The justify-content property in CSS is used to align and distribute space between flex items along the main axis of a flex container. This property is part of the Flexbox layout model, which helps create flexible and responsive layouts.

**Values of justify-content**

1. **flex-start**
   * **Description**: Aligns flex items at the start of the container.
   * **Example**: Items are aligned to the left (or top, depending on the flex direction).
2. **flex-end**
   * **Description**: Aligns flex items at the end of the container.
   * **Example**: Items are aligned to the right (or bottom, depending on the flex direction).
3. **center**
   * **Description**: Centers flex items within the container.
   * **Example**: Items are centered horizontally (or vertically, depending on the flex direction).
4. **space-between**
   * **Description**: Distributes flex items with equal space between them, with no space at the start or end.
   * **Example**: Items are spread out with equal spacing between them.
5. **space-around**
   * **Description**: Distributes flex items with equal space around them, including space before the first item and after the last item.
   * **Example**: Items have equal space around them, with half of the space at the start and end of the container.
6. **space-evenly**
   * **Description**: Distributes flex items with equal space between and around them.
   * **Example**: Items have equal spacing between them and also at the start and end of the container.

Example:

<!DOCTYPE *html*>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

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

    <title>Document</title>

     <style>

       .container {

        display:flex;

        border: 1px solid black;

        justify-content: space-between;

        flex-direction: row;

        height: 500px;

       }

       .container>div {

        margin: 10px;

        padding: 10px;

        border:1px solid blue;

        font-size: 30px;

        width:50px;

        height: 50px;

        background-color: yellowgreen;

       }

     </style>

</head>

<body>

        <div *class*="container">

            <div>1</div>

            <div>2</div>

            <div>3</div>

        </div>

</body>

</html>

<!DOCTYPE *html*>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

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

    <title>Document</title>

     <style>

       .container

       {

         display:flex;

         flex-direction: row;

         justify-content: space-between;

         border: 1px solid black;

         padding:20px;

         background-color: bisque;

      }

       .container>div {

          width:400px;

          background-color:azure;

       }

       .img {

        display: flex;

        flex-direction: column;

       }

       .container1 {

        display:flex;

        flex-wrap: wrap;

        }

     </style>

</head>

<body>

        <div *class*="container">

          <div>

          <h2>Gaming accessories</h2>

          <div *class*="container1">

              <div *class*="img">

              <img *src*="images/i1.jpg">

              <span>Headsets</span>

              </div>

              <div *class*="img">

              <img *src*="images/i2.jpg">

              <span>Headsets</span>

            </div>

            <div *class*="img">

            <img *src*="images/i3.jpg">

            <span>Headsets</span>

          </div>

          <div *class*="img">

            <img *src*="images/i4.jpg">

            <span>Headsets</span>

          </div>

          </div>

        </div>

        <div>

        <h2>New home arrivals under $50</h2>

          <div>

            <img *src*="images/k1.jpg">

            <img *src*="images/k2.jpg">

            <img *src*="images/k3.jpg">

            <img *src*="images/k4.jpg">

          </div>

        </div>

        <div>

        <h2>Shop deals in Fashion</h2>

          <div>

            <img *src*="images/f1.jpg">

            <img *src*="images/f2.jpg">

            <img *src*="images/f3.jpg">

            <img *src*="images/f4.jpg">

          </div>

        </div>

        <div>

          <h2>Refresh your space</h2>

          <div>

            <img *src*="images/d1.jpg">

            <img *src*="images/d2.jpg">

            <img *src*="images/d3.jpg">

            <img *src*="images/d4.jpg">

          </div>

        </div>

        </div>

</body>

</html>

Example

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Justify Content Example</title>

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

</head>

<body>

<div class="container flex-start">

<div class="item">1</div>

<div class="item">2</div>

<div class="item">3</div>

</div>

<div class="container flex-end">

<div class="item">1</div>

<div class="item">2</div>

<div class="item">3</div>

</div>

<div class="container center">

<div class="item">1</div>

<div class="item">2</div>

<div class="item">3</div>

</div>

<div class="container space-between">

<div class="item">1</div>

<div class="item">2</div>

<div class="item">3</div>

</div>

<div class="container space-around">

<div class="item">1</div>

<div class="item">2</div>

<div class="item">3</div>

</div>

<div class="container space-evenly">

<div class="item">1</div>

<div class="item">2</div>

<div class="item">3</div>

</div>

</body>

</html>

Style.css

body {

display: flex;

flex-direction: column;

align-items: center;

padding: 20px;

background-color: #f0f0f0;

}

.container {

display: flex;

width: 80%;

margin-bottom: 20px;

border: 1px solid #ccc;

background-color: #fff;

padding: 10px;

}

.item {

background-color: #4CAF50;

color: white;

text-align: center;

padding: 20px;

margin: 5px;

border-radius: 4px;

}

.flex-start {

justify-content: flex-start;

}

.flex-end {

justify-content: flex-end;

}

.center {

justify-content: center;

}

.space-between {

justify-content: space-between;

}

.space-around {

justify-content: space-around;

}

.space-evenly {

justify-content: space-evenly;

}

The position property in CSS is used to control the positioning of an element on a webpage. It determines how an element is placed within its containing element and how it interacts with other elements. The position property can take several values: static, relative, absolute, fixed, and sticky.

Here's a brief overview of each value with examples:

**1. static**

**Definition:** This is the default positioning for all elements. Elements with position: static are positioned according to the normal flow of the document. They are not affected by the top, right, bottom, or left properties.

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Static Position Example</title>

<style>

.static-box {

background-color: lightblue;

width: 200px;

height: 100px;

border: 2px solid blue;

}

</style>

</head>

<body>

<div class="static-box">Static Position</div>

</body>

</html>

**Explanation:**The .static-box element is positioned in the normal document flow. It will appear where it would naturally fall in the layout.

**2. relative**

**Definition:** An element with position: relative is positioned relative to its normal position. It can be moved using the top, right, bottom, and left properties, but the space originally occupied by the element is still preserved.

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Relative Position Example</title>

<style>

.relative-box {

position: relative;

background-color: lightcoral;

width: 200px;

height: 100px;

border: 2px solid red;

top: 20px; /\* Move down by 20px \*/

left: 30px; /\* Move right by 30px \*/

}

</style>

</head>

<body>

<div class="relative-box">Relative Position</div>

</body>

</html>

**Explanation:**The .relative-box is moved 20px down and 30px right from its normal position, but it still takes up the same amount of space in the layout.

Example:

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

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

<title>Relative Position Example</title>

<style>

    .container

    {

        width: 300px;

        height:300px;

        border: 1px solid black;

        background-color: aquamarine;

    }

    .div1 {

        position: relative;

        left:100px;

    }

    .div2 {

        position: relative;

        right: 50px;

        top:50px;

    }

</style>

</head>

<body>

    <div *class*="container div1">Division1</div>

    <div *class*="container div2">Division2</div>

</body>

</html>

Example:

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

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

<title>Relative Position Example</title>

<style>

   .s2 {

      position: relative;

      top:-5px;

      left:-10px;

   }

</style>

</head>

<body>

    <div *class*="container">

        <span *class*="s1">A</span>

        <span *class*="s2">2</span>

    </div>

</body>

</html>

**Example:**

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

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

<link *rel*="stylesheet" *href*="../node\_modules/bootstrap-icons/font/bootstrap-icons.min.css">

<title>Relative Position Example</title>

<style>

.bi-cart {

    font-size: 50px;

}

.value {

    font-size:20px ;

    position: relative;

    left:-30px;

    top:-35px;

    color:red;

}

.cart {

    font-size: 20px;

    font-weight: bold;

    left:-20px;

    top:10px;

    position: relative;

}

</style>

</head>

<body>

    <div>

        <span *class*="bi bi-cart"></span>

        <span *class*="value">0</span>

        <span *class*="cart">Cart</span>

    </div>

</body>

</html>

**3. absolute**

**Definition:** An element with position: absolute is positioned relative to its nearest positioned ancestor (i.e., an ancestor with position other than static). If no such ancestor exists, it is positioned relative to the initial containing block (usually the <html> element).

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Absolute Position Example</title>

<style>

.relative-container {

position: relative;

background-color: lightgrey;

width: 300px;

height: 200px;

border: 2px solid grey;

padding: 20px;

}

.absolute-box {

position: absolute;

background-color: lightgreen;

width: 150px;

height: 75px;

border: 2px solid green;

top: 20px; /\* 20px from the top of the .relative-container \*/

right: 20px; /\* 20px from the right of the .relative-container \*/

}

</style>

</head>

<body>

<div class="relative-container">

<div class="absolute-box">Absolute Position</div>

</div>

</body>

</html>

Example:

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

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

<link *rel*="stylesheet" *href*="../node\_modules/bootstrap-icons/font/bootstrap-icons.min.css">

<title>Relative Position Example</title>

<style>

    div {

        width:400px;

        height:300px;

        border: 1px solid black;

        position: absolute;

        left:20px;

        top:50px;

    }

    span {

        font-size: 40px;

        font-weight: bold;

        position: absolute;

        top:100px;

        left:50px;

    }

</style>

</head>

<body>

    <div>

        <span>Span</span>

    </div>

</body>

</html>

**Explanation:**The .absolute-box is positioned 20px from the top and right of its nearest positioned ancestor (.relative-container). The .relative-container itself has position: relative, which makes it the reference point for the absolute positioning.

**4. fixed**

**Definition:** An element with position: fixed is positioned relative to the viewport. It remains in the same position even when the page is scrolled.

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Fixed Position Example</title>

<style>

.fixed-box {

position: fixed;

background-color: lightpink;

width: 150px;

height: 75px;

border: 2px solid pink;

bottom: 20px; /\* 20px from the bottom of the viewport \*/

right: 20px; /\* 20px from the right of the viewport \*/

}

</style>

</head>

<body>

<div class="fixed-box">Fixed Position</div>

<div style="height: 1500px;">Scroll down to see the fixed box</div>

</body>

</html>

**Explanation:**The .fixed-box remains 20px from the bottom and right of the viewport, regardless of scrolling.

**5. sticky**

**Definition:** An element with position: sticky toggles between relative and fixed, depending on the user's scroll position. It acts like relative until it crosses a defined threshold, at which point it behaves like fixed.

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Sticky Position Example</title>

<style>

header {

background-color: lightblue;

padding: 10px;

text-align: center;

position: sticky;

top: 0; /\* Stick to the top of the viewport \*/

z-index: 1000; /\* Ensure it is on top of other content \*/

}

.content {

height: 1500px; /\* To make the page scrollable \*/

padding: 20px;

}

</style>

</head>

<body>

<header>

Sticky Header

</header>

<div class="content">

Scroll down to see the sticky header in action.

</div>

</body>

</html>

**Explanation:** The header element with position: sticky sticks to the top of the viewport as you scroll down the page.

**Example:**

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

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

<title>Sticky Position Example</title>

<style>

    .container {

       border: 1px solid red;

    }

    .child{

        border: inherit;

    }

</style>

</head>

<body>

    <div *class*="container">

        <div *class*="child">Div1</div>

        <div *class*="child">Div2</div>

        <div *class*="child">Div3</div>

    </div>

</body>

</html

CSS float

CSS float property is used to how element is floated within page (OR) how element is displayed.

The value of float is,

1. Left
2. Right
3. Inline-start
4. Inline-end
5. Inherit
6. None

Example:

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

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

<title>Sticky Position Example</title>

<style>

   .container1{

        float:right;

        width:200px;

        height:200px;

        background-color: aquamarine;

    }

    .container2{

        float:left;

        width:200px;

        height:200px;

        background-color: aquamarine;

    }

    .container3{

        float:left;

        width:700px;

        height:200px;

        background-color: aquamarine;

        margin-left: 150px;

    }

    .container4{

        float:none;

        width:200px;

        height:200px;

        background-color: aquamarine;

    }

</style>

</head>

<body>

    <div *class*="container1">

        Container1

    </div>

    <div *class*="container2">

        Container2

    </div>

    <div *class*="container3">

        Container3

    </div>

    <div *class*="container4">

        Container4

    </div>

</body>

</html>