

Auto-margin Alignment

Beware of `margin: auto` alignment on flex items.

When you use `margin: auto` on flex-items, things can look quite weird. You do need to understand what's going on. It may result in unexpected results, but I'm going to explain all that.

When you use `margin:auto` on a flex-item, the direction (left, right or both) that has the value `auto` will take up any empty spaces available.

That's a difficult one to catch. Here's what I mean:

Consider the navigation bar marked up and styled below:

```
<ul>
  <li>Branding</li>
  <li>Home</li>
  <li>Services</li>
  <li>About</li>
  <li>Contact</li>
</ul>
```

```
ul {
  display: flex;
}
li {
  flex: 0 0 auto;
}
```

Here's what we've got:

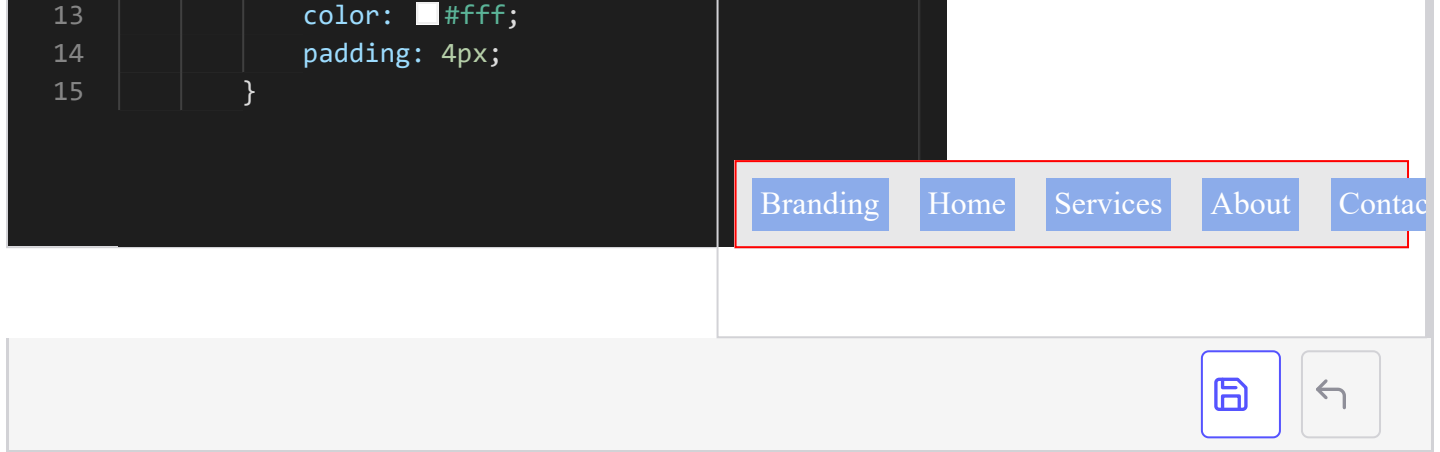
HTML

CSS

Output

```
1  ul {
2      display: flex;
3      border: 1px solid red;
4      padding: 0;
5      list-style: none;
6      background-color: #e8e8e9;
7  }
8
9  li {
10     flex: 0 0 auto;
11     background-color: #8cacea;
12     margin: 8px;
```

output



There are a couple of things to note here. So listen up!

1. The `flex-grow` value is set to zero. This explains why the list items don't grow
2. The flex-items are aligned to the start of the main-axis (the default behavior)
3. Owing to the items being aligned to the start of the main-axis, some extra space is left on the right. You see that?



Now use `margin-right: auto` on the first list item (branding) and see what happens.

```
li:nth-child(1) {  
    margin-right: auto; /*applied only to the right*/  
}
```



```

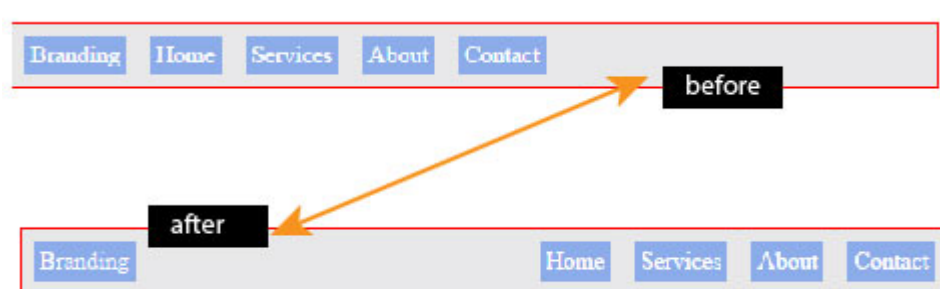
12
13 li {
14     flex: 0 0 auto;
15     background-color: #8cacea;
16     margin: 8px;
17     color: #fff;
18     padding: 4px;
19 }

```

Branding Home Services About Contact



What just happened? The extra space that existed has now been distributed to the right of the first flex-item.



Remember what I said earlier? When you use `margin:auto` on a flex-item, the direction (left, right or both) that has the value `auto` will take up any empty spaces available. It doesn't end with just one side. What if you wanted a margin auto alignment on both sides of a flex-item?

```

/*you may use the margin shorthand to set both sides of you wish*/
li:nth-child(1) {
    margin-left: auto;
    margin-right: auto;
}

```



HTML

CSS

Output

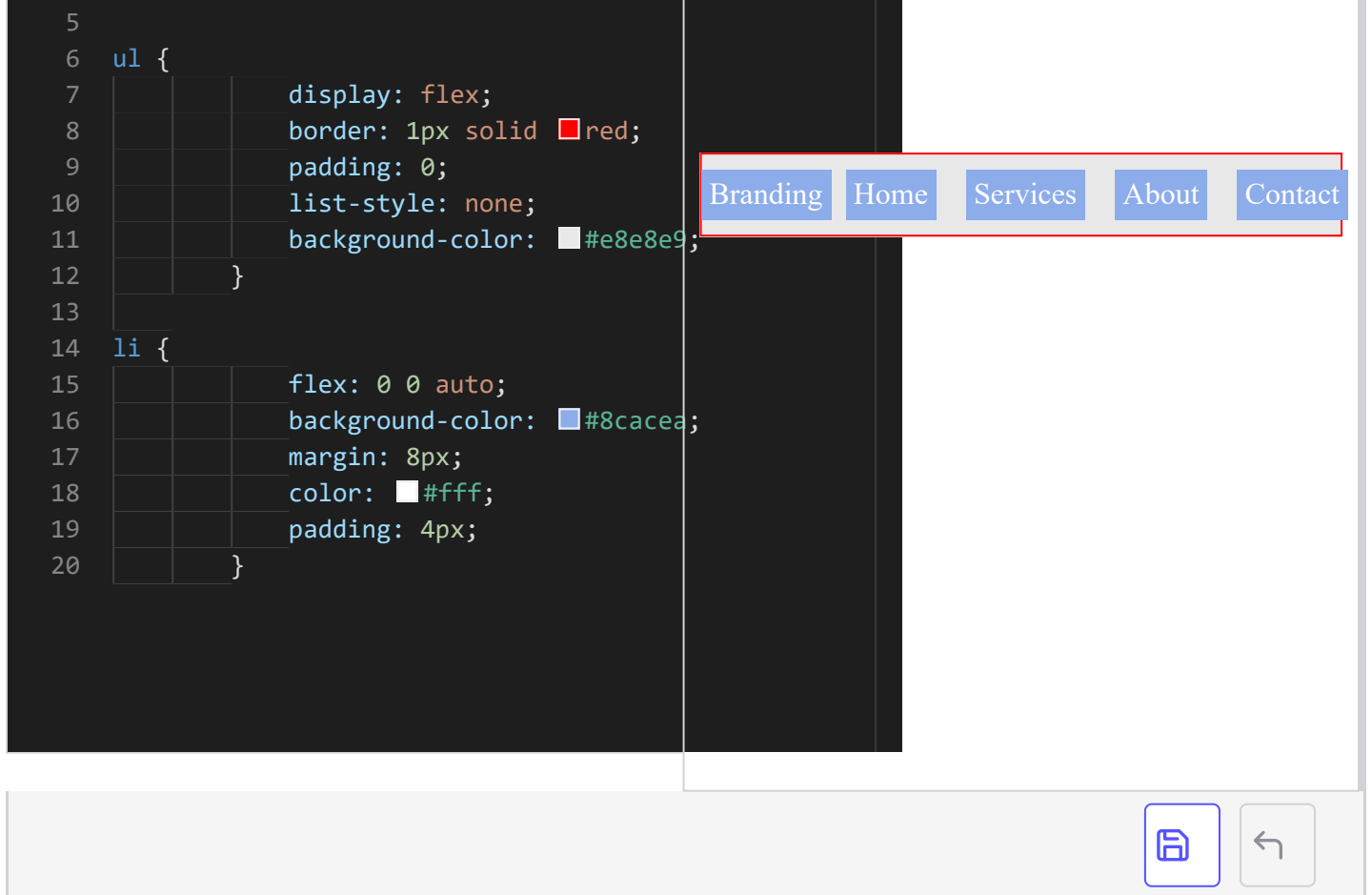
```

1 li:nth-child(1) {
2     margin-left: auto;
3     margin-right: auto;
4 }

```

CSS

output

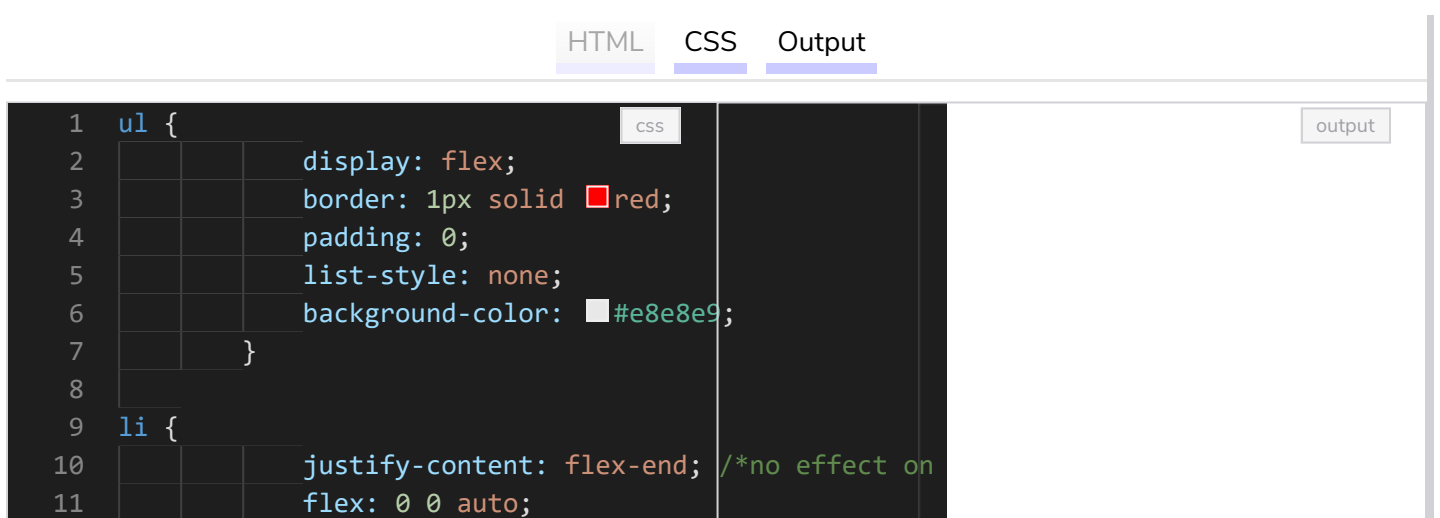
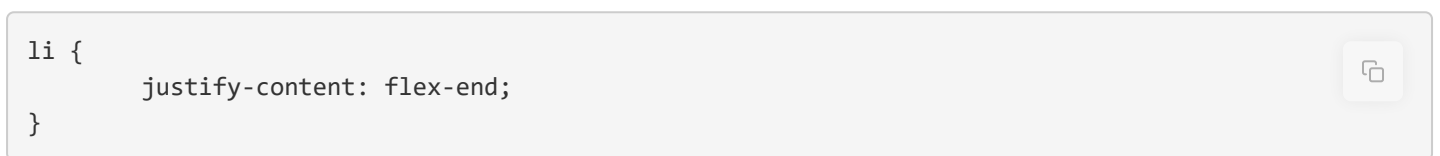


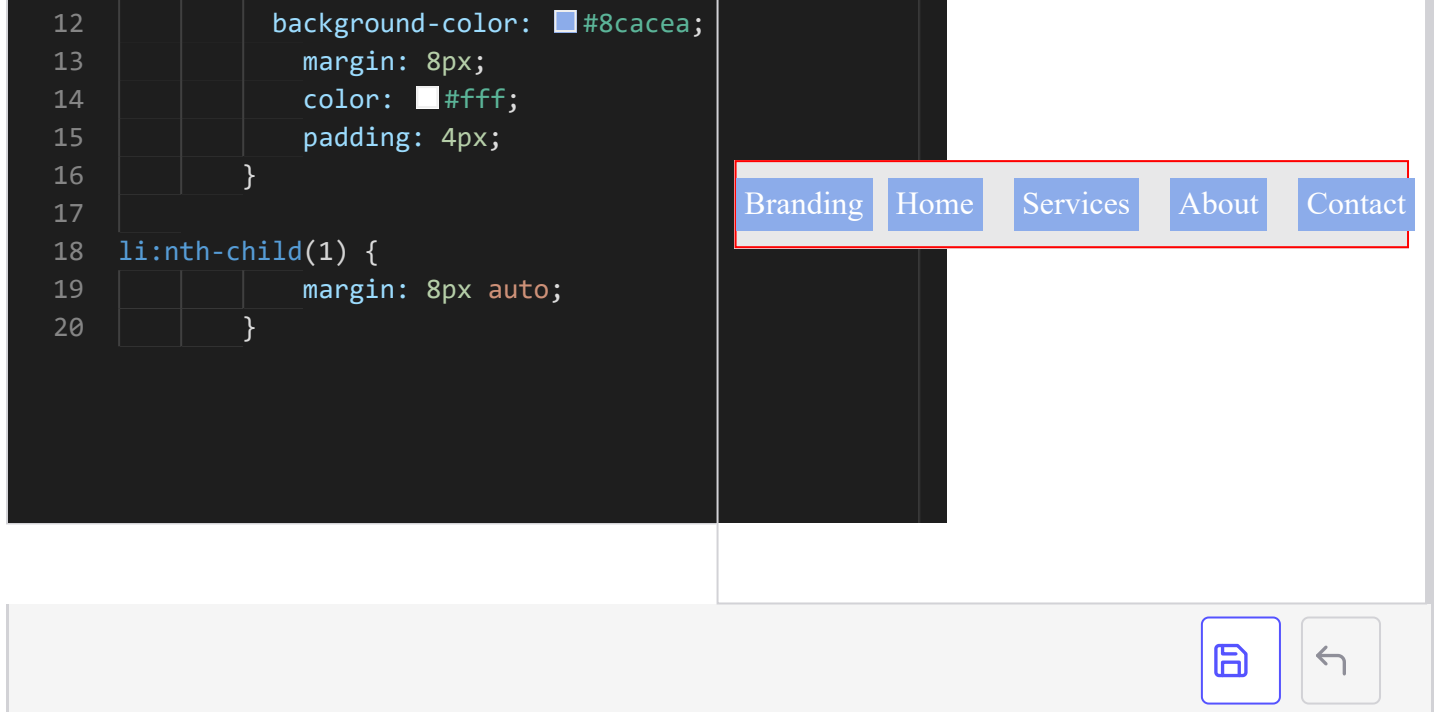
Now the space is distributed across both sides of the flex-item.

So, is there a trade off with the cool auto-margin alignment? It appears there's one. It can be a source of frustration if you don't pay attention too.

When you use the auto-margin alignment on a flex-item, the `justify-content` property no longer works on the flex-items. It just doesn't work.

For instance, setting a different alignment option on the flex-items above via the `justify-content` property, has no impact on the layout.



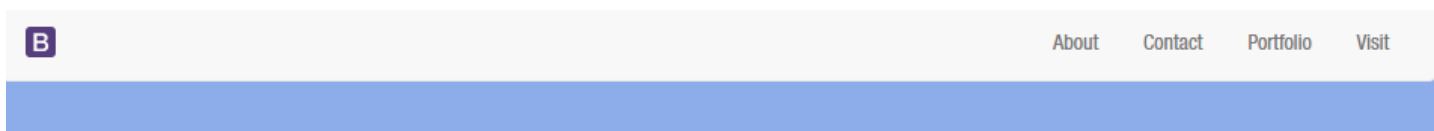


Practical Use cases

Navigations are a very big part of every website or application. Almost every website on the planet is got some sort of navigation system in place.

Take a look at these popular sites and how they approach their navigation systems. Do you see how flexbox can help you build these layouts more efficiently? Take a closer look to see where the auto-margin feature can come in very handy too.

1. Bootstrapped Navigation



2. AirBnB desktop Navigation



3. Twitter desktop Navigation



If you're feeling good, I advice you actually write codes. Try implementing the navigation systems yourself. You've got all the knowledge you need now. A bit of courage to start writing is all you need.

See you in the next lesson. Hopefully after you've completed the navigation system exercises :-)