

CSS provides several ways to control the alignment of text within elements. The most common methods are:-

1. **text-align** property (for horizontal alignment)
2. **vertical-align** property (for vertical alignment)
3. **Flexbox and Grid** (for more advanced layout control)

1. Horizontal Text Alignment with `text-align`

The `text-align` property is used to set the horizontal alignment of text within a block-level element (like a `div`, `p`, or `h1`).

Values for `text-align`:

- `left`: Aligns the text to the left of the container (default for most languages).
- `right`: Aligns the text to the right of the container.
- `center`: Centers the text horizontally.
- `justify`: Stretches the text to fill the width of the container, adding space between words to align both sides.

Example:

Css code

```
/* Align text to the left */
.left-align {
  text-align: left;
}

/* Align text to the right */
.right-align {
  text-align: right;
}

/* Center the text */
.center-align {
  text-align: center;
}

/* Justify the text */
.justify-align {
  text-align: justify;
}
```

Html code

```
<div class="left-align">
  <p>This text is aligned to the left.</p>
</div>

<div class="right-align">
  <p>This text is aligned to the right.</p>
</div>
```

```
<div class="center-align">
  <p>This text is centered horizontally.</p>
</div>
```

```
<div class="justify-align">
  <p>This text is justified. It will spread out to fill the container, adding
space between words as needed.</p>
</div>
```

2. Vertical Text Alignment with `vertical-align`

The `vertical-align` property is used to control the vertical positioning of inline-level elements. It works in conjunction with inline or inline-block elements, like text inside images or buttons.

Common Values for `vertical-align`:

- `baseline`: Aligns the element to the baseline of its parent (default).
- `middle`: Aligns the element to the middle of the line height.
- `top`: Aligns the element to the top of the line height.
- `bottom`: Aligns the element to the bottom of the line height.

Example:

```
Css code
/* Align text vertically in the middle of a container */
.middle-align {
  display: inline-block;
  vertical-align: middle;
  line-height: 100px; /* Define a height for the container */
  height: 100px;
  width: 100%;
}
html code
<div class="middle-align">
  <p>This text is vertically aligned in the middle.</p>
</div>
```

3. Centering Text with Flexbox

Flexbox provides a more modern and powerful way to align content both vertically and horizontally, especially when the container has a dynamic size.

Example:

```
Css code
.container {
  display: flex;
  justify-content: center; /* Horizontal centering */
  align-items: center; /* Vertical centering */
  height: 100vh; /* Full viewport height */
}
```

```
}  
  
.content {  
  text-align: center;  
}
```

html code

```
<div class="container">  
  <div class="content">  
    <h1>Centered Text with Flexbox</h1>  
    <p>This text is both vertically and horizontally centered.</p>  
  </div>  
</div>
```

- `justify-content: center;` horizontally centers the content.
- `align-items: center;` vertically centers the content.

4. Centering Text with Grid Layout

CSS Grid is another powerful layout system that allows easy centering of text both horizontally and vertically.

Example:

Css code

```
.grid-container {  
  display: grid;  
  place-items: center; /* Centers both horizontally and vertically */  
  height: 100vh;  
}
```

```
.grid-content {  
  text-align: center;  
}
```

Html code

```
<div class="grid-container">  
  <div class="grid-content">  
    <h1>Centered Text with Grid</h1>  
    <p>This text is both vertically and horizontally centered using CSS  
Grid.</p>  
  </div>  
</div>
```

- `place-items: center;` is shorthand for both `align-items` and `justify-items`, which centers the content both horizontally and vertically.

5. Aligning Text Inside a Button or Inline Elements

If you're aligning text within inline or inline-block elements like buttons or links, the `text-align` property and `vertical-align` property may be combined, but often you will just need **padding** and **line-height** to adjust alignment.

Example with a Button:

css

Copy code

```
.button {
  display: inline-block;
  padding: 15px 25px;
  font-size: 16px;
  text-align: center; /* Horizontally center the text inside the button */
  line-height: 50px; /* Vertically align the text */
  background-color: #007bff;
  color: white;
  border: none;
  border-radius: 4px;
}

.button:hover {
  background-color: #0056b3;
}
```

html

Copy code

```
<button class="button">Click Me</button>
```

6. Aligning Text within Table Cells

You can also use `text-align` and `vertical-align` for aligning text inside table cells.

Example:

css

Copy code

```
table {
  width: 100%;
  border-collapse: collapse;
}

td {
  text-align: center; /* Center text horizontally in table cells */
  vertical-align: middle; /* Center text vertically in table cells */
  padding: 20px;
  border: 1px solid #ddd;
}
```

html

Copy code

```
<table>
  <tr>
    <td>Row 1, Cell 1</td>
    <td>Row 1, Cell 2</td>
  </tr>
  <tr>
    <td>Row 2, Cell 1</td>
    <td>Row 2, Cell 2</td>
  </tr>
</table>
```

Summary of Key Properties:

- **text-align**: Aligns text horizontally (left, right, center, justify).
- **vertical-align**: Aligns inline elements vertically within their parent container (top, middle, bottom).
- **Flexbox & Grid**: Provide advanced methods for centering and aligning text both horizontally and vertically within containers.