUNCHBREAK isci dicta temporibus fugit explicabo ipsam nulla eveniet asperiores sequi voluptatum quia libero, dolorem vero ipsa esse labore tenetur suscipit rerum! Lorem ipsum dolor sit, amet consectetur adipisicing elit. Perspiciatis adipisci dicta temporibus fugit explicabo ipsam nulla eveniet asperiores sequi voluptatu

    b. Break-all

```
<p id="break">
      LUNCHBREAK isci dicta temporibus fugit explicabo ipsam nulla eveniet
      asperiores sequi voluptatum quia libero, dolorem vero ipsa esse labore
      tenetur suscipit rerum! Lorem ipsum dolor sit, amet consectetur
      adipisicing elit. Perspiciatis adipisci dicta temporibus fugit explicabo
      ipsam nulla eveniet asperiores sequi voluptatu
    </p>

#break {
  word-break: break-all;
  border: 2px solid black;
  color: red;
  width: 300px;
  resize: horizontal;
}
```

LUNCHBREAK isci dicta temporibus fugit ex plicabo ipsam nulla eveniet asperiores sequi v oluptatum quia libero, dolorem vero ipsa esse labore tenetur suscipit rerum! Lorem ipsum do lor sit, amet consectetur adipisicing elit. Perspi ciatis adipisci dicta temporibus fugit explicabo ipsam nulla eveniet asperiores sequi voluptatu

4. Writing-mode
    a. Horizontal
    b. Vertical
    c. Vertical-rl

d. Vertical-lr

5. Outline
   a. An outline is a line draw outside the elements border
      i. Outline-style
      ii. Outline-color
      iii. Outline-width
      iv. Outline : It is the Shorthand syntax

6. Resize
   - The elements can be resizable by the user.
   - Property name:resize

   Note
   The resize property does not apply to inline elements or to block elements where overflow="vissible". So the overflow is set to "scroll", "auto" or "hidden"

   Css syntax resize:none|both|horizontal|vertical|initial|inherit;

```
ex:
div{
    resize:vertical;
    overflow:auto;
}
```

7. CSS text-stroke

   a. Note: the text-stroke can only be used with the -webkit- prefix

8. Visibility- it just specifies whether the element is visible or not.
   a. Note: if the element is hidden it occupies or take space.

      Css syntax: visibility: visible|hidden|collapse|initial;

9. !important:
   - The !important rule in CSS is used to add more importance to a property/value than normal.
   - The !important rule, it will override ALL previous styling rules for that specific property or the element.

10. Initial : Represents the value specified as the property's initial value.
11. Inherit: Represents the computed value of the property on the element's parent.
12. Unset : This value acts as either inherit or initial, depending on whether the property is inherited or not. In other words, its sets all properties to their parent value if they are inheritable or to their initial value if not inheritable.
13. CSS display property is used to control the layout of the element. It specifies how the element is displayed.

Selectors

10 January 2022 21:45

- CSS Simple Selectors

1. Universal Selector(*)
2. Type/Element Selectors
3. Id Selector(#) ---------- id attribute
4. Class Selectors(.) ----- class attribute

Id and class are core attributes

- Combinator SELECTOR
- Pesudo element selectors
- Pesudo class selectors
- Attribute Selectors

Top Priority

1. Id
2. Class
3. Type/element
4. universal

Css Basic Selectors

A CSS Selectors are used to selecting HTML elements based on their elements name, id, attribute, etc. It can select one or more elements simultaneously.

Group the element and do styling, select some target element.

1. Universal Selectors((*)
    a. The asterisk(*) is known as the CSS universal selectors. It can be used to select any and all types of elements in an HTML page.
    b. This selector is useful when we want to select all the elements on the page.

Syntax:

```
*{
    property : value ;
}
```

Eg :

```
/* universal selector */
* {
  color: goldenrod;
  font-family: "Segoe UI", Tahoma, Geneva,
Verdana, sans-serif;
  font-size: 50px;
}
```

2. Type/Element Selectors
    a. The element selector in css is used to select HTML elements that are required to be styled.
    b. In a selector declaration, there is the name of the HTML element, and the CSS properties which are to be applied to that element are written inside the bracket{}.

Syntax :

```
Element_name{
        // CSS Property
    }
```

Eg :

```
div {
  color: aquamarine;
  background-color: blueviolet;
}
```

3. ID Selector(#)
    a. The id selector uses the id attribute of an HTML element to select a specific element.
    b. The id of an element is unique within a page, so the id selector is used to select one unique element!.
    c. To select an element with a specific id, write a hash(#) character, followed by the id of the element.

Syntax : #id_name{

```
Property : value ;
}
```

4. Class Selector
    a. The class selector selects HTML elements with a specific class attribute.
    b. To select elements with a specific class, write a period(.) character, followed by the

class name.

c. The class name is mostly used to set the CSS property to the given class
d. We can write multiple class file if we want to apply same property,

```
.demo,
.sample {
  background-color: green;
}
```

Syntax

```
.class_name{
    property: value;
}
```

Combinator Selectors

- CSS combinator are explaining the relation between 2 selectors.
- A CSS selector can contain more than one simple selector between the simple selectors, we can include a combinator.

➢ There are 4 different combinators in CSS:

- Descendant selector(space)

```
div   h1{

}
```

- Child selector(>)

```
div  >  h2{

}
```

- Adjacent sibling/immediate selector(+)

```
span  +   p{

}
```

- General sibling selector(~)

```
span  ~   p{

}
```

1. Descendant Selectors -
   Selects all its child element
   a. Typically represented by a single space () character
   b. Combines two selectors such that elements matched by the second selector are selected if they have an ancestor (parent, parent's parent , parent's parent's parent, etc) element matching the first selector.

Synatx:

```
selector1 selector2{
    property declarations
}
```

Eg:

```
div p{
        color: red;
    }
```

```
<!-- Descendant Selectors -->
   <p>Descendant Selectors</p>
   <div>
       <p>Home</p>
       <p>About</p>
       <span>
           <p>Contact</p>
       </span>
   </div>
```

Descendant Selectors

Home

About

Contact

2.Child Selectors (>)
Will select child only not its sub child

- The child selector selects all elements that are the children of a specific element
- The child combinator (>) is placed between two CSS selectors. It matches only those elements matched by the second selector that are the direct children of elements matched by the first.

Syntax :

```
Selector1>selector2{
    style properties
}
```

Eg:

```
div>p{
        color: red;
    }
```

```
<!-- child Selectors -->
 <p>Child Selectors</p>
 <div>
     <p>English</p>
     <p>Science</p>
     <span>
         <p>Maths</p>
```

Child Selectors

English

Science

Maths

Social

```
    </span>
    <p>Social</p>
  </div>
```

3. Adjacent/immediate Sibling Selectors(+)

- The adjacent sibling combinator (+) separates two selectors and matches the second element only if it immediately follows the first element , and both are children of the same parent element.

Syntax :

```
former_element + target_element {style properties }
```

Eg:

```
span+h1{
        color: red;
    }
```

```
<!-- Adjacent sibling combinator(+) -->
  <span>
      <p>Hello</p>
      <p>Hi</p>
  </span>
  <h1>Hello world</h1>
  <h2>MERN Stack</h2>
  <span>
      <p>React</p>
      <p>Vue</p>
  </span>
  <h1>Adjacent selector</h1>
  <h1>CSS</h1>
```

Hello

Hi

Hello world

MERN Stack

React

Vue

Adjacent selector

CSS

4. General Sibling Selectors(~)

- The general sibling selector selectors selects all elements that are next sibling of a specified element.
- The general sibling combinator (~) separates two selectors and matches all iterations of the second element, that are following the first element (though not necessarily immediately), and are children of the same parent element.

Syntax :

```
former_element ~ target_element { style properties }
```

Eg:

```
div~p{
        color: red;
    }
```

```
<!-- General sibling combinator -->
 <p>
     <div>
         <p>English</p>
         <p>Science</p>
     </div>
     <span>Kannada</span>
     <p>Physics</p>
     <h2>Chemistry</h2>
     <p>History</p>
 </p>
```

English

Science

Kannada

Physics

Chemistry

History

Attribute Selectors

- The CSS attribute selector matches elements based on the presence or value of a given attribute
- The CSS Attribute Selector is used to select an element with some specific attribute or attribute value.
- It is an excellent way to style the Html element by grouping them based on specific attribute and the attribute selector

1. [attr]: Represents elements with an attribute name of attr.

html

```
  <!-- attr -->
    <a href="http://www.facebook.com">
facebook
    </a>
    <a href="http://www.google.com" target="_blank">
Google
    </a>
    <a href="http://www.Adobe.com" target="_blank">
        Adobe
    </a>
```

css

```
/* [attr]: it checks only attribute name */
a[target] {
  color: green;
  font-family: Arial, Helvetica, sans-serif;
}
```

facebook Google Adobe

2. [attr=value]: Represents elements with an attribute name of attr whose value is exactly value.

```
 <!-- [attr=value] -->
    <a href="http://www.spotify.com" target="_blank">
        Spotify
            </a>
```

Spotify Gaana Wynk

2. [attr=value]: Represents elements with an attribute name of attr whose value is exactly value.

```
<!-- [attr=value] -->
   <a href="http://www.spotify.com" target="_blank">
        Spotify
            </a>
            <a href="http://www.gaana.com" target="_blank">
        Gaana
            </a>
            <a href="http://www.Wynk.com" target="_self">
                Wynk
            </a>
```

Spotify Gaana Wynk

Css

```
a[target="_self"] {
  background-color: yellow;
}
```

3. [attr~=value]: Represents elements with an attribute name of attr whose value is a whitespace separated list of words, one of which is exactly value.

html

```
<!-- [attr ~=value]  --> value with whitespace
            <div title="user home">Home</div>
            <div title="About">About</div>
            <div title="user Contact">Contact</div>
            <div title="user login">Login</div>
            <div title="userRegister">Register</div>
```

Home
About
Contact
Login
Register

Css

```
div[title~="user"] {
  font-family: cursive;
  color: blue;
}
```

4. [attr|=value] : Represents elements with an attribute name of attr whose value can be exactly value or can begin with value immediately followed by hyphen.

```
<!-- [attr|=value] --> exact value or value followed by hypen (-)
            <h1 class="top-text">Hello World</h1>
            <p class="topValues">Some Content</p>
            <p class="top-Content"> Some text is written</p>
              <h1 class="top data">Google</h1>
               <h1 class="top ">Facebook</h1>
```

Hello World

Some Content

Some text is written

google

Facebook

CSS

```
[class|="top"] {
  color: darkgoldenrod;
}
```

5. [attr^=value] : Represents elements with an attribute name of attr whose value is prefixed (preceded) by value.

```
<!-- [attr^=vale] --> consider only prefix value


<h1 class="right-text">React</h1>
            <p class="rightValues">Vue</p>
            <p class="right-Content"> Angular</p>
            <h1 class=" right data">Flutter</h1>
            <div class="right data"> Angular</div>
```

CSS

React

Vue

Angular

Flutter

Angular

```
[class^="right"] {
  text-shadow: 2px 4px blue;
  color: blueviolet;
}
```

6. [attr$=value] : Represents elements with an attribute name of attr whose value is suffixed (followed) by value.

```
<!-- [attr$=value] --> consider only the suffix value


<div class="first_test">first div element</div>
            <div class="second">second div element</div>
            <div class="my-test">third div element.</div>
            <p class="mytest">paragraph</p>
            <span class="all test">Im suffix data</span>
            <p class="mytest ">not take</p>
```

CSS

first div element
second div element
third div element.

paragraph

Im suffix data

not take

```
[class$="test"] {
  color: crimson;
}
```

7. [attr*=value] : Represents elements with an attribute name of attr whose value contains at least one occurrence of value within the string.

```
<!-- [attr*=value] --> atleast for the one occurrence of the value
            <div class="bottom_test">JSPIDER</div>
            <div class="secondTest">QSPIDER</div>
            <div class="Up-test">PYSPIIDER</div>
            <p class="Uptest">TESTYANTRA</p>
```

JSPIDER
QSPIDER
PYSPIIDER

TESTYANTRA

CSS

```
[class*="te"] {
  font-family: cursive;
  color: brown;
}
```

Pseudo-Classess Selector

A CSS pseudo-class is a keyword added to a selector that specify a special state of the selected element.
Syntax

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

1. Anchor pseudo class -

- The anchor pseudo-classes represent the state of links as visited, unvisited ,active/currently selected
    a. Visited - purple
    b. Active / currently selected - red
    c. Unvisited -blue
    d. Focus
- Anchor pseudo-classes also enables you to activate the HTML elements or apply a specified style to an element when the mouse pointer is kept over it

The anchor pseudo-classes include the following :

- :link (Applies styles to non visited links)
- :visited (Applies styles to non visited links)
- :hover (Applies styles to an element over which the mouse-pointer moves)
- :active (Applies styles to an active element)

Syntax

I) Link : once we visit the color will change to mentioned one.

```
a:link {
  color: darkgoldenrod;
}
```

Ii) Visited

```
<body>
    <a href="https://www.google.co.in/">click here</a>
</body>
```

click here

```
/* visited */
a:visited {
  color: chartreuse;
}
```

Iii) Hover : when u place cursor color will change

```
a:hover{
    color: brown;
}
```

Iv) Active :

```
a:active{
    color: black;
}
```

V) Focus : when its in active state we can add

```
input:focus {
  color: gray;
}
```

Vi) Checked

```
input:checked {
  height: 80px;
  width: 50px;
}
```

2. Pseudo child
    a. First-child

```
 <div>
    <p>First child1</p>
</div>
<div>
    <p>First child2</p>
    <p>First child3</p>
</div>
    <p>First child4</p>
```

First child1
First child2
First child3
First child4

```
p:first-child {
  color: rgb(11, 245, 11);
}
```

b. Last-child

```
<p>First child1</p>
</div>
<div>
```

First child1
First child2

b. Last-child

```
<p>First child1</p>
</div>
<div>
    <p>First child2</p>
    <p>First child3</p>
</div>
    <p>First child4</p>
```

First child1

First child2

First child3

First child4

```
p:last-child {
  color: rgb(11, 245, 11);
}
```

c. Nth-child

i. Odd

```
<p>First child1</p>
  <p>First child2</p>
  <p>First child3</p>
  <p>First child4</p>
```

First child1

First child2

First child3

First child4

```
p:nth-child(odd) {
  color: rgb(11, 245, 11);
}
```

ii. Even

```
<p>First child1</p>
  <p>First child2</p>
  <p>First child3</p>
  <p>First child4</p>
```

First child1

First child2

First child3

First child4

```
p:nth-child(even) {
  color: rgb(11, 245, 11);
}
```

iii. Any number

```
 <p>First child1</p>
  <p>First child2</p>
  <p>First child3</p>
  <p>First child4</p>
```

First child1

First child2

First child3

First child4

```
p:nth-child(3) {
  color: rgb(11, 245, 11);
}
```

iv. N+ some(value) - it will selected all elements from that value

```
<p>First child1</p>
    <p>First child2</p>
    <p>First child3</p>
    <p>First child4</p>
```

First child1

First child2

First child3

First child4

```
p:nth-child(N + 3) {
  color: rgb(11, 245, 11);
}
```

```
<p>First child1</p>
    <p>First child2</p>
    <p>First child3</p>
    <p>First child4</p>
```

First child1

First child2

First child3

First child4

```
p:nth-child(N + 1) {
  color: rgb(11, 245, 11);
}
```

Pseudo element selector

A CSS pseudo-element is a keyword added to a selector that lets you style a specific part of the selected elements.

It can be used to :

- Style the first letter or line of an element
- Insert content before or after the content of element

Syntax

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

1) ::first-line pseudo element

```
 <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit
recusandae
      libero maxime voluptatibus saepe tempore, fugiat soluta ad sit
fugit,
      veniam culpa voluptatem nobis amet enim labore. Recusandae,
labore rem.
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit
recusandae
      libero maxime voluptatibus saepe tempore, fugiat soluta ad sit
fugit,
      veniam culpa voluptatem nobis amet enim labore. Recusandae,
labore rem.
```

All CSS Pseudo Elements

Selector	Example	Example description
::after	p::after	Insert something after the content
::before	p::before	Insert something before the content
::first-letter	p::first-letter	Selects the first letter of each
::first-line	p::first-line	Selects the first line of each
::marker	::marker	Selects the markers of list items

```
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit
recusandae
      libero maxime voluptatibus saepe tempore, fugiat soluta ad sit
fugit,
      veniam culpa voluptatem nobis amet enim labore. Recusandae,
labore rem.
    </p>


p::first-line {
  color: red;
  background-color: aquamarine;
}
```

Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit recusandae libero maxime voluptatibus saepe tempore, fugiat soluta sit amet consectetur adipisicing elit. Velit recusandae libero maxime voluptatibus saepe tempore, fugiat soluta ad sit fugit, veniam

2) ::first-letter pseudo element

```
 <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit
recusandae
      libero maxime voluptatibus saepe tempore, fugiat soluta ad sit
fugit,
      veniam culpa voluptatem nobis amet enim labore. Recusandae,
labore rem.
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit
recusandae
      libero maxime voluptatibus saepe tempore, fugiat soluta ad sit
fugit,
      veniam culpa voluptatem nobis amet enim labore. Recusandae,
labore rem.
    </p>


p::first-letter {
  color: red;
  background-color: aquamarine;
}
```

Lorem ipsum dolor sit amet consect
sit amet consectetur adipisicing elit.

3) ::before pseudo element

- The ::before pseudo-element can be used to insert some content before the content of an element.

```
Syntax :
            selector::before {
                property:value;
            }
 <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit.
Velit recusandae
      libero maxime voluptatibus saepe tempore, fugiat soluta ad
sit fugit,
      veniam culpa voluptatem nobis amet enim labore. Recusandae,
labore rem.
      Lorem ipsum dolor sit amet consectetur adipisicing elit.
Velit recusandae
      libero maxime voluptatibus saepe tempore, fugiat soluta ad
sit fugit,
      veniam culpa voluptatem nobis amet enim labore. Recusandae,
labore rem.
    </p>

p::before {
  content: " ";
}
```

- Lorem ipsum dolor sit amet consectetu
  dolor sit amet consectetur adipisicing elit.

```
p::before {
  content: " ";
  content: url(https://encrypted-tbn0.gstatic.com/images?
q=tbn:ANd9GcTtZFJN1rG7233vqSCo-DF0xwsCo6MZaHohkg&usqp=CAU);
}
```


Lorem ipsum dolor sit amet consectetur adipisicing el
labore. Recusandae, labore rem. Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit recusa
Recusandae, labore rem.

4) ::after pseudo element :

- The ::after pseudo-element can be used to insert some content after the content of an element.

```
Syntax :
            selector::before {
                property:value;
            }
```

5) Marker :-

```
 <ul>
```

::marker	::marker	Selects the markers of list items
::selection	p::selection	Selects the portion of an element that is selected by a user

The following properties apply to the ::first-line pseudo-element:

-
- font properties
- color properties
- background properties
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

The following properties apply to the ::first-letter pseudo- element:

-
- font properties
- color properties
- background properties
- margin properties
- padding properties
- border properties
- text-decoration
- vertical-align (only if "float" is "none")
- text-transform
- line-height
- float
- clear

```
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>

li::marker {
  color: blueviolet;
}
```

- HTML
- CSS
- JS

```
<ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>

li::marker {
  content: "🧿";
}
```

🧿HTML
🧿CSS
🧿JS

6) ::Selection pseudo-element
   - The ::selection pseudo-element matches the portion of an element that is selected by a user.
   - The part will selecting will reflect changes

```
Syntax:
        selector::selection{
            property:value;
        }

   <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit.
Velit recusandae
      libero maxime voluptatibus saepe tempore, fugiat soluta ad
sit fugit,
      veniam culpa voluptatem nobis amet enim labore. Recusandae,
labore rem.
      Lorem ipsum dolor sit amet consectetur adipisicing elit.
Velit recusandae
      libero maxime voluptatibus saepe tempore, fugiat soluta ad
sit fugit,
      veniam culpa voluptatem nobis amet enim labore. Recusandae,
labore rem.
    </p>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>

::selection {
  color: blueviolet;
  background-color: aquamarine;
}
```

🎄 Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit recusan
dolor sit amet consectetur adipisicing elit. Velit recusandae libero maxim

🧿HTML
🧿CSS
🧿JS

Difference between pseudo class and pseudo element

Pseudo class is a selector that assist in the selection of something that cannot be expresses by a simple selector, for example :hover.

A pseudo-element however allows us to create items that do not normally exist in the document tree, for example ::after

Pseudo-classes

Pseudo-classes select regular elements but under certain conditions, like when their position relative to siblings or when their position relative to siblings or when they're under a particular state.
Here is a list of pseudo-classes in css3:

A) Dynamic pseudo-classes
:link
:visited
:hover
:active
:focus

B) UI element states pseudo-classes
:enabled
:disabled
:checked

C) Structural pseudo-classes
:first-child

:nth-child(n)
:nth-last-child(n)
:nth-of-type(n)
:nth-last-of-type(n)
:last-child
:first-of-type
:last-of-type

D) Other pseudo-classes
:not(x) :target :lang(language)

Pseudo-elements:
Pseudo-elements effectively create new elements that are not specified in the markup of the document and can be manipulated much like a regular element.

::before
::after
::first-letter
:first-line
::selection

Z-index & position

10 January 2022 21:50

Z-Index

- It specifies the stack order of an element.
- The elements can be placed in front of or behind or other side
- An element can have positive or negative stack order
- Z index work with position elements and with flex items.

```
<img
    src="https://cdn.pixabay.com/photo/2018/08/21/22/54/tallinn-3622413__340.jpg"
    alt=""
    height="300px"
    width="600"
  />
```


```
img {
  position: absolute;
  z-index: -1;
  top: 0;
}
```

Position

It sets how an element is positioned in a document.
The top, right,bottom, and left properties determine the final location of positioned elements.

There are five values the position property can take. They are:

- Static(by default)
- Relative - fixed- we can fix top right or bottom
- Absolute- for element , we can change
- Fixed
- Sticky

Note:

- If the position property is set to relative, absolute, sticky or fixed you can set the top, right,bottom and left properties
- If the position property is set to static top, right bottom and left properties.

Value	Description
static	Normal position for the element(where top ,right,bottom, and left have no effect) Div { position: static;}
relative	Positiion the element absolutely relative to its conatiner Div { position: absolute; top:10px;left:15px; }
absolute	Position the element absolutely relative to its container.
fixed	Positiion the element relative to the screen's viewport and stay fixed on screen when scrolling Div { position : fixed; top:10px;left:15px

inherit	Indicates that the element will inherit the position from its parent element div{ position: iinherit;}

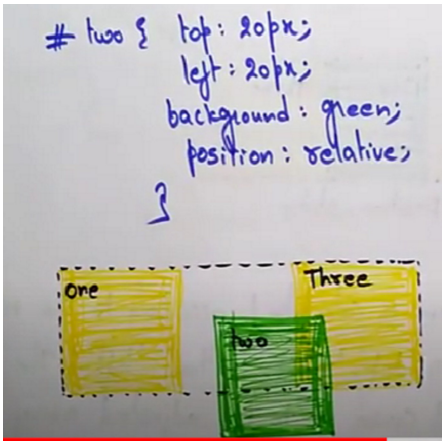
Static

By default HTML elements are static . They are not affected by top , bottom, left ,right properties.

Relative

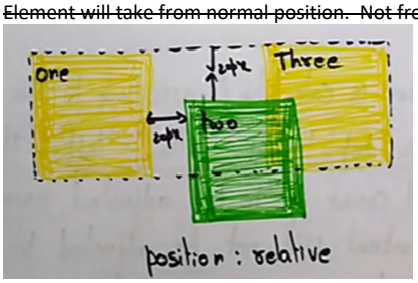
Relative is like static but we can place element by using property like top left, right, bottom
or
Places an element relative to current position.

Other elements will not be affected



It overflow element like we see in above example.
Consume whitespace

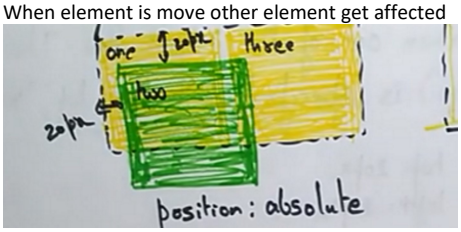
~~Element will take from normal position. Not from starting container.~~



Points : generally we dont use position property in relative

absolute

Absolute places an element relative to its parent position
When element is move other element get affected



Removes element from document flow

Absolute will not occupy white space.
Disadvantage - > another element will takeplace

Mostly parent is relative and we give child absolute as its relative to parent.
If we directly give absolute it will start from window

If parent is not relative or static it will start from window.

Fixed

Its nothing to do with parent.
Remains in the same place no matter how much we scroll.
Relative to browser window.
If there is something in that place the item fixed will overlapp.
Mostly used to build chat application.

Sticky

Same as fixed but its relative to container not relative to browser window.
Used mostly in menu
Even if we scroll it will be in same position in that container..
Element will scroll within the container

Float

It is used to position HTML elements horizontally.
Position left or right.

In how many ways can we position an HTML element? Or what are the permissible values of the position attribute?

There are mainly 7 values of position attribute that can be used to position an HTML element:

1. **static**: Default value. Here the element is positioned according to the normal flow of the document.
2. **absolute**: Here the element is positioned relative to its parent element. The final position is determined by the values of left, right, top, bottom.
3. **fixed**: This is similar to absolute except here the elements are positioned relative to the <html> element.
4. **relative**: Here the element is positioned according to the normal flow of the document and positioned relative to its original/ normal position.
5. **initial**: This resets the property to its default value.
6. **inherit**: Here the element inherits or takes the property of its parent.
7. Sticky

Units and gradients

10 January 2022 21:53

Different unit

1. Absolute unit
    a. (pixels)Px - 1px=1/96 inch
    b. Cm
    c. Mm
    d. (Inches)In
    e. (Points)Pt
    f. (Picas)pc
2. Relative unit
    a. %
    b. Elements and root elements(em &rem)
        i. Em - realative to the font size of current element
            1) Heading and paragraph have different values for 1em
        ii. Rem- the root element font size

```
html {
  font-size: 14px;
}
```

This means 1rem =16px
Everywhere in the document

    c. Character-size(ex-ch)
    d. Viewport dimension (vw & vh)
    e. Viewport Max (vmax)
    f. Viewport Min(vmin)

- 0(zero) is same in all the unit.
- Rem and em help with accessibility

```
.modal{
    width: 20rem;;
}
```

Gradients

There are 2 types of gradient

1. Linear-gradient(goes down/up/left/right/diagonally

a. Background-image: linear-gradient(direction, color-stop1, color-stop2,…);
    i. Background-image:linear-gradient(yellow,green)
2. Radial Gradient (defined by their center):
    i. Background-image:radial-gradient(shape size at position,start-color,….,last-color);

By default, shape is ellipse, size is farthest-corner, and position is center.
Background-image: radial-gradient(circle, orange, yellow, green)

Uses of different size keywords in gradient

The size parameter defines the size of the gradient. It can take four values:

- Closest-side
- Farthest-side
- Closest-corner
- Farthest-corner

Transform

10 January 2022 21:54

Transform

2D and 3D Transforms

- CSS transforms allow you to move, rotate, scale and skew elements

2D transformation methods:

1) Tanslate(): allows you to move elements

```
<h1>2D transform</h1>
```



```
h1 {
  border: 2px solid black;
  width: 300px;
  margin-left: auto;
  margin-right: auto;
  margin-top: 40px;
  background-color: aqua;
  transform: translate(100px, 150px);
}
```

- Css translate moves an element up and down or side-to-side:
- By indicating one value, you move the element to the right side.Negative values move elements to the left.
- The second value moves the element down. Negative values moves element up.

2) Rotate(deg): Rotates the element clockwise from its current position.

We have to use one value only

- Rotate()

3) Scale(): Affects the size of the element. This also applies to the font-size,padding,,height, and width of an element, we cannot use px ,we can use decimal value,normally we give just number.

- scaleX()


```
border: 2px solid black;
 width: 300px;
 margin-left: auto;
 margin-right: auto;
 margin-top: 40px;
 background-color: aqua;

 transform: scaleX(5);
```

- scaleY()

```
border: 2px solid black;
 width: 300px;
 margin-left: auto;
 margin-right: auto;
 margin-top: 40px;
```

```
  margin-left: auto;
  margin-right: auto;
  margin-top: 40px;
  background-color: aqua;

  transform: scaleY(5);
```

- Scale(x,y)

4) Skew(): a transformation that skews an element on the 2D plane
   - skewX()

```
h1 {
  border: 2px solid black;
  width: 300px;
  margin-left: auto;
  margin-right: auto;
  margin-top: 40px;
  background-color: aqua;
  transform: skewX(110deg);
}
```

   - skewY()

```
h1 {
  border: 2px solid black;
  width: 300px;
  margin-left: auto;
  margin-right: auto;
  margin-top: 40px;
  background-color: aqua;
  transform: skewY(120deg);
}
```

   - Skew(x,y)

```
h1 {
  border: 2px solid black;
  width: 300px;
  margin-left: auto;
  margin-right: auto;
  margin-top: 40px;
  background-color: aqua;
  transform: skew(110deg, 210deg);
}
```

Transition

10 January 2022 21:55

Transitions

CSS transitions allows you to change property values smoothly over a given duration

List of transition properties

1. Transition
   a. Syntax :

```
transition:  width  2s  ease;

  transition: all 0.3s ease-in;
```

2. Transition-delay : refers to how long you want to wait before starting the duration. When the transition effect will start. This value is written in seconds/milliseconds. 3s

```
transition-delay: 5s;
transition-delay: 15ms;
```

3. Transition-duration : refers to the duration of the transition

```
transition-duration: 3s;
```

4. Transition-property: refers to the CSS property you wish to transition.

```
transition-property: rotate;
                    :backgroundcolor,height,width
```

5. Transition-timing-function: refers to how to transitions occurs.
   - All transitions have a value of linear by default, which means the property changes evenly until the end of the transition.
   - And it describes about the speed curve of the transition effect.
   - It helps to change the speed of an transition over the duration.

```
Syntax  transition-timing-function: linear|ease,ease-in,ease-
out,ease-in-out,step-end,step-start;
```

Transition timing function	Description
Linear	Specifies a transition effect with the same speed from start To end
ease	Default value. Specifies a transition effect with a slow start, then Fast, then end slowly.
Ease-in	Specifies a transition effect with a slow start.
Ease-out	Specifies a transition effect with a slow end.

Ease-in-out	Specifies a transition effect with a slow start and end.

Ease-in-out (slow-start-slow end)

```
.card {
  width: 100px;
  border: 10px solid teal;
  padding: 10px;
  margin-top: 50px;
  background: corol;
  margin-left: auto;
  margin-right: auto;
  height: 50px;
  transition: background-color 2s ease-in-out;
  /* ease-in-out slow start slow end */
}
.card:hover {
  cursor: pointer;
  background-color: aquamarine;
  border: 10px solid crimson;
}
```

Animation

10 January 2022 21:55

Animation

CSS Animation property is used to create animation on the webpage.

Each animation must be defined with @Keyframes

Each keyframes @rules is define what should happen at specific moment during the animation

@keyframes can be controlled by either by short hand animation property or its 8 sub properties.

List of properties:

1. @Key frames
    a. It also helps in changing the animation from animation 1 to animation 2 or animation n.
    b. There are two ways to achieve the animations using keyframes
        i. Using from to keywords.(only 2 values)
            1) Syntax

```
@keyframes animation-name{
      from {
          properties: value;
      }
      to{
        properties: value;
      }
    }
```

        ii. Using the % assignment.(multiple values)

Syntax

```
@keyframes animation-name{
  0% {
      properties: value;
  }
  50% {
    properties: value;
}
  100%{
    properties: value;
  }
}
```

2. Animation-name
    a. Declares the name of @keyframes at-rule.

3. Animation duration
    a. The length of the time it takes for an animation to complete one cycle.

4. Animation delay
    a. Like transitions we can add the delay for the animations.
    b. The time between the element being loaded and the start of the animation sequence.

5. Animation-iteration-count
    a. It define the number of occurrences of animations. It can take count like 4 or infinity.

6. Animation direction
    a. It define the direction of the animation.
        i. Reverse
        ii. Alternate
        iii. Alternate-reverse
    b. Sets the direction of the animation after the cycle. Its default resets on each cycle.

7. Animation-timing-function
8. Animation-play-state : It is used to control the animation. It has paused, running. Pause/play the animation.
    a. Note: Even the JS can be used to control the animation
9. Animation-fill-mode
    a. Defines what values are applied by the animation outside the time it is executing.
    b. Sets which values are applied before/after the animation.
        i. Forwards|backwards|both|none
        ii. Short-hand syntax :
            1) Animation : name duration timing-function delay
            2) Iteration-count direction;
10. Animation

Each animation must be defined with @Keyframes

@Keyframes Rule

The animation is created in the @keyframe rule. It is used to control the intermediate steps in a css animation sequence.

Flex and Grid

10 January 2022 21:56

Flex

```
 <section id="flexContainer">
        <div class="block1">1</div>
        <div class="block2">2</div>
        <div class="block3">3</div>
    </section>
```

```
#flexContainer{
    background-color: crimson;
    width: 90%;
    margin: 50px auto;
    height: 100px;
    display: flex;

}
#flexContainer div{
    flex: 1;
}
.block1{
background-color: yellow;
}
.block2{
  background-color: skyblue;
}
.block3{
   background-color: darkmagenta;

}
```

Grid

container acts like a wrapper to store elements in 2d form

- Whenever we have grid or container we must have container
- We have to use property as display grid
    - Display: grid
        - If html element want to be grid container then it must be specified display as grid.

    - Display : grid-inline
- We have grid items
    - Elements store in grid container
- Grid line
    - The line between col is called column line
    - The line between row is called row line

- Grid gutter : space between rows and col , and the row gutter is different from grid col.
- Grid area : The grid area consist of multiple grid cells .It can always be square or a rectangle. It specifies where to place items
    - Grid-col-start :
    - Grid-col-end
    - Grid-row-start
    - Grid-row-end

- Grid-gap : shorthand syntax : row value col value;
    - Grid row gap
    - Grid col gap

- Grid auto rows/grid auto cols: display height

- Grid-template-area :
    - It is the another way to layout grid using grid area template property. Here no need of using line numbers.

    - We write the content within the grid template area using quotes.

- Grid-template :
    - Explicit values
    - Implicit values
    - Fractional units : define using fr
        - The fractional unit take fraction of space and divide it equally.
    - Repeat()

- Justify-item: it is use to position the items within the grid container horizontally
        - Stretch (default)
        - Center
        - Start
        - End

- Align-items :
    - Center
    - Start
    - End

Justify-self

Align-Self

Place-self : it is a combination/shorthand of align property(vertical) and justify property(horizontal).

What is for what?

1. Justify-items and align items are for container
2. Justify-self and align-self and place-self are for items, so we should not write in container

- Justify-content :
  - Space-around
  - Space-between
  - Space-evenly

Responsive web design(grid) -> css media queries

Library : collection of codes, less rule (react and jquerry)

Framework : collection of libraries, more rules (angular ,vue , js)

Bootstrap is css framework.

Flexbox

Before the flexbox these were the modules :

Flexlayout is based on flex flow directions

- Block , for sections in a webpage
- Inline for text
- Table for two dimensional table data
- Positioned for explicit positiion of an element.
- The main idea behind the flex layout is to give the container the ability to alter its items width/height (and order) to best fill the available space (mostly to accommodate kind of display devices and screen size).

Properties

Display : flex | inline-flex ;
Flex-direction : row |column | row-reverse | col-reverse
Flex-wrap : wrap | nowrap |wrapreverse
Flex-basis : <length>
Justify-content : flex-start | flex-end |center| space-around | space-between | space-evenly
Align-self : flex-start |flex-end | center |stretch
Align-items : flex-start | flex-end | center
Align content : flex-start | flex-end | center
Flex-grow : <number>
Flex-shrink : <number>
Flex : <integer>
Flex-flow :
Order: <integer>

Flexbox-container
Flex-direction
flex-wrap,
flex-flow,
justify-content,
align-items
,align contents

Flexbox items

Flex-grow - default value is 0
Flex-shrink : default value is 1
Order : default value is 0
Flex-basis : it defines the initial length of the flex items
Flex : first value will be for flex grow, flex shrink, flex basis
Syntax flex : flex grow flex shrink flex basis ;
Flex ;align-self :

CSS Flex Items

- The child elements in the flex container automatically becomes flexible.

The flex items properties are :

- order
- flex-grow
- flex-shrink
- flex-basis
- flex
- align-self

How flex properties works ?

Align items horizontally Justify content vertically.	When flex direction is row

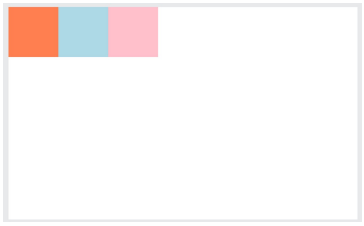
Flex properties

28 April 2022 14:28

1 justify-content(horizontly)

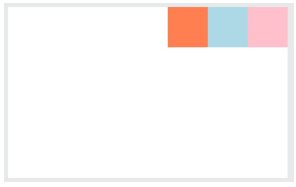
- **flex-start**: Items align to the left side of the container.
- **flex-end**: Items align to the right side of the container.
- **center**: Items align at the center of the container.
- **space-between**: Items display with equal spacing between them.
- **space-around**: Items display with equal spacing around them.

A. Flex start



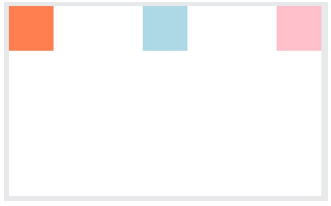
B. Flex end

a.

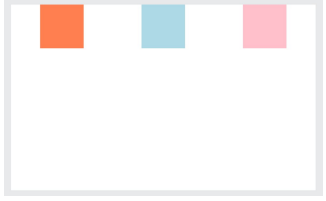


C. Center

D. Space between



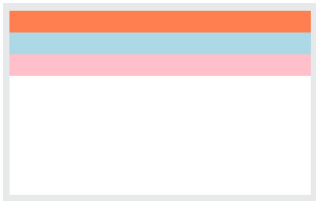
E. Space around



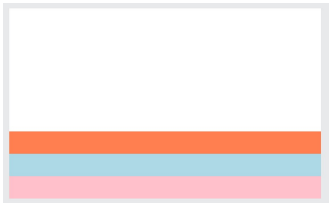
F. Space evenly

2) Align-content (vertically)

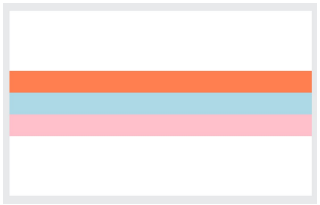
A. Flex start



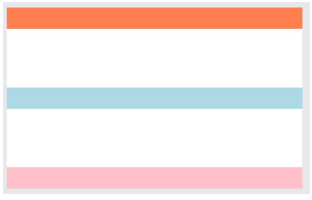
B. Flex end



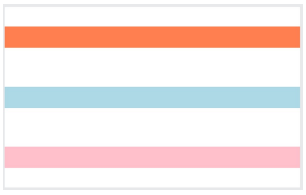
C. Center



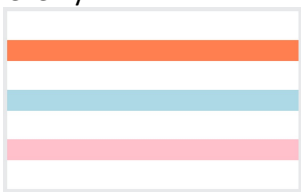
D. Space between



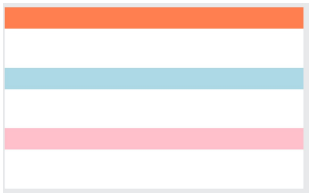
E. Space around



F. Space evenly



G. Stretch



3) align-items

- `flex-start`: Items align to the top of the container.
- `flex-end`: Items align to the bottom of the container.
- `center`: Items align at the vertical center of the container.
- `baseline`: Items display at the baseline of the container.
- `stretch`: Items are stretched to fit the container.

4) Flex-direction

Media queries

08 February 2022 10:01

Media queries is a feature of CSS 3 allowing content rendering to adapt to different conditions such as screen resolution. It became a W3C recommended standard in June 2012, and is a cornerstone technology of responsive web design.

Media queries are useful when you want to modify your site or app depending on a device's general type (such as print vs. screen) or specific characteristics and parameters (such as screen resolution or browser viewport width).

Media queries are used for the following:

- To conditionally apply styles with the CSS `@media` and `@import` at-rules.
- To target specific media for the `<style>`, `<link>`, `<source>`, and other HTML elements with the `media=` attribute.
- To test and monitor media states using the `Window.matchMedia()` and `MediaQueryList.addListener()` JavaScript methods.