Web Development

HTML

Web Development

Web development involves designing, building, and maintaining websites using a variety of programming languages, frameworks, and tools. HTML is the foundation of all websites, and CSS and JavaScript are used to create dynamic and interactive websites. Web developers use meta tags, Emmet, and a variety of HTML tags and attributes to create functional and visually appealing websites. To stay up to date with the latest trends and techniques, web developers need to be constantly learning and experimenting.

Web development is the process of creating and maintaining websites. It involves designing, building, and maintaining websites using a variety of programming languages, frameworks, and tools. Web development is a broad term that encompasses many different aspects of building and maintaining websites, including web design, front-end development, back-end development, and web content management.

An analogy to understand web development is to think of a website as a house. Just as a house needs a blueprint, foundation, walls, and a roof, a website needs a plan, structure, content, and design. Web development is the process of building and maintaining the house, making sure everything is functional and up to code.

HTML is the foundation of all websites, and it is essential to know the basics of HTML to create any website. HTML stands for Hypertext Markup Language, which is the standard markup language used to create web pages. It provides the structure and content of a web page and is the building block of every website.

Here are some of the most commonly used HTML codes in web development:

- `<html>`: This tag indicates the beginning of an HTML document.
- `<head>`: This tag contains information about the document, such as the title and the links to other resources.
- `<title>`: This tag specifies the title of the document, which appears in the browser's title bar.
- `<body>`: This tag contains the content of the document, such as text, images, and videos.
- `<h1>` to `<h6>`: These tags create headings of different sizes, with `<h1>` being the largest and `<h6>` being the smallest.
- `<p>`: This tag creates a paragraph of text.
- `<a>`: This tag creates a hyperlink to another web page or resource.
- `<img>`: This tag inserts an image into the document.
- `<ul>` and `<li>`: These tags create an unordered list, with `<ul>` indicating the beginning of the list and `<li>` indicating each item in the list.
- `<ol>` and `<li>`: These tags create an ordered list, with `<ol>` indicating the beginning of the list and `<li>` indicating each item in the list.

In addition to HTML, web developers use other languages, such as CSS and JavaScript, to create dynamic and interactive websites. CSS stands for Cascading Style Sheets, which is used to style HTML elements and give a website its visual appearance. JavaScript is used to create interactive features, such as dropdown menus, image sliders, and animations.

Web development is a constantly evolving field, and new tools and technologies are emerging all the time. To stay up to date with the latest trends and techniques, web developers need to be constantly learning and experimenting. There are many online resources available for learning web development, including tutorials, forums, and online courses.

By using HTML, CSS, JavaScript, and other tools and technologies, web developers can create dynamic, engaging, and visually appealing websites that are both functional and user-friendly.

To create a beautiful website, web developers can use a variety of CSS and JavaScript codes. Here are some commonly used ones:

CSS:

- `color`: This property sets the color of an HTML element's text content.
- `font-family`: This property sets the font of an HTML element's text content.
- `font-size`: This property sets the size of an HTML element's text content.
- `background-color`: This property sets the background color of an HTML element.
- `background-image`: This property sets the background image of an HTML element.
- `border`: This property sets the border of an HTML element.
- `border-radius`: This property sets the rounded corners of an HTML element.
- `box-shadow`: This property adds a shadow effect to an HTML element.

JavaScript:

- `addEventListener`: This method adds an event listener to an HTML element.
- `querySelector`: This method returns the first element that matches a specified CSS selector.
- `classList`: This property returns the class name(s) of an HTML element.
- `setAttribute`: This method sets the value of an attribute for an HTML element.
- `setTimeout`: This method calls a function after a specified amount of time has elapsed.
- `setInterval`: This method calls a function repeatedly at specified intervals.

Note that there are many more CSS and JavaScript codes that web developers can use to create beautiful websites, and the specific codes used will depend on the design and functionality of the website.

Emmet is a plugin for text editors, including VS Code, that allows web developers to write HTML and CSS code more quickly and efficiently. It uses a shorthand syntax that expands into full HTML and CSS code, allowing developers to write code faster and with fewer errors.

For example, instead of typing out the full HTML for a div element, a developer can simply type "div" and press the Tab key, and Emmet will automatically expand it into the full HTML code for a div element. Emmet also allows for the creation of nested HTML and CSS code, making it easier to organize and structure code.

Overall, Emmet is a powerful tool that can help web developers save time and increase productivity, making it a popular choice for many coder.

Meta tags are snippets of HTML code that provide information about a web page. They are located in the head section of an HTML document and are not visible to users. There are several different types of meta tags, including:

- `title`: This tag specifies the title of the web page, which appears in the browser's title bar and in search engine results.
- `description`: This tag provides a brief description of the web page, which appears in search engine results.
- `keywords`: This tag specifies keywords related to the content of the web page, which can help search engines understand the page's content.
- `author`: This tag specifies the author of the web page.
- `viewport`: This tag specifies the viewport settings for the web page, which can affect how the page is displayed on mobile devices.
- `charset`: This tag specifies the character encoding used in the HTML document.
- `robots`: This tag specifies instructions for search engine robots, such as whether or not to index the page or follow links on the page.
- `canonical`: This tag specifies the preferred version of a web page when there are multiple versions of the same page.
- `og:title`: This tag specifies the title of the web page for use in social media sharing.
- `og:description`: This tag provides a brief description of the web page for use in social media sharing.
- `og:image`: This tag specifies the image to be used when sharing the web page on social media.
- `og:url`: This tag specifies the URL of the web page for use in social media sharing.

Meta tags can help improve a web page's visibility in search engine results and can provide important information to users and search engines alike. It is important to use meta tags correctly and to update them regularly to ensure that they accurately reflect the content of the web page.

▼ This is how you include external CSS file in HTML code.

```
<link rel="stylesheet" href="FILE NAME">
```

▼ This is how you include external JAVASCRIPT file in HTML code.

```
<script src="FILE NAME"></script>
```

▼ When the audience type these keywords then your website will popup.

```
<meta name="keywords" content="python, html, java">
```

▼ When you don’t want your websites to popup in the search engine.

```
<meta name="Robots" content="NOINDEX, NOFOLLOW">
```

HTML tags and attributes are the building blocks of web development. They are used to create the structure and functionality of websites. HTML tags define different elements such as headings, paragraphs, links, images, lists, tables, forms, and more. Attributes provide additional information about the elements, such as their properties and behavior.

Some of the most commonly used HTML tags include:

- `<html>`: This tag indicates the beginning of an HTML document.
- `<head>`: This tag contains information about the document, such as the title and the links to other resources.
- `<title>`: This tag specifies the title of the document, which appears in the browser's title bar.
- `<body>`: This tag contains the content of the document, such as text, images, and videos.
- `<h1>` to `<h6>`: These tags create headings of different sizes, with `<h1>` being the largest and `<h6>` being the smallest.
- `<p>`: This tag creates a paragraph of text.
- `<a>`: This tag creates a hyperlink to another web page or resource.
- `<img>`: This tag inserts an image into the document.
- `<ul>` and `<li>`: These tags create an unordered list, with `<ul>` indicating the beginning of the list and `<li>` indicating each item in the list.
- `<ol>` and `<li>`: These tags create an ordered list, with `<ol>` indicating the beginning of the list and `<li>` indicating each item in the list.
- `<form>`: This tag creates a form for user input.
- `<input>`: This tag creates an input field within a form.
- `<select>` and `<option>`: These tags create a dropdown menu within a form.
- `<textarea>`: This tag creates a multi-line text input field within a form.
- `<button>`: This tag creates a button within a form.

- `<label>`: This tag creates a label for an input field within a form.
- `<div>`: This tag creates a container for other HTML elements.
- `<span>`: This tag creates an inline container for other HTML elements.
- `<table>`, `<tr>`, `<td>`, `<th>`: These tags create tables and their respective rows and cells.

HTML attributes provide additional information about the elements, such as their properties and behavior. Some of the most commonly used attributes include:

- `class`: This attribute assigns a class to an HTML element, which can be used for styling with CSS.
- `id`: This attribute assigns an ID to an HTML element, which can be used for identifying the element with JavaScript and for linking to specific parts of a web page.
- `href`: This attribute specifies the URL of a hyperlink.
- `src`: This attribute specifies the URL of an image or other media file.
- `alt`: This attribute provides alternative text for an image, which is displayed if the image cannot be loaded.
- `style`: This attribute applies inline styles to an HTML element.
- `target`: This attribute specifies whether the hyperlink should open in a new window or in the same window.
- `name`: This attribute specifies the name of an input field within a form.
- `value`: This attribute specifies the value of an input field within a form.
- `type`: This attribute specifies the type of an input field within a form, such as text, password, checkbox, or radio button.
- `rows` and `cols`: These attributes specify the number of rows and columns in a text area input field.
- `colspan` and `rowspan`: These attributes specify the number of columns or rows spanned by a table cell.

In addition to these commonly used HTML tags and attributes, there are many more that can be used in web development. The specific tags and attributes used will depend on the design and functionality of the website. It is important to understand the purpose and usage of each tag and attribute to create effective and efficient web pages.

Overall, HTML tags and attributes are essential components of web development. They allow developers to create structured, functional, and visually appealing websites that are both user-friendly and easy to navigate.

Headings and paragraphs

- Headings and paragraphs are two fundamental components of HTML (Hypertext Markup Language), which is used to structure the content of web pages.
- Headings are used to define the hierarchy of the content on a page, with the main heading being the most important and typically the largest, and subsequent headings decreasing in importance and size. Headings are defined using the `<h1>` to `<h6>` tags in HTML. For example, the main heading of a page might be enclosed in an `<h1>` tag, while subheadings might be enclosed in `<h2>` or `<h3>` tags.
- Paragraphs, on the other hand, are used to group together related sentences or pieces of text. They are defined using the `<p>` tag in HTML, which indicates to the browser that the enclosed text should be rendered as a separate paragraph.
- Using headings and paragraphs in web development helps to structure content and make it more readable and accessible to users. Additionally, search engines use heading tags to better understand the structure and content of a web page, which can improve search engine optimization (SEO) and help the page rank higher in search results.
- To add bold and italic formatting to text in HTML, you can use the `<b>`, `<strong>`, `<i>`, and `<em>` tags. These tags are used to provide emphasis to text and can be used to change the appearance of text in a web page.
- To add a new line while writing a paragraph we use `<br>` tag, it is a self closing tag.
- To insert a horizontal rule under a line we use `<hr>` tag.
- An anchor tag is an HTML element that is used to create hyperlinks within a web page. It is represented by the `<a>` tag in HTML and can be used to link to external web pages, other pages within the same website, or to specific locations within a web page.

```
<a href="https://openai.com/">Visit OpenAI</a>
```

- When you add the attribute `target=”_blank"` then by this attribute the link to other website will open on a new tab.
- To insert images we need to type `img`

```
<img src="image path" alt="text shown if we have the error loading image">
```

- Lists and Tables

In HTML, an ordered list and an unordered list are two types of lists that can be used to present information in a structured and organized manner.

An ordered list is a list where each item is numbered, usually with Arabic numerals (1, 2, 3, etc.). The HTML code for an ordered list starts with the `<ol>` tag and each item is marked with the `<li>` tag. For example:

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
```

This will produce an ordered list that looks like this:

1. Item 1
2. Item 2
3. Item 3

An unordered list, on the other hand, is a list where each item is bulleted or marked with a symbol, such as a dot or a dash. The HTML code for an unordered list starts with the `<ul>` tag and each item is marked with the `<li>` tag. For example:

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

This will produce an unordered list that looks like this:

- Item 1
- Item 2
- Item 3

In summary, an ordered list is used when the items have a specific sequence or order, while an unordered list is used when the items do not have a specific sequence or order.

In HTML, a table is a way to display data in rows and columns. A table is created using the `<table>` tag, which contains one or more `<tr>` tags (table rows), and each row can contain one or more `<td>`

tags (table data/cells). The `<th>`
tag (table header) is used to define header cells for a table.

```
<table>
        <thead>
            <tr>
                <th>Name</th>
                <th>Employee ID</th>
                <th>Employee Role</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Harry</td>
                <td>234567</td>
                <td>programmer</td>
             </tr>
             <tr>
                <td>Mukul</td>
                <td>4369876</td>
                <td>programmer</td>
             </tr>
             <tr>
                <td>Kunal</td>
                <td>9876345</td>
                <td>programmer</td>
             </tr>

        </tbody>
    </table>
```

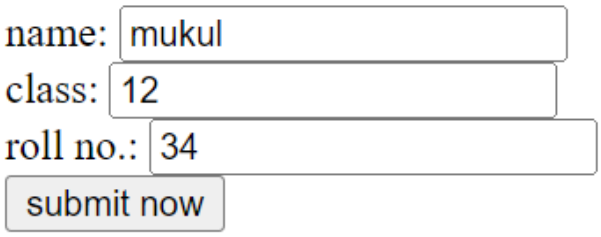
Forms

In HTML, there are several types of forms that you can use to create user input fields, buttons, and other interactive elements. Here are some of the most commonly used form types:

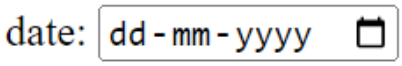
1. Text input: Allows users to enter single or multi-line text. You can use this for fields like name, email, and comments.
2. Password input: Hides the user's input with asterisks or dots to keep passwords secure.
3. Checkbox: Allows users to select one or more options from a list of choices.
4. Radio buttons: Similar to checkboxes, but allows users to select only one option from a list of choices.
5. Select dropdown: Allows users to select one option from a list of predefined options.
6. Text area: Allows users to enter multi-line text, typically used for longer responses like comments or feedback.

7. Button: Allows users to submit the form or trigger an action, like a search or reset button.
8. File upload: Allows users to upload files, such as images, documents, or videos.
9. Hidden input: Hides the input field from the user, typically used to pass information between pages or to store data on the server.

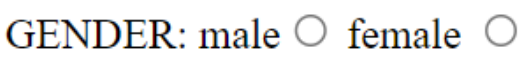
```
<form action="backend.php">
    <div> name: <input type="text" name="type your name"></div> ox
    <div> class: <input type="text" name="enter your class"></div>
    <div> roll no.: <input type="text" name="enter your roll no."></div>
    <div> <input type="submit" value="submit now"></div>

    </form>
```




```
<div> date: <input type="date" name="date" ></div>o
```




```
<div> GENDER: male<input type="radio" name="mygender" > female <input type="radio" name="mygender"></div>
```



- You can use `<lable>` tag if you want the textbox to be selected while you type on the text of the particular box.

```
<label for="class">class</label>

    <div><input type="text" name="enter your class" id="class"></div>
```

What are inline and block elements? Give examples.

In HTML and CSS, elements can be classified as either inline or block-level elements based on how they are rendered on a web page.

Block-level elements are those which take up the full width of their parent container and create a new line after the element. Examples of block-level elements include `div`, `h1`, `p`, `ul`, `ol`, and `form`.

Inline elements, on the other hand, do not start on a new line and only take up as much width as necessary. Examples of inline elements include `a`, `span`, `img`, `strong`, `em`, `label`, and `input`.

It's important to note that some elements can be both inline and block-level depending on the context in which they are used. For example, the `img` tag is typically an inline element, but if it is given a `display: block` style, it will behave like a block-level element. Similarly, the `a` tag can be styled to behave like a block-level element by adding `display: block` to its style.

IDs and Classes

- Use of Emmet : `.` is for class and `#` is for id.

` ` tag is used if you want spaces between the letters of the same sentence. You will have to type ` ` the times you want the spaces.

There are many HTML entities, but here are some of the most commonly used ones:

- `&lt;` represents the less-than symbol `<`.

- `>` represents the greater-than symbol `>`.
- `&` represents the ampersand `&`.
- `&quot;` represents the double quotation mark `"`.
- `&apos;` represents the single quotation mark `'` (although this entity is technically not part of the HTML specification and should be avoided).
- ` ` represents a non-breaking space.
- `&copy;` represents the copyright symbol ©.
- `&reg;` represents the registered trademark symbol ®.
- `&trade;` represents the trademark symbol ™.
- `&mdash;` represents an em dash —.
- `&ndash;` represents an en dash –.
- `&ldquo;` represents a left double quotation mark “.
- `&rdquo;` represents a right double quotation mark ”.
- `&lsquo;` represents a left single quotation mark ‘.
- `&rsquo;` represents a right single quotation mark ’.
- `&euro;` represents the euro currency symbol €.
- `&pound;` represents the pound currency symbol £.
- `&yen;` represents the yen currency symbol ¥.
- `&cent;` represents the cent currency symbol ¢.

What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.

Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

Semantic Elements in HTML

Many web sites contain HTML code like: to indicate navigation, header, and footer.

In HTML there are some semantic elements that can be used to define different parts of a web page:

<article>

<aside>

<details>

<figcaption>

<figure>

<footer>

<header>

<main>

<mark>

<nav>

<section>

<summary>

<time>

CASCADING STYLE SHEETS

- CSS helps style to raw HTML.
- CSS is used to make our websites responsive.
- CSS is used to give style to our websites.

Three ways to add CSS to the markup are :

1. Inline CSS: CSS is added to the elements directly using style attribute.
2. Internal CSS: CSS is kept inside the head tags in <style> tags.
3. External CSS: CSS is kept separately inside a .CSS style sheet.

Selectors

- A CSS selector is used to select an HT,L elements for styling

```
body{
     color:red;
     background:pink;
    }
```

▼ Element selectors

element selectors are used to target and style specific HTML elements on a web page. An element selector is written as the name of the HTML element, such as "p" for paragraph, "h1" for heading 1, "div" for a division or section of the web page

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

▼ ID selectors

An ID selector is written using the "#" symbol followed by the ID name of the element you want to target. ID names must be unique on a web page, and they are often used to identify specific elements that need to be styled differently from others.

For example, to style an HTML element with the ID name "header", you would use the following CSS code:

```
#header {
  background-color: blue;
  color: white;
}
```

▼ Class Selectors

A class selector is written using the "." symbol followed by the class name of the element you want to target. Class names can be used multiple times on a web page, and they are often used to identify groups of elements that need to be styled similarly.

For example, to style all HTML elements with the class name "button", you would use the following CSS code:

```
.button {
  background-color: green;
  color: white;
  padding: 10px;
}
```

This code would select all HTML elements with the class name "button" and set their background color to green, text color to white, and add 10px padding.

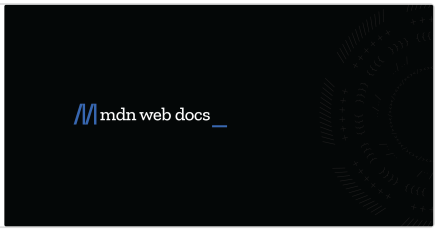
Both ID and class selectors are powerful tools for styling web pages and offer a lot of flexibility in how you can target and style specific elements.

> visit this website for reference “great CSS reference’

CSS reference - CSS: Cascading Style Sheets | MDN

Use this CSS reference to browse an alphabetical index of all of the standard CSS properties, pseudo-classes, pseudo-elements, data types, functional notations and at-rules. You can also browse key CSS concepts

https://developer.mozilla.org/en-US/docs/Web/CSS/Reference



Colors in CSS

1. Hexadecimal values: A six-digit code that represents the amount of red, green, and blue (RGB) in a color. For example, #FF0000 represents pure red.
2. RGB values: A set of three numbers that represent the amount of red, green, and blue in a color. Each value can range from 0 to 255. For example, rgb(255, 0, 0) represents pure red.
3. RGBA values: Similar to RGB values, but with an additional fourth value that represents the opacity of the color. The opacity value can range from 0 (completely transparent) to 1 (completely opaque). For example, rgba(255, 0, 0, 0.5) represents semi-transparent red.
4. HSL values: A color model that defines colors based on their hue, saturation, and lightness. Hue is represented by a degree value from 0 to 360, saturation is represented as a percentage from 0% (gray) to 100% (fully saturated), and lightness is also represented as a percentage from 0% (black) to 100% (white). For example, hsl(0, 100%, 50%) represents pure red.
5. HSLA values: Similar to HSL values, but with an additional fourth value that represents the opacity of the color. The opacity value can range from 0 (completely transparent) to 1 (completely opaque). For example, hsla(0, 100%, 50%, 0.5) represents semi-transparent red in the HSL color model.

Background property

▼ Background color property

The `background-color` property in CSS is used to set the background color of an element. It specifies the color that will be used as the background for the content area of the element.

The `background-color` property can be applied to any HTML element, such as `<body>`, `<div>`, `<p>`, etc. The value of this property can be specified using a color name, a hexadecimal color code, an RGB or RGBA color value, or a HSL or HSLA color value.

```
background-color: brown;
background-color: #74992e;
```

▼ Background image property

The `background-image` CSS property sets one or more background images on an element.

```
background-image: url();
```

▼ Background repeat property

The `background-repeat` property sets how background images are repeated. A background image can be repeated along the horizontal and vertical axes, or not repeated at all.

▼ Background size property

The `background-size` CSS property sets the size of the element's background image. The image can be left to its natural size, stretched, or constrained to fit the available space.

▼ Background attachment

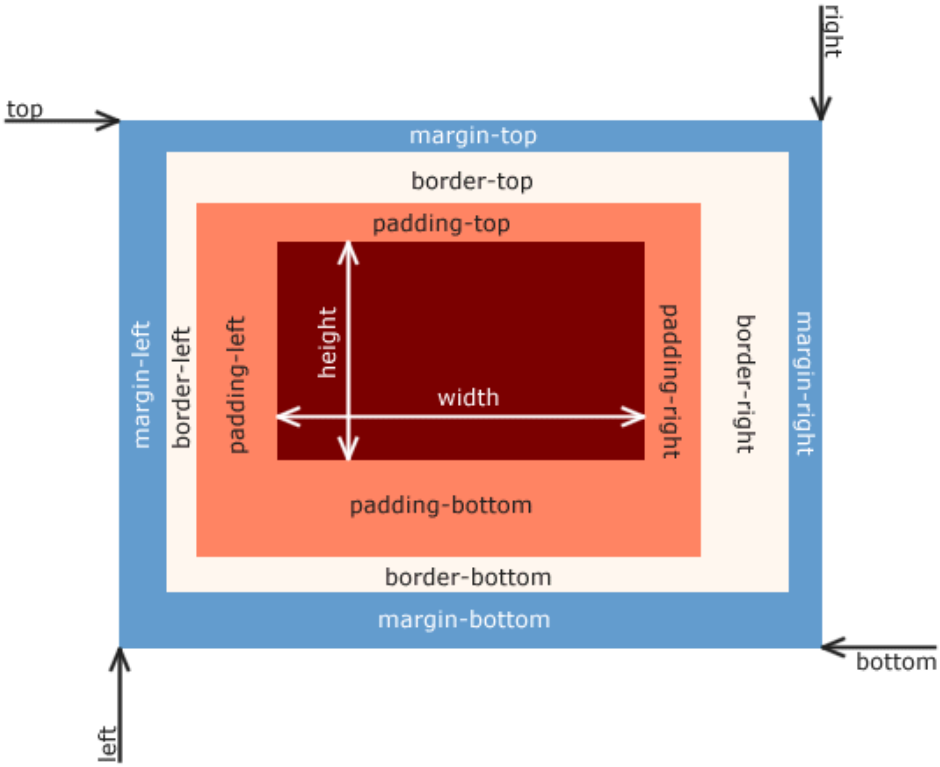
The `background-attachment` CSS property sets whether a background image's position is fixed within the viewport, or scrolls with its containing block.

- FOR MORE INFO VISIT

https://developer.mozilla.org/en-US/docs/Web/CSS/background-attachment

CSS BOX MODEL

The CSS box model looks at all the HTML elements as boxes.



Margin collapse

Margin collapse is a phenomenon that occurs when the vertical margins of adjacent elements collapse into each other, resulting in a single margin that equals the larger of the two original margins.

Margin collapse can happen in various situations, but it's most common between adjacent block-level elements, such as paragraphs, headings, and divs. When two such elements are adjacent and have margins that touch each other, the margins collapse into a single margin.

The rules of margin collapse are as follows:

- If the top margin of an element and the bottom margin of its first child collapse , the resulting margin is the larger of the two margins.
- If the bottom margin of an element and the top margin of its last child collapse, the resulting margin is the larger of the two margins.
- If two adjacent elements have margins that touch each other, the resulting margin is the larger of the two margins.

The display property

The CSS display property is used to determine weather an element is treated as a block/inline element and the layout used for flexbox, grid etc.

- Display Inline : Takes only space required by the element. No line break before and after.
- Display Block : Takes full space available in width and leaves a newline before and after the element.
- Display Inline-Block : Similar to inline but setting height, width, margin and padding is allowed, elements can sit next to each other.
- Display none vs Visibility hidden : With display none the element is removed from the document flow. Its space is not blocked. With Visibility hidden the element is hidden but its space is reserved.

Text align property

`text-align` property is used to specify the horizontal alignment of text within its containing element. It can be used on any block-level element, such as `div`, `p`, `h1`, etc.

The `text-align` property accepts several values:

- `left`: Aligns the text to the left of the container.
- `center`: Centers the text horizontally within the container.
- `right`: Aligns the text to the right of the container.
- `justify`: Stretches the lines of text so that they align with both the left and right edges of the container. This is commonly used for paragraphs of text.

CSS FLEXBOX

The `float` property in CSS is used to specify whether an element should float to the left or right of its container. When an element is floated, it is taken out of the normal flow of the document and positioned alongside the edge of its container, allowing content to flow around it. The syntax for the `float` property is:

```
float: left | right | none | inherit;
```

The `clear` property in CSS is used to specify which sides of an element should be cleared of floating elements. When an element is cleared, it is positioned below any preceding floated elements. The syntax for the `clear` property is:

```
clear: left | right | both | none | inherit;
```

- `left`: The element is moved below any preceding left-floated elements.
- `right`: The element is moved below any preceding right-floated elements.
- `both`: The element is moved below any preceding left-floated or right-floated elements.
- `none`: The element is not moved below any preceding floated elements.
- `inherit`: The value of the property is inherited from the parent element.

Flex-Direction Property

Defines the direction forwards which items are laid. Can be row, row-reverse, column, column-reverse.

Flex Properties for parent (Flex container)

1. Flex-wrap : can be wrap, no wrap, wrap-reverse. Wrap items as needed with this property.
2. Justify-content : Defines alignment along main axis.
3. align-items : Define alignment along cross axis.
4. Align-content : Aligns a flex container’s lines when there is extra space in the cross axis.

Flex Properties for children (Flex items)

1. Order : Controls the order in which the item appear in the flex container.
2. align-self : Allows default alignment to be overridden for the individual flex items.
3. Flex-grow : defines the ability for a flex item to grow.
4. flex-shrink : Specifies how much a flex item will shrink relative to the rest of the flex items.

CSS Grid and Media Queries

A CSS grid can be initialized using

```
container{
 display: grid;
        }
```

All direct children automatically becomes grid items.

The Grid-column-gap property

Used to adjust the space between the columns of a CSS grid.

The grid-row-gap property

Used to adjust the space between the rows of a CSS grid.

The grid-gap property

Shorthand property for grid-row-gap & grid-column-gap

```
container{
          display: gird;
          gird-gap: 40px(row), 100px(column)
          }
```

NOTE: For a single value of grid-gap, both row and column gaps can be set in one value.

Following are the properties for grid container:

1. The grid-template-column property can be used to specify the width columns.

```
.container{
       Display: grid;
       Grid-template-columns: 80px 120px auto;
          }
```

2. The grid-template-rows property can be used to specify the height of each row.

```
.container{
       display: grid;
       grid-template-rows: 70px 150px;
          }
```

3. Justify content property is used to align the whole grid inside the container.
4. The align content property is used to vertically align the whole grid inside the container.

Following are the properties for grid item:

1. The grid-column property defines how many columns an item will span.

```
.grid-item{
        grid-column: 1/5;
           }
```

2. The grid-row property defines how many rows an item will span.
3. We can make an item to start on column 1 and span 3 columns like this:

```
.item{
    grid-column: 1/span 3;
     }
```

CSS media Queries

Used to apply CSS only when a certain condition is true.

SYNTAX

```
@media only screen and (ma-width: 800px) {
          body{
              background: red;
              }
            }
```

Transformation, Transitions & Animations

Transformations are used to relate, move, skew or scale elements. They are used to create a 3-D effects.

The transform property

Used to apply a 2-D or 3-D transformation to an element.

The transform-origin property

Allows to change the position of transformed elements

2D transform ~ Can change x and y axis

3D transform ~ Can change z axis as well

CSS 2D transform methods

You can use the following 2D transform in CSS:

1. translate()
2. rotate()
3. scale x()
4. scale y()
5. skew ()
6. matrix ()
7. scale ()

CSS 3D transform method

1. rotate X()
2. rotate Y()
3. rotate Z()

CSS transitions

Used to change the property values smoothly, over a given duration.

The transition property

The transition property is used to add transition in CSS:

Following are the properties used for CSS transition.

1. Transition-property = The property you want to transition.
2. transition-duration = Time for which you want transition to apply.

3. transition-timing-function = How you want the property to transition.
4. transition-delay = Specifies the delay for the transition.

All these properties can be set using a single shorthand property.

```
transition: width 35 ease-in 25;
```

Transitioning multiple properties

We can transition multiple properties as follows:

Transition : Opacity 1s ease-out 1s, transform 2s ease-in;

CSS animations

Used to animate CSS properties with more control. We can use @keyframes rule to change the animations from a given style to a new style.

```
@keyframes mukul{
           from{ width: 20px;}
             to{ width: 31px;}
   }
```

Properties to add animations

Following are the properties used to set animations in CSS:

1. animation-name = name of the animation
2. animation-duration = how long does the animation run?
3. animation-timing-function = Determines speed curve of the animation
4. animation-delay = Delay for the start of the animation
5. animation-iteration-count = Number of times an animation should run
6. animation-direction = specifies the direction of the animation

The animation shorthand

All the animation properties from 1-6 can be applied like this:

```
animation: mukul 65 linear 15 infinite reverse;
```

Using percentage value states with animation

We can use % values to indicate what should happen when a certain percent of animation is completed.

```
@keyframes mukul{
0%{
   width: 20px;
  }
50%{
  width: 80px;
  }
100%{
  width: 200px
  }
}
```