CS390 WEB APPLICATION DEVELOPMENT

NISARG KOLHE

JavaScript

- A language for adding **functionality** to your websites
- Examples?
  - Handling Click Events
  - DOM manipulation

History of JavaScript

- Written in 10 days by **Brendan Eich** in 1995
- Created to provide for a way to help make sites more **dynamic**
- Submitted JavaScript for specification to ECMA in 1997

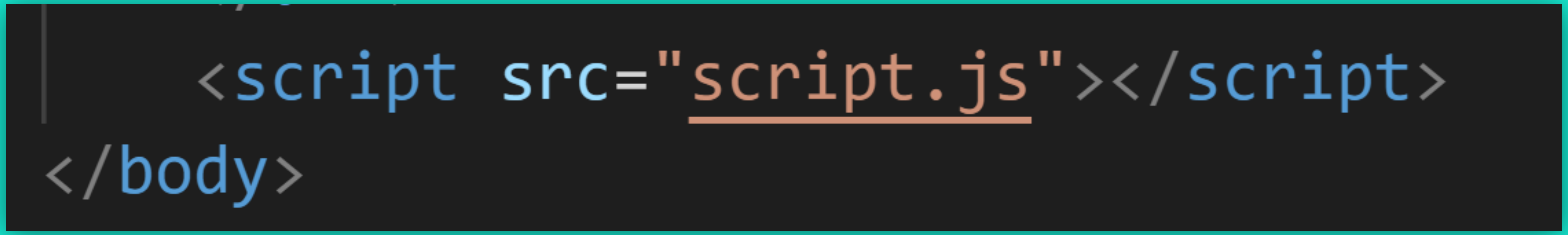
ECMAScript

- Specification by **ECMA**
- Standardized JavaScript features
- Widely used **ES5** released in 2009, then ES5.1 in 2011
- ES2015 (ES6) released in June 2015
- Latest version is **ES2019**

Getting Started

What do we need?

- **script** tag in the bottom of the body
- **src** attribute set equal to your js file's name



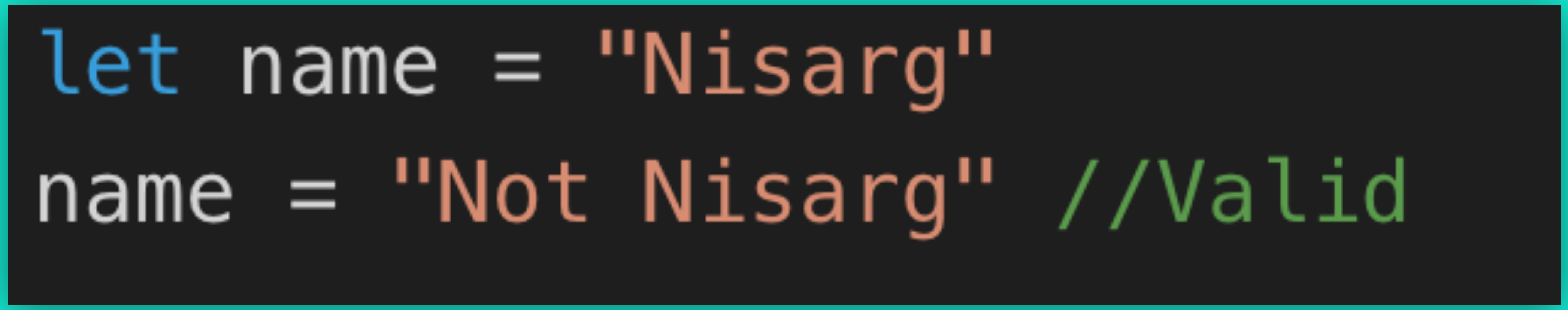
```
    <script src="script.js"></script>
</body>
```

JS Fundamentals

- Syntax is similar to Java
- No classes needed, just start typing!
- No types
- this keyword isn’t lexical

Creating a variable

- Use **let**
- **DON'T** use var
- No need to set type
- You can reassign a variable too



```
let name = "Nisarg"
name = "Not Nisarg" //Valid
```

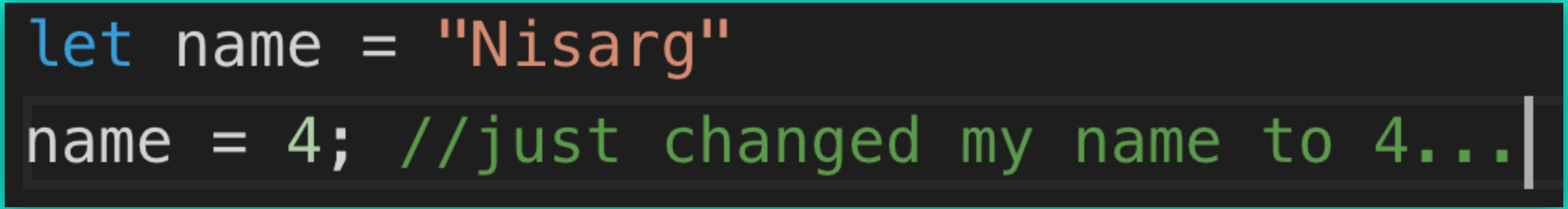
Printing

```
let var = “hello world”;

console.log(var);
```

Check Dev console in your browser!

What about changing the type?

Also valid.



```
let name = "Nisarg"
name = 4; //just changed my name to 4...
```

Truthy/Falsy values

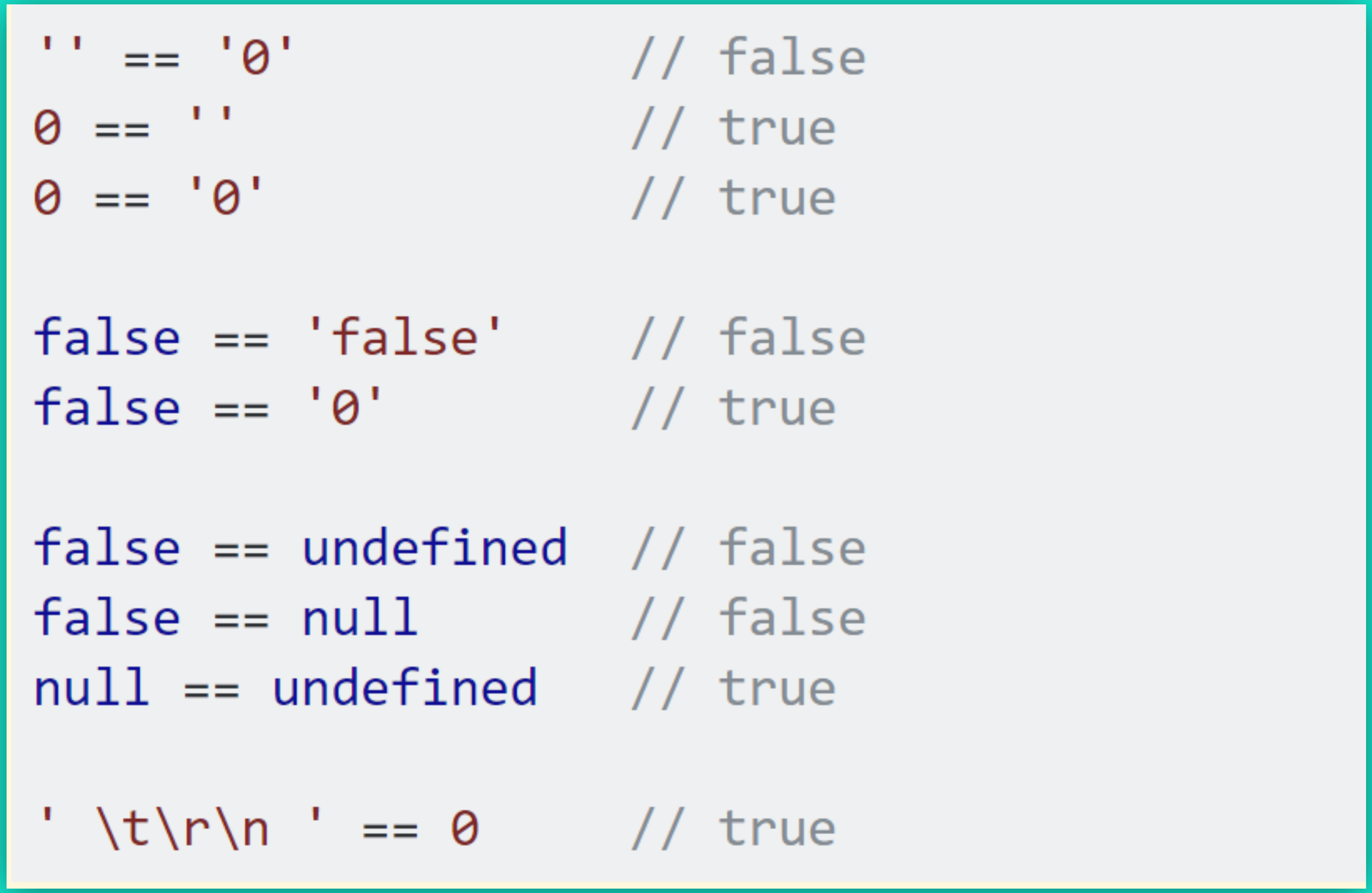
Expression	Value
[‘a’]	truthy
0	falsy
Number > 0	truthy
TRUE	truthy
Any non empty string	truthy

Truthy/Falsy values

Expression	Value
null	falsy
undefined	falsy
“”	falsy
[]	truthy
{}	truthy

== or ===?

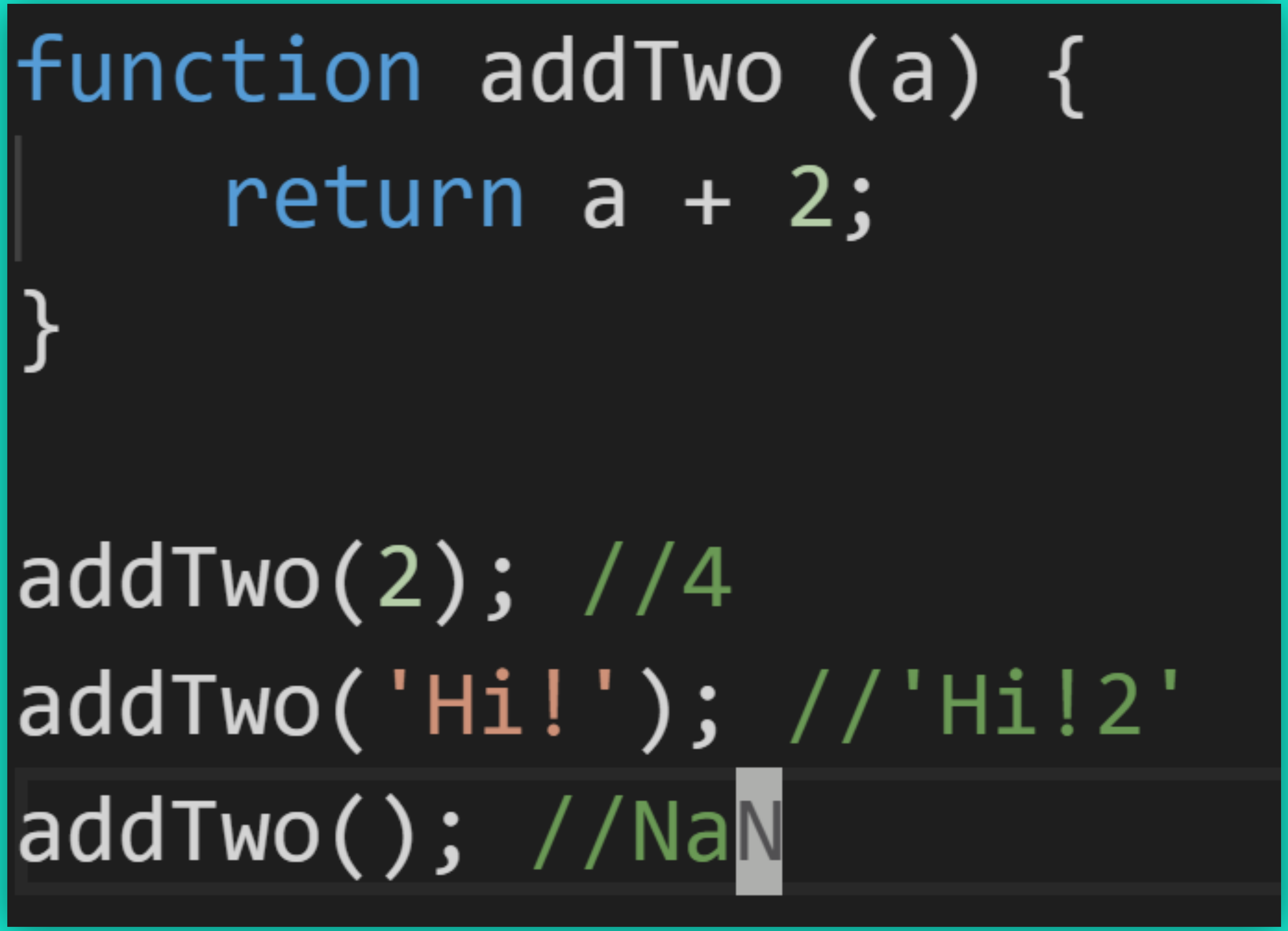
- == does **type casting**, which is awesome
  - Not if there are two different types you’re comparing
- So..which one to use?
  - === doesn’t type cast



```
'' == '0'           // false
0 == ''             // true
0 == '0'            // true

false == 'false'    // false
false == '0'        // true

false == undefined  // false
false == null       // false
null == undefined   // true

' \t\r\n ' == 0     // true
```

Functions

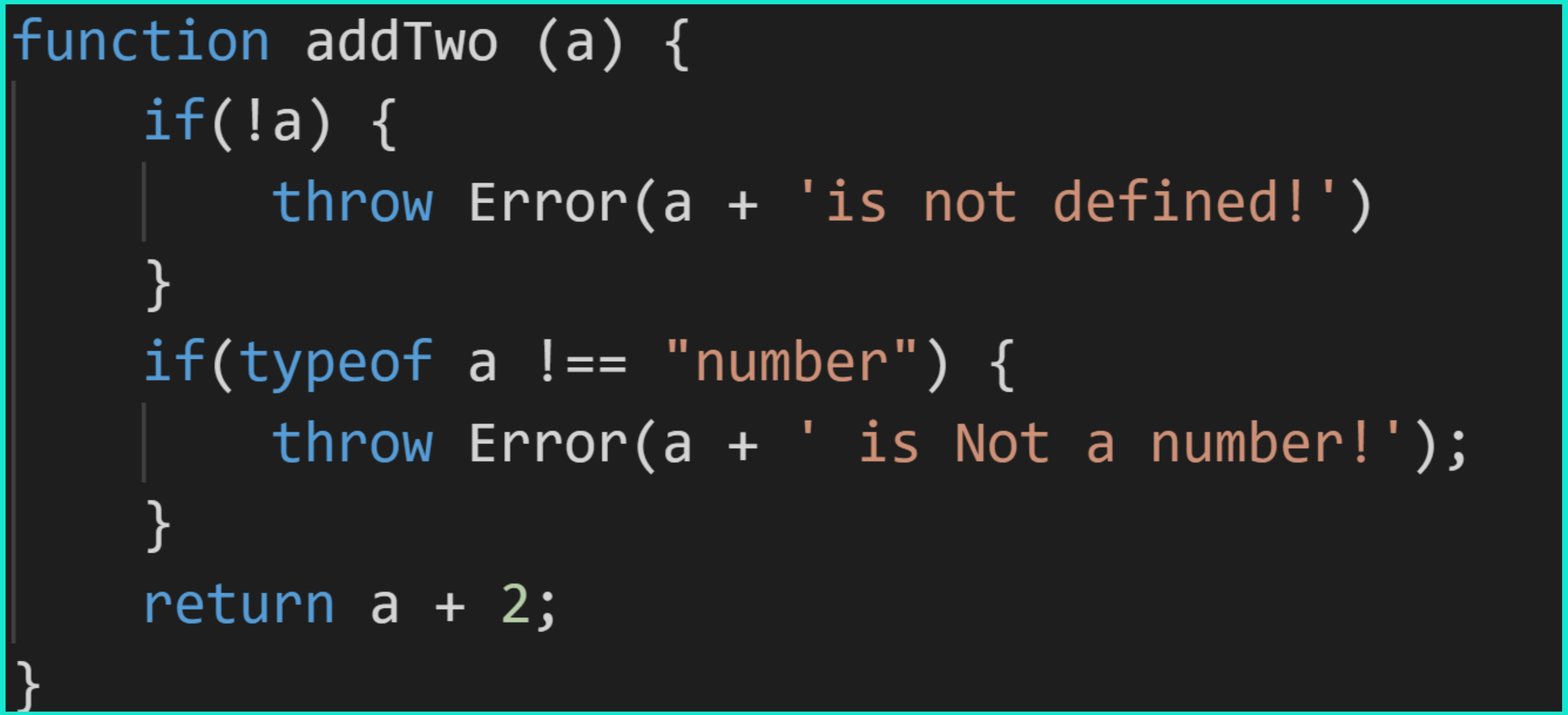
- Don’t enforce number of arguments
- OR TYPE



```
function addTwo (a) {
    return a + 2;
}

addTwo(2); //4
addTwo('Hi!'); //'Hi!2'
addTwo(); //NaN
```

How do I fix this?

- Don’t enforce number of arguments
- OR TYPE

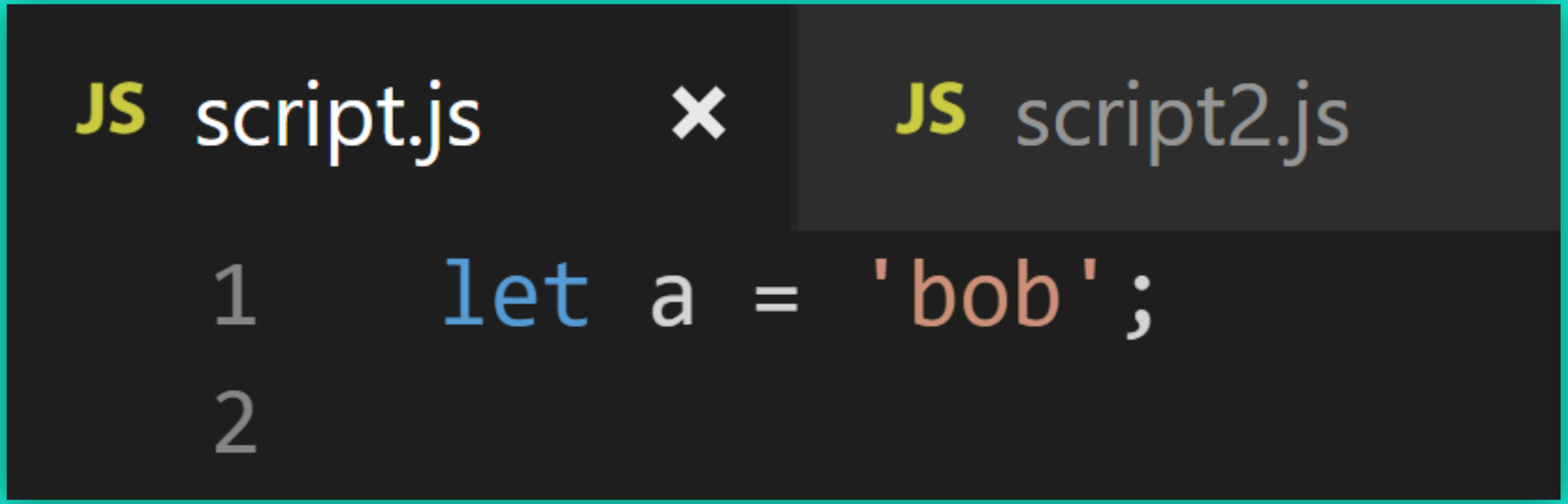


```
function addTwo (a) {
    if(!a) {
        throw Error(a + 'is not defined!')
    }
    if(typeof a !== "number") {
        throw Error(a + ' is Not a number!');
    }
    return a + 2;
}
```

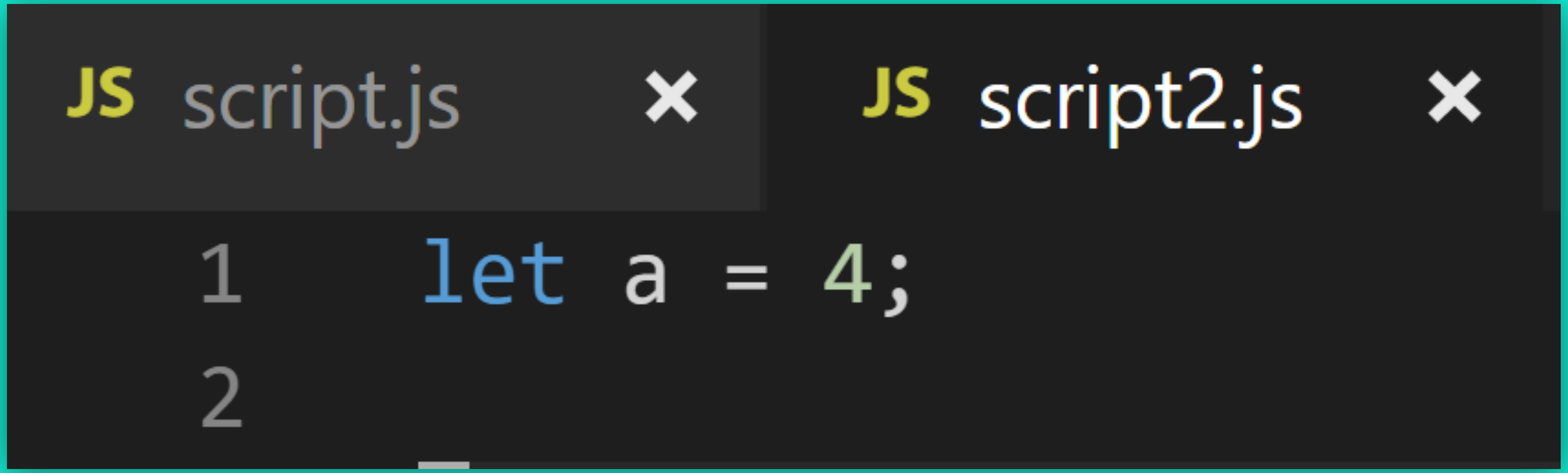
How do I REALLY fix this?

- Use a type checker like flow
- Avoid pure JavaScript
  - TypeScript is basically JavaScript with type checking built in

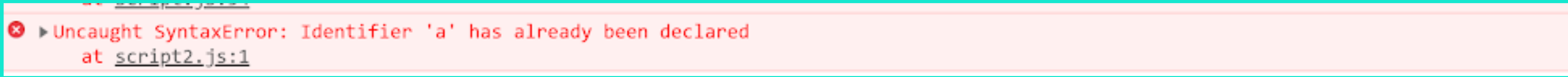
More JS pitfalls!



```
let a = 'bob';
```



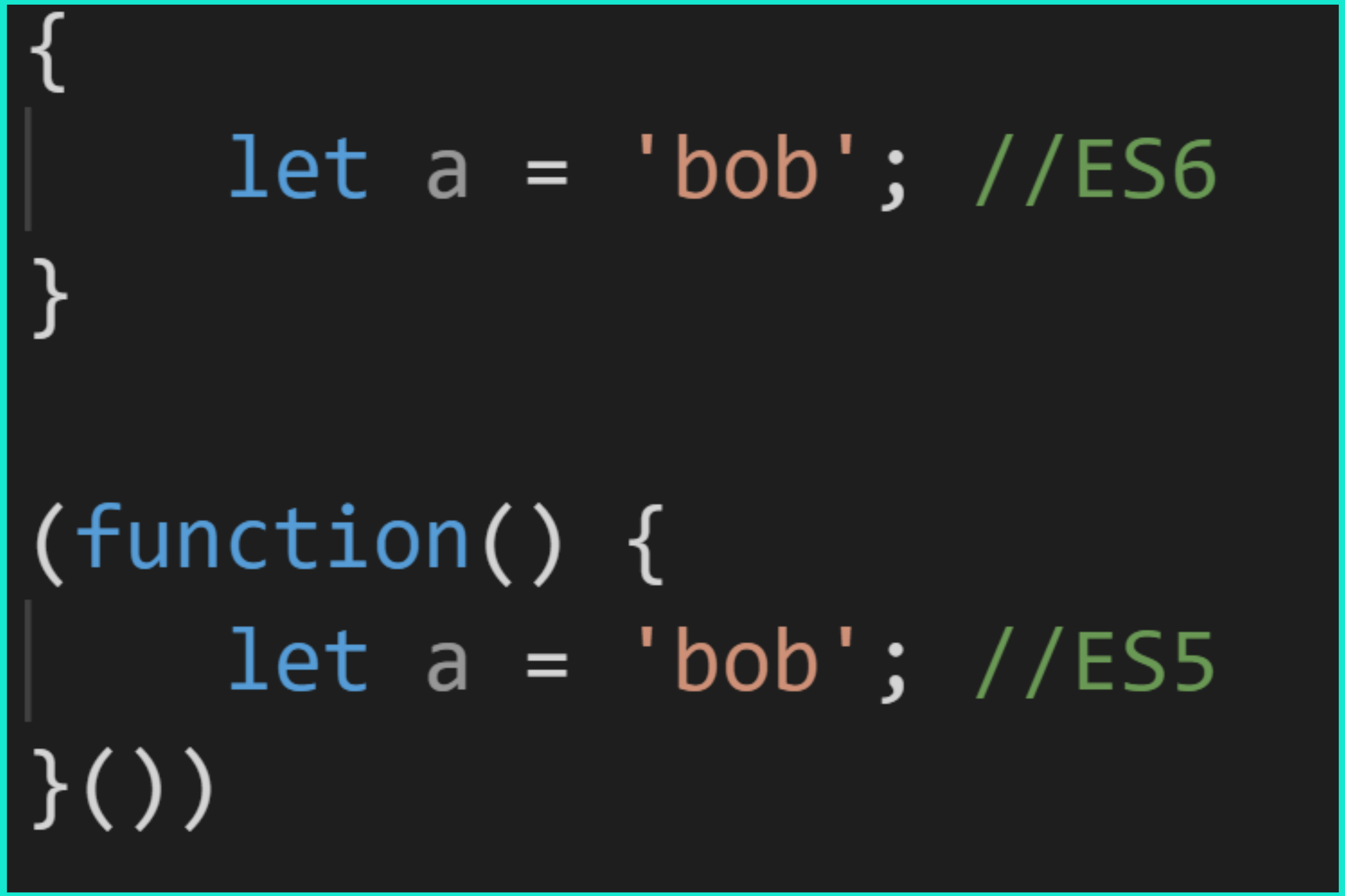
```
let a = 4;
```



```
▶Uncaught SyntaxError: Identifier 'a' has already been declared
    at script2.js:1
```

One way to fix it

Immediately Invoked Function Expression



```
{
    let a = 'bob'; //ES6
}

(function() {
    let a = 'bob'; //ES5
}())
```

Resources

- https://www.learn-js.org
- W3Schools
  - https://www.w3schools.com/js/
- CodeAcademy (very comprehensive)
  - https://www.codecademy.com/learn/introduction-to-javascript