CSS notes

What is CSS ?

CSS stands for ***Cascading Style Sheets***. It is a style sheet language used to describe the presentation and formatting of HTML, XHTML, and XML documents. CSS allows web developers to separate the content and structure of a web page from its presentation, allowing for more flexibility and control over the appearance of the page. With CSS, developers can define styles for elements such as fonts, colors, margins, borders, and layout, and apply those styles to specific elements or groups of elements throughout the web page.

Importance :

- ***Separation of content and presentation:*** CSS allows you to separate the content and presentation of a web page. This means you can write clean HTML code that describes the structure of the content, and use CSS to define the visual style and layout of the page. This separation makes it easier to maintain and update a website.
- ***Consistency:*** By defining a consistent style for your website, you can create a cohesive look and feel that helps to build your brand identity. CSS also makes it easier to ensure consistency across multiple pages of a website.
- ***Responsiveness:*** With CSS, you can create responsive web pages that adjust to different screen sizes and devices. This is crucial for ensuring that your website is accessible to a wider audience, including those using mobile devices.
- ***Flexibility:*** CSS provides a lot of flexibility when it comes to styling web pages. You can control the appearance of individual elements, apply styles to groups of elements, and create reusable styles that can be applied across multiple pages.
- ***Accessibility**:* CSS makes it easier to create accessible websites by allowing you to control the presentation of content without compromising the underlying structure. This makes it easier for screen readers and other assistive technologies to navigate and interpret your website.

CSS naming conventions and organization :

CSS naming conventions and organization are essential for creating maintainable and scalable CSS code. They help to make the code more understandable, reduce code duplication, and improve collaboration between team members. Here are some of the commonly used CSS naming conventions and organization techniques:

- ***BEM (Block Element Modifier)*** - BEM is a naming convention that uses a class naming system to define blocks, elements, and modifiers in CSS. It helps to create reusable and maintainable code by providing a clear structure and hierarchy.
- ***SMACSS (Scalable and Modular Architecture for CSS)*** - SMACSS is a set of guidelines for organizing and managing CSS code. It emphasizes on the separation of concerns by dividing styles into base, layout, module, state, and theme.
- ***Atomic CSS*** - Atomic CSS is a naming convention that uses atomic classes to define specific CSS properties. It helps to create reusable and modular code by providing a limited set of classes that can be combined to create any style.
- ***OOCSS (Object-Oriented CSS)*** - OOCSS is a methodology that emphasizes on creating reusable and modular CSS code by separating structure and skin. It uses objects to represent reusable and independent components in CSS.
- ***ITCSS (Inverted Triangle CSS)*** - ITCSS is a methodology that organizes CSS code based on specificity and inheritance. It defines styles in a hierarchical order from global to local, making it easier to manage and maintain large CSS codebases.
- ***SMIT (Single-Responsibility Media-Independent CSS)*** - SMIT is a naming convention that separates CSS code based on device-independent properties and media-specific properties. It helps to create flexible and scalable CSS code that can adapt to different devices and media types.

Basic syntax and structure :

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

- Selector: It is the HTML element, class, ID, or attribute to which the CSS styles are applied. It specifies the target element(s) for styling.
- Property: It is the attribute of the element that you want to style. It represents the aspect of the element that you want to change, such as color, font-size, margin, padding, etc.
- Value: It is the assigned value for the property. It specifies how you want to change the appearance or behavior of the element.

Example :

```
p {
 font-size: 16px;
}
```

- Selector: "p" is the selector that targets all the paragraphs in the HTML document.
- Property: "font-size" is the property that specifies the size of the font.
- Value: "16px" is the value assigned to the "font-size" property, indicating the font size in pixels.

Types Of CSS :

*1. **Inline CSS:*** Inline CSS is applied directly to an HTML element using the style attribute. The syntax is as follows:

```
<element style="property: value;">
```

*2. **Internal CSS:*** Internal CSS is defined within the head section of an HTML document using the *<style>* tag. The syntax is as follows:

```
<head>
 <style>
  selector {
   property: value;
  }
 </style>
</head>
```

*3. **External CSS**:* External CSS is defined in a separate CSS file and linked to an HTML document using the *<link>* tag. The syntax is as follows:

```
<head>
 <link rel="stylesheet" type="text/css" href="path/to/file.css">
</head>
```

Selectors

CSS selectors are used to target HTML elements for styling. There are several types of CSS selectors:

*1. **Tag selectors:*** These selectors target elements based on their tag name. For example, to target all *p* elements, you would use the *p* selector:

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

*2. **Class selectors:*** These selectors target elements based on their class attribute. To create a class selector, you prefix the class name with a period (.). For example, to target all elements with the class "my-class", you would use the *.my-class* selector:

```
.my-class {
  color: blue;
}
```

*3. **ID selectors:*** These selectors target elements based on their ID attribute. To create an ID selector, you prefix the ID name with a hash (#). For example, to target the element with the ID "my-id", you would use the *#my-id* selector:

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

*4. **Attribute selectors:*** These selectors target elements based on their attributes. For example, to target all *input* elements with the *type* attribute set to "text", you would use the *[attribute=value]* selector:

```
input[type="text"] {
  border: 1px solid black;
}
```

*5. **Descendant selectors:*** These selectors target elements that are descendants of other elements. For example, to target all *a* elements that are descendants of *li* elements, you would use the *li a* selector:

```
cssCopy codeli a {
  text-decoration: none;

}
```

Note : *Classes* are often used to apply styles to *multiple* elements that share a common set of characteristics, while *IDs* are typically used to apply styles to a *single*, unique element on a page.

Box Model :

The CSS box model is a fundamental concept in CSS that defines the layout and design of web pages. It consists of four parts: margin, border, padding, and content. Here is a brief explanation of each part:

1. ***Margin****:* The margin is the space outside of an element's border. It can be set using the *margin* property and can have different values for each side of the element (top, right, bottom, left).

2. ***Border:*** The border is a line that surrounds an element's padding and content. It can be styled and customized using the *border* property, which can have different values for width, style, and color.

3. ***Padding:*** The padding is the space between an element's content and its border. It can be set using the *padding* property and can have different values for each side of the element (top, right, bottom, left).

4. ***Content:*** The content is the actual content of the element, such as text, images, or other HTML elements. Its size can be set using the *width* and *height* properties.

Color Properties

Color properties refers to the visual aspects of an element such as the background color, foreground color, and transparency. The three main color properties in CSS are:

1. ***background-color:*** This property sets the background color of an element. It can be set to any valid CSS color value, such as a hexadecimal color code, RGB value, or a color keyword.

```
body {

  background-color: #f1f1f1;
}
```

2. ***color:*** This property sets the foreground color of an element, which affects the color of the text inside the element. It can also be set to any valid CSS color value.

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

3. ***opacity:*** This property sets the opacity of an element, which determines the degree of transparency of the element. It is a value between 0 and 1, with 0 being completely transparent and 1 being completely opaque.

```
div {
  background-color: blue;
  opacity: 0.5;
}
```

<u>Text Properties :</u>

Text properties can be used to style text content within HTML elements. Some commonly used text properties are:

1. ***font-size:*** Sets the size of the font. Syntax: *font-size: <size>*

2. ***font-family:*** Sets the font family for the text. Syntax: *font-family: <family>*

3. ***font-weight:*** Sets the weight (thickness) of the font. Syntax: *font-weight: <weight>*

4. ***text-align:*** Sets the horizontal alignment of the text. Syntax: *text-align: <alignment>*

5. ***text-transform:*** Sets the capitalization of the text. Syntax: *text-transform: <transform>*

6. ***Color :*** Sets the color of the text. Syntax : *color : <color_code>*

Border Properties :

The *border* property is used to specify the style, width, and color of an element's border. It can be applied to any HTML element and has the following sub-properties:

- ***border-style:*** sets the style of the border (e.g. solid, dotted, dashed, etc.)
- ***border-width:*** sets the width of the border in pixels, ems, or other length units
- ***border-color:*** sets the color of the border
- ***border-top, border-right, border-bottom, border-left****:* allows you to set a specific border for each side of an element
- ***border-radius:*** sets the roundedness of the border corners
- ***border-image:*** allows you to use an image for the border
- ***box-shadow:*** creates a shadow effect around the border of an element

---

Margin And Padding :

Margin and padding are properties that allows you to add space around or within an element.

Margin is the space outside of the border, while padding is the space inside the border.

Background Properties

The background properties allows you to control the appearance of the background of an element, including the background color, image, and other related properties. Some of the important background properties are:

- ***background-color:*** sets the background color of an element.
- ***background-image:*** sets the background image of an element.
- ***background-repeat:*** controls how the background image is repeated.
- ***background-position:*** controls the position of the background image.
- ***background-size:*** sets the size of the background image.

1. Styling Tables:

Changing the font, color, and size of the table text

Adding borders and background color to the table

Changing the font, color, and size of the table headers

Styling the table rows and columns differently

2. Styling Forms :

Changing the font, color, and size of the form elements (e.g., input fields, buttons, etc.)

Adding borders and background color to the form elements

Styling the form labels

3. Styling Lists :

Changing the font, color, and size of the list items

Adding background color or borders to the list items

Changing the style of the bullet points or numbers