

CSS (Cascading Style Sheets)

CSS is a style sheet language used to design and format HTML web pages. It controls the layout, colors, fonts, spacing, animations, and responsiveness of a website.

CSS Syntax

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

1. Types of CSS

CSS can be added to HTML documents in three ways:

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

1. Inline CSS

Inline CSS is used to apply styles directly inside an HTML element using the style attribute.

Syntax:

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

Example:

```
<h1 style="color: red; font-size: 30px;">
```

Welcome to CSS

```
</h1>
```

2. Internal CSS

Internal CSS is written inside the <style> tag within the <head> section of an HTML document.

Syntax:

```
<head>
```

```
    <style>
```

```

p {
    color: blue;
    font-size: 18px;
}
</style>

</head>

<body>

<p>This is Internal CSS</p>

</body>

```

3. External CSS

External CSS is written in a separate .css file and linked with the HTML file using the <link> tag.

CSS File (style.css):

```

body {
    background-color: lightgray;
}

h1 {
    color: green;
}

```

html file (index.html)

```

<head>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <h1>This is Internal CSS</p1>

</body>

```

2. CSS Selectors

1. Element Selector
2. Universal Selector
3. Class Selector
4. ID Selector
5. Group Selector
6. Descendant Selector
7. Pseudo-class

1. Element Selector (Type Selector)

The Element Selector is used to select HTML elements based on their tag name and apply CSS styles to all elements of that type.

Syntax:

```
html-element {  
  property: value;  
}
```

Example

```
<html>  
<head>  
<style>  
  h1 {  
    color: green;  
  }  
  p {  
    font-size: 20px;  
    color: blue;  
  }
```

```
</style>
</head>
<body>
    <h1>CSS Element Selector</h1>
    <p>This paragraph is styled using Element Selector.</p>
    <p>All paragraph elements get the same style.</p>
</body>
</html>
```

2. Universal Selector (*)

The Universal Selector is used to select all HTML elements on a webpage. It applies the same CSS style to every element.

Syntax:

```
* {
    property: value;
}
```

example

```
<html>
<head>
    <style>
        * {
            color: blue;
            font-family: Arial;
        }
    </style>
</head>
<body>
    <h1>Welcome</h1>
    <p>This is a paragraph.</p>
    <div>This is a division.</div>
```

```
</body>
```

```
</html>
```

3. Class Selector (.)

The Class Selector is used to select HTML elements with a specific class attribute. It is represented using a dot (.) symbol.

Syntax:

```
.classname {  
    property: value;  
}
```

example

```
.box {  
    background-color: lightblue;  
    padding: 20px;  
}  
  
<div class="box">Content 1</div>  
<div class="box">Content 2</div>
```

4. ID Selector (#)

The ID Selector is used to select a single HTML element with a specific id attribute. It is represented using a hash (#) symbol.

Syntax:

```
#idname {  
    property: value;  
}
```

Example:

```
#header {  
    background-color: black;  
    color: white;  
}  
  
<h1 id="header">Welcome</h1>
```

5. Group Selector (,)

The Group Selector is used to apply the same CSS style to multiple selectors at the same time. It uses a comma (,).

Syntax:

```
selector1, selector2, selector3 {  
    property: value;  
}
```

Example:

```
h1, h2, p {  
    font-family: Arial;  
    color: blue;  
}  
<h1>Heading</h1>  
<h2>Sub Heading</h2>  
<p>Paragraph</p>
```

6. Descendant Selector (Space)

The Descendant Selector selects all elements that are inside another element, at any level.

Syntax:

```
parent child {  
    property: value;  
}
```

Example:

```
div p {  
    color: red;  
}  
<div>  
    <p>This paragraph is selected.</p>
```

```

    <section>

        <p>This paragraph is also selected.</p>

    </section>

</div>

```

7. Pseudo-Class Selector (:)

A Pseudo-Class Selector is used to style an element when it is in a specific state or condition.

Syntax:

```

selector:pseudo-class {
    property: value;
}

```

Example:

```

button:hover {
    background-color: green;
}

<button>Submit</button>

```

3.CSS Box Model

Every HTML element is treated as a box containing:

```

+-----+
|  Margin  |
| +-----+ |
| | Border | | | |
| | +-----+ | |
| | |Padding| | |
| | | Content| | |
| | +-----+ | |
| +-----+ |
+-----+

```

Box Model Components:

- Content
- Padding
- Border
- Margin

Types of CSS Box Layouts

1. Block Layout
2. Flexbox Layout
3. Grid Layout

1. Block Layout (display: block)

The `<div>` tag is a container element used to create sections or boxes in a webpage. By applying CSS properties like width, height, background, margin, padding, border, and display, we can design different layouts.

```
<head>

  <title>Box Layout</title>

  <style>

    .box-container {

      display: block;

      width: 300px;

      height: 150px;

      background-color: lightblue;

      border: 2px solid black;

      padding: 20px;

      margin: 20px;

    }

  </style>

</head>

<body>
```

```

<div class="box-container">This is a basic box </div>

<div class="box-container">This is a basic box </div>

<div class="box-container">This is a basic box </div>

</body>

```

2. Flexbox Layout (display: flex)

- Flexbox (Flexible Box Layout) is a CSS layout system used to arrange elements inside a container in a row or column direction. It makes alignment and spacing easier.

- It has single row multiple columns or single column multiple rows
- by default its row flex
- We can Wrap the flex row or column to the next line

Example:

Row flex (flex-direction: row;)

```

+-----+ +-----+ +-----+
| Box 1 | | Box 2 | | Box 3 |
+-----+ +-----+ +-----+

```

Column flex (flex-direction: column);

```

+-----+
| Box 1 |
+-----+
+-----+
| Box 2 |
+-----+
+-----+
| Box 3 |
+-----+

```

Wrap Row or Column Flex (flex-wrap: wrap;)

```

+-----+ +-----+ +-----+
| Box 1 | | Box 2 | | Box 3 |
+-----+ +-----+ +-----+
+-----+ +-----+ +-----+

```

| Box 4 | | Box 5 | | Box 6 |

+-----+ +-----+ +-----+

Row Flex Example

```
.container {  
    display: flex;          /* Enables flex layout */  
    flex-direction: row;    /* Places boxes horizontally */  
    gap: 20px;              /* Space between boxes */  
    justify-content: center; /* Centers the row */  
}
```

program

```
<head>  
  
<style>  
    .container {  
        display: flex;  
        flex-direction: row;  
        gap: 20px;  
        justify-content: center;  
    }  
    .box {  
        width: 200px;  
        height: 200px;  
        border: 2px solid black;  
    }  
    .box img {  
        width: 100%;  
        height: 100%;  
        object-fit: cover;  
    }
```

```
        </style>
</head>
<body>
    <h2>Rose Gallery - Flex Row Layout</h2>
    <div class="container">
        <div class="box"></div>
        <div class="box"></div>
        <div class="box"></div>
        <div class="box"></div>
    </div>
</body>
```

column Flex Example

program

```
<head>
```

```
    <style>
```

```
        .container {
```

```
            display: flex;
```

```
            flex-direction: column;
```

```
            gap: 20px;
```

```
            justify-content: center;
```

```
        }
```

```
        .box {
```

```
            width: 200px;
```

```
            height: 200px;
```

```
            border: 2px solid black;
```

```
        }
```

```

        .box img {
            width: 100%;
            height: 100%;
            object-fit: cover;
        }
    </style>
</head>
<body>
    <h2>Rose Gallery - Flex Row Layout</h2>
    <div class="container">
        <div class="box"></div>
        <div class="box"></div>
        <div class="box"></div>
        <div class="box"></div>
    </div>
</body>

```

row or column flex wrap

add flex-wrap: wrap; in the container

3. Grid Layout (display: grid)

CSS Grid is a two-dimensional layout system used to arrange elements in rows and columns. It is useful for creating complex webpage layouts like galleries, dashboards, and card layouts.

There are 2 methods to create grid

1. Number method

2. Layout method

1. Number method

<head>

<style>

.container{

display: grid;

border: 1px solid green;

grid-row: 25% 25% 25% 25%;

grid-column: 25% 25% 25% 25%;

gap:1px;

}

.header{

background-color: orangered;

height: 25vh;

grid-row: 1/2;

grid-column: 1/5;

}

.sb1{

background-color: rgb(0, 183, 255);

height: 25vh;

grid-row: 2/3;

grid-column: 1/2;

}

.sb2{

background-color: rgb(0, 183, 255);

height: 25vh;

grid-row: 2/3;

grid-column: 2/3;

}

.sb3{

background-color: rgb(0, 183, 255);

```
    height: 25vh;
    grid-row: 2/3;
    grid-column: 3/4;
}
.sb{
    background-color: rgb(56, 230, 128);
    height: 50vh;
    grid-row: 2/4;
    grid-column: 4/5;
}
.main{
    background-color: rgb(229, 239, 91);
    height: 25vh;
    grid-row: 3/4;
    grid-column: 1/4;
}
.footer{
    background-color: rgb(53, 56, 239);
    height: 25vh;
    grid-row: 4/5;
    grid-column: 1/5;
}
</style>
</head>
<body>
<div class="container">
    <div class="header">Header element
        <span>Home</span>
        <span>About</span>
```

```

    <span>Login</span>

  </div>

  <div class="sb1">SmallBox1</div>
  <div class="sb2">SmallBox2</div>
  <div class="sb3">SmallBox3</div>
  <div class="sb"> Side Bar</div>
  <div class="main">MainContent</div>
  <div class="footer">Footer</div>
</div>
</body>

```

Grid Layout Method

```

style.css

.container {
  display: grid;
  grid-template-areas:
    "header header header"
    "sidebar main main"
    "footer footer footer";
  grid-template-rows: 60px 1fr 60px;
  grid-template-columns: 200px 1fr 1fr;
  height: 90vh;
  width:100vw;
}

.header {
  grid-area: header;
  background: #333;
  color: white;
  display: flex;

```

```
align-items: center;
padding: 0 20px;
}

.sidebar {
grid-area: sidebar;
background: #f4f4f4;
padding: 20px;
}

.footer {
grid-area: footer;
background: #333;
color: white;
text-align: center;
line-height: 60px;
}

.main {
grid-area: main;
padding: 20px;
display: grid;
grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));
gap: 20px;
}

.product {
border: 1px solid #ddd;
padding: 10px;
text-align: center;
}

.product img {
width: 100%;
```

```
height: auto;
}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>E-Commerce</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<div class="container">
    <div class="header">
        <h1>E-Commerce</h1>
    </div>
    <div class="sidebar">
        <ul>
            <li>About Us</li>
            <li>Contact</li>
            <li>About Us</li>
            <li>Contact</li>
            <li>About Us</li>
            <li>Contact</li>
        </ul>
    </div>
    <div class="main">
        <div class="product">
            
            <h4>Product 1</h4>
            <p>$10.00</p>
        </div>
    </div>
</div>
```

</div>

<div class="product">

<h4>Product 2</h4>

<p>\$20.00</p>

</div>

<div class="product">

<h4>Product 3</h4>

<p>\$15.00</p>

</div>

<div class="product">

<h4>Product 4</h4>

<p>\$25.00</p>

</div>

</div>

<div class="footer">

© E-Commerce

</div>

</div>

</body>

</html>

=====

Sample DataBase Connection code

=====

```
<div id='home' style="display: none">
```

```
<h3> welcomes to KI Deemed to be University Guntur</h3>
```

```
<form method="post" enc type="">
```

```
    <h1> Welcomes to Dept Of CSE-4</h1>
```

```
    Enter the student Name:<input type="text" name="name"/><br>
```

```
    Enter the student father Name:<input type="text" name="fname"/><br>
```

```
    Enter the Student Address:<input type="text" name="address"/><br>
```

```
    Enter the cell number:<input type="text" name="cellnumber"/><br>
```

```
        <input type="submit" name="submit"/>
```

```
</form>
```

```
<?php
```

```
    if(isset($_REQUEST['submit']))
```

```
    {
```

```
        $aa=$_REQUEST['name'];
```

```
        $bb=$_REQUEST['fname'];
```

```
        $cc=$_REQUEST['address'];
```

```
        $dd=$_REQUEST['cellnumber'];
```

```
        mysql_connect('localhost','root');
```

```
        $conn=mysql_select_db('ists');
```

```
        $sql="INSERT INTO
```

```
`student`(`name`,`fname`,`address`,`cellnumber`)VALUES(\"$aa\", \"$bb\", \"$cc\", \"$dd\");
```

```
        $result=mysql_query($sql);
```

```
        if($result)
```

```
        {
```

```
            echo "<script>alert('inserted sucessfully');</script>";
```

```
        }
```

```
else
{
echo "<script>alert('not sucessfully inserted wrong');</script>";
}
}
?>
</h3>
</div>
```