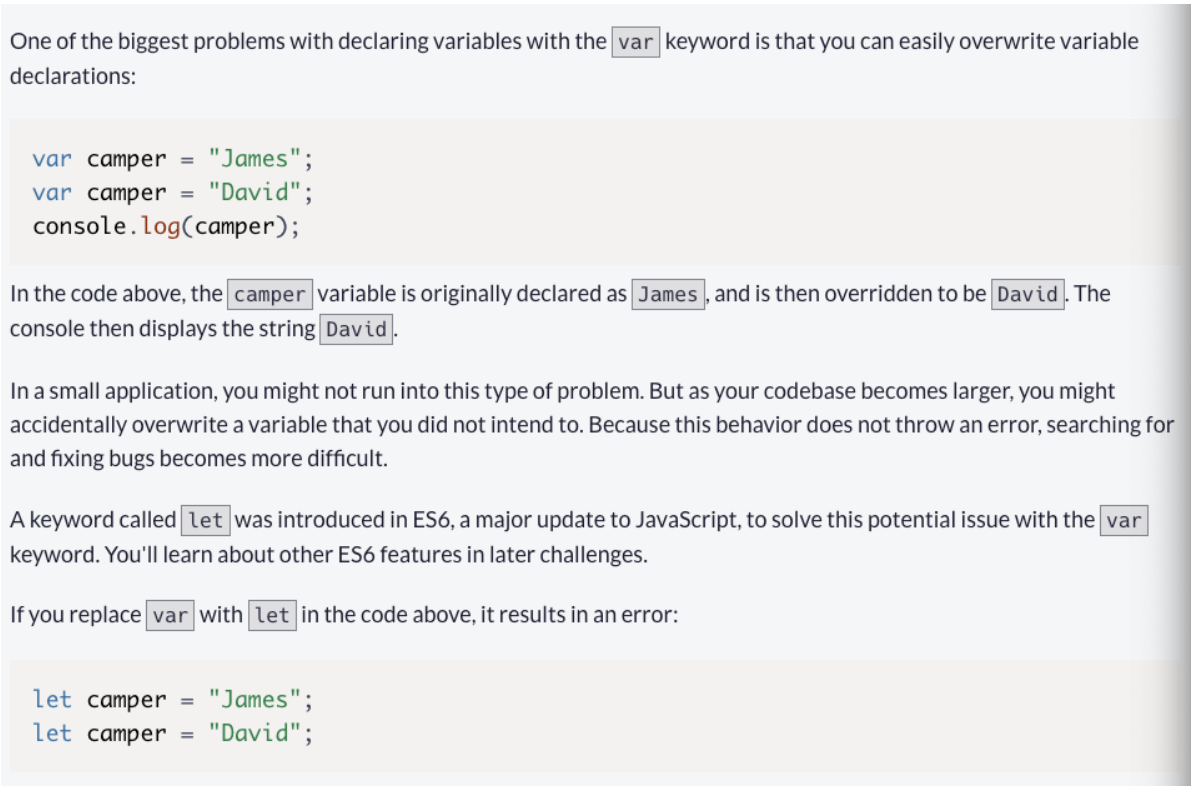
Practice

Contains concepts from practice question solving.

1. Initially the code used to run in the “var” format, but since using var keyword, the variables can be overwritten, hence “let” was introduced to remove this overwriting nature in JS.
   This was going to be an issue in Big codebases. Hence in ES6 documentations, “let” was introduced.



One of the biggest problems with declaring variables with the `var` keyword is that you can easily overwrite variable declarations:

```
var camper = "James";
var camper = "David";
console.log(camper);
```

In the code above, the `camper` variable is originally declared as `James`, and is then overridden to be `David`. The console then displays the string `David`.

In a small application, you might not run into this type of problem. But as your codebase becomes larger, you might accidentally overwrite a variable that you did not intend to. Because this behavior does not throw an error, searching for and fixing bugs becomes more difficult.

A keyword called `let` was introduced in ES6, a major update to JavaScript, to solve this potential issue with the `var` keyword. You'll learn about other ES6 features in later challenges.

If you replace `var` with `let` in the code above, it results in an error:

```
let camper = "James";
let camper = "David";
```

2. Just after let, “const” was defined:

Declare a Read-Only Variable with the const Keyword

The keyword `let` is not the only new way to declare variables. In ES6, you can also declare variables using the `const` keyword.

`const` has all the awesome features that `let` has, with the added bonus that variables declared using `const` are read-only. They are a constant value, which means that once a variable is assigned with `const`, it cannot be reassigned:

```
const FAV_PET = "Cats";
FAV_PET = "Dogs";
```

The console will display an error due to reassigning the value of `FAV_PET`.

You should always name variables you don't want to reassign using the `const` keyword. This helps when you accidentally attempt to reassign a variable that is meant to stay constant.

3. Everything to the right of the assignment sign = is evaluated first.