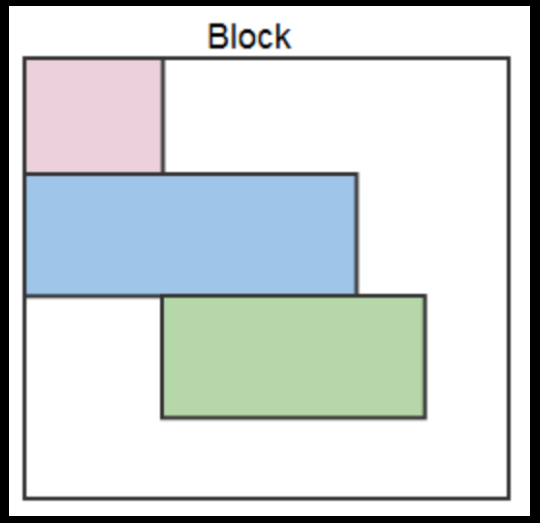
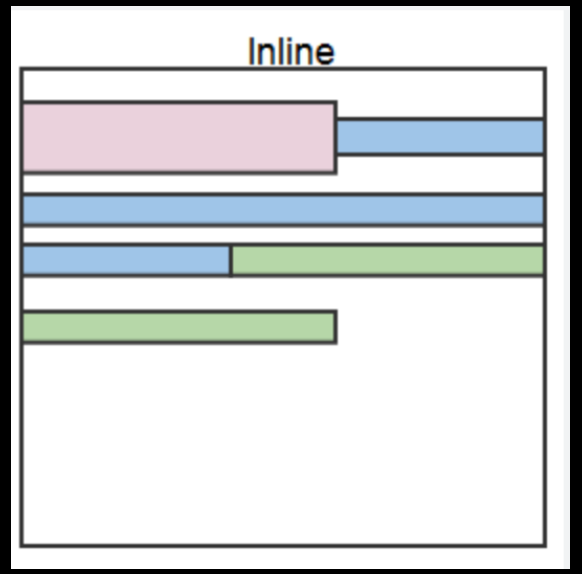
Display Property (Block / Inline Elements)





Block Elements

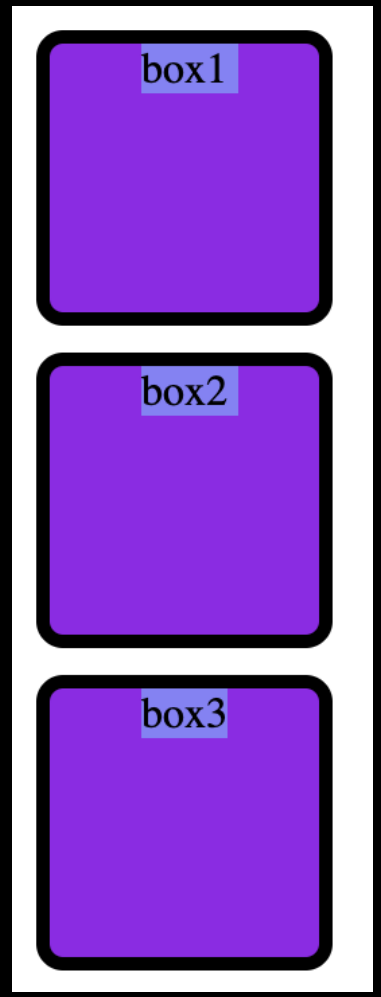
- New Line: Start on a new line.
- Full Width: Take up all horizontal space.
- Styling: Can have margins and padding.
- Size: Width and height can be set.
- Examples: , <p>, <h1>, <ul>, <li>.

Inline Elements

- Flow: Stay in line with text.
- Width: Just as wide as the content.
- No Break: No new line between elements.
- Limited Styling: Can't set size easily.
- Examples: , <a>, , .

Display Property (Block)

```css
.box {
    height: 100px;
    width: 100px;
    background-color: blueviolet;
    margin: 10px;
    text-align: center;
    border: 5px solid black;
    border-radius: 10px;

    display: block;
}
```




```html
<head>
    <title>Display Block</title>
    <link rel="stylesheet" href="../../css/level 5/display.css">
</head>
<body>
    <div id="parent">
        <div id="div1" class="box">box1</div>
        <div id="div2" class="box">box2</div>
        <div id="div3" class="box">box3</div>
    </div>
</body>
```

Display Property (Inline)

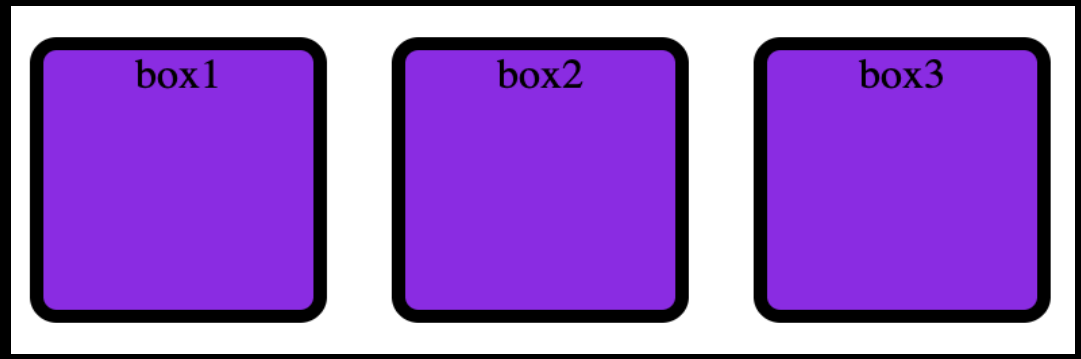
```css
.box {
    height: 100px;
    width: 100px;
    background-color: blueviolet;
    margin: 10px;
    text-align: center;
    border: 5px solid black;
    border-radius: 10px;

    display: inline;
}
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Display Inline</title>
    <link rel="stylesheet" href="../../css/
    level 5/display_inline.css">
</head>
<body>
    <div id="parent">
        <div id="div1" class="box">box1</div>
        <div id="div2" class="box">box2</div>
        <div id="div3" class="box">box3</div>
    </div>
</body>
</html>
```

Display Property (Inline-Block)

```css
.box {
    height: 100px;
    width: 100px;
    background-color: blueviolet;
    margin: 10px;
    text-align: center;
    border: 5px solid black;
    border-radius: 10px;

    display: inline-block;
}
```

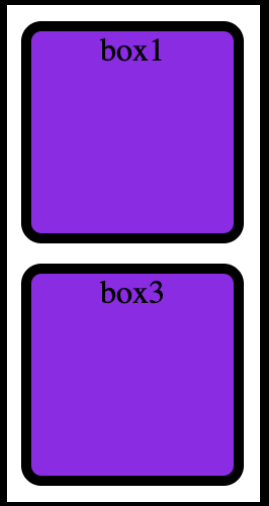
```html
<head>
    <title>Display Inline Block</title>
    <link rel="stylesheet" href="../../css/level 5/display_inline_block.css">
</head>
<body>
    <div id="parent">
        <div id="div1" class="box">box1</div>
        <div id="div2" class="box">box2</div>
        <div id="div3" class="box">box3</div>
    </div>
</body>
```



Display Property (None)

```css
.box {
    height: 100px;
    width: 100px;
    background-color: blueviolet;
    margin: 10px;
    text-align: center;
    border: 5px solid black;
    border-radius: 10px;
}

#div2 {
    display: none;
}
```

```html
<head>
    <title>Display None</title>
    <link rel="stylesheet" href="../../css/
    level 5/display_none.css">
</head>
<body>
    <div id="parent">
        <div id="div1" class="box">box1</div>
        <div id="div2" class="box">box2</div>
        <div id="div3" class="box">box3</div>
    </div>
</body>
```



Relative Units

📏 CSS Units Cheat Sheet

px

Absolute pixel value

%

A percentage of the parent element.
100% is the width of the parent element

em

Relative to the font size of the element

vh

Relative to 1% of the viewport's height

rem

Relative to the font size of
the root element

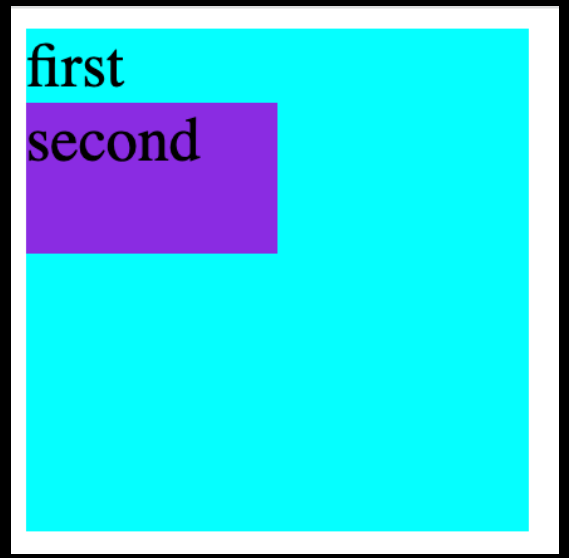
vw

Relative to 1% of the viewport's width

Relative Units (Percentage)

```css
#first {
    height: 200px;
    width: 200px;
    background-color: aqua;
    font-size: 25px;
}

#second {
    background-color: blueviolet;
    width: 50%;
    height: 30%;
}
```

```html
<body>
    <div id="first">
        first
        <div id="second">
            second
        </div>
    </div>
</body>
```

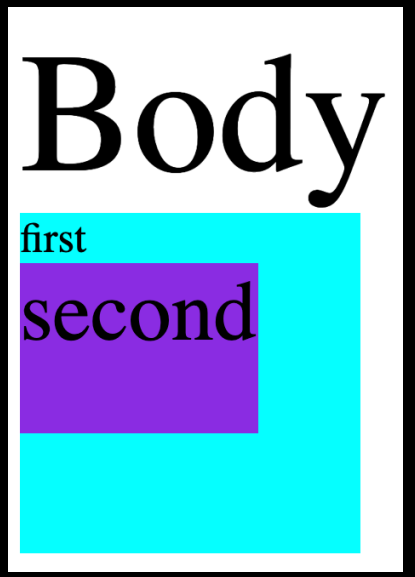


- **Relative Sizing**: Facilitates dynamic sizing relative to parents.
- **Adaptability**: Ensures responsiveness across various screens.
- **Dimensions**: Quickly set width and height as a percentage.

Relative Units (EM)

```css
body {
    font-size: 100px;
}

#first {
    height: 200px;
    width: 200px;
    background-color: aqua;
    font-size: 25px;
}

#second {
    background-color: blueviolet;
    width: 70%;
    height: 50%;
    font-size: 2em;
}
```

```html
<body>
    Body
    <div id="first">
        first
        <div id="second">
            second
        </div>
    </div>
</body>
```

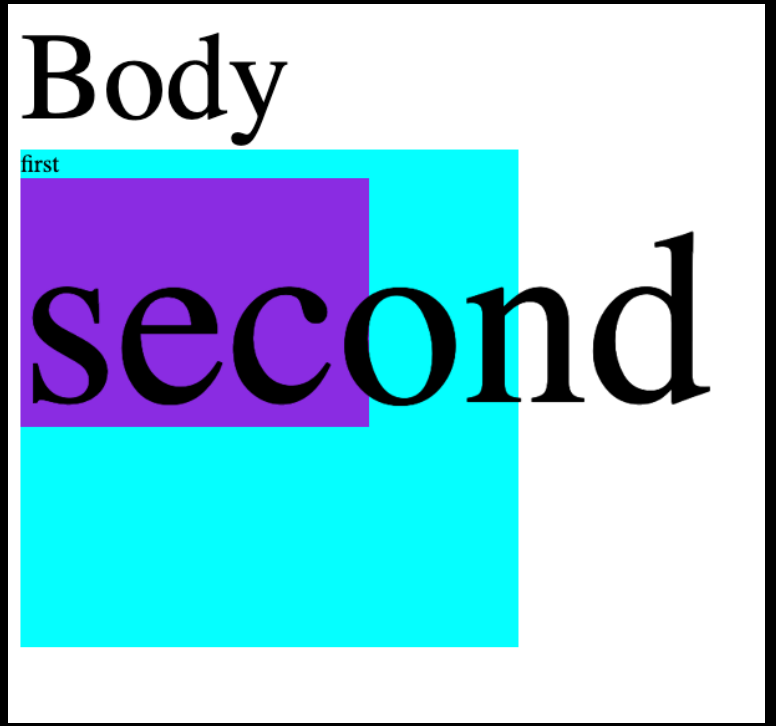


- **Relative Unit**: Sized relative to the parent element's font size.
- **Scalability**: Facilitates easy scaling of elements for responsive design.
- **Font Sizing**: Commonly used for setting font sizes adaptively.

Relative Units (REM)

```css
* {
    font-size: 50px;
}

#first {
    height: 200px;
    width: 200px;
    background-color: aqua;
    font-size: 10px;
}

#second {
    background-color: blueviolet;
    width: 70%;
    height: 50%;
    font-size: 2rem;
}
```

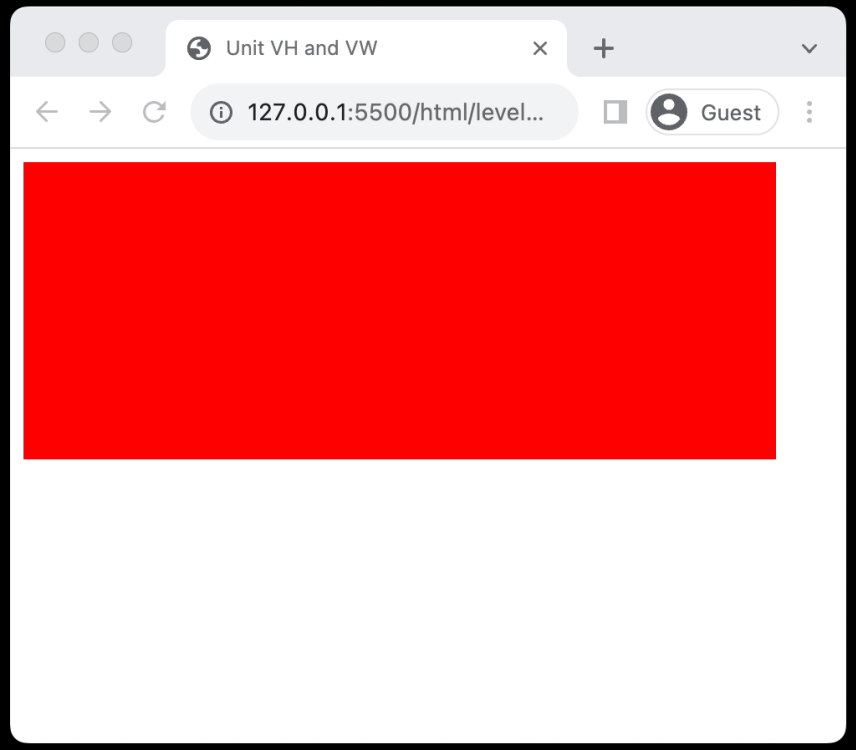
```html
<body>
    Body
    <div id="first">
        first
        <div id="second">
            second
        </div>
    </div>
</body>
```



- **Relative Sizing**: Facilitates dynamic sizing relative to root element.
- **Adaptability**: Ensures responsiveness across various screens.
- **Dimensions**: Quickly set width and height as a percentage.

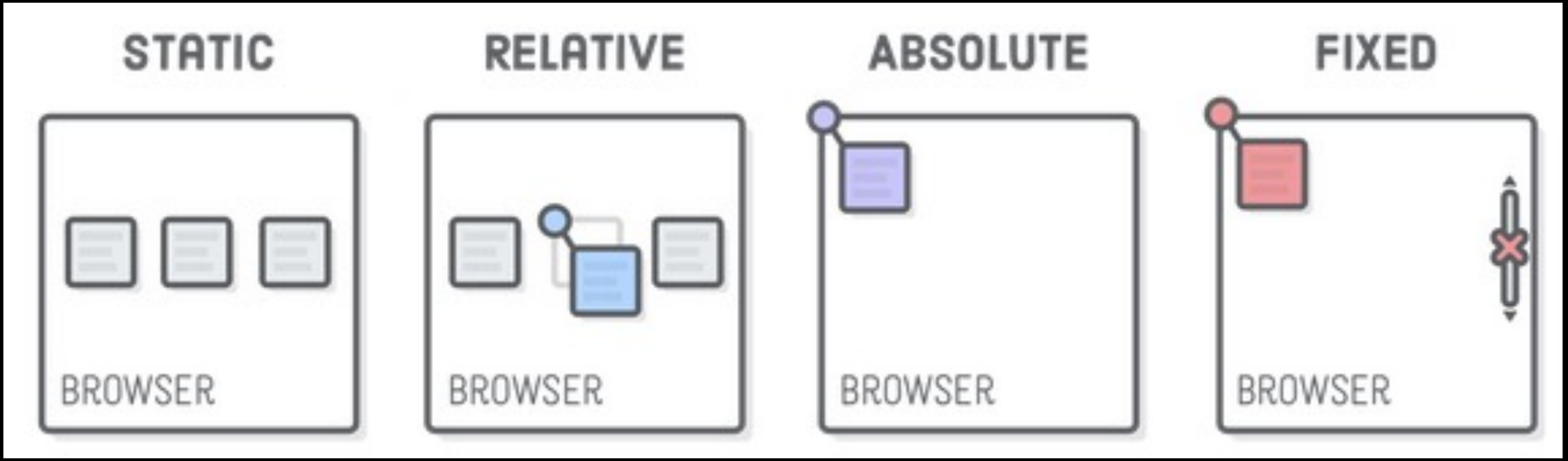
Relative Units (VW/VH)

```html
<head>
    <title>Unit VH and VW</title>
    <style>
        #first {
            height: 50vh;
            width: 90vw;
            background-color: red;
        }
    </style>
</head>
<body>
    <div id="first"></div>
</body>
```



- **Viewport Relative Units**: Units based on viewport's width (vw) or height (vh) for responsive design.
- **Responsive Layouts**: Essential for creating adaptive layouts; e.g., height: 100vh for full-screen sections.
- **Element Sizing**: Useful for defining heights and widths that scale

Position Property

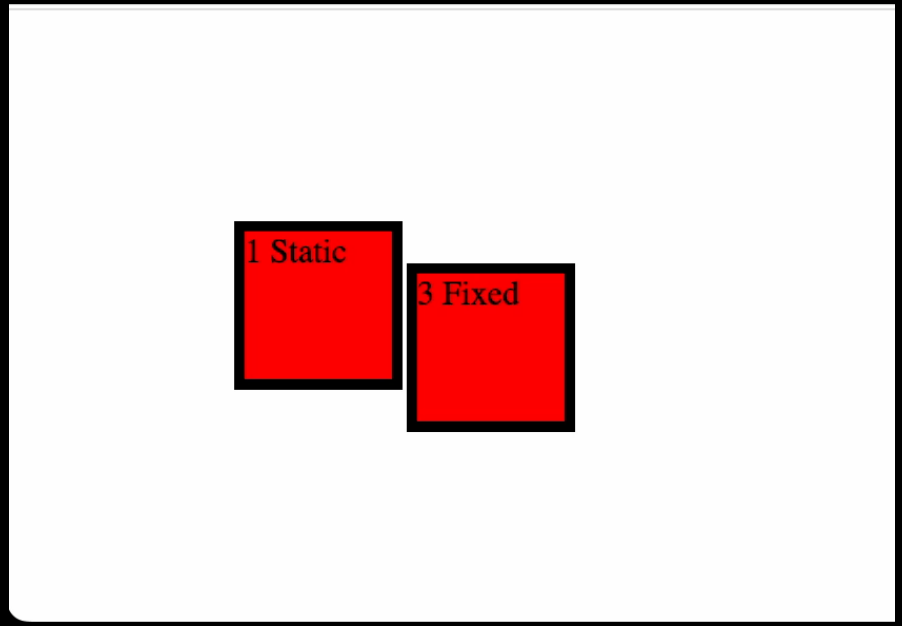


- **Static** (default) : Elements follow the normal document flow. (top, right, bottom, left, z-index would not work)
- **Relative**: Element's position adjusted from its normal position.
- **Absolute**: Positions element relative to the nearest positioned ancestor.
- **Fixed**: Element positioned relative to the viewport, does not move on scroll.

Position Property

```css
div {
    height: 70px;
    width: 70px;
    background-color: red;
    border: 5px solid black;
    margin: 20px;
}
#div1 {
    position: static;
}
#div2 {
    position: relative;
    top: 20px;
    left: 90px;
}
#div3 {
    position: fixed;
    top: 20px;
    left: 90px;
}
#div4 {
    position: absolute;
    top: 200px;
    left: 200px;
}
```

```html
<head>
    <title>Position</title>
    <link rel="stylesheet" href="../../
    css/level 5/positions.css">
</head>
<body>
    <div id="div1">1 Static</div>
    <div id="div2">2 Relative</div>
    <div id="div3">3 Fixed</div>
    <div id="div4">4 Absolute</div>
</body>
```

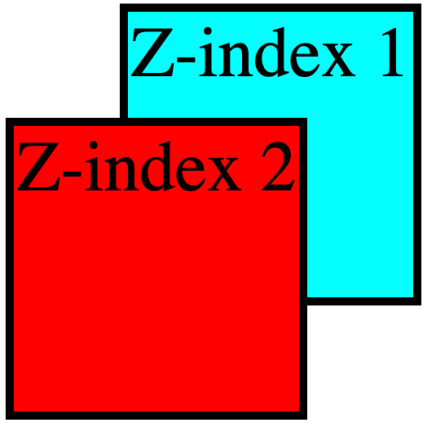


Position Property (z index)

```css
.container {
    position: relative;
}
.box1, .box2 {
    position: absolute;
    border: 3px solid black;
    width: 100px;
    height: 100px;
    text-align: center;
    font-size: 25px;
}
.box1 {
    background-color: red;
    left: 20px;
    top: 60px;
    z-index: 2;
}
.box2 {
    background-color: aqua;
    left: 60px;
    top: 20px;
    z-index: 1;
}
```

- **Stacking Order**: Determines the stacking order of elements along the Z-axis.
- **Position Context**: Only applies to elements with position set to relative, absolute, fixed, or sticky.
- **Integer Values**: Accepts integer values, including negative numbers.
- **Higher Values**: An element with a higher z-index value appears above others.

```html
<head>
    <title>Z-Index</title>
    <link rel="stylesheet" href="../../css/level 5/z-index.css">
</head>
<body>
    <div class="container">
        <div class="box1">Z-index 2</div>
        <div class="box2">Z-index 1</div>
    </div>
</body>
```

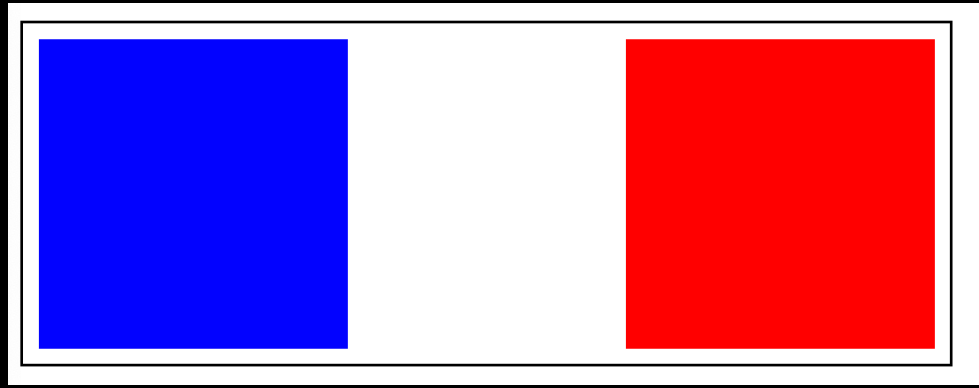


Float Property

```css
.container {
    height: 110px;
    width: 300px;
    border: 1px solid #000;
}

.box {
    width: 100px;
    height: 100px;
    margin: 5px;
}

.box1 {
    background-color: red;
    float: right;
}

.box2 {
    background-color: blue;
    float: left;
}
```

- **Element Alignment**: Allows elements to be aligned to the left or right within their containing element.
- **Values**: Can take values like "left", "right", or "none" to determine the floating direction.
- **Old Layout Technique**: Less commonly used with the advent of Flexbox.

```html
<head>
    <title>Float Property</title>
    <link rel="stylesheet" href="../../css/level 6/float.css">
</head>
<body>
    <div class="container">
        <div class="box box1"></div>
        <div class="box box2"></div>
    </div>
</body>
```



Practice Set Level 5

Display and Position

- Create a webpage with header, footer, and a content area.
  - Header
    - Create a nav bar with links.
  - Main
    - Create a div with width and height, Background green and border radius 50%
    - Create Three divs with container height and width 100px. Display inline block.
    - Set the correct position property for the single div element to ensure it remains at the right side of the page and does not shift when scrolling.
    - Use z-index to place the div on top of another div.
  - Footer
    - Add text in footer.