THIS KEYWORD IN JS

A COMPREHENSIVE GUIDE

zeshan.codes

Understanding “this”

The this keyword in JavaScript can be tricky!

In the global context, this refers to the global object:

index.js JavaScript

```javascript
console.log(this); // In a browser, this will be the window object
```

zeshan.codes

“this” in Object Methods

Using this inside an object method refers to the object itself.

index.js JavaScript

```javascript
const person = {
  name: "Alice",
  greet() {
    console.log(`Hello, my name is ${this.name}`);
  }
};

person.greet(); // Hello, my name is Alice
```

“this” inside greet refers to the person object. How do you use this in your code?

“this” in Constructor Functions

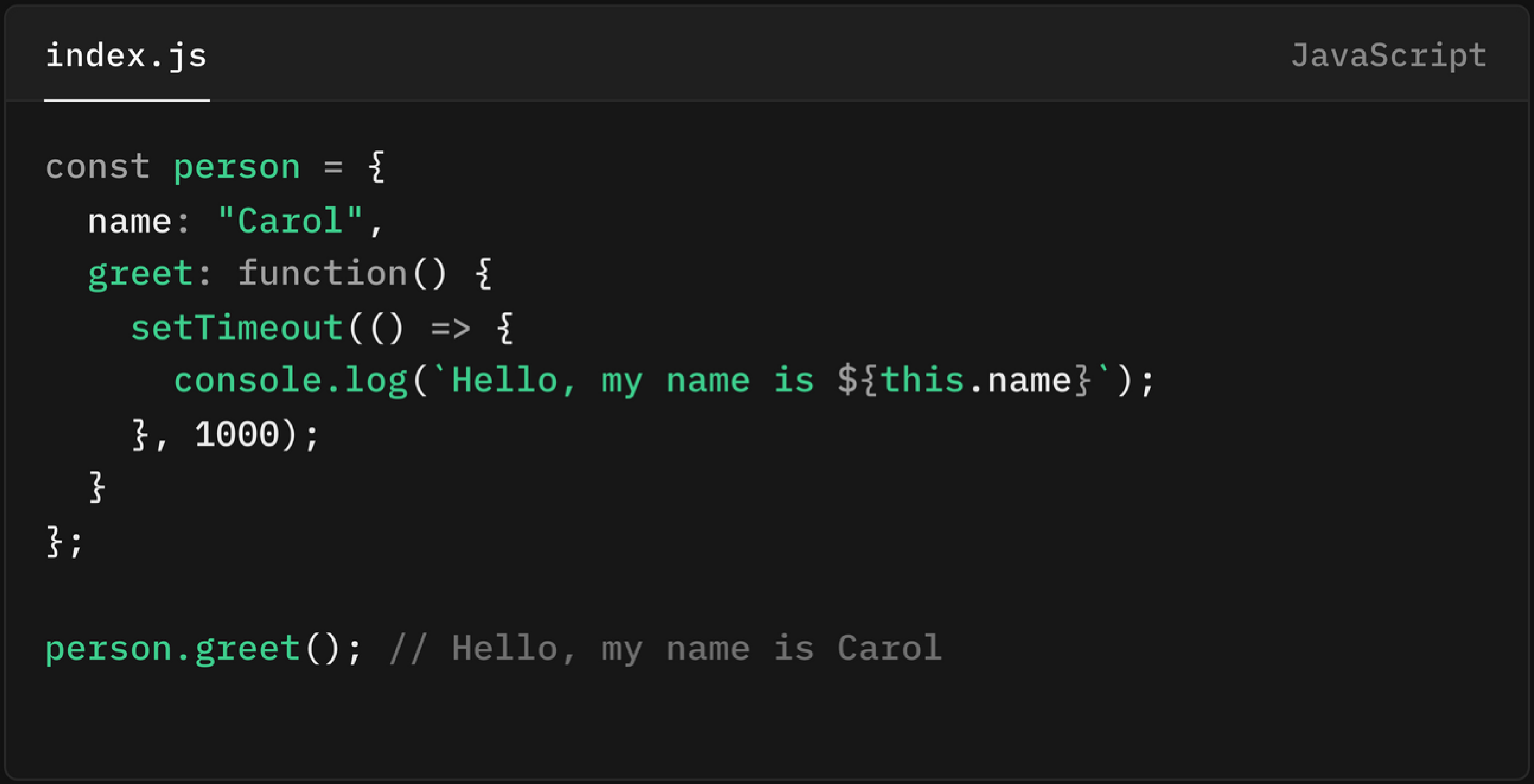
In constructor functions, “this” refers to the new instance being created.

index.js JavaScript

```javascript
function Person(name) {
  this.name = name;
}

const person1 = new Person('Bob');
console.log(person1.name); // Bob
```

Arrow Functions and “this”

Arrow functions have a lexical this—they don't have their own this context.



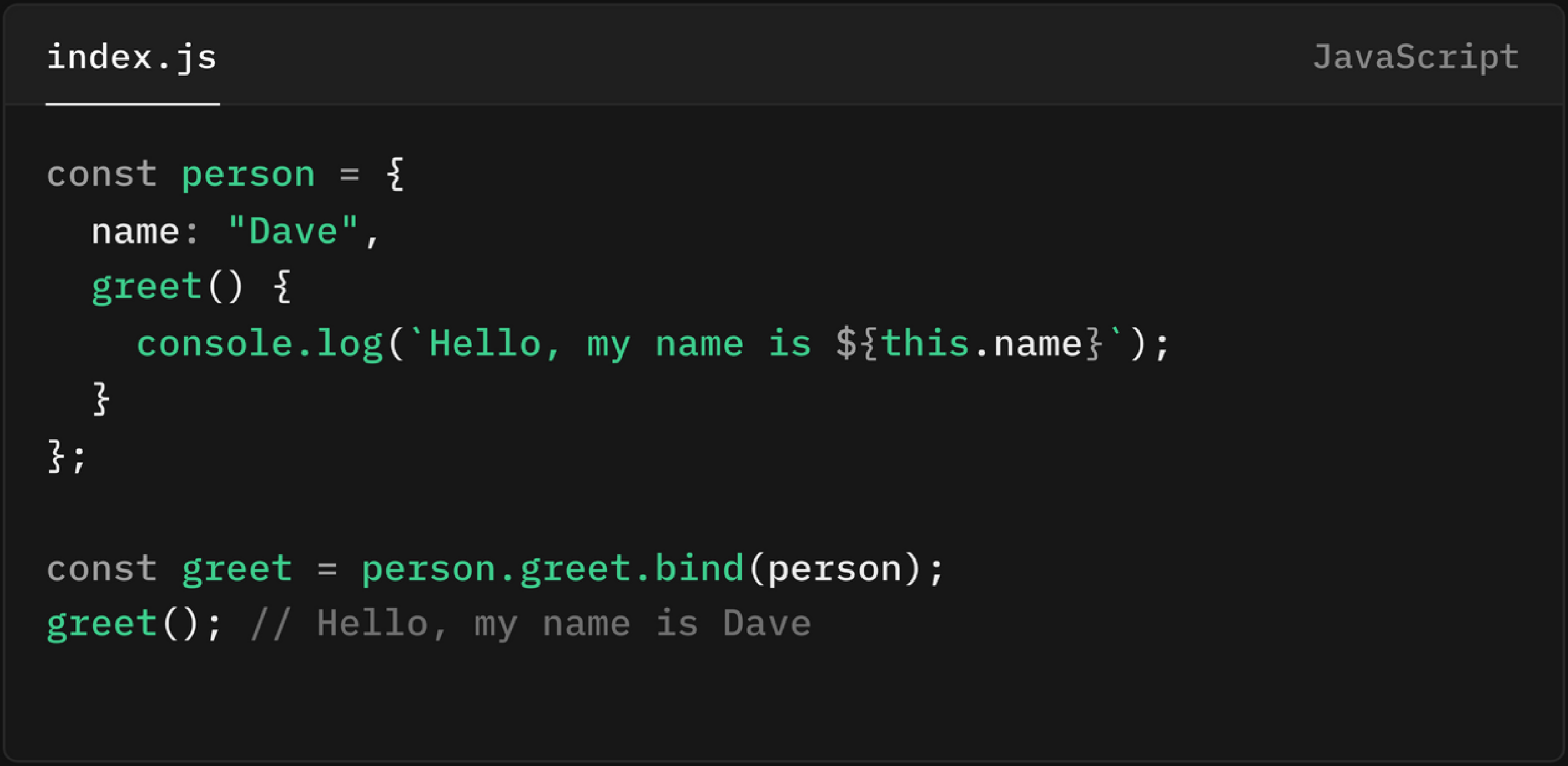
```javascript
const person = {
  name: "Carol",
  greet: function() {
    setTimeout(() => {
      console.log(`Hello, my name is ${this.name}`);
    }, 1000);
  }
};

person.greet(); // Hello, my name is Carol
```

it inherits this from the greet function’s scope.

zeshan.codes

Binding “this” with .bind()

You can explicitly set the value of this using .bind().



```javascript
const person = {
  name: "Dave",
  greet() {
    console.log(`Hello, my name is ${this.name}`);
  }
};

const greet = person.greet.bind(person);
greet(); // Hello, my name is Dave
```

Binding this can be especially useful in event handlers.

zeshan.codes

in linkedin/in/zeshancodes zeshan.codes

For Upcoming posts on JS

Do follow

Zeeshan Shakeel
Software engineer