Introduction to CSS

1. Introduction to CSS

CSS (Cascading Style Sheets) is a style sheet language used for describing the look and formatting of a document written in HTML. CSS describes how HTML elements should be displayed. This covers everything from layout to colors and fonts.

2. CSS Syntax

A CSS rule-set consists of a selector and a declaration block:

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

The selector points to the HTML element you want to style. Each declaration includes a CSS property name and a value, separated by a colon. A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

3. Inline, Internal, and External CSS

There are three ways of inserting a style sheet:

- Inline style: Styles are added directly to the HTML elements using the style attribute. This method can quickly become hard to manage on larger projects.
-

```
<p style="color:red;">This is a red paragraph.</p>
```

- Internal style: Styles are embedded in the HTML document within the `<style>` tag in the `<head>` section.

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

```
  </style>
</head>
```

- External style: Styles are placed in external CSS files. External stylesheets are the most powerful way to manage CSS because the styles can be applied across multiple pages. To use an external style sheet, add a link to it in the `<head>` section of the HTML file.

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

4. CSS Selectors

CSS selectors are used to select the HTML elements you want to style. Here are some basic selectors:

- `p`: The type selector matches elements by node name.
- `.classname`: The class selector matches elements based on the contents of their class attribute.
- `#idname`: The ID selector matches elements based on the contents of their ID attribute.

5. 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.

6. CSS Positioning

CSS positioning properties allow you to control the position of elements on the webpage. The `position` property can take one of five values: `static`, `relative`, `fixed`, `absolute`, or `sticky`.

7. CSS Flexbox

CSS Flexbox is a layout module that makes it easier to design flexible and efficient layouts. With flexbox, you can align elements horizontally and vertically, regardless of their original size and the size of their parent container.

Flexbox (Flexible Box) is a layout model that allows easy control over direction, order, alignment, and size of boxes. Flexbox can manage both one-dimensional layouts (either in a row or a column). Let's dive into the basics of Flexbox.

1. **Display: Flex**
2. To start using the Flexbox model, you need to first define a flex container. This is done by setting the `display` property of an element to `flex` or `inline-flex`.

```
.container {
  display: flex;
}
```

1. **Flex Direction**
2. The `flex-direction` property is used to define the direction in which the flex items are placed in the flex container. By default, it is set to `row`, which means that the flex items are lined up horizontally.

```
.container {
  display: flex;
  flex-direction: row; /* default value; can be omitted */
}
```

To line up the flex items vertically, set `flex-direction` to `column`.

```
.container {
  display: flex;
  flex-direction: column;
}
```

1. **Flex Wrap**

By default, flex items will all try to fit onto one line. You can change this and allow the items to wrap onto multiple lines with the `flex-wrap` property.

```
.container {
  display: flex;
  flex-wrap: wrap;
}
```

1. **Justify Content**

The `justify-content` property is used to align the flex items along the horizontal line that runs in the direction of the flex-direction.

```
.container {
  display: flex;
  justify-content: center; /* aligns flex items in the center */
}
```

1. **Align Items**

The `align-items` property is used to align the flex items along the vertical line (perpendicular to the direction of the flex-direction).

```
.container {
  display: flex;
  align-items: center; /* aligns flex items in the middle */
}
```

units of measurements in CSS

There are several types of units of measurements in CSS that can be divided into two categories: absolute and relative.

Absolute Units:

Absolute units are fixed and do not scale according to any other value. They are usually used when the output medium is known, such as on a printed page.

- `px`: pixel units.
- `in`: inches — 1in is equal to 96px.
- `cm`: centimeters.
- `mm`: millimeters.
- `pt`: points — 1pt is equal to 1/72th of 1in.
- `pc`: picas — 1pc is equal to 12pt.

Relative Units:

Relative units are relative to another length value. They scale better than absolute units when designing responsive websites, which is generally the best practice for web design.

- `em`: This unit is relative to the font-size of its closest or parent element. If the parent element doesn't have a set font size, it will keep looking up the DOM tree until it finds an element with a set font size.
- `rem`: This unit is relative to the font-size of the root element (`html`).
- `ch`: This is equal to the width of the "0" (zero) character.
- `vw`: This stands for "viewport width". One unit is equal to 1% of the width of the viewport.
- `vh`: This stands for "viewport height". One unit is equal to 1% of the height of the viewport.
- `vmin`: This is equal to 1% of the smaller dimension (height or width) of the viewport.
- `vmax`: This is equal to 1% of the larger dimension (height or width) of the viewport.
- `%`: This unit is a percentage value that is relative to the parent element.

CSS3 introduced new units, including viewport-relative units (`vh`, `vw`, `vmin`, and `vmax`) and the `rem` unit for typography. These units offer more flexibility and control in the responsive design context.