function Counter() {
 this.num = 0;

this.timer = setInterval(function add() {
 console.log(this);
 }, 1000);
}

var b = new Counter();

As you'll see, the window object is logged out every second. Clear the interval

Back to our original function. It was logging NaN because this.num was

referring to the num property on the window object (window.num which

So how do we fix this? With an arrow function! We need a function that

doesn't bind this. With an arrow function, the this binding keeps its

original binding from the context. Lets take our original counter function

doesn't exist), and not the b object (b.num) we had just created.

and replace our setInterval with an arrow function.

this.timer = setInterval(() => {

console.log(this.num);

by running:

clearInterval(b.timer);

function Counter() {
 this.num = 0;

this.num++;

var b = new Counter();

}, 1000);

created b object!

// true
// true
// ...

clearInterval(b.timer);

Conclusion

1. Shorter Syntax

2. No binding of this

subject for others to explore.

You can clear the interval with:

}

// 1 // 2 // 3 // ...

Let's back up. Our setInterval function isn't being called on a declared

object. It also isn't being called with the new keyword (only the counter()

function is). And lastly, we're not using call, bind, or apply. setInterval

is just a normal function. In fact, the value of this in setInterval is being

bound to the global object! Lets test this theory by logging the value of this:

For proof of concept we can again try logging this from within our arrow
function. We'll create a variable called that in our counter function. We'll
then log out true if the value of this in our setInterval function is equal to
the value of this (via that) in the parent counter function:

function Counter() {
 var that = this;

this.timer = setInterval(() => {
 console.log(this === that);
 }, 1000);
}

var b = new Counter();

As expected, we log true each time! Again, clear the interval with:

Hopefully this article helped you see the two main benefits of arrow functions:

As a disclaimer, there is more to arrow functions than what was explained in

learning! As always, leave a comment if you have any great resources on the

. . .

If this post was helpful, please hit the little blue heart! And don't

• And when you're ready to really dive into Web Development—<u>5 Best</u>

Udemy Courses for Learning Full Stack Web Development

forget to check out my other recent articles:

JavaScript: Regular Expressions For Beginners

JavaScript: The Keyword this for Beginners

• Find the Best Website Builders

Brandon Morelli

Aug 2017

Twitter too:

2K 🗌

More from codeburst

heavy website

5 steps to speed up your image

Responses

116

right?

3

tcvieira May 2, 2017

Brandon Morelli

May 2, 2017

Leon Forte May 6, 2017

Conversation between tcvieira and Brandon Morelli.

Brandon Morelli

9 min read

@BrandonMorelli

Medium member since

Creator of @codeburstio

 Frequently posting web development tutorials & articles. Follow me on

this article. But this should give you a great base of knowledge for further

As you'll see, the console begins logging increasing numbers—it works! The

preserved. Inside the setInterval function, this is still bound to our newly

original this binding created by the counter constructor function is

JavaScript ES6 Web Development Programming Learning To Code

Like what you read? Give Brandon Morelli a round of applause.

From a quick cheer to a standing ovation, clap to show how much you enjoyed this story.

Follow

More from codeburst

Deepak Gupta

Application

Javascript- Currying VS Partial

codeburst

1.8K 🗌

Bursts of code to power

through your day. Web Development articles,

tutorials, and news.

Follow

More from codeburst

How to get started with coding

396

(and 2 blunders to avoid)

Shivek Khurana

5 min read

1 response 🔍

Applause from Brandon Morelli (author)

Nicholas Pedroso
May 2, 2017

I like it.

Fast, interesting and informative reading FTW;)

Its nice to notice that if you want to return a literal object, you just parenthesize the body like

param => ({firstKey: param})

And if you have no arguments you can use underscore:

_=> { ... }

We can use the setTimeout() function. setTimeout() is a function that

Great article! In this part I think you mean setInterval() and not setTimeout(),

calls another function after a set number of milliseconds.

Ahh thank you! Such a silly mistake. I've fixed it in the post.

Conversation between Leon Forte and Brandon Morelli.

I'm really enjoying this JS series. You have a great knack of explaining complicated things simply. 2 1 response 🔍 Brandon Morelli May 19, 2017 Thank you for the kind words! Glad to help. ELES) Conversation between Mudia Imasuen and Brandon Morelli. Mudia Imasuen May 2, 2017 Pretty simple although Ialmost got lost when you began saying "this" "that". Thanks for the piece, I understand arrow functions better.. Already optimizing my code to use them. Good work. **1999** 2 1 response 🔍 Brandon Morelli Thanks for the feedback. I'll update that paragraph to hopefully improve understanding for future readers. ELES) Applause from Brandon Morelli (author) Schalavadi May 2, 2017 Yet another excellent article. Sincere thanks for sharing your knowledge 1

Show all responses