Working With Fat Arrows In ES6 Activity

#### **Learning Outcomes Addressed**

1. Create and use anonymous functions in JavaScript
2. Identify the different JavaScript syntax variations

Follow the instructions in Next Tech to complete the activity.

### Arrow Function - ES6

JavaScript ES6 has new syntax and features to make your code more modern and readable. It allows you to write fewer lines of code and accomplish more.

"Fat arrow functions" (called fat arrows because -> is a thin arrow and => is a fat arrow) are one of these features. They provide a new way to write concise functions. Here's an example of how writing a function has changed:

**ES5:**

function calcTip(amount) {

return amount+(amount\*.2);

}

calcTip(10)

12

**ES6:**

let tip = amount => amount+(amount\*.2)   
   
tip(10)  
12

You'll notice a few things in these examples: The ES6 code is shorter, it didn't require the word function at the start, and the curly braces and the return statement are omitted.

However, this only works when the fat arrow function is one line. If you had a more complex function and needed to break it up into a multiline statement, you do need to use curly braces and include a return statement, like so:

let tip = amount => {  
    return amount+(amount\*.2);  
}  
   
tip(10)  
12

You can explore more ways in which the ES6 syntax is different by going through the [ES6 documentation](http://es6-features.org/).

**Your task in this activity is to convert the functions in the arrowfunction.js file into ES6 functions.**

Task

convert the function in the arrowfunction.js file into ES6 functions

function addFive(num) {

  return num + 5;

}

// TODO: Convert addFive() to arrow function

function divideBy(x, y) {

  return x / y;

}

// TODO: Convert divideBy() to arrow function

function logToConsole() {

  let msg = "MIT Coding Certificate";

  console.log(msg);

}