Using the Exercise Files

Accessing the Example Files

Most of the lectures in this course will have you work with JavaScript code. For each lecture of this nature we will usually provide both a set of starting files and a set of finished files.

The starting files can be used by you to follow along and to try things out on your own. I would encourage that you do that.

The finished files contain the same set of files, but it shows what they look like at the end of that lecture. This allows you to check your own work, or replace your own code if you have made a mistake and you are not sure how to fix it.

When a lecture comes with downloadable files, you will see the link beneath the lecture. Unzip the file. For lectures that have both a set of starting files and a set of finished files, you will find two folders. The name of the folder for the starting files will end with "start". The name of the folder for the finished files will end with "final". Some lectures may only contain starting or finished files because that is all that is needed for that lecture.

A Note on the Use of var vs. let

The question came up recently as to why in the vast majority of the coding examples I use var instead of let. Well, the simple answer is that I do it for backwards compatibility.

Yes, it is true that there is only a small percentage of browsers that are still being used that do not support let, but I still find myself in 2017 doing work for clients that requires me to support those browsers. Therefore, I use var in many of my examples.

However, you should know that things are moving away from var. So in these examples, you could replace var with let.

If you need a review of the difference, you can watch this tutorial from my YouTube channel: https://youtu.be/9_10SWEW3-g

Now what about const? As a reminder, const prevents that reassignment of the variable. That can be very valuable as it can prevent a variable from being reassigned. So why don't I use const more in my examples? The reason is I feel it can mis-communicate. Because so much in JavaScript relies on objects, if const is used to declare those objects, it will only prevent the reassigning of an object, not the changing of it. (We address techniques later in the course that can be used to prevent the changing of an object.)

For example, if I declare the following:

```
const obj = {
```

```
    name: "Steve"
}
```

The JavaScript engine prevents me from doing this:

```
obj = {};
```

But I can still do this:

```
obj.name = "James";
```

So const is valuable, but make sure you are aware of its limitations. With primitives it truly acts as a constant. But not necessarily with objects.

Keep in mind as you move through the examples, my intent is not to communicate the best use of var. And I hope it doesn't distract you from learning what the example is really trying to teach.