

Advanced CSS3



chapter 5 / 7

Advanced CSS3

Sections in this chapter:

1. Layout
2. Flexbox

5-1. Layout

Now we'll take our nose off the floor and start looking at the bigger picture. How do we **lay things out** in CSS?

5-1-1

To start our journey, let's look at a few different solutions to something that's really hard to do in CSS - **centering stuff vertically**!

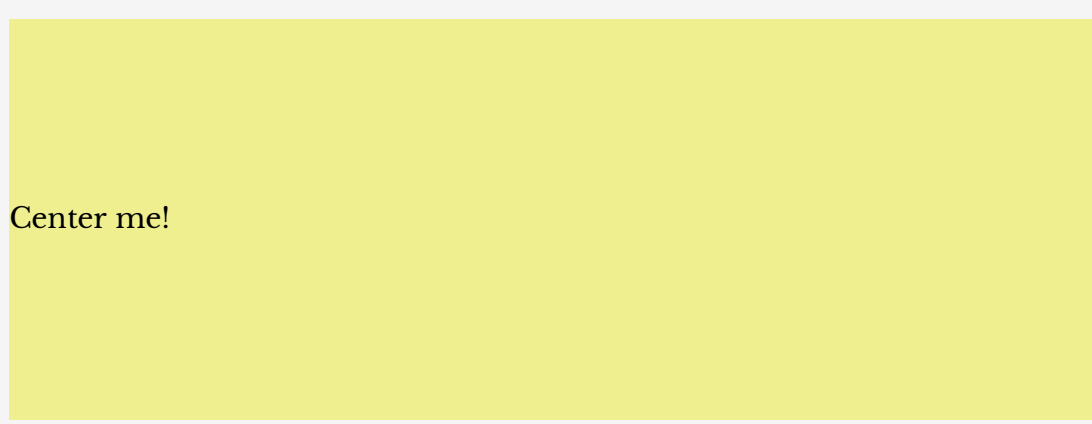
5-1-2

Variant 1/6:

5-1-3

```
<section>
  <article>Center me!</article>
</section>
```

```
section {
  height: 100px;
  background-color: #8799db;
  display: table;
  width: 100%;
}
article {
  background-color: #fef8f8;
  display: table-cell;
  vertical-align: middle;
}
```



Center me!

vertical-align is only respected inside inline and table-cell elements. It can't be used to vertically align block elements.

Variant 2/6:

5-1-4

```
section {  
  height: 100px;  
  background-color: #8799db;  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
}  
article {  
  background-color: #fef8e7;  
}
```



We can use flex!

Variant 3/6:

5-1-5

```
section {  
  height: 100px;  
  background-color: #8799db;  
  position: relative;  
}  
article {  
  background-color: #fef8e7;  
  position: absolute;  
  top: 50%;  
  transform: translateY(-50%);  
}
```



We can use transform to move the block element in the y-direction.

Variant 4/6:

5-1-6

```
section {  
  height: 100px;  
  background-color: #8799db;  
  line-height: 100px;  
}  
article {  
  background-color: #fef8f8;  
  display: inline-block;  
  vertical-align: middle;  
}
```




Setting the line-height to a value that is **greater** than the font-size will by default distribute the space above and below the text equally.

Variant 5/6:

5-1-7

```
section {  
  height: 100px;  
  background-color: #8799db;  
  position: relative;  
}  
article {  
  background-color: #fef8f8;  
  display: inline-block;  
  height: 40px;  
  position: absolute;  
  bottom: 0;  
  top: 0;  
  margin: 0 auto;  
}
```




Center me!

Will only work with set height on the child element.

Variant 6/6:

5-1-8

```
section {  
  height: 100px;  
  background-color: #8799db;  
}  
section:before {  
  content: ' ';  
  display: block;  
  height: 50%;  
  margin-bottom: -20px;  
}  
article {  
  background-color: #fef8e7;  
  display: inline-block;  
  height: 40px;  
}
```



Center me!

This also needs a set child element height.

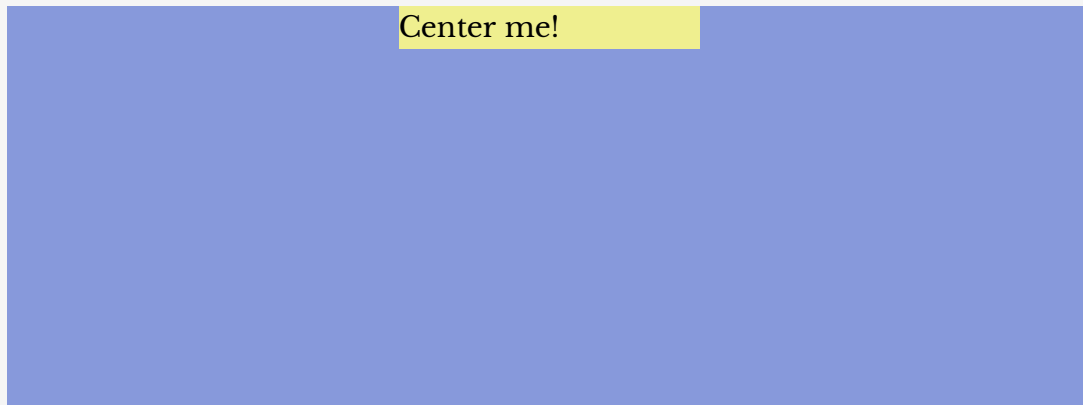
Centering stuff **horizontally** is easier, but still worth exploring:

5-1-9

Variant 1/4:

5-1-10

```
section {  
  height: 100px;  
  background-color: #8799db;  
  text-align: center;  
}  
article {  
  background-color: #fefef8;  
  display: inline-block;  
  width: 150px;  
  text-align: left;  
}
```

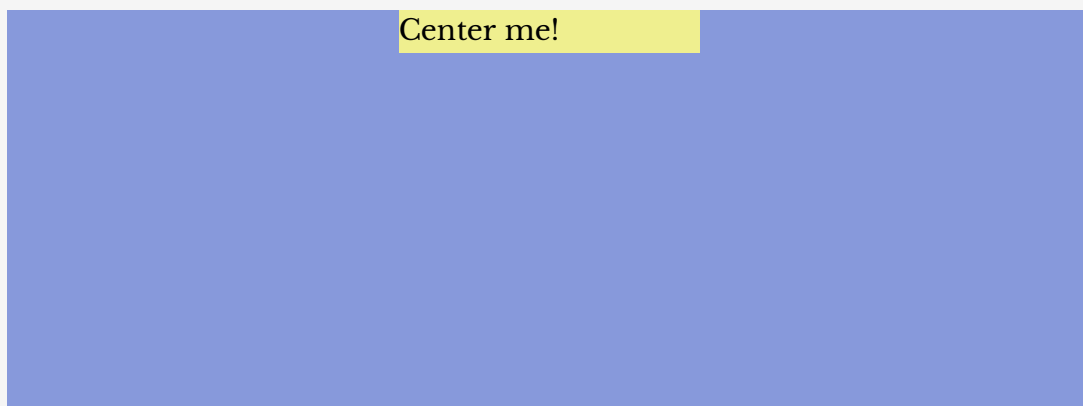


Easiest is to make block text centered.

Variant 2/4:

5-1-11

```
section {  
  height: 100px;  
  background-color: #8799db;  
}  
article {  
  background-color: #fefef8;  
  width: 150px;  
  margin: 0 auto;  
}
```

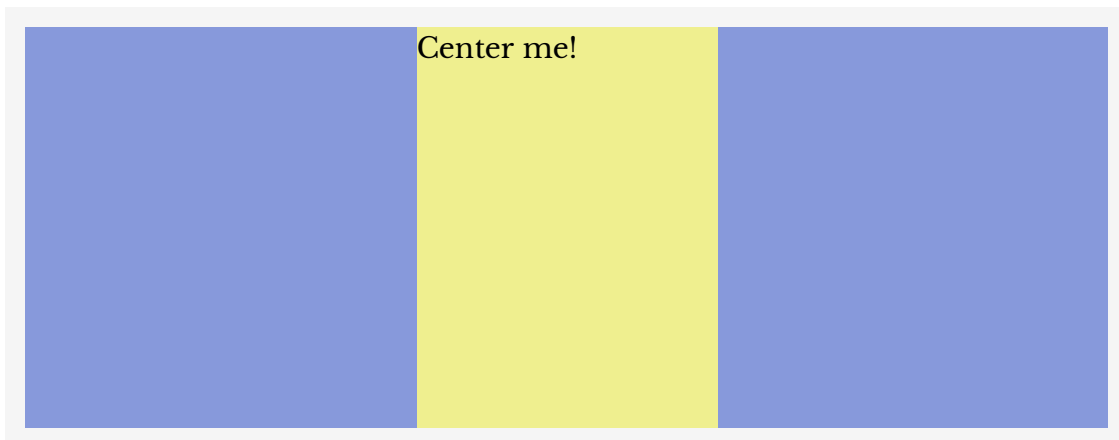


We can also do auto margins. These won't work without fixed width.

Variant 3/4:

5-1-12

```
section {  
  height: 100px;  
  background-color: #8799db;  
  display: flex;  
  justify-content: center;  
}  
article {  
  background-color: #fefef8;  
  width: 150px;  
}
```



We can use flex!

Variant 4/4:

5-1-13

```
section {  
  height: 100px;  
  background-color: #8799db;  
  position: relative;  
}  
article {  
  background-color: #fefef8;  
  width: 150px;  
  position: absolute;  
  left: 50%;  
  transform: translateX(-50%);  
}
```



Here we are using transform to move the block element in the x-direction.

We also need to discuss the concept of **floats**:

5-1-14

Variant 1/3:

5-1-15

```
<figure>
  
</figure>
<section>The fox jumped over the little stream.</section>
```

```
figure img {
  display: inline-block;
}
section {
}
```



The fox jumped over the little stream.

The logo is an inline block.

Variant 2/3:

5-1-16

```
figure img {
  display: inline-block;
  float: left;
}
section {
}
```



The fox jumped over the little
stream.

Now note how all lines wrap around it!

Variant 3/3:

5-1-17

```
figure img {  
  display: inline-block;  
  float: left;  
}  
section {  
  clear: left;  
}
```



The fox jumped over the little stream.

We can expressively clear our way below floats.

5-2. Flexbox

There are **two common problems** that are embarrassingly hard to solve using regular CSS:

5-2-1

- **centering things vertically**
- **having columns of equal height**

CSS3 finally brought us the remedy: **Flexbox**!

The basic idea is simple: in this **new display mode** you...

5-2-2

- opt in to the **new display mode** (`display: flex`)
- choose **vertical or horizontal direction** (`flex-direction`)
- choose how the **content should be aligned along the direction axis** (`justify-content`)
- and also **perpendicular to the direction axis** (`align-items`)

There are **some more flex properties**, and also some **for the children** in a box, which gives you even more control.

5-2-3

This is **hard to visualize from textual explanations**, so try out an interactive tutorial here:

5-2-4

<http://blog.krawaller.se/flexboxdemo/>

There are also two popular games that demonstrate the concept:

5-2-5

- Flexbox Froggy @ <http://flexboxfroggy.com/>
- Flexbox Defense @ <http://www.flexboxdefense.com/>