UNIT 3

INTRODUCTION TO CSS

3.1 Introduction to stylesheet

CSS (Cascading Style Sheets) is a language designed to simplify the process of making web pages presentable. It allows you to apply styles to HTML documents, describing how a webpage should look by prescribing colors, fonts, spacing, and positioning. CSS provides developers and designers with powerful control over the presentation of HTML elements.

HTML uses tags and CSS uses rulesets. CSS styles are applied to the HTML element using selectors. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

Advantages of CSS

1. **Consistency:** One style code may apply across several pages, giving your website a unified look and feel.
2. **Flexibility:** Without changing the underlying HTML structure, CSS enables you to quickly modify your web page's look by changing styles in a single CSS file.
3. **Efficiency:** CSS helps to reduce repetitive code, resulting in smaller, quicker-loading web pages.
4. **Accessibility:** CSS provides better assistance for building web pages that are simpler to use and understand for those who have disabilities.
5. **Faster Page Speed:** Slower page performance results from more code. And CSS makes it possible to write less code. With CSS, you may apply a single CSS rule to each instance of a certain tag throughout an HTML text.
6. **Better User Experience:** CSS facilitates a user-friendly layout and makes websites to be pleasant to the eye. The user experience is enhanced whenever buttons and text are placed logically and are well-organized.

7.quicker Development Time: With CSS, you may use a single code to apply precise formatting guidelines and styles to several pages. Several internet pages can use the same cascading style sheet. Creating CSS rules for one page will be sufficient for every page of the same kind, such as product pages, which should all have the same layout, appearance, and feel.

8.Easy Formatting Changes: With CSS, it's simple to alter the format of a particular group of pages. There's no need to correct each page separately. Make modifications to the correct CSS stylesheet, and all the pages that are utilizing that stylesheet will be updated.

9.Compatibility Across Devices: Website responsiveness is important. Web pages must be completely viewable and user-friendly on all devices in the modern world. CSS and HTML work together to provide responsive design on all devices, including smartphones, tablets, desktops, and even smart TVs.

3.2 Basic syntax and structure

CSS Syntax

The CSS provides the style to the HTML element, which the browser interprets. After being interpreted by the browser, the CSS style property will be applied to all the elements of the HTML. We can provide style property to the HTML element in three parts. These three parts are as follows.

1. Selector

It is an HTML tag. All the style properties of the CSS will be applied to the selector. The selector tag like <h1> or <table> etc.

2. Property

It is a type of attribute that is present in HTML tags. All the attributes of the HTML will be converted to the CSS properties. The CSS properties like color, border, etc.

3. Value

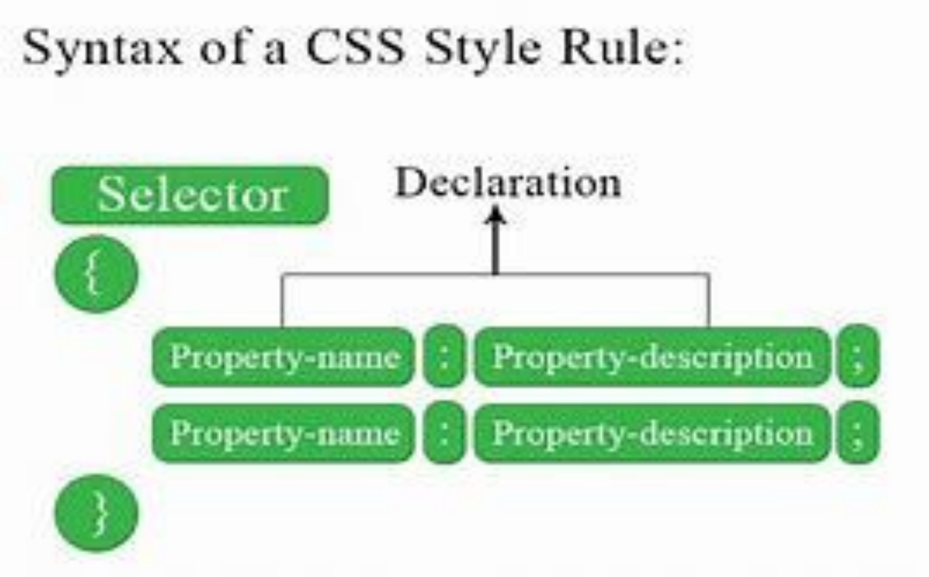
In HTML, these are assigned to the properties. For example, the color property can have a value of either red or #F1F1F1, etc.

Syntax:

We must provide the CSS property to the HTML element in a proper way. We must follow the syntax below to implement the CSS property.

selector { property: value }

Example:



Types of selectors

1.Universal Selectors

Styles applied using universal selectors are applied to all elements on the HTML page. Applying styles this way is considered inefficient as browsers have to parse all elements in the HTML document to apply the style for.

```
* { color: red; }
```

2.Element Selectors

HTML elements can be styled using their tag names. With the below CSS, all <h2> tags would be rendered in red color.

```
h2 { color: red; }
```

3.Class Selectors

Class selectors are useful if similar styles need to be applied to different parts of the application. Class selectors are defined using . followed by the class name like shown below:

```
.a-class { color: red; }
```

4.ID Selectors

Id attribute in HTML is used to uniquely identify an HTML element. These ids could also be used to style an element using # followed by the id.

```
#my-id { color: red; }
```

5.Attribute Selectors

HTML elements could be styled based on the presence of an attribute using the Attribute selector. For example below declaration would add border to any element with disabled attribute.

```
[disabled] { border: 2px solid black; }
```

6.Group Selectors

If more than one class or element share the same styles they could be comma separated and defined with a single declaration to avoid duplication.

```
#my-id, .b-class { color: red; }
```

7.Descendant Selectors

All elements that are nested inside a given element are called descendants. For example, with the below selector we can style all paragraph elements that are nested in the <body> tag.

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

8.Child Selectors

Parent Child selector is a little more specific than the descendant selector. It only styles the elements that are direct children of the selector defined on the left side. Below declaration applies only to anchor tags that are immediate children of paragraph tag. If <a> tag happens to be nested inside another <span> tag it will not receive the style.

```
p > a { color: blue; }
```

3.3 Types of style sheet

1.Inline CSS

Inline CSS involves applying styles directly within an HTML element's style attribute. This method is often used for quick styling or when styling needs to be applied to a specific element without affecting others.

Example:

```
<p style="color: blue; font-size: 18px;">This is an
inline styled paragraph.</p>
```

2. Internal (Embedded) CSS

Internal CSS is defined within a <style> tag in the <head> section of an HTML document. This method allows styling rules to be applied to the entire document, making it useful for single-page applications.

Example:

```
<style>
h1 {
color: darkgreen; text-align: center;
} </style>
```

3. External CSS

External CSS is defined in separate .css files and linked to HTML documents using the <link> tag in the <head> section. This method is ideal for larger projects where styles need to be applied consistently across multiple pages.

Example:

```
<head>
<link rel="stylesheet" href="styles.css">
</head>
```

Styles.css

```
p { color: gray; line-height: 1.5; }
```

3.3 CSS Border

- **Different Border Related Properties**

We have the following commonly used border-related properties:

1. border-style: specifies the styles of the border
2. border-width: sets the width of the border
3. border-color: sets the color of the border
4. border: shorthand property to
   specify border-style, border-width, and border color
5. border-radius: sets the radius of the border
6. border-image: allows to set an image as a border

Border style

Dotted	Creates a series of dots.
Dashed	Forms a dashed line.
Solid	Produces a continuous line.
Double	Renders two parallel lines.
Groove	Creates 3D grooved effect.
Ridge	Creates 3D ridged effect.
Inset	Adds 3D inset border.
Outset	Adds 3D outset border.
None	Removes the border.
Hidden	Hides the border.

```
<html >
<head>
<link rel="stylesheet"
href="style.css" /> <title>CSS
border-style</title> </head>
<body>
<p class="solid">border-style:
solid;</p>
<p
class="dotted">border-style:
dotted;</p>
<p
class="dashed">border-style:
dashed;</p> </body> </html>
```

Style.css

```
/* create a solid border */
p.solid { border-style: solid; } /*
create a dotted border */
p.dotted { border-style: dotted;
} /* create a dashed border */
p.dashed { border-style:
dashed; } p { padding: 8px; }
```

- **CSS Border Width**
- Border width sets the width of the border. The width of the border can be in px, pt, cm or thin, medium, and thick.

```
<head> <style>
 p
{
border-width: 8px;
} </style> </head>
```

- **CSS Border Color**
- This property is used to set the color of the border. Color can be set using the color name, value, or RGB value.
- <head> <style>

```
 p
 {
border-color:red;
} </style> </head>
```

Border Color property

- When one value is specified, it applies the same color to all four sides.
- When two values are specified, the first color applies to the top and bottom, the second to the left and right.
- When three values are specified, the first color applies to the top, the second to the left and right, the third to the bottom.
- When four values are specified, the colors apply to the top, right, bottom, and left in that order (clockwise).

- **Border radius property**
- The CSS border-radius property rounds the corners of an element’s border, creating smoother edges, with values specifying the curvature radius.
- **border-image property** allows you to specify an image to be used as the border around an element.

```
<head> <style>
 p {
border-image:
url(border.png) 30% round;

border-radius: 35px;
background: green;
}
 </style> </head>
```

- The border-radius property defines the radius of the element's corners.
- This property allows you to add rounded corners to elements!
- This property can have from one to four values. Here are the rules:
- **Four values - border-radius: 15px 50px 30px 5px;** (first value applies to top-left corner, second value applies to top-right corner, third value applies to bottom-right corner, and fourth value applies to bottom-left corner):
- **Three values - border-radius: 15px 50px 30px;** (first value applies to top-left corner, second value applies to top-right and bottom-left corners, and third value applies to bottom-right corner):
- **Two values - border-radius: 15px 50px;** (first value applies to top-left and bottom-right corners, and the second value applies to top-right and bottom-left corners):
- **One value - border-radius: 15px;** (the value applies to all four corners, which are rounded equally:

CSS Margins

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).

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

margin-top

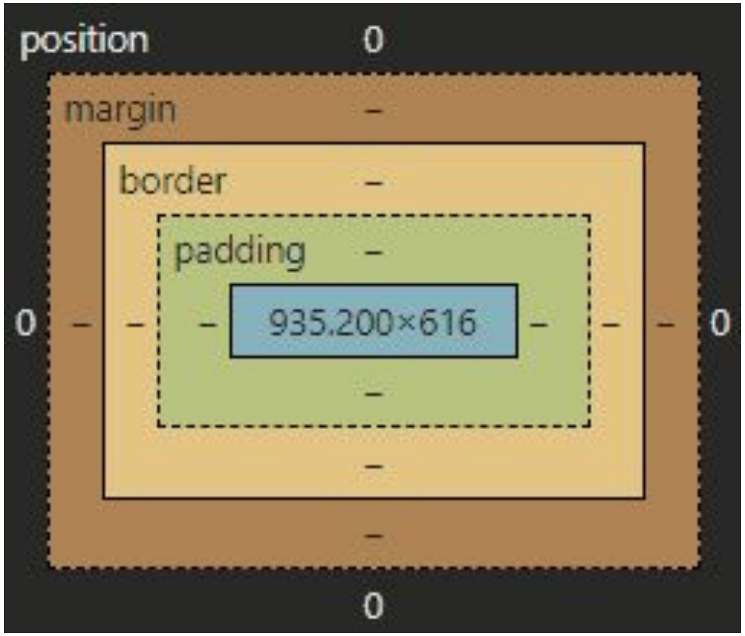
margin-right

margin-bottom

margin-left

```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

3.3 Css margin



- Margins, as defined by the CSS margin property, create spaces around an element, setting it apart from neighbouring elements. You can individually set margins for each side: top, right, bottom, and left. The margin values can be specified in various units (e.g., pixels, rems, ems, percentages). margins also allow negative values.

- **Margin Properties**

1. **margin-top**: Sets the top margin of an element.
2. **margin-right**: Sets the right margin of an element.
3. **margin-bottom**: Specifies the margin at the bottom of an element.
4. **margin-left**: Determines the width of the margin on the left side of an element.

- **top and bottom =** 80px;
- **left and right =** 100px;

```
<head>
 <style>
p {
 margin: 80px 100px 50px 80px;
}
h1 {
 margin: 80px 100px;
}

 </style>
 </head>
```

3.4 CSS Positioning

- CSS positioning allows you to control the layout of HTML elements using properties such as top, right, bottom, and left. Understanding the different types of position properties in CSS static, relative, absolute, fixed, and sticky can enhance the design and functionality of web pages. This article explores these properties with examples, illustrating how each can be applied to achieve various layout effects.

Position Property	Description
Fixed	An element with position: fixed property remains in the same position relative to the viewport even when the page is scrolled.
Static	Default positioning method. Elements with position: static are positioned according to the normal flow of the document.
Relative	Elements with position: relative are positioned relative to their normal position in the document flow. Other elements will not fill the gap left by this element when adjusted.
Absolute	Positioned concerning its nearest non-static ancestor. Elements with position: absolute are taken out of the normal document flow.
Sticky	Combines features of position: relative and position: fixed. The element is treated as position: relative until it reaches a specified threshold, then it becomes position: fixed.

1.Background properties

The **background** shorthand CSS property sets all background style properties at once, such as color, image, origin and size, or repeat method. Component properties not set in the background shorthand property value declaration are set to their default values.

1.CSS Background color

The background-color property is specified as a single <color> value.

```
/* Keyword values */
background-color: indigo;
/* Hexadecimal value */
background-color: #bbff00; /* Fully opaque */
background-color: #bf0; /* Fully opaque shorthand */
/* RGB value */
background-color: rgb(255 255 128);
background-color: rgb(117 190 218 / 50%); /* 50% transparent */
/* Special keyword values */
background-color: currentcolor;
background-color: transparent;
```

2.CSS Background Image

- The background-image property specifies an image to use as the background of an element.
- By default, the image is repeated so it covers the entire element.

Ex:

```
div {
 background-image: url("mdn_logo_only_color.png"); }
.catsandstars
 {
background-image: url("startransparent.gif"), url("catfront.png");
 background: no-repeat, repeat;
}
```

3.CSS Background Position

The background-position property sets the starting position of a background image.

Value	Description
left top left center left bottom right top right center right bottom center top center center center bottom	If you only specify one keyword, the other value will be "center“
x% y%	The first value is the horizontal position and the second value is the vertical. The top left corner is 0% 0%. The right bottom corner is 100% 100%. If you only specify one value, the other value will be 50%. Default value is: 0% 0%
xpos ypos	The first value is the horizontal position and the second value is the vertical. The top left corner is 0 0. Units can be pixels (0px 0px) or any other CSS units. If you only specify one value, the other value will be 50%. You can mix % and positions

Ex:background-repeat

```
body {
background-image:
   url(“abc.jpg");
 background-repeat: repeat-x;
background-size: 200px 150px;
}
```

Ex: background Position

```
   body{
  background-position: top;
    background-position: bottom;
   background-position: left;
    background-position: right;
background-position: center;
background-position: 25% 75%;
background-position: 25px 75px;
}
```

4.CSS Background Repeat

The background-repeat property in CSS is used to repeat the background image both horizontally and vertically. It also decides whether the background-image will be repeated or not.

Property Values:

repeat: This property is used to repeat the background image both horizontally and vertically. The last image will be clipped if it is not fit in the browser window.

repeat-x: This property is used to repeat the background image horizontally.

repeat-y: This property is used to set the background image repeated only vertically.

no-repeat: This property does not repeat the background image. It displays the background image only

5.background-attachment

The background-attachment CSS property sets whether a background image's position is fixed within the viewport or scrolls with its containing block.

```
 background-attachment: scroll;
 background-attachment: fixed;
 background-attachment: local;
p {
background-image: url("starsolid.gif");
background-attachment: fixed;
}
```

2.CSS Font Properties

1. font-family

The font-family CSS property specifies a prioritized list of one or more font family names and/or generic family names for the selected element.

Ex:

```
P{
font-family: monospace;
 font-family: cursive;
font-family: fantasy;
}
```

2.font-size

The **font-size** CSS property sets the size of the font. Changing the font size also updates the sizes of the font size-relative <length> units, such as em, ex.

Ex:

```
P{
    font-size: xx-small;
    font-size: x-small;
     font-size: small; font-size: medium;
    font-size: large; font-size: 20px;
    font-size: 50%;
  }
```

3.Font –Style

The font-style CSS property sets whether a font should be styled with a normal, italic,oblique.

Ex:

```
P{
font-style: normal;
 font-style: italic;
Font-style:oblique;
}
```

4. font-variant

We can set font variant using the font variant property.

Ex:

```
P
{
font-variant: small-caps;
font-variant:normal;
}
```

5.font-weight

The **font weight** CSS property sets the weight (or boldness) of the font.

Ex

```
P{
font-weight: normal;
 font-weight: bold;
font-weight: lighter;
font-weight: bolder;
font-weight: 100;
}
```

3.CSS Text properties

CSS Text Formatting refers to applying styles to text elements to control appearance and layout. This includes properties for color, alignment, decoration, indentation, justification, shadows, spacing, and direction. These properties enhance readability and improving the presentation of textual content on web pages.

1.CSS letter spacing

The letter-spacing property in CSS is used to control the spacing between text characters, allowing you to increase or decrease the space between characters.

Property Values:

normal: The normal letter spacing for the current font i.e no extra space between characters. This is the default value.

length: This mode specifies extra inter-character space in addition to the default space between characters. Negative values are also allowed.

initial: This mode sets this property to its default value.

Ex:

```
p {
 letter-spacing: normal;
letter-spacing: 5px;
}
```

2.CSS text decoration

The text-decoration property is used to add a decoration line to text.

```
    text-decoration: value;
```

Ex:

```
h1{
    text-decoration: overline;
    text-decoration: line-through;
    text-decoration: underline;
    text-decoration: underline overline;
}
```

3.CSS text align

The text-align property specifies the horizontal alignment of text in an element.

Ex

```
div.a {
    text-align: center;
   }

   div.b {
    text-align: left;
   }

   div.c {
    text-align: right;
   }

   div.c {
    text-align: justify;
   }
```

4.CSS text transform

The **text-transform property** controls the capitalization of text. It can make text uppercase, lowercase, or capitalize the first letter of each word.

none Default value. No capitalization.

Capitalize Transforms the first character of each word to uppercase.

Uppercase Converts all characters in each word to uppercase.

Lowercase Converts all characters in each word to lowercase.

initialSets the property to its default value.

Ex:

```
p {
 text-transform: capitalize;
 }
```

5.CSS Lists

- In HTML, there are two main types of lists:
- unordered lists (<ul>) - the list items are marked with bullets
- ordered lists (<ol>) - the list items are marked with numbers or letters
- The CSS list properties allow you to:
- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker
- Add background colors to lists and list items

1. **list-style-type** property specifies the type of list item marker.

Ex:

```
ul
{
List-style-type:square;
}
```

2. **list-style-image** property specifies an image as the list item marker.

Ex:

```
ul {
  list-style-image: url('sqpurple.gif');
}
```

3**. list-style-position** property specifies the position of the list-item markers (bullet points).

i)list-style-position: outside; means that the bullet points will be outside the list item. The start of each line of a list item will be aligned vertically.

ii) list-style-position: inside; means that the bullet points will be inside the list item. As it is part of the list item, it will be part of the text and push the text at the start.

Ex:

```
ul.a {
  list-style-position: outside;
}

ul.b {
  list-style-position: inside;
}
```

6.CSS Link

links can be styled differently depending on what state they are in.

The four links states are:

a:link - a normal, unvisited link

a:visited - a link the user has visited

a:hover - a link when the user mouses over it

a:active - a link the moment it is clicked

Example

```
/* unvisited link */
a:link {
  color: red;
}

/* visited link */
a:visited {
  color: green;
}

/* mouse over link */
a:hover {
  color: hotpink;
}

/* selected link */
a:active {
  color: blue;
}
```

7. CSS table

1.table-layout property defines the algorithm used to lay out table cells, rows, and columns.

Ex:

```
table {
  table-layout: auto;
  width: 180px;
}

.b {
  table-layout: fixed;
  width: 180px;
}
```

2. border property is used to specify borders for table elements.

Ex:table, th, td {

border: 1.5px solid blue; }

3. border-collapse property tells us whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.

Ex:

Table{

border-collapse: collapse/separate;

}

4.Border Spacing property specifies the space between the borders of the adjacent cells.

Ex:

Table{border-spacing:5px;}

5.Caption Side

Caption Side property is used for controlling the placement of caption in the table. By default, captions are placed above the table.

Syntax:

```
caption-side: top/bottom;
```

Ex:

```
table{caption-side: bottom; }
<table ><caption>Caption at the top of the table.</caption></table>
```

6.Empty cell

Empty cell property specifies whether or not to display borders and background on empty cells in a table.

Syntax:

```
empty-cells:show/hide;
```

Ex:

```
table{empty-cells: hide;}
table{empty-cells: show; }
```

8.CSS image opacity

The opacity property specifies the opacity/transparency of an element.

The opacity property can take a value from 0.0 - 1.0 . The lower the value, the more transparent.

Default value of image 1.

Ex:

```
img {
  opacity: 0.5;
}
img:hover {
  opacity: 1.0;
}
```

9.CSS Display and visibility properties

display: none - unlike the first property, this means the element will not appear on the page at all. In this case, the tag is removed from the normal flow of the page, which allows other elements to fill it in.

Ex:

```
.block1 { background-color: rgb(224, 110, 49);
margin-right: 20px; display: none; }
```

visibility: hidden - this CSS property makes the text *invisible*, but the space allocated for it will remain. In other words, the element is hidden from view but not the page flow.

Ex:

```
.block2 {
 background-color: rgb(77, 77, 234);
margin-right: 20px;
visibility: hidden; }
```

3.5 Use of Id & classes in CSS

ID Selector in CSS

The id selector is used to select the id attribute of an HTML element for selecting a particular element. An id is always unique within the page, so it is chosen to select a single, unique element.It is written with the hash character (#), followed by the id of the element.

Example

```
#para {
text-align: center;  color: blue;
}  <body>
<p id = "para">This paragraph will be affected.</p>
</body>
```

Class Selector

The class selector is used to select the HTML elements with a specific class attribute. It is written with a period character (full stop symbol) followed by the class name.

Ex:

```
.example {
font-size: 25px;
}
<body>
<h1 class="example">This heading is blue and center-aligned.</h1>
```

The difference between the id and class

Class	Id
We can apply a class to various elements so that it could be numerous times on a single page.	The Id is unique in a page, and we can only apply it to one specific element.
The class is assigned to an element and its name starts with "." followed by the name of the class.	The name of the Id starts with the "#" symbol followed by a unique id name.
We can attach multiple class selectors to an element.	We can attach only one ID selector to an element.

Syntax:
.class{
// declarations of CSS
}
	Syntax:
	#id{
	// declarations of CSS
	}

3.6 Use of &

In CSS, the ampersand (&) is not directly used in traditional CSS syntax but is a significant part of the syntax in pre-processors like Sass (Syntactically Awesome Style Sheets) and Less. Here's how the ampersand is used in these pre-processors:

In Sass (SCSS Syntax)

1.Nesting Selectors: The ampersand is often used in Sass to refer to the parent selector when nesting styles. This allows you to write more organized and readable CSS.

Ex:

```
.button { background-color: blue;
 color: white;
&:hover { background-color: darkblue; }
&.active { background-color: green; } }
```

3.7 Introduction to CSS3

CSS3 is the latest evolution of the Cascading Style Sheets language, which is used to style and layout web pages. CSS3 brings a host of new features and improvements over its predecessors, allowing for more advanced and flexible styling. Here are some of the key features of CSS3:

1. Some of the key modules of CSS3 are:
2. Box model
3. Image values and replaced content
4. Text effects
5. Selectors
6. Backgrounds and borders
7. Animations
8. User interface (UI)
9. Multiple column layouts
10. 2D/3D transformations

Why use CSS3

- CSS3 is used because it allows you to make websites look better and work more smoothly. It offers:
- **Better Design**: With new features, you can create more visually appealing and creative designs.
- **Responsive Layouts**: It helps websites look good on all devices, from phones to desktops.
- **Improved Layouts**: New tools make it easier to arrange content neatly and flexibly.
- **Smooth Animations**: You can add cool animations and transitions without needing extra scripts.

1.Gradients:

CSS gradients let you display smooth transitions between two or more specified colors.

CSS defines three types of gradients:

Linear Gradients (goes down/up/left/right/diagonally)

Radial Gradients (defined by their center)

Conic Gradients (rotated around a center point)

1. CSS Linear Gradients

To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect.

background-image: linear-gradient(*direction*, *color-stop1*, *color-stop2, ...*);

Direction - Top to Bottom (this is default)

Ex:

```
#grad {
  background-image: linear-gradient(to right, red , yellow);
}
```

Direction - Diagonal

You can make a gradient diagonally by specifying both the horizontal and vertical starting positions.

The following example shows a linear gradient that starts at top left (and goes to bottom right). It starts red, transitioning to yellow:

Ex:

```
#grad {
  background-image: linear-gradient(to bottom right, red, yellow);
}
```

Using Angles

If you want more control over the direction of the gradient, you can define an angle, instead of the predefined directions (to bottom, to top, to right, to left, to bottom right, etc.). A value of 0deg is equivalent to "to top". A value of 90deg is equivalent to "to right". A value of 180deg is equivalent to "to bottom".

Syntax

background-image: linear-gradient(*angle*, *color-stop1*, *color-stop2*);

Ex:

```
#grad {
  background-image: linear-gradient(180deg, red, yellow);
}
```

2.CSS Radial Gradients

A radial gradient is defined by its center.

To create a radial gradient you must also define at least two color stops.

Syntax

background-image: radial-gradient(*shape size* at *position, start-color, ..., last-color*);

Radial Gradient - Evenly Spaced Color Stops (this is default)

Ex:

```
#grad {
  background-image: radial-gradient(red, yellow, green);
}
```

Radial Gradient - Differently Spaced Color Stops

Ex:

```
#grad {
  background-image: radial-gradient(red 5%, yellow 15%, green 60%);
}
```

Set Shape

The shape parameter defines the shape. It can take the value circle or ellipse. The default value is ellipse.

Ex:

```
#grad {
  background-image: radial-gradient(circle, red, yellow, green);
}
```

3.Conic Gradients (rotated around a center point)

A conic gradient is a gradient with color transitions rotated around a center point.

To create a conic gradient you must define at least two colors.

Syntax

background-image: conic-gradient([from *angle*] [at *position*,] *color* [*degree*]*,*
color [*degree*]*, ...*);

By default, *angle* is 0deg and *position* is center.

If no *degree* is specified, the colors will be spread equally around the center point.

Conic Gradient: Three Colors

```
Ex:#grad {
  background-image: conic-gradient(red, yellow, green);
}
```

Conic Gradient: Five Colors

Ex:

```
#grad {
  background-image: conic-gradient(red, yellow, green, blue, black);
}
```

Conic Gradient: Three Colors and Degrees

Ex:

```
#grad {
  background-image: conic-gradient(red 45deg, yellow 90deg, green 210deg);
}
```

2.Transitions

Transitions: transition allows smooth changes from one style to another over a specified duration.

The **transition-property CSS property** sets which CSS properties will have a transition effect, enabling smooth animations for specified properties when they change, without altering the layout or requiring JavaScript.

This property is a combination of four sub-properties

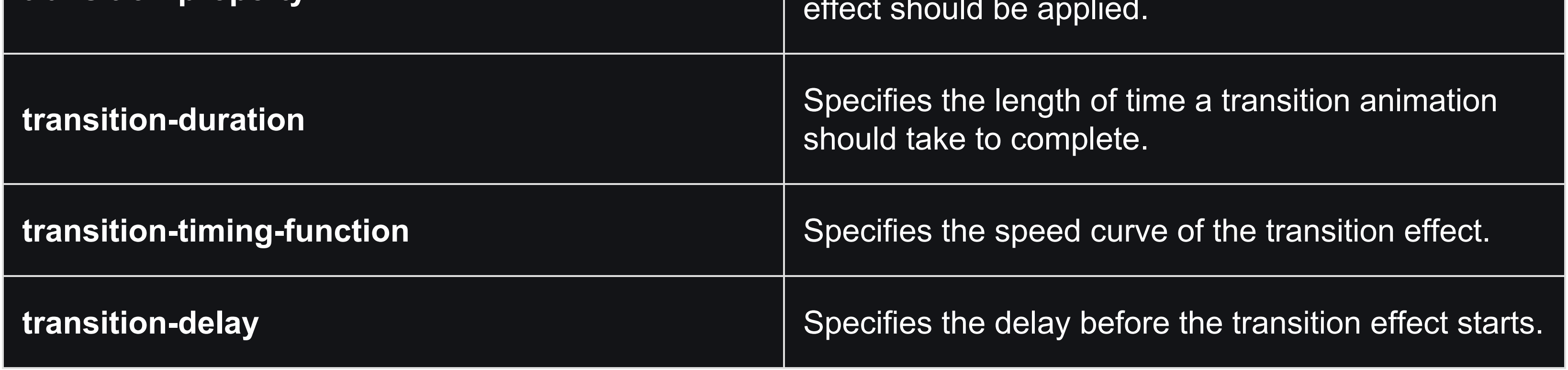
transition-property

transition-duration

transition-timing-function

transition-delay

Note: The transition effect can be defined in two states (hover and active) using pseudo-classes like hover or active or classes dynamically set by using JavaScript.



	effect should be applied.
transition-duration	Specifies the length of time a transition animation should take to complete.
transition-timing-function	Specifies the speed curve of the transition effect.
transition-delay	Specifies the delay before the transition effect starts.

Ex:

```
div {
  width: 100px;
  height: 100px;
  background: red;
  transition-property: width;
  transition-duration: 2s;
 transition: width 2s, height 4s;
}

div:hover {
  width: 300px;
}
```

Specify the Speed Curve of the Transition

The transition-timing-function property specifies the speed curve of the transition effect.

The transition-timing-function property can have the following values:

ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)

linear - specifies a transition effect with the same speed from start to end

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

cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

Ex:

```
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}
```

Delay the Transition Effect

The transition-delay property specifies a delay (in seconds) for the transition effect.

The following example has a 1 second delay before starting:

Ex:

```
div {
  transition-delay: 1s;
}
```

3.Animations

Animations: @keyframes enables creating complex animations by defining intermediate steps between styles. An animation lets an element gradually change from one style to another.

You can change as many CSS properties you want, as many times as you want.

To use CSS animation, you must first specify some keyframes for the animation.

Keyframes hold what styles the element will have at certain times.

```
<style>
div {
 width: 100px;
 height: 100px;
 background-color: red;
 animation-name: example;
 animation-duration: 4s;
}

@keyframes example {
 from {background-color: red;}
 to {background-color: yellow;}
}
```

EX:

```
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}

@keyframes example {
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}
</style>
</head>
<body>

<h1>CSS Animation</h1>

<div></div>
</body>
</html>
```

4.The multiple columns

The multiple columns are used to create column layout in the web pages. There are many column property in CSS which are listed below:

column-count

column-gap

column-rule-style

column-rule-width

column-rule-color

column-rule

column-span

column-width

1.The column-count property: This property is used to count the number of column element in document that should be divided.

Ex:

```
.abc
{
        -webkit-column-count: 3;
        -moz-column-count: 3;
        column-count: 3;
}
```

2. **The columns-gap property:** This property is used to define the gap between columns.

Ex:

```
.abc {
        -webkit-column-count: 3;
        -moz-column-count: 3;
        column-count: 3;
        padding-top: 35px;
        text-align: justify;
        -webkit-column-gap: 50px;
        -moz-column-gap: 50px;
        column-gap: 50px;
        width: 100%;
     }
```

3. The column-rule-style property: This property is used to specify the style between the columns.

Ex:

```
.abc{column-count: 3;
        padding-top: 35px;
        column-rule-style: solid;
        text-align: justify;}
```

4.The column-rule-color property: This property is used to define the color between the columns.

Ex:

```
p{
column-rule-color: green;
        column-count: 3;
        padding-top: 35px;
          column-gap: 50px;
         column-rule-style: solid;
        column-rule-width: 10px;
          text-align: justify;
}
```

5.The column-rule-width property: This property is used to specify the width of rule between columns.

Ex:
```
.geek{
     column-count: 3;
    padding-top: 35px;
       column-gap: 50px;
     column-rule-style: solid;
    column-rule-width: 10px;
      text-align: justify;}
```

6.The column-rule property: This property is used to define the style, width, and color of the rule between columns.

Ex:
```
.geeks_content {
     column-count: 3;
    column-rule: 10px solid green;
     width: 100%;
     text-align: justify;
  }
```

3.8 Introduction of Bootstrap

Bootstrap is a widely used open-source framework designed to facilitate web development. It provides a collection of tools and components that streamline the process of building responsive, mobile-first websites and web applications. Here's a comprehensive introduction to Bootstrap:

Overview

Created By: Bootstrap was originally developed by Mark Otto and Jacob Thornton at Twitter in 2011. It was open-sourced in 2011 and has since become one of the most popular front-end frameworks.

Current Version: As of the latest update, Bootstrap 5 is the current major version. It includes significant improvements and new features compared to previous versions.

Key Features

1. **Responsive Grid System:** Bootstrap includes a flexible grid system that uses a series of containers, rows, and columns to layout and align content. The grid system is built with a 12-column layout, which adapts to different screen sizes, allowing for responsive design.
2. **Predefined CSS Classes:** The framework provides a wide range of predefined CSS classes for common design elements like buttons, forms, tables, and typography. These classes help speed up development by reducing the need for custom CSS.
3. **Components:** Bootstrap comes with a variety of reusable components such as navigation bars, modals, carousels, and alerts. These components are designed to be easy to use and customize, providing consistent styling across different parts of your website.
4. **JavaScript Plugins:** Bootstrap includes a set of JavaScript plugins that enhance the functionality of its components. These plugins cover features like tooltips, popovers, and modals, and are built using jQuery.
5. **Customizable:** Bootstrap is highly customizable. You can modify its default styles by overriding CSS classes or using Bootstrap’s built-in Sass variables to adjust themes and components to fit your design needs.
6. **Accessibility:** Bootstrap is designed with accessibility in mind, providing features and components that help ensure your website is usable by people with disabilities.

Basic Usage

To start using Bootstrap, you can include it in your project in a few different ways:

1.Via CDN (Content Delivery Network): You can link to Bootstrap's CSS and JavaScript files directly from a CDN, which is a quick and easy way to get started.

```
<link
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"
></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scri
pt>
```

2.Via npm (Node Package Manager): If you're using a build tool like Webpack or a framework like React, you can install Bootstrap via npm.

```
npm install bootstrap
```

```
import 'bootstrap/dist/css/bootstrap.min.css';
import 'bootstrap/dist/js/bootstrap.bundle.min';
```

3.Downloading Files: You can also download the Bootstrap files from its official website and include them in your project manually.