NiTDA

FEDERAL MINISTRY OF COMMUNICATIONS AND DIGITAL ECONOMY
Leveraging ICT for National Development

Introduction to JavaScript
Part 2

Calistus Igwilo
https://linkedin.com/in/calistus-igwilo
https://twitter.com/CalistusIgwilo

DOMINEUM

BSV BLOCKCHAIN

JavaScript Variable Practice

```
fruit = "orange";

myNum = "5";

otherNum = 20;

sumNum = otherNum + myNum;
```

JavaScript is loosely typed.

It makes it more expressive

+ sign does 2 things:

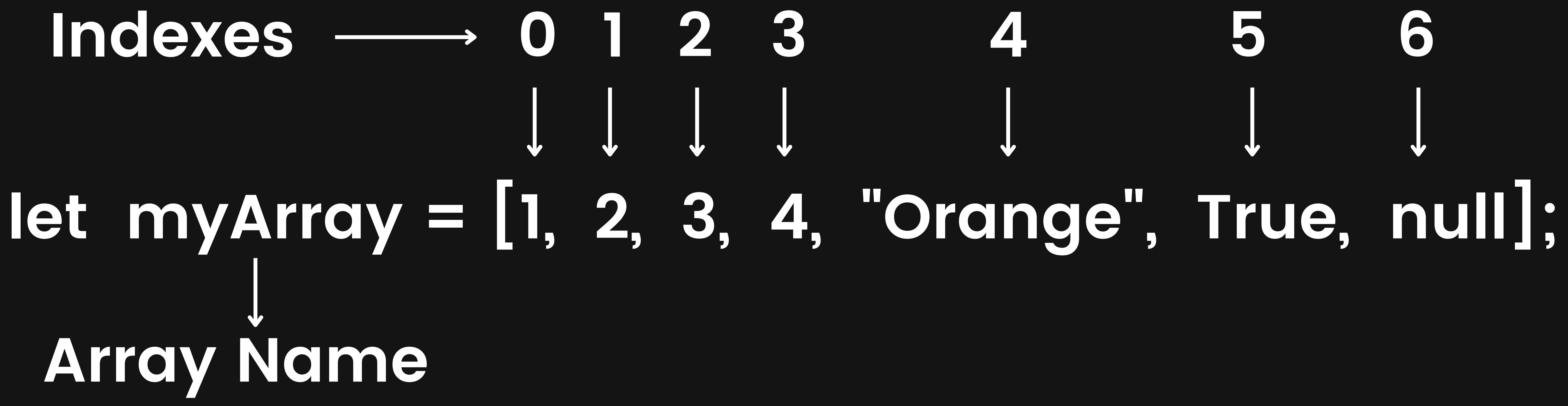
- Add
- Concatenate

typeof

JavaScript Array

Array is a container with subcontainers

Stores multiple items under a single variable name



Indexes → 0 1 2 3 4 5 6

```
let  myArray = [1,  2,  3,  4,  "Orange",  True,  null];
```

Array Name

Creating an Array in JavaScript

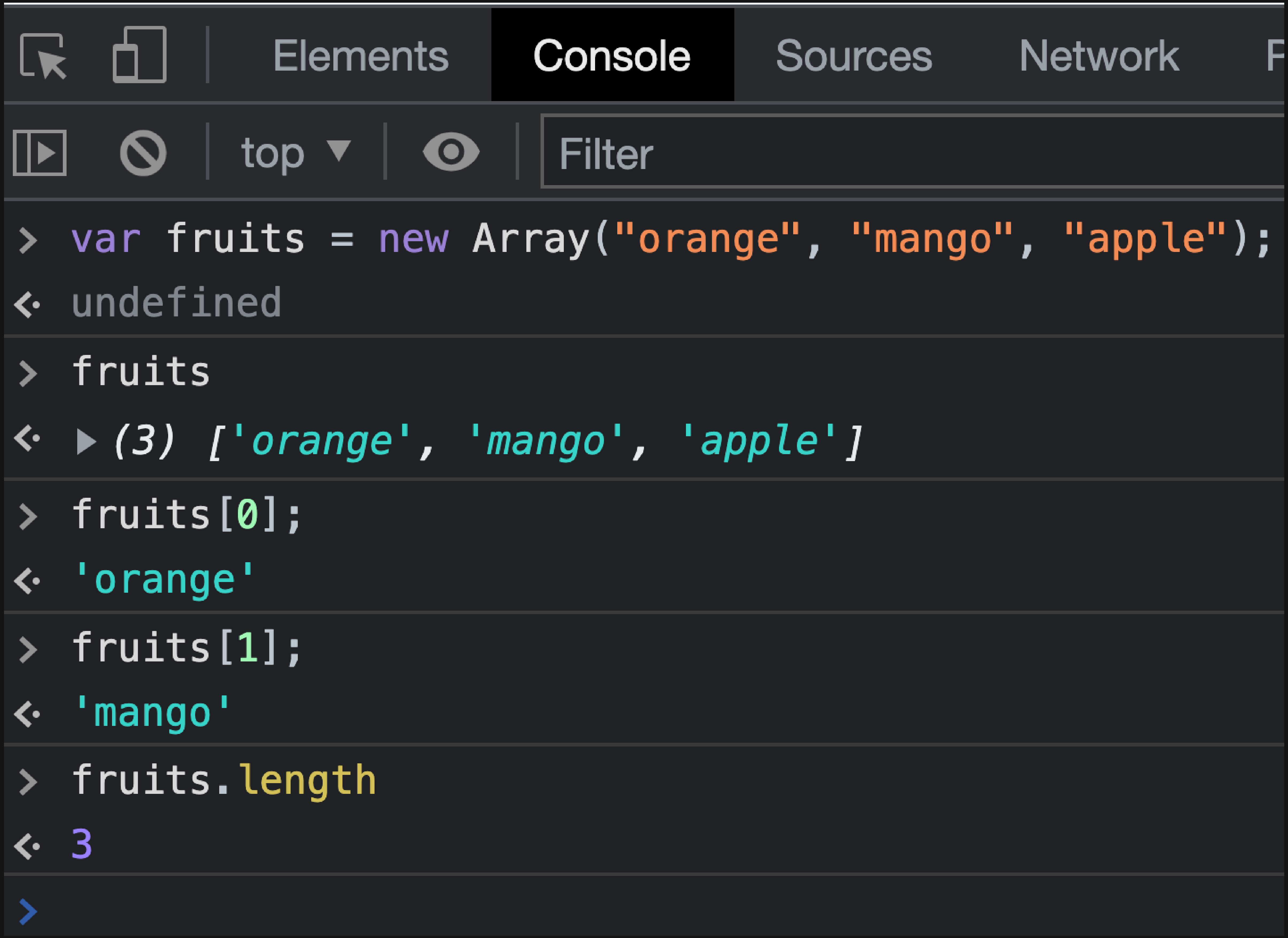
There are 2 methods of creating array in Javascript

```
var fruits = new Array("orange", "mango", "apple");
```

```
var fruits = ["orange", "mango", "apple"];
```

Both produce the same result

JavaScript Array Practice



```
> var fruits = new Array("orange", "mango", "apple");
<· undefined
> fruits
<· ▶ (3) ['orange', 'mango', 'apple']
> fruits[0];
<· 'orange'
> fruits[1];
<· 'mango'
> fruits.length
<· 3
>
```

JavaScript Array Practice

You can put anything in an array

```
var vehicles = ["innosson", "massila", "toyota"];
```

```
var vehicles =
["innosson", ["ikenga", "fox", "umu"],"massila", ["elicpse", "pajero",
"l200"],"toyota", ["camry", "corolla", "venza"]];
```

```
console.log(vehicles[1][2])  --> You will get "umu"
```

Array Methods

push - adds element(s) to an array

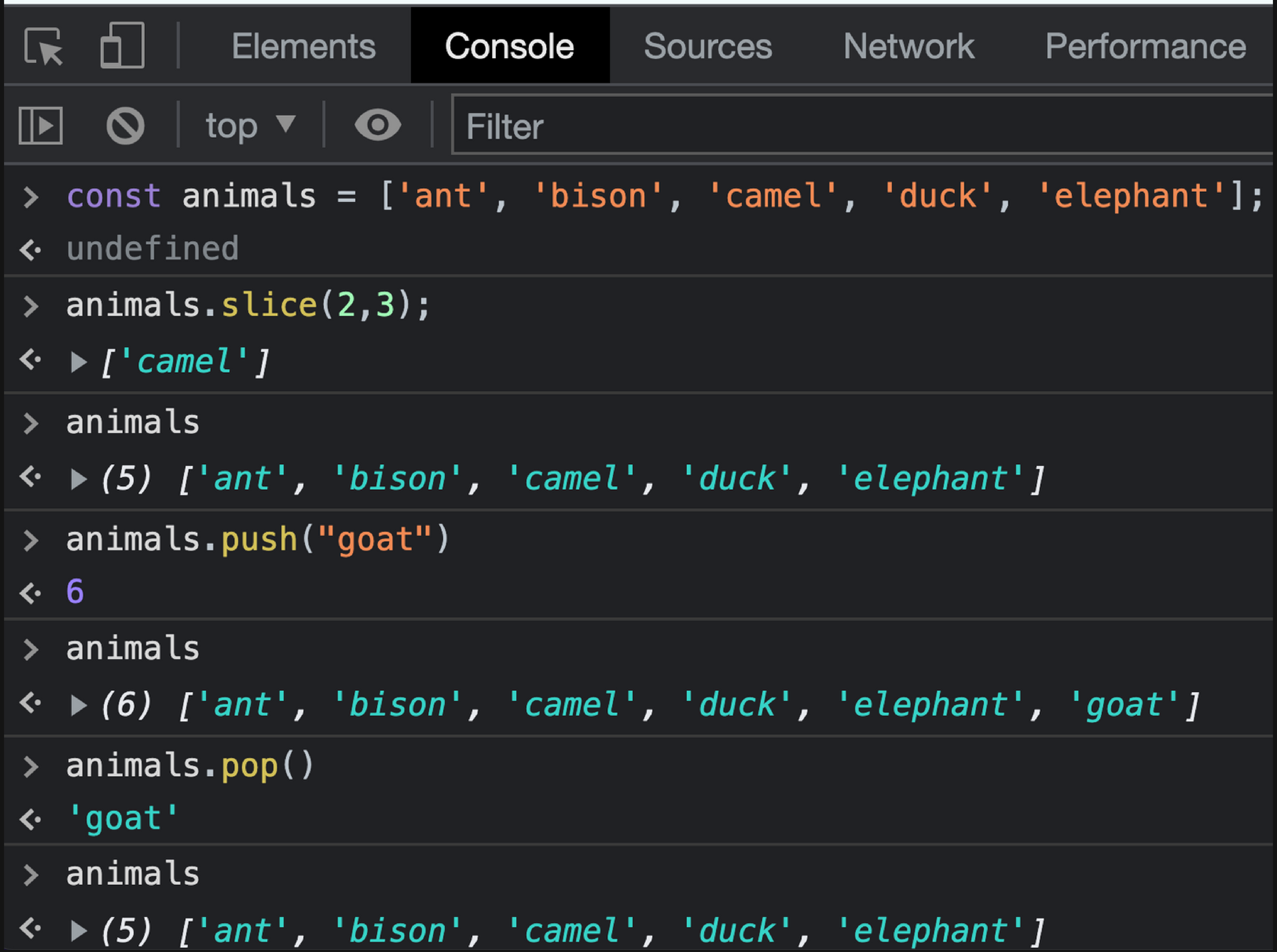
pop - removes the last element from an array and return it

splice - returns a shallow copy of a portion of an array

sort - arranges items in ascending order

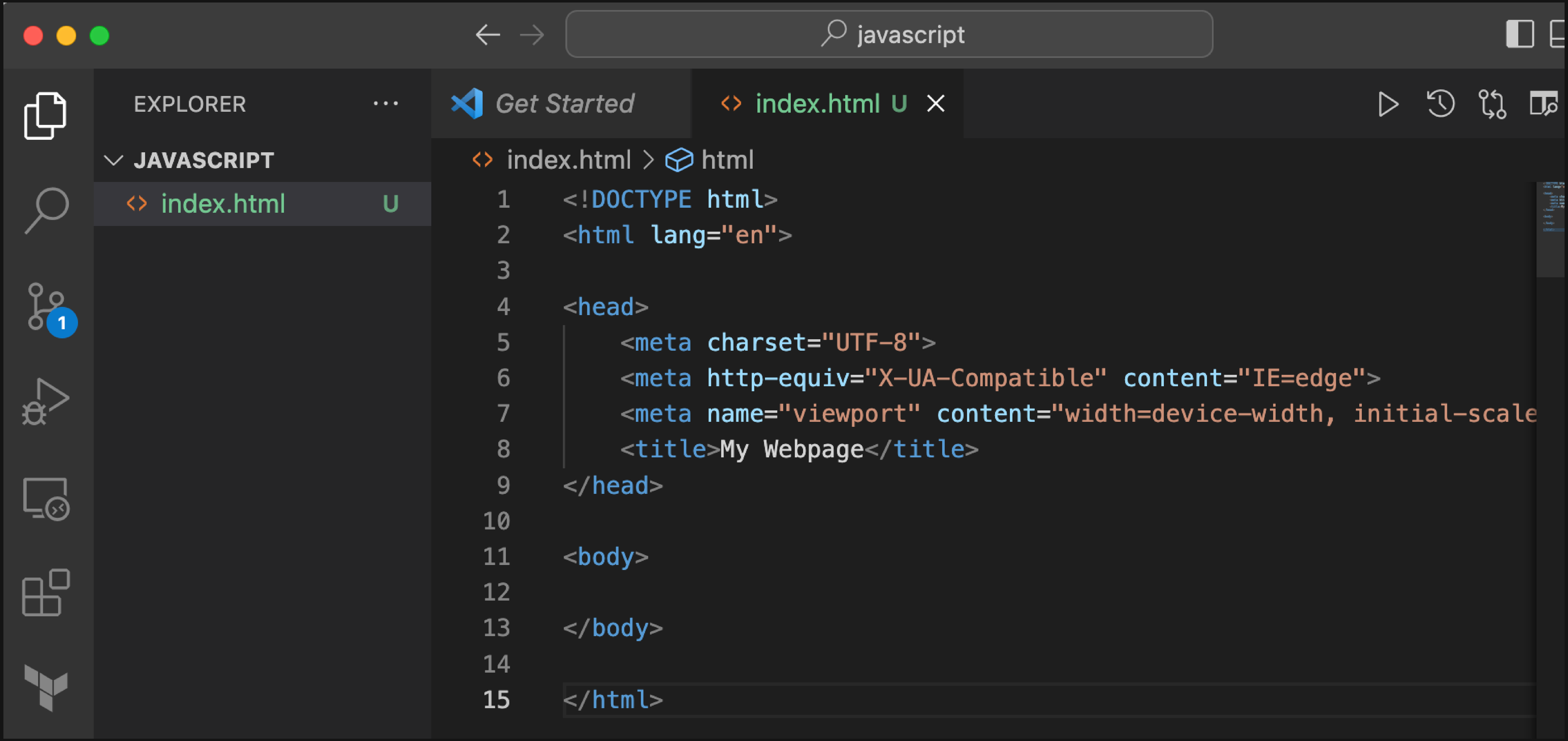
length - returns the length of the array

Array Methods Practice



```
> const animals = ['ant', 'bison', 'camel', 'duck', 'elephant'];
<· undefined
> animals.slice(2,3);
<· ▶ ['camel']
> animals
<· ▶ (5) ['ant', 'bison', 'camel', 'duck', 'elephant']
> animals.push("goat")
<· 6
> animals
<· ▶ (6) ['ant', 'bison', 'camel', 'duck', 'elephant', 'goat']
> animals.pop()
<· 'goat'
> animals
<· ▶ (5) ['ant', 'bison', 'camel', 'duck', 'elephant']
```

Working with Code Editor

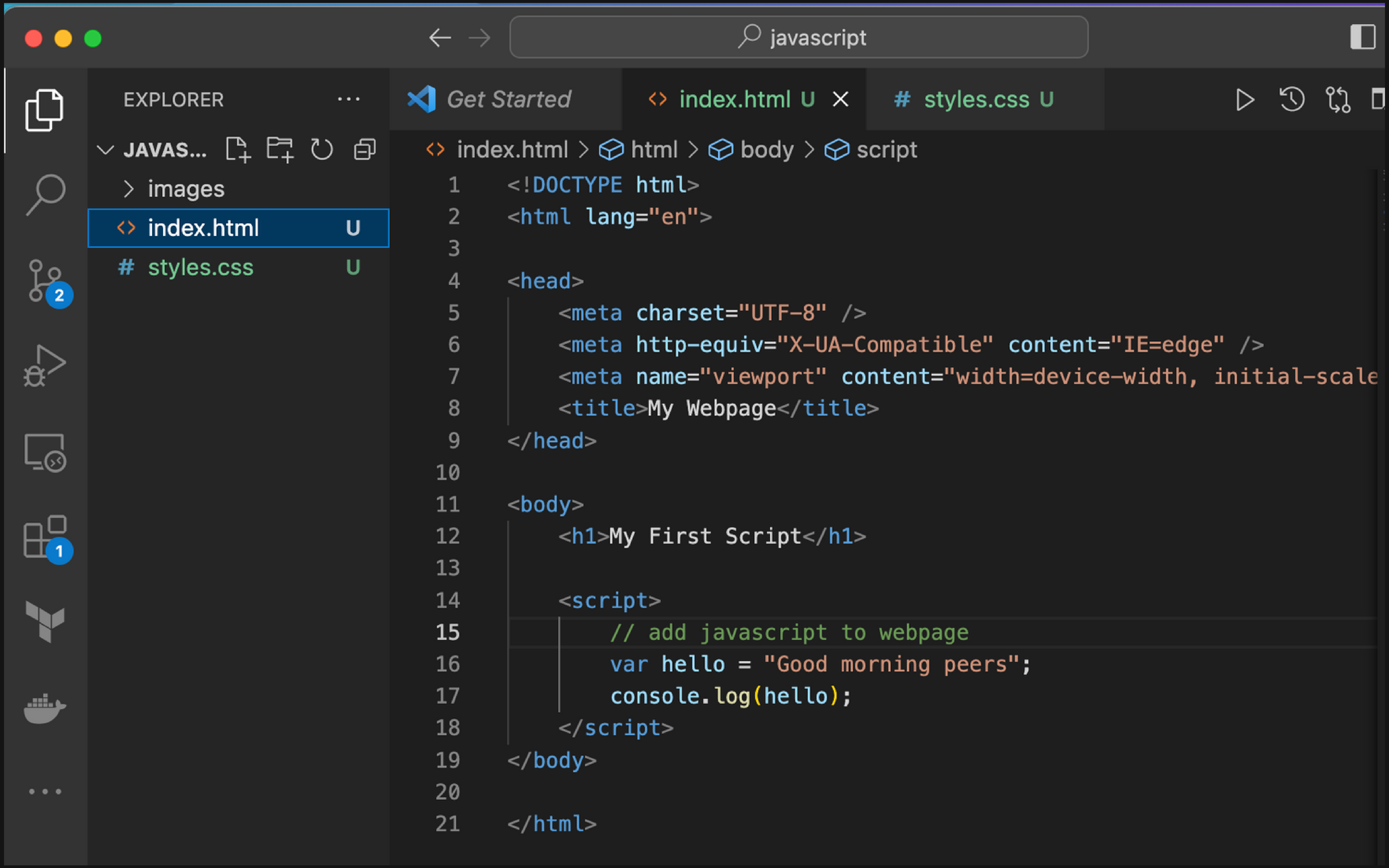


```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale
    <title>My Webpage</title>
</head>

<body>

</body>

</html>
```

Working with Code Editor

- Create files and directories from explorer

- install plugins: Atom One Light Theme



```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale
    <title>My Webpage</title>
</head>

<body>
    <h1>My First Script</h1>

    <script>
        // add javascript to webpage
        var hello = "Good morning peers";
        console.log(hello);
    </script>
</body>

</html>
```

JavaScript Operators

+	addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo (remainder after division)

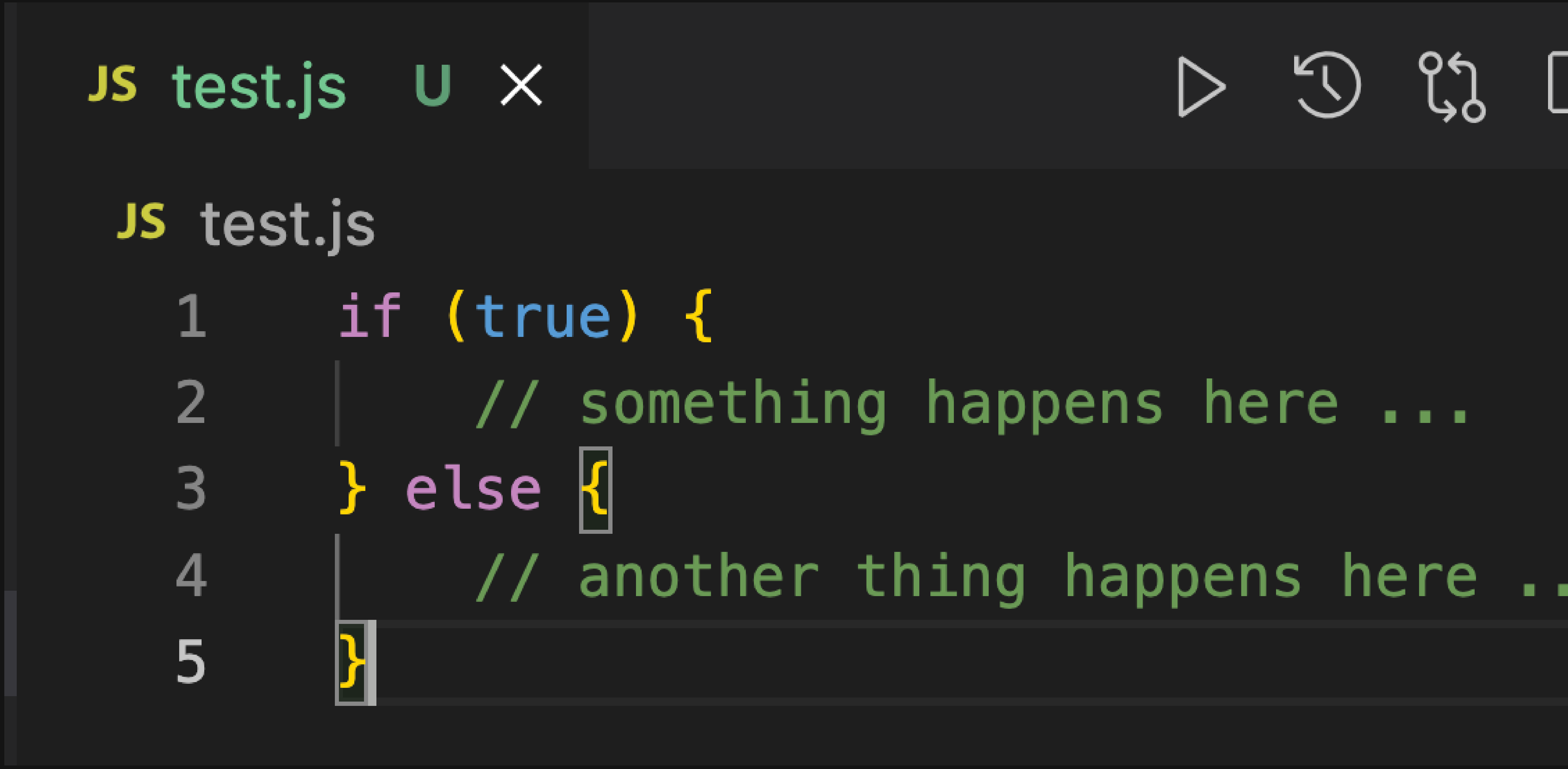
JavaScript Operators

https://www.w3schools.com/js/js_operators.asp

+	addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (remainder after division)

Boolean in JavaScript

- **True**
- **False**
  - 0
  - null
  - undefined



```
if (true) {
    // something happens here ...
} else {
    // another thing happens here ..
}
```

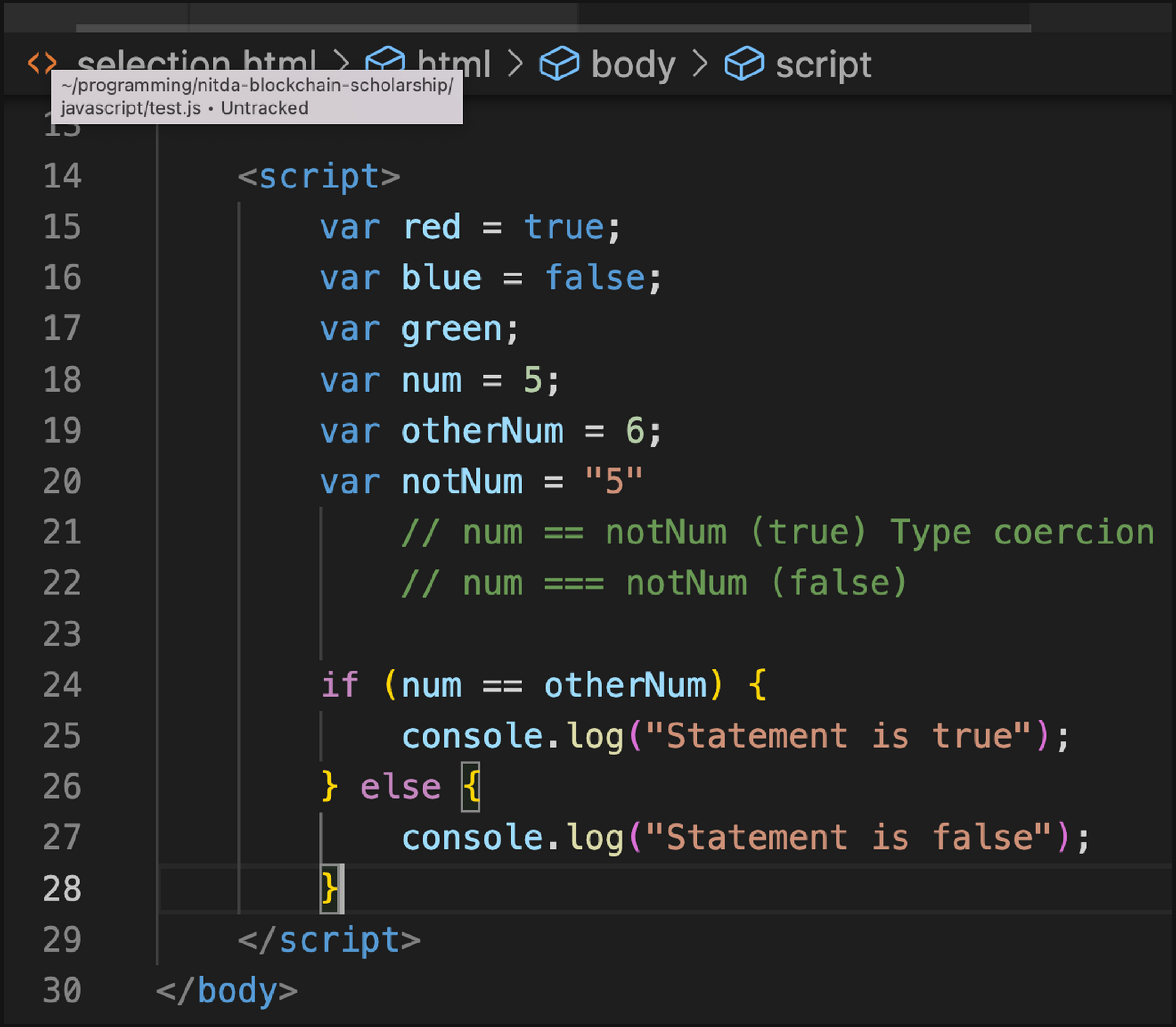
Comparing Values

==

!=

<

>



~/programming/nitda-blockchain-scholarship/ javascript/test.js • Untracked

```
    <script>
        var red = true;
        var blue = false;
        var green;
        var num = 5;
        var otherNum = 6;
        var notNum = "5"
            // num == notNum (true) Type coercion
            // num === notNum (false)

        if (num == otherNum) {
            console.log("Statement is true");
        } else {
            console.log("Statement is false");
        }
    </script>
</body>
```