Trainer guide
Examples and exercises
JS-2 – Arrow functions

Contents

JS-2a | Exercise: Arrow functions

Link to environment

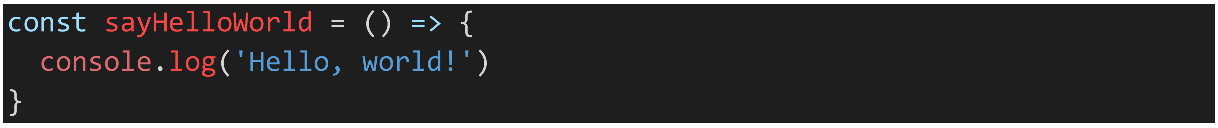
This exercise gives learners practice on converting traditional JavaScript functions into arrow functions.

index.js

- In this exercise, you'll practice converting regular functions into arrow functions.
- It's important to understand the arrow syntax because arrow functions are used all the time in JavaScript. They are a very concise way of expressing functions. We'll see arrow functions in practice in a later exercise.

Demo

- Let's convert this first function, `sayHelloWorld()`, into an arrow function.
    - Firstly, when we write an arrow function, we don't use the `function` keyword. We can, however, declare an arrow function like any other variable using `let` or `const`.
        - *Replace "`function sayHelloWorld()`" with "`const sayHelloWorld = `"*
    - Now let's write the arrow function itself.
    - The general syntax for an arrow function is:
        - A pair of parentheses
            - *After the equals sign, write "()"*
            - Inside the parentheses is where you place the function parameters. In this case, there are no parameters, so we will leave the brackets empty.
        - Followed by an arrow
            - *Write "=>"*
            - The arrow is formed of an equals sign followed by a greater-than sign.
        - Followed by the body of the function.
            - *Ensure the function body is present.*
- This code is now functionally equivalent to the version we started with.



```
const sayHelloWorld = () => {
  console.log('Hello, world!')
}
```

Exercise

- Your task is to convert the functions in index.js to arrow functions.

Solve the exercise live before moving on.

Solution reference

Worth mentioning

If an arrow function only has a single parameter, then the parentheses around that parameter are sometimes omitted. This is relevant to the second example.