COURSEWARE

Alignment

Contents

- Overview
  - Center
    - Align Elements
    - Align Text
    - Image
  - Left and Right align
    - Using position
    - Using float
  - Center Vertically
    - Using Padding
    - Using line-height
    - Using Flexbox
- Tutorial
- Exercises

Overview

In this module, we will be looking at aligning elements using CSS.

Center

Align Elements

To horizontally center a block element (like `<div>`), use `margin:auto;`.

Setting the width of an element will prevent it from stretching out to the edges of its container.
The element will then take up the specified width, and the remaining space will e split equally between two margins:

```
.center {
    margin:auto;
    width:50%;
    border: 3px solid blue;
    padding 10px;
}
```

Note: Center aligning has no effect if the `width` property is not set (or set to 100%)

Align Text

To just center the text inside an element, use `text-align:center;`

```
.center {
    text-align:center;
    border: 3px solid blue;
}
```

Image

To center an image, make it a `block` element and set left and right margin to `auto`.

```css
img {
    display: block;
    margin: 0 auto;
    width: 40%;
}
```

Left and Right align

Using position

One method for aligning elements is to use `position:absolute;`:

```css
.right {
    position: absolute;
    right: 0px;
    width: 300px;
    border: 3px solid blue;
    padding: 10px;
}
```

Note: Absolute positioned elements are removed from the normal low, and can overlap elements.

Using float

Another method for aligning elements is to use the `float` property:

```css
.right {
    float: right;
    width: 300px;
    border: 3px solid blue;
    padding: 10px;
}
```

Note: If an element is taller than the element containing it, and it is floated, it will overflow outside of its container.

Center Vertically

Using Padding

There are many ways to center an element vertically in CSS.
A simple solution is to use top and bottom `padding`:

```css
.center {
    padding: 70px 0;
    border: 3px solid blue;
}
```

To center both vertically and horizontally, use `padding` and `text-align:center`

```css
.center {
    padding:70px 0px;
    border: 3px solid blue;
    text-align:center;
}
```

Using line-height

Another trick is to use the `line-height` property with a value that is equal to the `height` property:

```css
.center {
    line-height: 200px;
    height: 200px;
    border: 3px solid green;
    text-align: center;
}

/* If the text has multiple lines, add the following: */
.center p {
    line-height: 1.5;
    display: inline-block;
    vertical-align: middle;
}
```

Using Flexbox

You can also use flexbox to center things.

Just note that flexbox is not supported in IE10 and earlier versions:

```css
.center {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 200px;
  border: 3px solid green;
}
```

Tutorial

There is no tutorial for this module.

Exercises

1. Position the `<div>` element all the way to the right using absolute positioning

2. Center align the `<div>` using margins

```html
<!DOCTYPE html>
<html>
<head>
    <style>
        div {
            width: 300px;
            background-color: #b0e0e6;
        }
    </style>
</head>
<body>
    <div>
        <p>In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.</p>
        <p>'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'</p>
    </div>
</body>
</html>
```

▼ Solution 1.

```css
div {
    position: absolute;
    right: 0px;
    width: 300px;
    background-color: #b0e0e6;
}
```

▼ Solution 2.

```css
div {
    margin-left: auto;
    margin-right: auto;
    width: 300px;
    background-color: #b0e0e6;
}
```

```css
div {
    margin-left: auto;
    margin-right: auto;
    width: 300px;
    background-color: #b0e0e6;
}
```