

Responsive Design with Flexbox

Books have been written on responsive design, good books at that.

Since this course focuses on the Flexbox model, I wouldn't be taking a deep plunge into the general state of responsive designs.

Like I stated somewhere earlier, we do get some responsiveness out of the box with the flexbox model.

Flexbox as in “flexible box”. However, it's possible to target various screen sizes via media queries and then change the flex behavior.

Here's an example.

The handy unordered list comes to the rescue again:

```
<ul>
  <li>Home</li>
  <li>About</li>
  <li>Contact</li>
  <li>Register</li>
  <li>Login</li>
</ul>
```



and with a bit of styling:

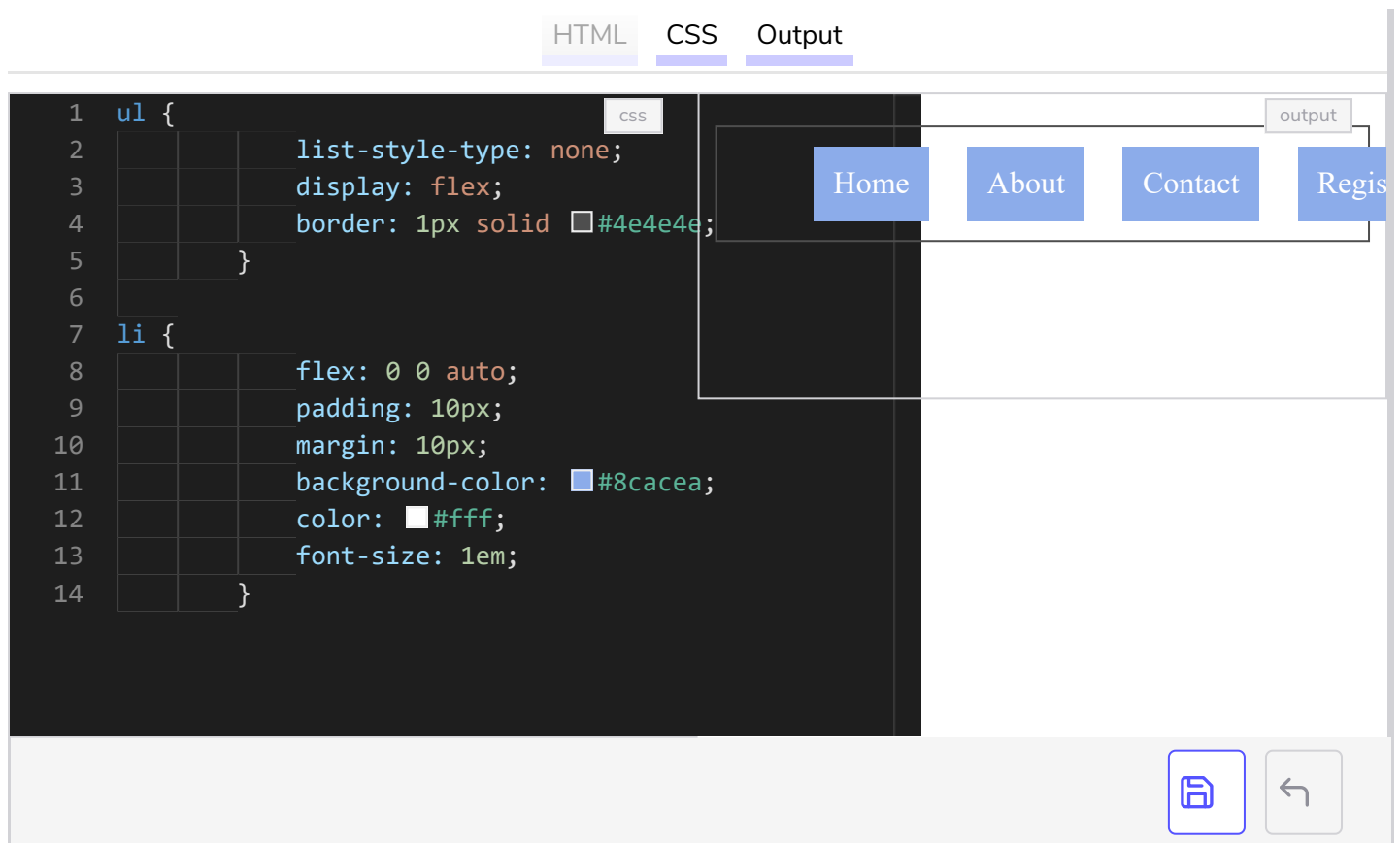
```
ul {
    list-style-type: none;
    display: flex;
    border: 1px solid #4e4e4e;
}

li {
    flex: 0 0 auto;
    padding: 10px;
    margin: 10px;
    background-color: #8cacea;
    color: #fff;
    font-size: 1em;
}
```



You're a pro at this flex stuff now so you understand what's going on up there.

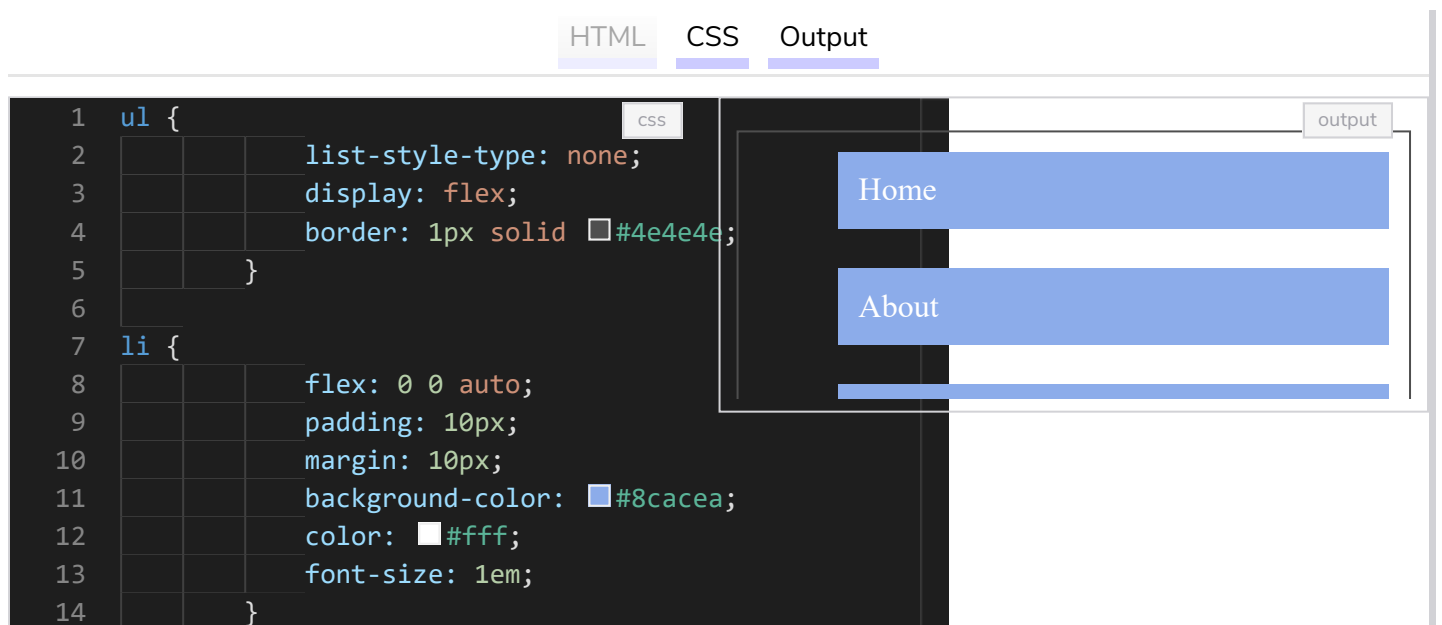
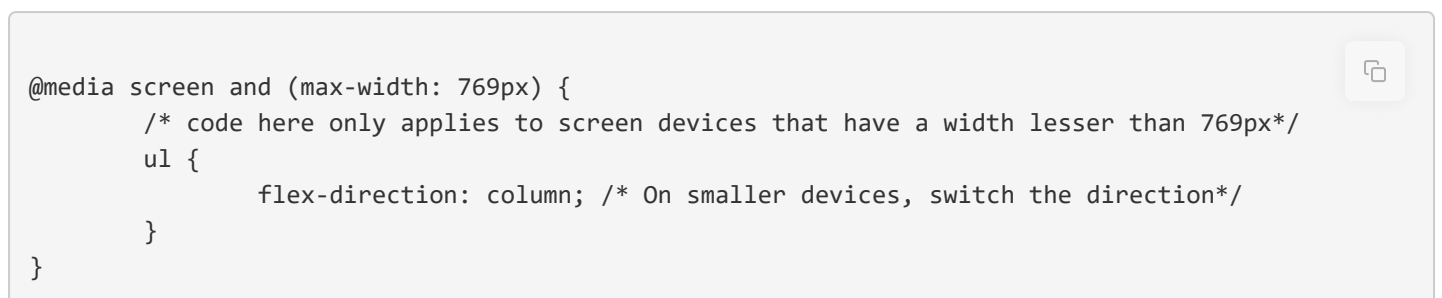
So, here's how the navigation bar looks.



While this is cool for desktops and tablets, at certain screen sizes it particularly doesn't look good.

On mobile, you'd want to stack the nav items vertically.

Then comes in media queries



```

15
16      /*=====
17      Resize browser to see at work
18      =====*/
19
20 @media screen and (max-width: 769px) {
21   ul {
22     flex-direction: column;
23   }
24 }

```



If you knew a few things about responsive designs before now, that's great.

Just transpose the Flexbox model unto your existing knowledge and you're good to go.

By the way, I made the assumption that you understand what media queries are. If you don't, see the quick brief below.

Media Queries

Media queries are at the heart of responsive design. They let you target specific screen sizes and specify codes to be run on the devices alone.

The most popular form in which media queries are used is something called the @media rule.

It looks like this:

```

@media screen and (max-width: 300px) {
  /*write your css in this code block*/
}

```



Looking at it, you can almost guess what that does.

“For a screen device with a maximum width of 300px ... do this and that ”

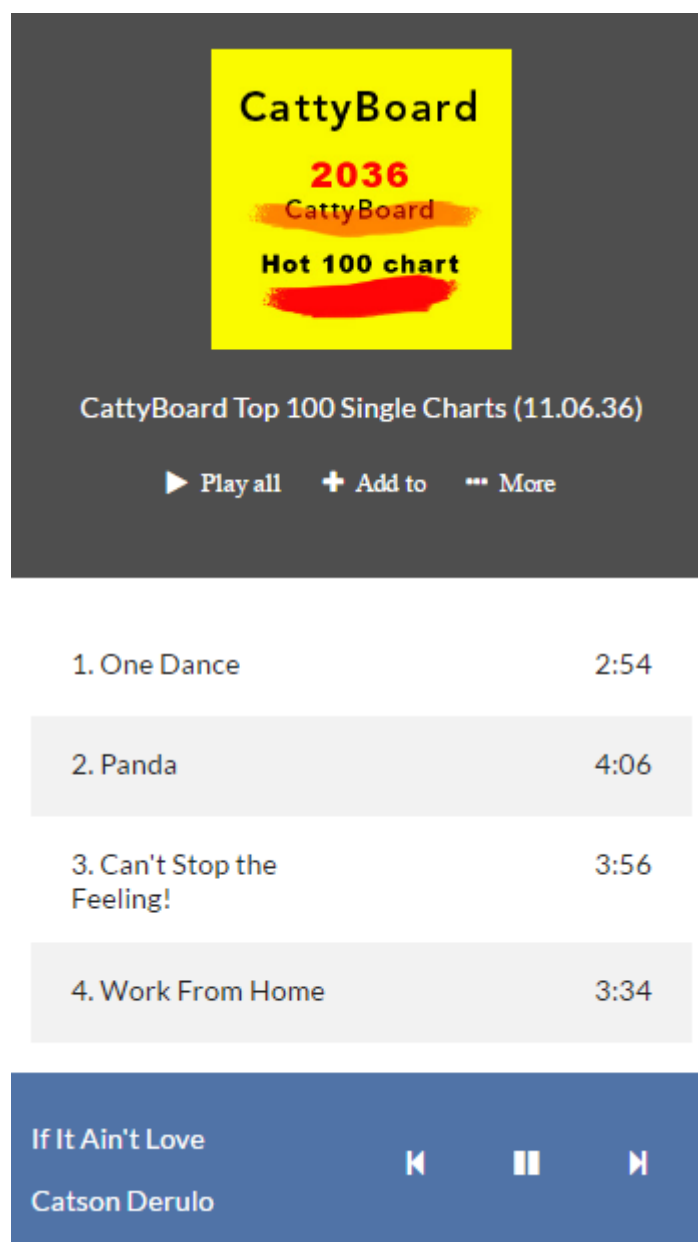
Any styles within the code block will only apply to devices that match the

expression, `screen and (max-width: 300px)`

I guess that helped clear up some confusion.

Quick Exercise

Catty music is displayed differently on mobile devices. That's great news. What's even better is you should try to recreate this.



You're almost at the end!

In the concluding lesson, I'll discuss browser support, helpful links and resources to get you moving. I'm sure you'd love it.