

Alignment basics

Let's explore how to align text and HTML elements using CSS.

Let's first focus on horizontal alignment. Vertical alignment is more difficult so you'll explore that later on.

Text Alignment

Aligning text within an HTML element is very simple. To do this, you use the `text-align` CSS property. In the following example, the CSS rule is setting the text of all paragraph elements to be center aligned.

```
1  p {  
2      text-align: center;  
3  }
```

Text alignment can be set to `left`, `right`, `center` and `justify`.

The `justify` alignment spreads the text out so that every line of the text has the same width.

The default alignment is `left` for languages that are left-to-right such as English. For right-to-left languages such as Arabic, the default alignment is `right`.

HTML Element Alignment

HTML element alignment is more complicated than text alignment. To align HTML elements, you must consider the box model and document flow from previous lessons. Aligning an HTML element is done by changing the properties of its box model and how it impacts the document flow.

HTML Element Center Alignment

To center align an element, you set a width on the element and push its margins out to fill the remaining available space of the parent element as in the following HTML structure:

```
1  <div class="parent">  
2      <div class="child">  
3      </div>  
4  </div>
```

In your CSS, you'll set the `parent` element to have a red border to visualize the space it occupies:

```
1  .parent {  
2      border: 4px solid red;  
3  }
```

The `child` element will have a width equal to 50% of the `parent` element with a padding of 20 pixels. Note that `padding: 20px` is shorthand for setting the padding top, bottom, left and right to `20px`. To visualize the space it occupies, set the border to green:

```
1 .child {
2   width: 50%;
3   padding: 20px;
4   border: 4px solid green;
5 }
```

To align the element to the center, set its `margin` property to `auto`. The `auto` will tell the browser to calculate the margin automatically based on the space available.

```
1 .child {
2   width: 50%;
3   padding: 20px;
4   border: 4px solid green;
5   margin: auto;
6 }
```

The result is the `child` element is centered within the `parent` element:



It is important to note that this works because the `div` element is a block-level element.

If you want to align an inline element like `img`, you will need to change it to a block-level element. Similar to the `div` example, you add the `img` to a parent element:

```
1 <div class="parent">
2   
3 </div>
```

The CSS rule then changes the `img` element to a block-level element and sets its margin to `auto`:

```
1 .child {
2   display: block;
3   width: 50%;
4   margin: auto;
5 }
```

To be more precise, in CSS you can set only the left and right margins to auto. This allows you to set the top and bottom margins to specific values if needed.

```
1  .child {
2    display: block;
3    width: 50%;
4    margin-left: auto;
5    margin-right: auto;
6  }
```

HTML Element Left / Right Alignment

The two most common ways to left and right align elements are to use the `float` property and the `position` property.

The `position` property has several value options that impact how the element displays in the document flow. You'll explore how to use the `position` property later on. For now, let's focus on the `float` property.

The `float` property sets an element's position relative to the text content within a parent element. Text will wrap around the element.

In the following example, the image will be aligned to the right of the `div` element. The text content will wrap around the image:

HTML

```
1  <div class="parent">
2     Lorem ipsum dolor sit amet, consectetur adipiscing e
3  </div>
```

CSS

```
1  .child {
2    float: right;
3  }
```

The following displays in the web browser:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur eu odio eget leo auctor porta sit amet sit amet justo. Donec fermentum quam in diam volutpat, at lacinia diam placerat. Aenean quis feugiat sem. Suspendisse a dui massa. Phasellus scelerisque, mi vestibulum iaculis tristique, orci tellus gravida nisi, in pellentesque elit massa ut lorem. Sed elementum ornare nunc vel cursus. Duis sed enim in nulla efficitur convallis sed eget dolor. Curabitur scelerisque eros erat, in vulputate dolor consequat vel. Praesent ac sapien condimentum, ultricies libero at, auctor orci. Curabitur ut augue ac massa convallis faucibus sed in magna. Phasellus scelerisque auctor est auctor. Nam laoreet sem sapien, porta imperdiet urna laoreet eu. Morbi dolor turpis, congue id bibendum eget, viverra et risus. Quisque vitae erat id tortor ullamcorper maximus.

