

<b>Time</b>	~ 1 hour
<b>Learning Goals</b>	<ul style="list-style-type: none"> <li>• Practice HTML and CSS</li> <li>• Learn about SASS</li> <li>• Learn about mobile responsive layout (widths, @media screens etc.)</li> </ul>

## Layout Challenge: Navigation Bar

Building a responsive web app is critical to providing good user experience for your app across various screen sizes. Let's take a short break from the URL shortener and work on the given repo first. We will get back to your URL shortener in the next challenge.

### Start Challenge

1. Fork the repo from: <https://github.com/NextAcademy/airbnb-frontend>
2. Clone the repo and complete the responsive challenges below. Only modify the html views and `application.scss` pages.

*You're strictly not allowed to use any bootstrapping frameworks here!*

## Frontend and Responsive Challenges

### 1. Learn about SASS

Take a peek at the Gemfile. As you may have noticed, Rails by default installs the `sass` for you: `gem 'sass-rails', '~> 5.0'` A badly structured CSS file can result in future users of your code getting confused and lost in the sea of spaghetti CSS code! As our web pages and apps get more complex, our stylesheets get longer and harder to maintain.

CSS preprocessors like Sass help by keeping our style sheets concise and allowing us to modularize our code while offering a whole slew of features such as assigning variables, allow nesting and utilisation of mixins.

*For this exercise, read briefly on how [Sass's nesting tool](#) works, as you'll need it for your next exercise.*

### 2. Add viewport

A viewport is the visible area of a webpage and varies depending on the device, e.g. it would be smaller on a mobile vs. a tablet or a desktop screen.

Including the `<meta tag>` in our html `<head>` for viewport will allow web developers to take control of the screen size. Click on `inspect element` and the mobile view icon next to the

magnifying glass icon to see how our web app would look in a mobile screen. Now add the follow `<meta tag>` in our `<head>` and check out the difference.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

What should I do if I want to allow user to zoom in at small screen? What should I do if I *don't* want users to zoom in at small screen?

### 3. Shrink large text in small screen

Ewww, the `PAIR BNB` header looks hideous in small screen. Use the [@media rule](#) to reduce it's font size in smaller screen.

### 4. Adjust columns in small screen

Using `@media` and `width`, set the `About Us` and `Why use us` paragraphs to full-width on small screen (400px).

### 5. Hide items in small screen

Hmmm, our navbar ain't looking so great in small screen. Ideally the navbar should be converted into a condensed dropdown for users to navigate around the site. For now, hide some of the links on small screen so it doesn't clutter up our view. *Hint:* look at some of the [display](#) functions, can any be used to hide a link?

### Submit Challenge

Congratulations! You now have a better idea on how to make your website responsive! Push your completed challenge to your repo to submit challenge. Feel free to implement what you have learnt in your URL shortener and PairBnB app later.