CSS

Cascading Style Sheet

By- Rameshkumar Yadav

CSS Introduction

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on web pages.
- CSS is the language we use to style a Web page.

- **Why Use CSS?**
- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Solved a Big Problem

- **HTML was created to describe the content of a web page, like: heading, paragraph, image, Etc.**
- **When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.**
- **To solve this problem, the World Wide Web Consortium (W3C) created CSS.**

How To Add CSS

- There are three ways of inserting a style sheet:

1. Inline CSS
2. Internal CSS
3. External CSS

Inline CSS

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.
- Inline styles are defined within the "style" attribute of the relevant element.
- Example:-

```
<h1 style="color:blue;text-align:center;">
                                        Coding Seekho</h1>
<p style="color:red;">This is a paragraph.</p>
```

Internal CSS

- An internal style sheet may be used if one single HTML page has a unique style.
- The internal style is defined inside the <style> element, inside the head section.

```
<head>
        <style>
                h1 {
                        color: maroon;
                   }
        </style>
</head>
```

- <h1>This is a heading</h1>

External CSS

- With an external style sheet, you can change the look of an entire website by changing just one file!
- Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.
- The external .css file should not contain any HTML tags.
- Note: Do not add a space between the property value and the unit:

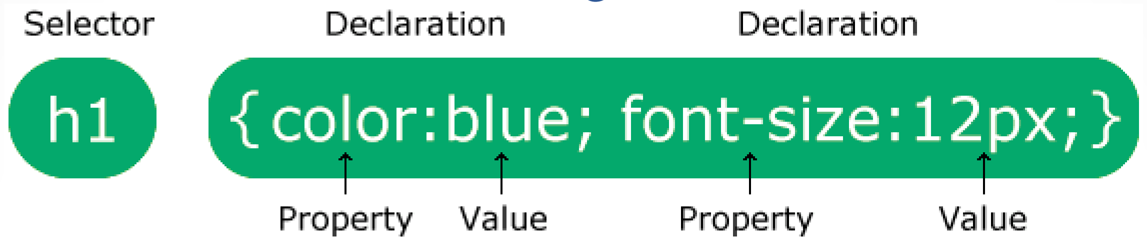
```
<head>
	<link rel="stylesheet" href="styles.css”>
</head>
```

CSS Comments

- CSS comments are not displayed in the browser, but they can help document your source code.
- Comments are used to explain the code, and may help when you edit the source code at a later date.
- Comments are ignored by browsers.

- A CSS comment is placed inside the <style> element, and starts with /* and ends with */

CSS Syntax



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

CSS Selectors

- CSS selectors are used to "find" (or select) the HTML elements you want to style.
- The CSS element Selector
- The element selector selects HTML elements based on the element name.

```
<style>
        p {
        text-align: center;
        color: red;
        }
</style>
```

The CSS id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.
- Note: An id name cannot start with a number!

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

- `<p id="para1">Hello World!</p>`

The CSS class Selector

- The class selector selects HTML elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the class name.

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

- `<h1 class="center">Coding Seekho</h1>`
- `<p class="center">Coding Seekho</p>`

The CSS class Selector

- You can also specify that only specific HTML elements should be affected by a class.
- In this example only <p> elements with class="center" will be red and center-aligned:

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

- `<h1 class="center">Coding Seekho</h1>`
- `<p class="center">Coding Seekho</p>`

The CSS class Selector

- **HTML elements can also refer to more than one class.**
- **Note: A class name cannot start with a number!**
- **In this example the <p> element will be styled according to class="center" and to class="large":**

```
p.center {
        text-align: center;
        color: red;
        }
p.large { font-size: 300%; }
```

- <h1 class="center">Coding Seekho</h1>
- <p class="center">Coding Seekho</p>
- <p class="center large">Coding Seekho</p>

The CSS Universal Selector

- The universal selector (*) selects all HTML elements on the page.

```
* {
        text-align: center;
        color: blue;
}
<h1>Hello world!</h1>
<p>Every element on the page will be affected by the style.</p>
<p>Coding Seekho</p>
```

The CSS Grouping Selector

- The grouping selector selects all the HTML elements with the same style definitions.
- To group selectors, separate each selector with a comma.

```
h1, h2, p {
        text-align: center;
        color: red;
        }
```

- <h1>Hello World!</h1>
- <h2>Coding Seekho</h2>
- <p>This is a paragraph.</p>

CSS Backgrounds

- The CSS background properties are used to add background effects for elements.
- In these lecture, you will learn about the following CSS background properties:

- background-color
- background-image- Url(‘’)
- Background-size
- background-repeat
- background-attachment
- background-position
- background (shorthand property)

CSS background-color

- The background-color property specifies the background color of an element.
- You can set the background color for multiple HTML elements.
- With CSS, a color is most often specified by a valid color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)“
- Example:-

```
    body {
            background-color: lightblue;
         }
```

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.
- Example:-

```
body {  background-image: url("paper.gif");  }
```

- Note: When using a background image, use an image that does not disturb the text.
- The background image can also be set for specific elements.
- Example:-

```
p { background-image: url("paper.gif");  }
```

CSS background-repeat

- By default, the background-image property repeats an image both horizontally and vertically.
- Some images should be repeated only horizontally or vertically, or they will look stranger.
- To repeat an image vertically, set background-repeat: repeat-y;
- To repeat an image horizontally, set background-repeat: repeat-x;
- Showing the background image only once is also specified by the background-repeat: no-repeat;

```
body {
  background-image: url("gradient_bg.png");
        background-repeat: repeat-x;
        background-repeat: repeat-y;
        background-repeat: no-repeat;
}
```

CSS background-position

- The background-position property is used to specify the position of the background image.
- The background image is placed in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.
-
```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
}
```
- Position- left, right, center, top, bottom, or in px.

CSS background-attachment

- The background-attachment property specifies whether the background image should scroll or be fixed.

- Example:-

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: fixed;
  background-attachment: scroll;
}
```

background - Shorthand property

- To shorten the code, it is also possible to specify all the background properties in one single property. This is called a shorthand property.
- Example:-

-
  ```
  body {
    background: #ffffff url("img_tree.png") no-repeat right
  top;
  }
  ```
- When using the shorthand property the order of the property values is:
- background-color, background-image, background-repeat, background-attachment, background-position.
- It does not matter if one of the property values is missing, as long as the other ones are in this order.

CSS border-style

- The CSS border properties allow you to specify the style, width, and color of an element's border.
- The border-style property specifies what kind of border to display.
- The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).
- Example:-

```
<style>
        p {border-style: solid;}
</style>
```

CSS border-style

- The following values are allowed:
- dotted - Defines a dotted border
- dashed - Defines a dashed border
- solid - Defines a solid border
- double - Defines a double border
- groove - Defines a 3D grooved border.
- ridge - Defines a 3D ridged border.
- inset - Defines a 3D inset border.
- outset - Defines a 3D outset border.
- none - Defines no border
- hidden - Defines a hidden border
- Mix- {border-style: dotted dashed solid double;}

CSS border-width

- **The border-width property specifies the width of the four borders.**
- **The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.**

-
  ```
  p {
    border-style: solid;
    border-width: medium;
  }
  ```
- **The border-width property can have from four values (top, right, bottom, and left border):**

-
  ```
  p {
    border-style: solid;
    border-width: 25px 10px 4px 35px;
  }
  ```

CSS border-color

- The border-color property is used to set the color of the four borders.

-
  ```
  p {
    border-style: solid;
    border-color: green;
  }
  ```
- The border-color property can have four values (top, right, bottom, and left border).

-
  ```
  p {
    border-style: solid;
    border-color: red green blue yellow;
  }
  ```

CSS Border Sides

- You can specify a different border for each side.
- In CSS, there are also properties for specifying each of the borders (top, right, bottom, and left).

-
  ```
  p {
    border-top-style: dotted;
    border-right-style: solid;
    border-bottom-style: dotted;
    border-left-style: solid;
  }
  ```
- border-style: dotted solid double;
- border-style: dotted solid;
- border-style: dotted;

CSS Border - Shorthand Property

- The border property is a shorthand property for the following individual border properties:
- border-width
- border-style (required)
- border-color

-
  ```
  p {
    border: 5px solid red;
  }
  ```
- You can also specify all the individual border properties for just one side.

-
  ```
  p {
    border-left: 6px solid red;
  }
  ```

CSS Rounded Borders

- The border-radius property is used to add rounded borders to an element.

-
  ```
  p {
    border: 2px solid red;
    border-radius: 5px;
  }
  ```
- You can also specify all the individual rounded border value.

-
  ```
  p {
    border: 2px solid red;
    border-radius: 5px 10px;
  }
  ```

CSS Margins

- **Margins 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-top: 20px;**
- **margin-right: 40px;**
- **margin-bottom: 30px;**
- **margin-left: 25px;**
- **margin:auto;**
- **margin: 25px 50px 75px 100px;**

CSS Padding

- Padding is used to create space around an element's content, inside of any defined borders.
- CSS has properties for specifying the padding for each side of an element.

-
  ```
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
  ```
- `padding: 25px 50px 75px 100px;`
- `padding: 25px 50px;`

CSS Height, Width

- The CSS height and width properties are used to set the height and width of an element.
- The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.

-
  ```
  div {
    height: 200px;
    width: 50%;
    background-color: powderblue;
  }
  ```

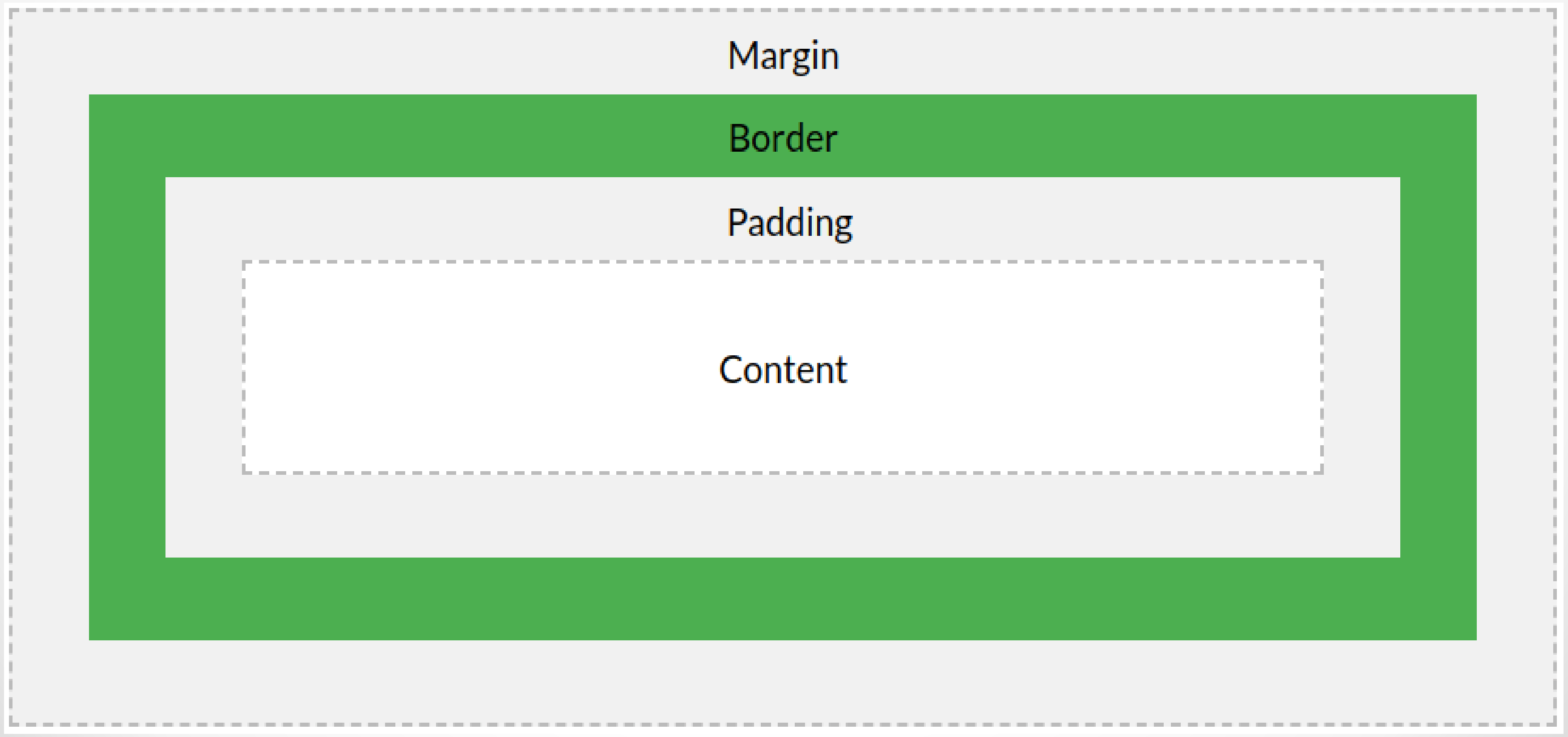
CSS Box Model

- All HTML elements can be considered as boxes.
- In CSS, the term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.

-
  ```
  div {
    width: 300px;
    border: 15px solid green;
    padding: 50px;
    margin: 20px;
  }
  ```

CSS Box Model

- **In CSS, the term "box model" is used when talking about design and layout.**



CSS Outline

- An outline is a line that is drawn around elements, OUTSIDE the borders, to make the element "stand out".
- CSS has the following outline properties:
- outline-style: solid;
- outline-color: red;
- outline-width: 5px;
  - thin (typically 1px)
  - medium (typically 3px)
  - thick (typically 5px)
- outline-offset: 15px;
- outline:5px solid yellow;

CSS Text

- Text color & Text Background
- **Text Alignment**
- In this chapter you will learn about the following properties:
- text-align
- text-align: center;
- text-align: left;
- text-align: right;
- text-align: justify;

Text Alignment

- The text-align property is used to set the horizontal alignment of a text.
- A text can be left or right aligned, centered, or justified.
- When the text-align property is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

CSS Text Decoration

- **Text Decoration**

- **In this chapter you will learn about the following properties:**

- **text-decoration-line**
- **text-decoration-color**
- **text-decoration-style**
- **text-decoration-thickness**
- **text-decoration**

Text decoration line

- Add a Decoration Line to Text
- The text-decoration-line property is used to add a decoration line to text.
- Tip: You can combine more than one value, like overline and underline to display lines both over and under a text.
- text-decoration-line: overline;
- text-decoration-line: line-through;
- text-decoration-line: underline;
- text-decoration-line: overline underline;

Text decoration color

- Specify a Color for the Decoration Line
- The text-decoration-color property is used to set the color of the decoration line.

- text-decoration-line: overline;
- text-decoration-line: line-through;
- text-decoration-line: underline;
- text-decoration-line: overline underline;

- text-decoration-color: red;

Text decoration style

- Specify a Style for the Decoration Line
- The text-decoration-style property is used to set the style of the decoration line.

- text-decoration-line: overline;
- text-decoration-line: line-through;
- text-decoration-line: underline;
- text-decoration-line: overline underline;

- text-decoration-style: solid;

Text decoration thikness

- Specify the Thickness for the Decoration Line
- The text-decoration-thickness property is used to set the thickness of the decoration line.

- text-decoration-line: overline;
- text-decoration-line: line-through;
- text-decoration-line: underline;
- text-decoration-line: overline underline;
- text-decoration-thickness: 5px;
- text-decoration-thickness: auto;

Text decoration

- The Shorthand Property
- The text-decoration property is a shorthand property for:
- text-decoration: underline red double 5px;

- All links in HTML are underlined by default. Sometimes you see that links are styled with no underline. The text-decoration: none; is used to remove the underline from links.
- text-decoration: none;

CSS Text Transformation

- The text-transform property is used to specify uppercase and lowercase letters in a text.
- It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

- text-transform: uppercase;
- text-transform: lowercase;
- text-transform: capitalize;

CSS Text Spacing

- **Letter Spacing**
- The letter-spacing property is used to specify the space between the characters in a text.
- The following example demonstrates how to increase or decrease the space between characters:
- letter-spacing: 5px;
- **Line Height**
- The line-height property is used to specify the space between lines:
- line-height: 0.8;

CSS Text Spacing

- **Word Spacing**
- The word-spacing property is used to specify the space between the words in a text.
- The following example demonstrates how to increase or decrease the space between words:
- word-spacing: 10px;
- **Text Indentation**
- The text-indent property is used to specify the indentation of the first line of a text:
- text-indent: 50px;

CSS Text-Box Shadow

- Text Shadow
- The text-shadow property adds shadow to text.
- In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px).
- text-shadow: 2px 2px;
- Next, add a color (red) to the shadow.
- text-shadow: 2px 2px red;
- Then, add a blur effect (5px) to the shadow.
- text-shadow: 2px 2px 5px red;
- text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue;
- Box-shadow: 2px 2px 5px green;

CSS Fonts

- Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.
- In CSS, we use the font-family property to specify the font of a text.
- If the font name is more than one word, it must be in quotation marks, like: "Times New Roman".

- Example:-
- font-family: "Times New Roman", Times, serif;

CSS Fonts

- **Font Style**
- The font-style property is mostly used to specify italic text.
- font-style: normal; - italic, oblique.
- **Font Weight**
- The font-weight property specifies the weight of a font:
- font-weight: bold;
- font-weight: lighter;
- font-weight: 900;

CSS Fonts

- **Font Size**
- The font-size property sets the size of the text.
- Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.
- Note: If you do not specify a font size, the default size for normal text, like paragraphs, is 16px (16px=1em).
- Example:-
- font-size: 40px;

CSS Google Fonts

- If you do not want to use any of the standard fonts in HTML, you can use Google Fonts.
- Google Fonts are free to use, and have more than 1000 fonts to choose from.
- How To Use Google Fonts
- Just add a special style sheet link in the <head> section and then refer to the font in the CSS.
- <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Audiowide">
- font-family: "Audiowide", sans-serif;

The float Property

- The float property is used for positioning and formatting content.
- The float property can have one of the following values:
- left - The element floats to the left of its container
- right - The element floats to the right of its container
- none - The element does not float (will be displayed just where it occurs in the text). This is default
- div {

```
  float: left;
  float: right;

   float: none;
}
```

CSS :hover Selector

- The :hover selector is used to select elements when you mouse over them.
- The :hover selector can be used on all elements, not only on links.

- Syntax:- :hover
- Example:-
- h1:hover{

```
  background-color: yellow;
}
```

CSS :hover for link

- Links can be styled with any CSS property (e.g. color, font-family, background, etc.).
- In addition, 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

CSS :hover for link

- **Styling Links**
- **Mouse over the links and watch them change layout.**
- **a:hover {color:#ffcc00;}**
- **a:hover {font-size:150%;}**
- **a:hover {background:#66ff66;}**
- **a:hover {font-family:monospace;}**
- **a:hover {text-decoration:underline;}**
- **border: 2px solid green;**
- **padding: 10px 20px;**

The display Property

- **The display property specifies if/how an element is displayed.**
- **Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.**
- **Block-level Elements**
- **A block-level element always starts on a new line and takes up the full width available**
- **Examples of block-level elements:**
- **<div>, <h1> - <h6>, <p>, etc.**

The display Property

- **Inline-block Elements**
- An inline-block element does not start on a new line.
- Examples of inline-block elements:
- <span>, <a>, <img>
- Every element has a default display value. However, you can override this.
- Changing an inline element to a block element, or vice versa.
- One common use for display: inline-block is to display list items horizontally instead of vertically.

The position Property

- The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).
- position: static;
- HTML elements are positioned static by default.
- Static positioned elements are not affected by the top, bottom, left, and right properties.
- An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page.

The position Property

- position: relative;
- Left: 20px;   right: 10px;
- Top: 20px;   bottom: 30px;
- An element with position: relative; is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

The position Property

- **position: fixed;**

- An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.
- A fixed element does not leave a gap in the page where it would normally have been located.

The position Property

- position: absolute;
- An element with position: absolute; is positioned relative to the nearest positioned founder (instead of positioned relative to the viewport, like fixed).
- However; if an absolute positioned element has no positioned founder, it uses the document body, and moves along with page scrolling.
- Note: Absolute positioned elements are removed from the normal flow, and can overlap elements.

The position Property

- **position: sticky;**

- **An element with position: sticky; is positioned based on the user's scroll position.**
- **A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).**

CSS Overflow

- The overflow property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.
- The overflow property has the following values:
- visible - Default. The overflow is not clipped. The content renders outside the element's box
- hidden - The overflow is clipped, and the rest of the content will be invisible
- scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content
- auto - Similar to scroll, but it adds scrollbars only when necessary

CSS Opacity / Transparency

- 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.
- The opacity property is often used together with the :hover selector to change the opacity on mouse-over.
- Transparency using RGBA
- Transparent Box
- opacity: 0.5;

CSS Navigation Bar

- **Vertical Navigation Bar**
- **Horizontal Navigation Bar**
- **CSS Dropdowns Navigation Bar**
- You can create navigation bar in
- Div tag as well as using
- ul tag

CSS visibility Property

- The visibility property specifies whether or not an element is visible.
- Tip: Hidden elements take up space on the page. Use the display property to both hide and remove an element from the document layout!
- visibility: visible;
- Default value. The element is visible
- visibility: hidden;
- Hidden – The element is hidden (but still takes up space)

CSS Transitions

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

- We will learn about the following properties:
- transition-delay
- transition-duration
- transition-property
- transition-timing-function
- transition

CSS Transitions

- **How to Use CSS Transitions?**

- To create a transition effect, you must specify two things:
  - the CSS property you want to add an effect to
  - the duration of the effect
- Note: If the duration part is not specified, the transition will have no effect, because the default value is 0.

Delay the Transition Effect

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

```
div {
        width: 100px;
        height: 100px;
        background: red;
        transition-delay: 5s;
    }
div:hover {
         width: 300px;
        }
```

transition-duration Property

- The transition-duration property specifies how many seconds (s) or milliseconds (ms) a transition effect takes to complete.
- Default value is 0s, meaning there will be no effect

```
div {
        width: 100px;
        height: 100px;
        background: red;
        transition-duration: 5s;
    }
div:hover {
        width: 300px;
        }
```

transition-property

- The transition-property property specifies the name of the CSS property the transition effect is for (the transition effect will start when the specified CSS property changes).
- Tip: A transition effect could typically occur when a user hover over an element.
- Note: Always specify the transition-duration property, otherwise the duration is 0, and the transition will have no effect.
- transition-property: width, height;
- transition-duration: 2s;

transition-timing-function

- The transition-timing-function property specifies the speed curve of the transition effect.
- This property allows a transition effect to change speed over its duration.
- 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

CSS Animations

- An animation lets an element gradually change from one style to another.
- @keyframes
- animation-name
- animation-duration
- animation-delay
- animation-iteration-count
- animation-direction
- animation-timing-function
- animation

CSS Animations

- animation-name:-
- You have to set the name for animation.
- animation-duration:-
- The animation-duration property defines how long an animation should take to complete. If the animation-duration property is not specified, no animation will occur, because the default value is 0s (0 seconds).
- animation-delay
- The animation-delay property specifies a delay for the start of an animation.

@keyframes

- @keyframes
- When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.
- To get an animation to work, you must bind the animation to an element.

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

@keyframes

- In the example above we have specified when the style will change by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).
- It is also possible to use percent. By using percent, you can add as many style changes as you like.
- @keyframes example {
  ```
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
  ```
- You can also provide position to change the direction.

CSS Animations

- animation-iteration-count
- The animation-iteration-count property specifies the number of times an animation should run.
- animation-direction
- The animation-direction property specifies whether an animation should be played forwards, backwards or in alternate cycles.
- reverse - The animation is played in (backwards)
- alternate - The animation is played forwards first, then backwards
- alternate-reverse - The animation is played backwards first, then forwards

CSS Transforms

- CSS transforms allow you to move, rotate, scale, and skew elements.
- CSS Transforms Methods
- translate()
- rotate()
- scaleX()
- scaleY()
- scale()
- skewX()
- skewY()
- skew()

CSS Transforms

- translate()
- The translate() method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).
- transform: translate(50px,100px);

- rotate()
- The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.
- transform: rotate(20deg);

CSS Transforms

- scale()
- The scale() method increases or decreases the size of an element (according to the parameters given for the width and height).
- transform: scale(2, 3);
- The scaleX() method increases or decreases the width of an element.
- transform: scaleX(2);
- The scaleY() method increases or decreases the height of an element.
- transform: scaleY(3);

CSS Transforms

- skew()
- The skew() method skews an element along the X and Y-axis by the given angles.
- transform: skew(20deg, 10deg);
- The skewX() method skews an element along the X-axis by the given angle.
- transform: skewX(20deg);
- The skewY() method skews an element along the Y-axis by the given angle.
- transform: skewY(20deg);

CSS Gradients

- CSS gradients let you display smooth transitions between two or more specified colors.
- CSS defines two types of gradients:
- Linear Gradients (goes down/up/left/right/diagonally)
- Radial Gradients (defined by their center)
- background-image: linear-gradient(*direction*, *color-stop1*, *color-stop2*, ...);
- background-image: radial-gradient(*shape size* at *position*, *start-color*, ..., *last-color*);

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.
- Direction - Top to Bottom (this is default)
- background-image: linear-gradient(red, yellow);
- Direction - Left to Right
- background-image: linear-gradient(to right, red , yellow);

CSS Linear Gradients

- Direction - Diagonal
- You can make a gradient diagonally by specifying both the horizontal and vertical starting positions.
- background-image: linear-gradient(to bottom right, red, yellow);
- Using Angles
- background-image: linear-gradient(180deg, red, yellow);
- Repeating a linear-gradient
- background-image: repeating-linear-gradient(red, yellow 10%, green 20%);

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.
- Radial Gradient - Evenly Spaced Color Stops (this is default)
- background-image: radial-gradient(red, yellow, green);
- Radial Gradient - Differently Spaced Color Stops
- background-image: radial-gradient(red 5%, yellow 15%, green 60%);

CSS Radial Gradients

- Set Shape
- The shape parameter defines the shape. It can take the value circle or ellipse. The default value is ellipse.
- background-image: radial-gradient(circle, red, yellow, green);
- The size parameter defines the size of the gradient. It can take four values:
- closest-side, farthest-side, closest-corner
- farthest-corner
- background-image: radial-gradient(closest-side at 60% 55%, red, yellow, black);

Form CSS

- If you only want to style a specific input type, you can use attribute selectors.
- input[type=text] - will only select text fields
- input[type=password] - will only select password fields
- input[type=number] - will only select number fields
- Many Other ways you can select the form tag.
- You can style form by using all CSS style properties.

CSS Flexbox

- CSS Flexbox Layout Module

- Block, for sections in a webpage
- Inline, for text
- Positioned, for explicit position of an element

- The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.

CSS Flex Container

- To start using the Flexbox model, you need to first define a flex container.
- <div class="flex-container"> (Parent)
-   <div>1</div>(Child)
-   <div>2</div>
- </div>
- The flex container becomes flexible by setting the display property to flex.
- Direct child elements(s) of the flexible container automatically becomes flexible items.

CSS Flex Container

- **The flex container properties are:**

- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

The flex-direction Property

- **The flex-direction property defines in which direction the container wants to stack the flex items.**
- **The column value stacks the flex items vertically (from top to bottom)- flex-direction: column;**
- **The column-reverse value stacks the flex items vertically (but from bottom to top)**
- **flex-direction: column-reverse;**
- **The row value stacks the flex items horizontally (from left to right)-flex-direction: row;**
- **The row-reverse value stacks the flex items horizontally (but from right to left)**
- **flex-direction: row-reverse;**

The flex-wrap Property

- **The flex-wrap property specifies whether the flex items should wrap or not.**
- The wrap value specifies that the flex items will wrap if necessary.
- flex-wrap: wrap;
- The nowrap value specifies that the flex items will not wrap (this is default).
- flex-wrap: nowrap;
- The wrap-reverse value specifies that the flexible items will wrap if necessary, in reverse order:
- flex-wrap: wrap-reverse;

The flex-flow Property

- **The flex-flow property is a shorthand property for setting both the flex-direction and flex-wrap properties.**
- **flex-flow: row wrap;**

The justify-content Property

- The justify-content property is used to align the flex items.
- The center value aligns the flex items at the center of the container.
- justify-content: center;
- The flex-start value aligns the flex items at the beginning of the container (this is default).
- justify-content: flex-start;
- The flex-end value aligns the flex items at the end of the container.
- justify-content: flex-end;

The justify-content Property

- The space-around value displays the flex items with space before, between, and after the lines.

- justify-content: space-around;

- The space-between value displays the flex items with space between the lines.

- justify-content: space-between;

The align-items Property

- The align-items property is used to align the flex items.
- The center value aligns the flex items in the middle of the container.
- align-items: center;
- The flex-start value aligns the flex items at the top of the container.
- align-items: flex-start;
- The flex-end value aligns the flex items at the bottom of the container.
- align-items: flex-end;

The align-content Property

- The align-content property is used to align the flex lines.
- The space-between value displays the flex lines with equal space between them.
- align-content: space-between;
- The space-around value displays the flex lines with space before, between, and after them.
- align-content: space-around;
- The stretch value stretches the flex lines to take up the remaining space (this is default).

The align-content Property

- align-content: center;
- The flex-start value displays the flex lines at the start of the container.
- align-content: flex-start;
- The flex-end value displays the flex lines at the end of the container.
- align-content: flex-end;

CSS Flex Items

- **The flex item properties are:**
- order
- flex-grow
- flex-shrink

The order Property

- The order property specifies the order of the flex items.
- The order value must be a number, default value is 0.
- The *order* property can change the order of the flex items:

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

The flex-grow Property

- The flex-grow property specifies how much a flex item will grow relative to the rest of the flex items.
- The value must be a number, default value is 0.
- `<div style="flex-grow: 8">3</div>`
- **The flex-shrink Property**
- The flex-shrink property specifies how much a flex item will shrink relative to the rest of the flex items.
- The value must be a number, default value is 1.
- `<div style="flex-shrink: 0">3</div>`