

Detailed Explanation of `justify-content` with Examples

The `justify-content` property is used in CSS to align items along the main axis in flexbox or grid layouts. The main axis is determined by the `flex-direction` property in flexbox and the writing mode in grid layouts.

Syntax

```
.container {  
  display: flex; /* or display: grid; */  
  justify-content: value;  
}
```

Values

1. `flex-start`

Items are aligned at the beginning of the main axis (default behavior in most cases).

Example:

```
<div class="container">  
  <div class="item">Item 1</div>  
  <div class="item">Item 2</div>  
  <div class="item">Item 3</div>  
</div>
```

```
.container {  
  display: flex;  
  justify-content: flex-start;  
  gap: 10px;  
}  
.item {  
  background: lightblue;  
  padding: 10px;
```

```
border: 1px solid blue;
}
```

Visual Output: All items will start from the left (or the start of the main axis).

2. flex-end

Items are aligned at the end of the main axis.

Example:

```
.container {
  display: flex;
  justify-content: flex-end;
  gap: 10px;
}
```

Visual Output: All items will align to the right (or the end of the main axis).

3. center

Items are centered along the main axis.

Example:

```
.container {
  display: flex;
  justify-content: center;
  gap: 10px;
}
```

Visual Output: Items are evenly spaced around the center of the container.

4. space-between

Items are evenly distributed along the main axis, with the first item at the start and the last item at the end of the container.

Example:

```
.container {  
  display: flex;  
  justify-content: space-between;  
}
```

Visual Output: Equal space between items, but no space at the edges of the container.

5. space-around

Items are evenly distributed with equal space around them. This means items will have half-sized spaces at the edges compared to the space between items.

Example:

```
.container {  
  display: flex;  
  justify-content: space-around;  
}
```

Visual Output: Items are spaced with equal padding around, but the edges of the container have smaller gaps.

6. space-evenly

Items are evenly distributed along the main axis, with equal space between and around them.

Example:

```
.container {  
  display: flex;  
  justify-content: space-evenly;  
}
```

Visual Output: Equal space around all items, including the edges of the container.

Comparison Chart

Value	Spacing Behavior
flex-start	All items align at the start (no extra spacing).
flex-end	All items align at the end (no extra spacing).
center	Items are centered with no extra space at edges.
space-between	Equal space between items, none at the edges.
space-around	Equal space around items, smaller spaces at edges compared to between items.
space-evenly	Equal space around all items, including edges.

Interactive Example (HTML + CSS)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Justify Content</title>
  <style>
    .container {
      display: flex;
      width: 100%;
      height: 100px;
      border: 2px solid black;
      margin-bottom: 20px;
    }
    .item {
      background: lightcoral;
      padding: 10px;
      border: 1px solid darkred;
    }
  </style>
</head>
<body>
  <div class="container" style="justify-content: flex-start;">
    <div class="item">Item 1</div>
    <div class="item">Item 2</div>
```

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

<div class="container" style="justify-content: center;">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>

<div class="container" style="justify-content: space-between;">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>

<div class="container" style="justify-content: space-evenly;">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
</body>
</html>
```

Open this code in a browser to visually understand the differences between each value.