

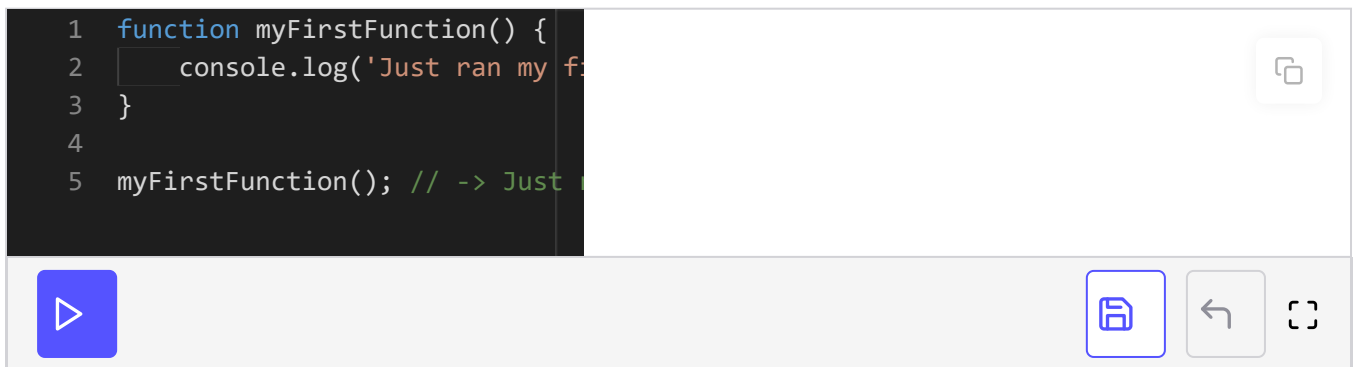
# Introduction to Functions

Learn one of the fundamental building blocks of all languages: functions. Learn how to write a function, why they're useful, and what we can do with them.

We now get to one of the topics that new programmers find most difficult. Once the understanding sinks in, however, it'll feel trivial. You'll write functions pretty much every single day that you write code.

Let's show some code and then discuss what's happening.

```
1 function myFirstFunction() {  
2   console.log('Just ran my f  
3 }  
4  
5 myFirstFunction(); // -> Just
```



## Creating a Function

To write a function, we start with the `function` keyword. We write the name we want to give the function after it. We write a pair of parentheses after that.

```
function functionName()
```

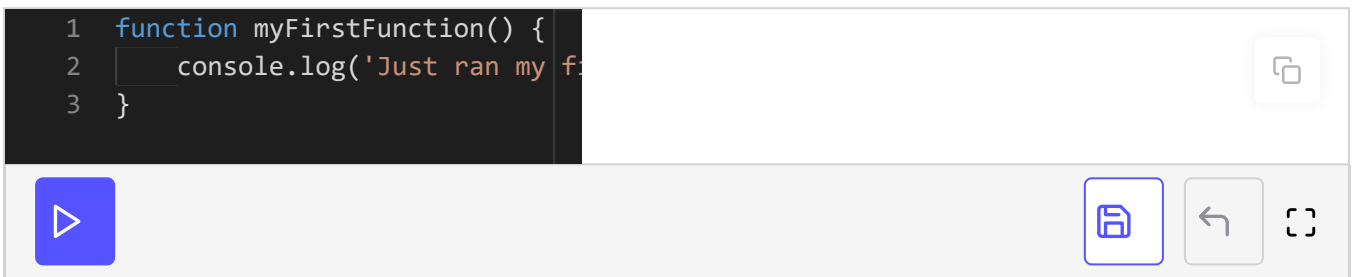
Function naming rules are the same as variable naming rules. Once we create a function, the function name can be treated like a normal variable.

We then write a pair of curly brackets `{ }`. Inside these curly brackets, we put some code. The code inside those brackets is called the **function body**.

```
function functionName() {  
    // function body  
    // some code here  
}
```

When the engine is running our code, nothing inside the function body runs until we do something special. If all we have in a file is the function `myFirstFunction` from above, nothing will happen. See for yourself.

```
1 function myFirstFunction() {  
2   console.log('Just ran my f  
3 }
```



The code inside the function doesn't run.

To make the code in the function body run, we have to **call** the function. **Invoke** is another word for call. This is done by writing the function name with a pair of parentheses after it.

```
function functionName() {  
  // some code here  
}  
  
functionName();
```

Reference the first code block in the article to see this in action.

## Why They're Useful

The value of functions is that they reduce the total amount of code that we have to write and copy. This is because we can call a function as many times as we like.

If there's some code that needs to be run multiple times, instead of copying the code over and over, we can put that code in a function. We can then simply call the function whenever we need to.

If we were copying our code and some piece of it needed to be changed, we'd have to change it in every place that we copied the code.

If the code is in a function, we just have to change it in that one location.

Here's a simple example.

```
function greet() {  
  console.log('Hello! My name is Alex.');
```



```
}  
  
greet(); // -> Hello! My name is Alex.  
greet(); // -> Hello! My name is Alex.  
greet(); // -> Hello! My name is Alex.
```



Complex code often requires hundreds of lines to be run repeatedly. Functions are the only practical way to implement something like that.

Functions might still not seem too useful. Let's move on to the next lesson where we'll go over how to make functions more powerful. Try to make sure that this lesson makes sense before moving on.

## Code Challenge

This is your first code challenge. It's a simple one. Feel free to test your understanding.

### INSTRUCTIONS

Write a function named `func` that logs something to the console.

