The z-index property in CSS controls the stack order of elements. Elements with a higher z-index value appear in front of those with a lower value. This is useful when you want to manage overlapping elements on a webpage.

---

Syntax:

selector {

z-index: value;

}

value: Can be a number (positive, negative, or zero) or auto.

Key Points:

1. The z-index only works on elements with a position property set to relative, absolute, fixed, or sticky.

2. Higher z-index values bring elements to the front.

3. Default z-index is auto.

---

Example Program:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>CSS Z-Index Example</title>

<style>

body {

margin: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #f0f0f0;

}

.box {

position: absolute;

width: 150px;

height: 150px;

border: 2px solid black;

color: white;

display: flex;

justify-content: center;

align-items: center;

font-family: Arial, sans-serif;

}

.box1 {

background-color: red;

z-index: 1;

}

.box2 {

background-color: blue;

z-index: 2;

}

.box3 {

background-color: green;

z-index: 3;

}

</style>

</head>

<body>

<div class="box box1">Box 1</div>

<div class="box box2">Box 2</div>

<div class="box box3">Box 3</div>

</body>

</html>

---

Explanation:

1. Box 1 (Red): Has a z-index of 1, so it appears behind the other two.

2. Box 2 (Blue): Has a z-index of 2, appearing in the middle.

3. Box 3 (Green): Has the highest z-index (3), so it appears on top.

Output:

Green (z-index: 3) will overlap both red and blue.

Blue (z-index: 2) will overlap red but be behind green.

Red (z-index: 1) will be at the bottom of the stack.