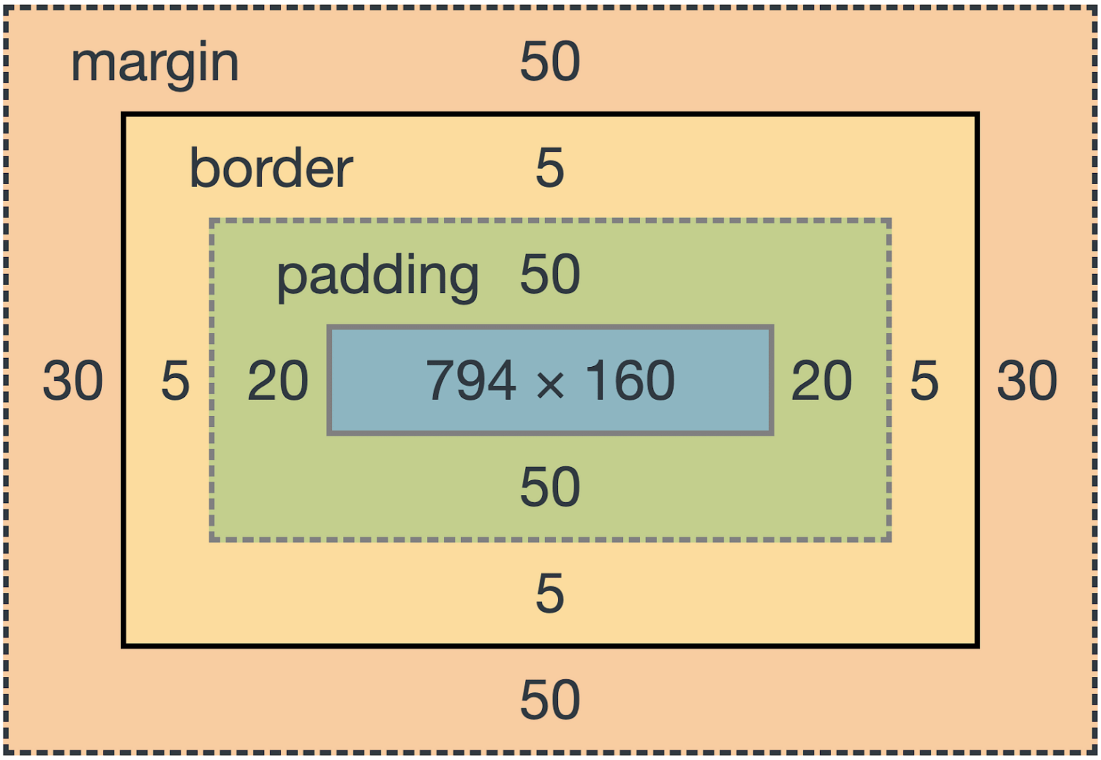
Date: 14-06-2024

Cascading Stylesheets(CSS):

1. CSS Box Model



It is the basic framework of web development whether you are making a website using any other language such as Angular or React. The box model helps us to define the padding**,** border, and **margin** around an element. So from the above diagram we can see where all these things lie around the element. The element is in the center surrounded by padding, border and margin. These parts can be explained as-

- **Content-** The content of the box, where text and images appear.
- **Padding-** It clears an area around the content. The padding is transparent.
- **Border-** A border is one that covers the padding and content.
- **Margin-** It clears an area outside the border. The margin is also transparent.

2. Float & Clear Property

The CSS float property specifies how an element should float. The CSS clear property specifies what elements can float beside the cleared element and on which side. The float property is used for positioning and formatting content, for example, let an image float left to the text in a container.

The float property can have one of one of the following values−

- **Left-** The elements floats to the left of its container.
- **Right-** The elements floats to the right of its container.

- **None-** The element does not float (it will be displayed just where it occurs in the text). This is default.
- **Inherit-** The element inherits the float value of its parent.

3. Pseudo Selectors

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

- **Hover** is used to change the color of text or background of a button as soon as you hover that part. The code for this is as below.

  - **Syntax:-** a.hover{ color: rgb(5,0,0);

```
                    Background-color: rgb(221, 250, 263);
                  }
```

- The next Pseudo selector is Visited. As soon as you visit the anchor tag button and click the link mentioned, it changes its color.
  - **Syntax-:**

```
          a:visited{
        background-color: yellow;
    }
```

- The next selector is Active. If you visit any button, and click it, it becomes active and showcases with different properties.
  - **Syntax-:**

```
          a:active{
        background-color:darkblue;
    }
```

4. Navigation Bar

A navigation bar is usually a list of links, so using the <ul> and <li> elements can help in obtaining it.

- **Syntax-:**

```
               <header>
            <nav class="navbar">
               <ul>
                  <li><a href="#">Home</a></li>
                  <li><a href="#">About</a></li>
                  <li><a href="#">Services</a></li>
                  <li><a href="#">Contact us</a></li>
                  <div class="search">
                     <input type="text" name="search" id="search" placeholder="Search this
website">
                  </div>
               </ul>
            </nav>
         </header>
```

- The Life-style property is used to remove all the bulleted points in the navigation items.

  Synatx-:

```
          .navbar li{
    float:left;
```

```
            list-style: none;
            margin: 13px 20px;
        }
```

- We can also add the Search bar in the navigation menu.

Syntax:-

```
            <div class="search">
            <input type="text" name="search" id="search" placeholder="Search this
website">
        </div>
```

- Within the 'navbar', for styling the input tag we can include the border, border-radius, padding, and width.

5. **CSS Display Property**

The display CSS property sets whether an element is treated as a block or inline element and the layout used for its children, such as flow-layout, grid or flex.

Syntax-:

```
            img {
        margin: auto;
        display: block;
        width: 34px;
}
```

- Display Inline means it will take the space according to the size of the element.
- Display Block means we can set its width and by margin manually.
- The Inline-block property here allows us to change the width of inline elements also.
- To ensure that all the three blocks come in a single line, we can use the property box-sizing.

6. **Position Property**

With the help of CSS Position Property, you can put your HTML elements at the position of your choice.

Types of Position properties:-

- Static:- it is the default position of the HTML elements.

- Relative:- it is used when we need to position an HTML element relative to its normal position. We can set the top, right, bottom, and left properties that will cause the element to adjust away from the normal position.

- Absolute:- An element with absolute position will move according to the position of its parent element. In the absence of any parent element, the HTML element will be placed relative to the page.

- Fixed:- An element is fixed position; will remain stuck to a specific position even after the page is scrolled. This position property is used when we want to keep an HTML element at a fixed spot no matter where on the page the user is.

- Sticky:- It is a hybrid of relative and fixed position. An HTML element with position sticky; will just sit there until a given position offset is met.

  Syntax:-
  ```
  #box1{ position: static;}
  #box2{position: relative;}
  #box3{position: absolute;}
  #box4{position: fixed;}
  #box5{position: sticky;}
  ```

7. CSS Visibility & Z-index Property

- Visibility property is used to hide or show an HTML element without changing the layout of the page.
- The hidden element uses the space on the page because it is still there, but it is not visible to the user.
- There is a minor difference between the display: none; and visibility:hidden; property of CSS.
- **Display** : none- It completely removes an HTML tag from the web page like it was never there.
- **Visibility**: hidden- It makes the tag invisible but will not remove the element, and it will still occupy the space on the page.

- **Syntax-:**

```
    <style>
.box {
   width: 170px;
   height: 170px;
   border: 2px solid black;
}
#box1 {
   background-color: greenyellow;
   display:none;
}
#box2 {
   background-color: rebeccapurple;
   visibility: hidden;
}
#box3 {
   background-color: blue;
}
#box4 {
   background-color: lightcoral;
}
```

- Z-index does not work on static position value. It only works on the elements with position: relative, absolute, fixed, or sticky

Syntax:-

```
          #box1 {
    top: 69px;
    position: relative;
    background-color: greenyellow;
    z-index: 1;
 }
 #box2 {
    top: 34px;
    position: relative;
    background-color: rebeccapurple;
    z-index: 0;
 }
```

8. **CSS Flex Box**

CSS flex box module was designed as a one-dimensional(1-D)Layout Model, and as a method that could offerspace distribution between items in an interface with powerful alignment capabilities.

- **Properties:-**
  - ➢ **flex-direction:** it gives directionto a flex item.
  - ➢ **flex-wrap:** items should wrap or not.
  - ➢ **flex-flow:** gives direction to items as well as wrap.
  - ➢ **justify-axis:** space between browser & content along main axis.
  - ➢ **align-items:** specifies default alignment inside flex box.
  - ➢ **align-content:** To horizontally center block elements.

9. **Resposiveness design**

- Responsive design is way for a web developer to make a website adapt to all devices and resolutions.
- Endless new resolutions and devices are challenging to support separately for a web developer; therefore responsive design is the best approach to develop a website that can adjust itself according to the screen size.
- Responsive design is necessity and not a luxury anymore!

Various Ways to Achieve Responsive Design:

- ➢ By using rem/vh/vw units over pixels.

- By using max-width/max-height.
- Using CSS Media Queries.
- Setting up the viewport.

I. **Em-**
- Font-sizes are inherited relatives to the parent element.
- 10em means ten times of the parent element.

II. **Rem**
- Font-size gets according to the font-size of the root element.
- In general,<html> is the root element.
- In rem, "r" stands for "root".

III. **Vh**
- It stans for viewport height
- Vh is equal to the 1/100 times of the viewport height.
- Suppose height of the browser is 1000px, so 1vh is equaled to (1/100)*1000=10px.

IV. **Vw**
- It stands for viewport width.
- Similar tovh, vw is eual to the 1/100 times of the viewport width.
- Suppose width of the browser is 1000px, so 1vw is equaled to (1/100)*1000= 10px.

10. Media Query

- Media queries are used when we want to customize our website's presentation according to the user's screen size. With the help of media queries, you can display different markups based upon the device's general type(mobile, desktop, tablet).
- A media query is a logical operation. Whenever a media query becomes true, then the related CSS is applied to the target element.
- Synatx:-

```
            @media screen and (max-width: 800px)
{
#contents{width:90%}
}
```

  - @media: A media query is always start with @media.
  - Screen: It is the applicable media type.
  - Max-width: It is the media feature.
  - #contents{width:90%}: It is the CSS to be applied when the conditions are met.

11. CSS Advanced selectors

- Div p:- Add styling to all the paragraphs that are inside the div.
- Div >p:- It is applying to all the <p> elements which are the direct child of any div element.
- Div +p:- It selects the <p> tags that are immediately after the <div> elements.

12. CSS Advanced pseudo selectors

- **Attribues:-**

```
a. Input[type= “text”] { border:23px;
                         Padding: 9px;
                        }
b. Input[type= ‘email’] {color: yellow;
                         Border: 4px solid black;
                         }
```

- **Selectors :-**

```
a. Li:nth-child(3){color: green;}
b. Li:nth-child(3n+3){color: red}
c. Li:nth-child(even){background-color: yellow;}
```

13. Pseudo selectors

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

14. CSS Shadow & Blur Property

- **Box-shadow** CSS property adds shadow effects around an element’s frame. We can set multiple effects separated by commas. It is described by X and Y offsets relative to the element, blur and spread radius, and colour.
- **Syntax:-**

```
{ box-shadow: 10px 13px green;}
```

- **Text-shadow** CSS property adds a shadow to text. It accepts a comma-separated list of shadows to be applied to the text and any of its decorations. It is also described by some combination of X and Y offsets from the element, blur radius, and colour.
- **Syntax:-**

```
        .card h2{
    text-shadow: 3px 4px red;
}
```

- **Blur-radius** property to the boxes. This property is used to make the border blur
- other property is **Spread-radius**. It is used to spread the color around the box.

15. Variables & Custom Property

- **‘--’ symbol: Variables** in CSS helps us to assign the same properties to different elements.
- **-- root:** it can be used within its scope only. we can make a global variable in terms of programming language.
- **Syntax:-**

```
 .box{
--box-color: violet;
width:200px;
height: 200px;
background-color: var(--box-color);
border: 2px solid var(--box-color);
```

```
        box-shadow: 3px 3px var(--box-color);
        margin: 2px 9px;
    }:root{
        --primary-color: blue;
        --danger-color: red;
        --maxw: 333px;
    }
```

- Any custom properties written in the root variable can be accessed anywhere in the code

16. Animations & Keyframes

- **Animations:**

- we need to start by giving the **Animation-name.**
- The next property used is **Animation-duration.**
- **Animation-Iteration:-** It is used to decide the number of times the animation will run. This is a last property.

- **@Keyframes:** Keyframes are used to make the animation. From and to are used to decide how the animation will move in the webpage.
- These types of animations are used to design scroll bars or progress bars on the webpage.
- There are some other properties also to customize the animations like-

  - **Animation-fill-mode:** If we want to keep the last property applied to the animation then we can set the animation-fill-property.
  - **Animation-timing-function:** We can define this property with three different values-
    - I. **ease-in:** After applying this, the animation will start slowly and becomes fast towards the end.
    - II. **ease-out:** After applying this, the animation will begin fastly and become slow towards the end.
    - III. **ease-in-out:** After applying this, the animation will start slowly, then become fast in the midway, and ends slowly.
  - **Animation-delay:** It is used to define the time after which the animation will start.
  - **Animation-direction:** This property is used to define the direction of the animation. For example, if we select it as reverse, it will move the animation in reverse direction.

17. CSS Transitions

- **Transitions-properties**: It is used to decide which transition property we want to use.
- **Transition-duration:** If we want to see the duration which is required to make the change, we can use this property.
- **Transition-timing-function:** This property is used to decide the speed of transition from beginning to end. These are of three types as follows-

I. **ease-in:** After applying this, the animation will start slowly and becomes fast towards the end.

II. **ease-out:** After applying this, the animation will begin fastly and become slow towards the end.

III. **ease-in-out:** After applying this, the animation will start slowly, then become fast in the midway, and ends slowly.

- **Transition-delay:** It is that particular time interval after which the transition effect will start. For example, if we set it as 2s, then the transition effect will start after 2 seconds only.

18. Transform Property

- The Transform property applies a 2D or 3D transformation to an element. This property allows you to rotate, scale, move, skew., elements. This property is also used in creating animations.
  - ➢ **Rotate:** If we want to rotate the box then use it. It will completely rotate the box to 360 degrees.
  - ➢ **Skew:** We can also **skew** the box through certain degrees. The skew property is used sometimes when we want to put the content on one side or we want to show the 3D effect.
  - ➢ **Scale:** We can also scale the box. The box will become large depending on the values we provide.
- **Translate:** Move the box in the x or y axis respectively.

19. CSS Grid Layout

- The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.
- CSS grids are the display properties that allow us to transform any box into the grid. The main difference between flexbox and grid is, in flexbox, we can either move the box in horizontal or vertical directions but in the grid system, we can make the two-dimensional grid systems
- The grid system should not be considered same as table because table is just a normal combination of rows and columns whereas grid is a full layout system.

Properties:

- ➢ Gap
- ➢ Grid-templates-columns
- ➢ Grid-template-rows
- ➢ Justify-content
- ➢ Align-content
- ➢ Align-items

I. **Spanning:** It can increase the number of items in the HTMl.

Properties:

- ➢ Grid-column-start{}
  Grid-column-end{}
- ➢ Grid-row-start{}
  Grid-column-end{}
- ➢ Grid-column {1/span column}
  Grid-row {1/span row}

II. **Autofit:** is that will repeat the number of times it is necessary.

III. **Minmax:** It will decide the minimum width to give to any rows.