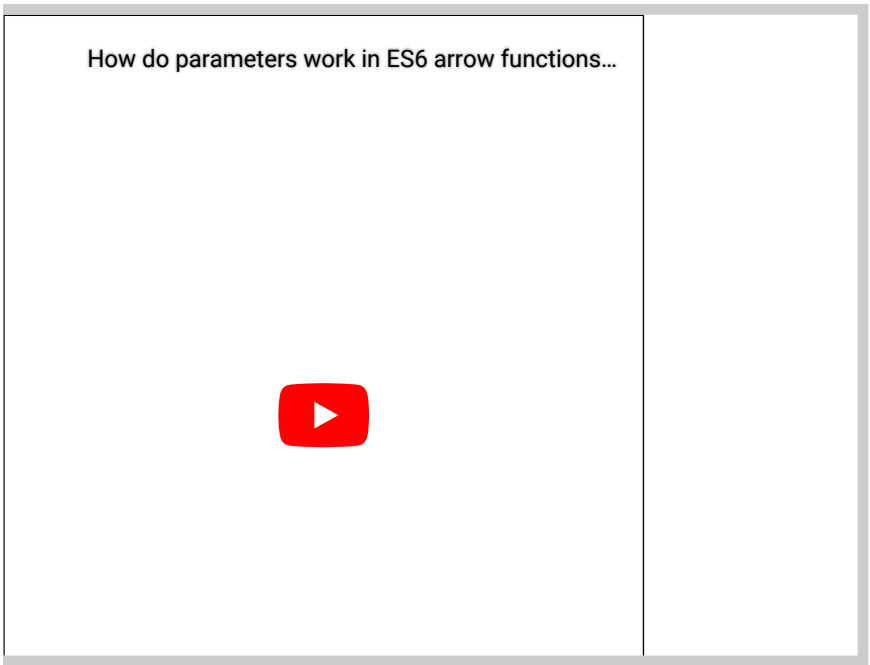
scraperapi

Getting Started with ECMAScript 6 Arrow function Parameters

There is a tremendous amount of flexibility when it comes to parameters in Arrow Functions.

In the previous article: "Getting Started with ECMAScript 6 Arrow functions – Basic Syntax," we had a very basic overview of ECMAScript 6's Arrow Functions and their syntax. In this article, I'll dive a little deeper into the topic of parameter.

In the previous article, we learned how the parameter(s) now come first, and then the "=>" characters, which effectively replace the "function" keyword. This is very efficient and is a key aspect to the simplicity of arrow functions. But it won't take long before you'll want to accept multiple parameters in your arrow function. And of course, you may want to accept no parameters.



Example # 1

```
1 var addTwoNumbers = (numOne,NumTwo) => numOne + NumTwo;
2
```

```
addTwoNumbers(2,3); // 5
```

In Example # 1, the function addTwoNumbers accepts two parameters. This differs from all of the examples in the previous article, each of which dealt with only one parameter. In ECMAScript 6's arrow function, when you need to accept multiple parameters, you leverage syntax which should be quite familiar to you: enclose the parameters in parentheses.

Arrow Functions that Take No Parameters

In situations in which you'll want to accept no parameters, the syntax will be empty parentheses.

Example # 2

```
var returnHello = () => "hello";

returnHello(); // "hello"
```

In Example # 2, we have the function: returnHello. This function takes no parameters and simply returns the string: "hello". It is certainly not a very useful function, but the main point here is that if you need to create an arrow function that takes no parameters, simply use a set of empty parentheses in place of what would have been a single parameter or parentheses that contained multiple parameters.

The Arguments Object

According to the ECMAScript Language Specification ECMA-262 6th Edition – DRAFT, "Arrow functions never have an arguments objects." At the time of this writing, version 33.1 of FireFox does allow access to the arguments object inside of an arrow function.

Example # 3

```
var returnArgsObj = () => arguments ;

console.dir( returnArgsObj('a','b','c') ); // {0: 'a',1: 'b',2: 'c'}
```

Running Example # 3 in the JavaScript console of FireFox 33.1 demonstrates that the argument object is, in fact, available inside of an arrow function. I'm not sure if this is a mistake or an intentional disregard for the specification. For now, it's probably best to not attempt access to the arguments object inside of an arrow function.

Summary

In this article we talked about parameters in ECMAScript 6 Arrow Functions. We discussed how to accepts multiple parameters, as well as how to specify none. We also talked a bit about access to the arguments object inside an arrow function. In the next article, I will discuss the value of "this" inside an arrow function.

© Copyright 2022 Kevin Chisholm - Blog

WordPress Theme | Hashone b