Functions

Learning Objectives

- How can we write good functions?
- What's a pure function?
- Coding Style/Conventions
- … What else?

Professional Code

SRP

DRY

Tests

Conventions

Pure Functions

Comments

Naming conventions

- Why have conventions?
- What makes good function name?
- Booleans – Prefix?
- Arrays – Plural or Arr Suffix?

Functions Recap

- We don’t want to repeat ourselves!
  - We can write a function ONCE and use it as many times as we like, with as many different parameters as we like

- It also helps to document our code
  - Not only us reading our code
  - Easier to fix bugs

What makes a good function

A good functions is made up of the following…

- Descriptive name
- Avoid globals
- Stick to a strict coding style
- Testable?
- Comment as much as needed but not more
- Use shortcut notation when it makes sense
- Modularize – one function per task

How big is too big?

- So when do I start breaking don my functions…. After 10 lines, 15 lines….?
  - There is no hard and fast rule on the size of function (max number of lines), but if you feel a function getting large, take a step back and go back to the most important rule
    - Is my function still doing just one job?

What is a pure function?

- Pure Functions
  - Given the same input, will always return the same output.
  - Produces no side effects.
  - … Examples of impure functions?

- Pure functions have many beneficial properties, and form the foundation of functional programming.
- Pure functions are completely independent of outside state
  - They are therefore immune to entire classes of bugs that have to do with shared mutable state.

```
const pureFunction = (num1, num2) => {
  const sum = num1 + num2;
  return sum;
}
```

Types of Functions

- ES5 vs ES6
  - ES5 uses the function key word
  - Can be used before initiated
    - All functions are ‘hoisted’ to the top of the JS file

```
function testFunction (number) {
  console.log(number)
}
```

  - ES6 is written as an arrow function
  - Cleaner looking and this is the one we’ll be using on the course
    - These are not hoisted, you must declare the function above wherever you have invoked it in the JS file

```
const testFunction = (number) => {
  console.log(number)
}
```