JS – Functions

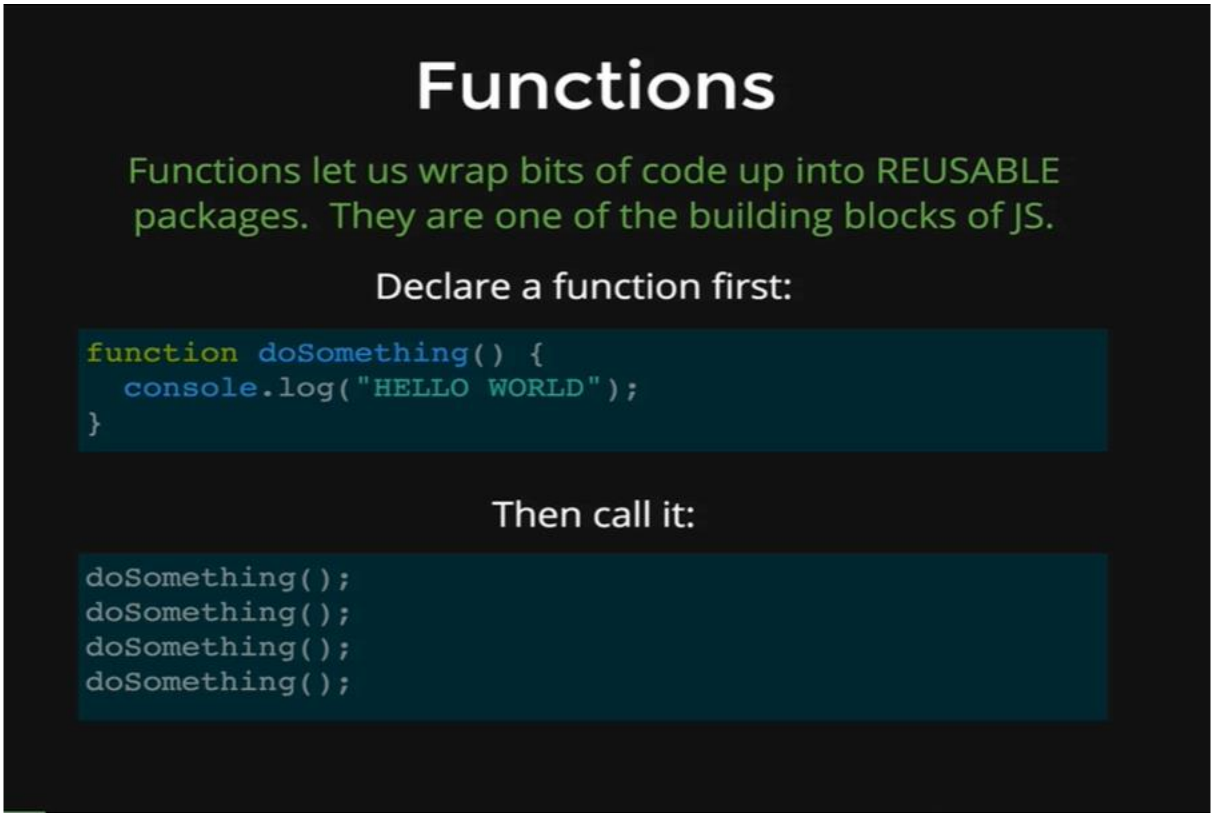
A Fundamental Building Block

Objectives:

- *Understand why we use functions*
- *Define a function without arguments*
- *Define a function with arguments*
- *Return a value from a function*

Functions are probably the single most important aspect of JavaScript. If a normal variable stores a value like “Hello” or 27, a function can store 20 lines of code and then we can recall that function and that 20 lines of code will run. Functions are just reusable bits of code.

What we said above is a very simple definition of function. There are more to it that we will be seeing as we move forward.



Functions

Functions let us wrap bits of code up into REUSABLE packages. They are one of the building blocks of JS.

Declare a function first:

```
function doSomething() {
  console.log("HELLO WORLD");
}
```

Then call it:

```
doSomething();
doSomething();
doSomething();
doSomething();
```

The first thing about function is that we must declare them first, so we define a block of code and we give it a name and that makes the function and then we run it later. It’s a two-step process. First declare a function and then call it. We can see above the syntax of declaring a function. *Function* is the keyword just like *var, if* or *while,* and then it is followed by the name of the function, then we have opening and closing parentheses and then curly braces. Inside the curly braces we can put as much code as we want.

Then we can call the function by referencing its name and parentheses at the end. So, lets see how that looks like.

```
function sayHi() {

        console.log(“HELLO”);

        console.log(“GOODBYE!”);

        }
```

After we type this function, JavaScript is going to see the function, but it is not going to execute the code inside the function. We are just registering a function called *sayHi,* but it doesn’t execute the code. So, now if we want to refer to the function, we call its name and we add parentheses and then we press *Enter*.

```
sayHi()
```

Output:

```
HELLO!
GOODBYE!
```

And now it runs whatever is inside the *sayHi()* function and produces the output. So, we can call this function as much as we want by writing its name and hitting *enter* button and every time it will produce the output as above.

```
sayHi()
```

Output:

```
HELLO!
GOODBYE!
```

```
sayHi()
```

Output:

```
HELLO!
GOODBYE!
```

```
sayHi()
```

Output:

```
HELLO!
GOODBYE!
```

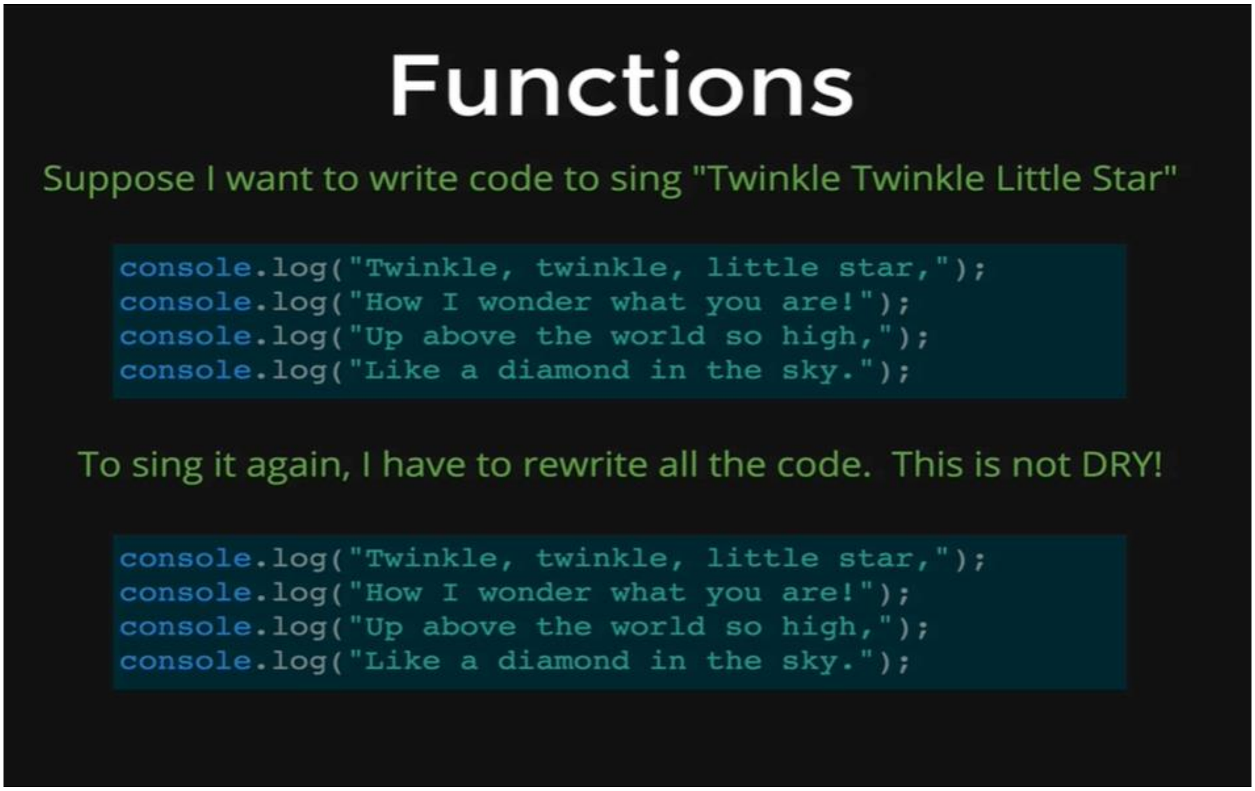
Now, if we just type sayHi without its parentheses then JavaScript returns us the full code of the function instead of executing the code inside the function.

```
sayHi
```

Output:

```
function sayHi() {
        console.log(“HELLO!”);
        console.log(“GOODBYE!”);
        }
```

It is important to have the parentheses after our function call to get the code inside the function executed, instead of getting the code back.



Functions

Suppose I want to write code to sing "Twinkle Twinkle Little Star"

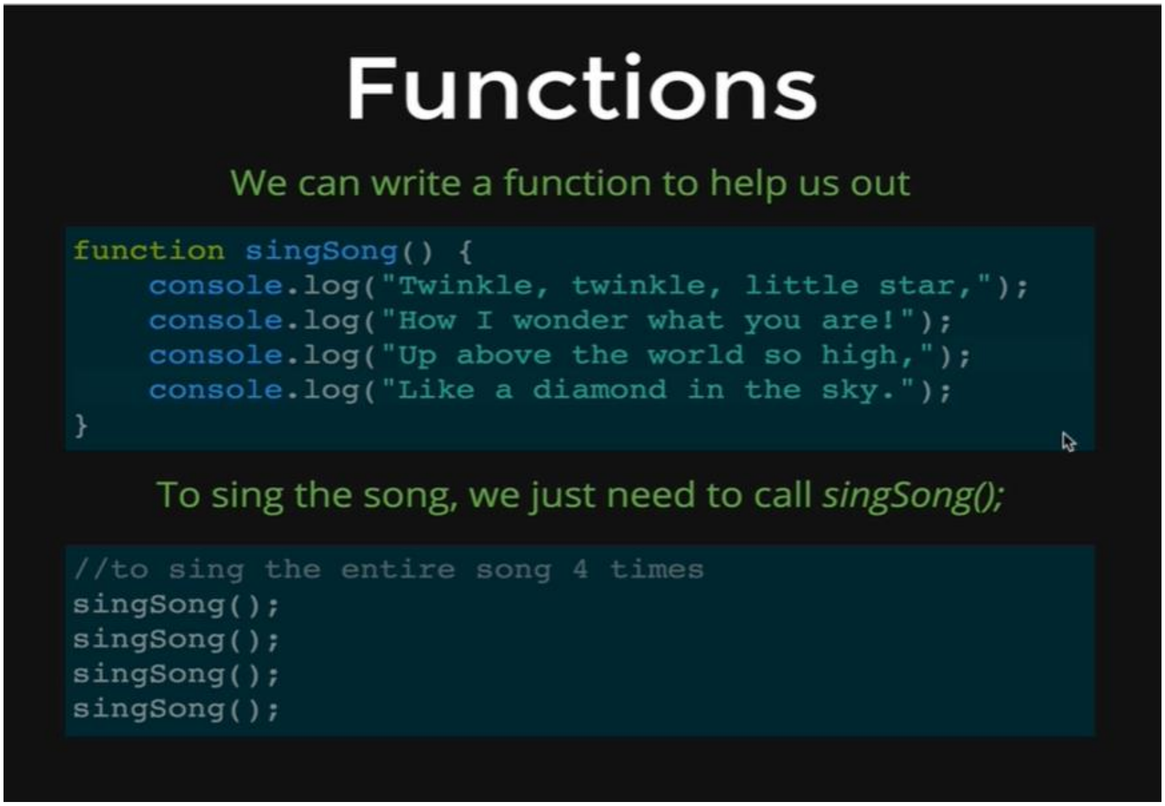
```
console.log("Twinkle, twinkle, little star,");
console.log("How I wonder what you are!");
console.log("Up above the world so high,");
console.log("Like a diamond in the sky.");
```

To sing it again, I have to rewrite all the code. This is not DRY!

```
console.log("Twinkle, twinkle, little star,");
console.log("How I wonder what you are!");
console.log("Up above the world so high,");
console.log("Like a diamond in the sky.");
```

We have another code of how function can help us DRY up our code. Remember DRY stands for Don’t Repeat Yourself. So, if we wanted to sing this song, “Twinkle Twinkle Little Star”, we have four console.log statements, so to sing the song once we need all the console.logs to be typed one after another and then to sing it again we need to type them again and so on every time we want to sing the full song we need to run the code individually.

This is obviously not very DRY and there are a lot of repeated code.



Functions

We can write a function to help us out

```
function singSong() {
    console.log("Twinkle, twinkle, little star,");
    console.log("How I wonder what you are!");
    console.log("Up above the world so high,");
    console.log("Like a diamond in the sky.");
}
```

To sing the song, we just need to call *singSong();*

```
//to sing the entire song 4 times
singSong();
singSong();
singSong();
singSong();
```

Using a function, we can just write those codes onetime. First we write the *function* keyword followed by the function name *singSong*, followed by parentheses and then we have our opening curly braces followed by the four console.logs containing each line of the Twinkle Twinkle song inside the parentheses and we can just call the functions as many times as we want to sing the song.

```
function singSong () {
        console.log(“Twinkle, twinkle, little star,”);
        console.log(“How I wonder what you are!”);
        console.log(“Up above the world so high,”);
        console.log(“Like a diamond in the sky.”);
        }
```

First, we type our function definition code and then we press *Enter,* and it lets JavaScript know that we have created a function named *singSong*.

And now if we just type *singSong* without the parentheses, we get the full function code back from JavaScript.

```
singSong
```

Output:

```
function singSong () {
        console.log(“Twinkle, twinkle, little star,”);
        console.log(“How I wonder what you are!”);
        console.log(“Up above the world so high,”);
        console.log(“Like a diamond in the sky.”);
        }
```

And now if we just want to sing the song then we make the function call followed by the parentheses.

```
 singSong()
```

Output:

```
Twinkle, twinkle, little star,
How I wonder what you are!
Up above the world so high
Like a diamond in the sky.
```

So, we just called the function once and get the song. Now if we want our song four times, its super easy we just type the function name four times followed by the parentheses.

```
singSong()
singSong()
singSong()
singSong()
```

Output:

```
Twinkle, twinkle, little star,
How I wonder what you are!
Up above the world so high
Like a diamond in the sky.
Twinkle, twinkle, little star,
How I wonder what you are!
Up above the world so high
Like a diamond in the sky.
Twinkle, twinkle, little star,
How I wonder what you are!
Up above the world so high
Like a diamond in the sky.
Twinkle, twinkle, little star,
How I wonder what you are!
Up above the world so high
Like a diamond in the sky.
```

This illustrates the power of functions in programming, though the Twinkle Twinkle song is a simple example but imagine if we have 50 lines of code, that we need to use multiple times. Having a function with all that code saves a lot of time and effort.