Web Dev Bootcamp - Day 4

Topics

1. Strings
2. Functions
3. Objects
4. var vs. let vs. const
5. Difference between '==' and '==='
6. Document Object Model (DOM)


CHALIA SURU KARTE HAI
HAR HAR JAVASCRIPT

STRINGS

A string is zero or more characters written inside double quote

Ex:

```
var name = "Babu Rao";
var ch = "a";
var num = "12";
var sentence = "JavaScript is one of the most popular language";
var isAccepted = "true";
```

MORE ON STRINGS

```
var name = "Babu rao"

// Returns the datatype of a variable
console.log(typeof name);   // string

// Returns the length of a string
console.log(name.length);   // 8

// Converts a string to uppercase
console.log(name.toUpperCase());    // BABU RAO

// Converts a string to lowercase
console.log(name.toLowerCase());    // babu rao
```

Learn more about Strings in JavaScript

After understanding Strings my reaction

FUNCTIONS

A JavaScript function is a block of code designed to perform a particular task.

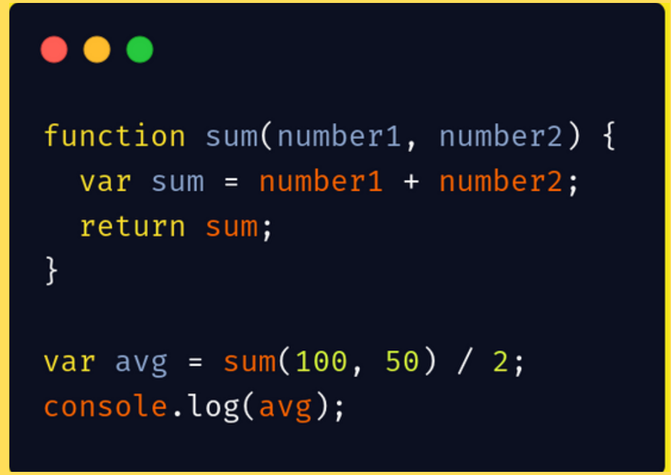
Ex:

```
function printName() {
  console.log("Hello World");
}

printName();
```

```
function printName(name) {
  console.log("Hello " + name);
}

printName("Babu Rao");
```



```
function sum(number1, number2) {
  var sum = number1 + number2;
  return sum;
}

var avg = sum(100, 50) / 2;
console.log(avg);
```

Learn more about Functions

OBJECTS

A JavaScript object is a collection of named values

Ex:

```
var car = {
  model: "Audi R8",
  price: 1200000,
  twoSeater: true,
};

console.log(car);           // { model: 'Audi R8', price: 1200000, twoSeater: true }
console.log(car.model);     // Audi R8
console.log(car.price);     // 1200000
console.log(car.twoSeater); // true
```

Learn more about Objects

var vs. let vs. const

const: const is used to declare constants. It can't be modified after initialization.

```
const name = "Babu rao";
name = "hero";  // TypeError: Assignment to
                   constant variable
```

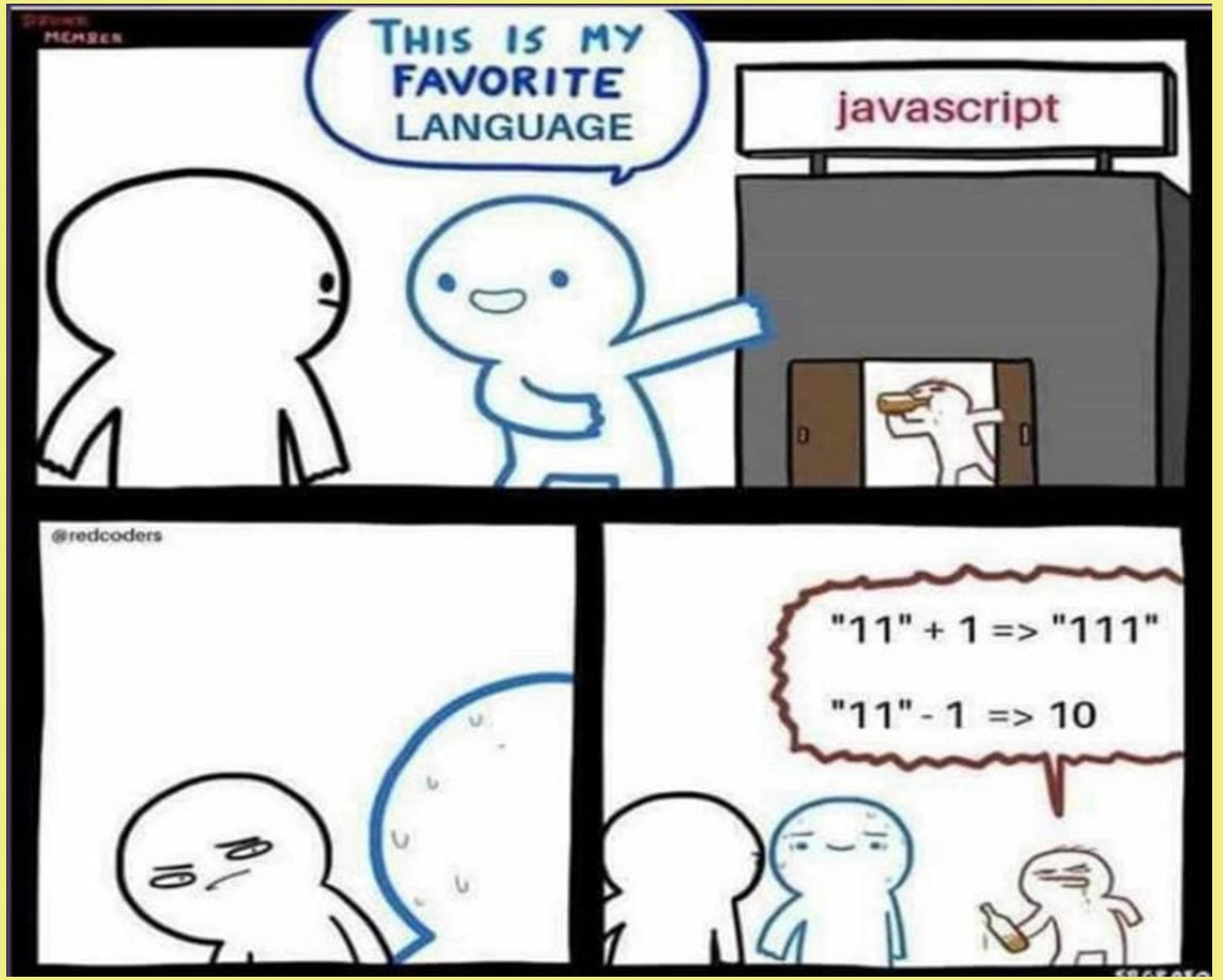
var and let: scope of a variable defined with let is limited to the block in which it is declared while variable declared with var has the global scope

```
for (let i = 0; i < 3; i++) {
  console.log(i);
}

console.log(i); //ReferenceError: i is not
                  defined
```

```
for (var i = 0; i < 3; i++) {
  console.log(i);
}
console.log(i);
/*
0
1
2
3
*/
```

Learn more about them


THIS IS MY FAVORITE LANGUAGE
javascript
@redcoders
"11" + 1 => "111"
"11" - 1 => 10

== vs ===

== == converts the variable values to the same type before performing a comparison. This is called type coercion.
=== does not do any type conversion (coercion) and returns true only if both values and types are identical for the two variables being compared.

Ex:

```
if (4 == "4") {
  console.log("same");
} else {
  console.log("Not same");
}

// same
```

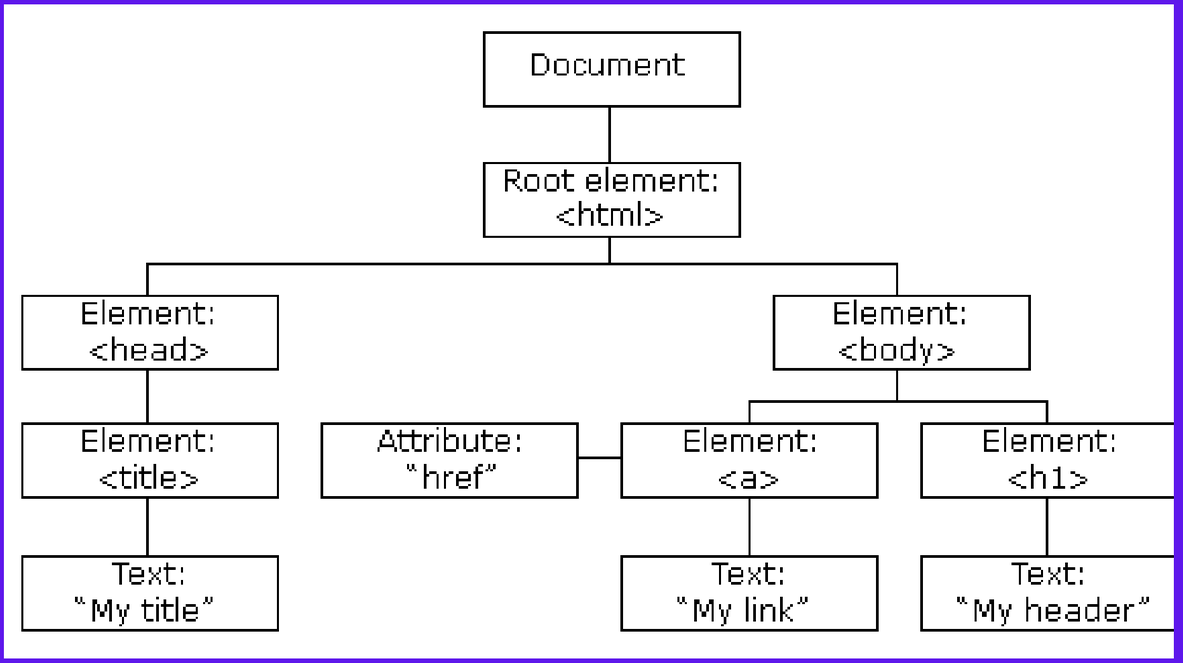
```
if (4 = = = "4") {
  console.log("same");
} else {
  console.log("Not same");
}

// Not same
```

Learn more about == and ===

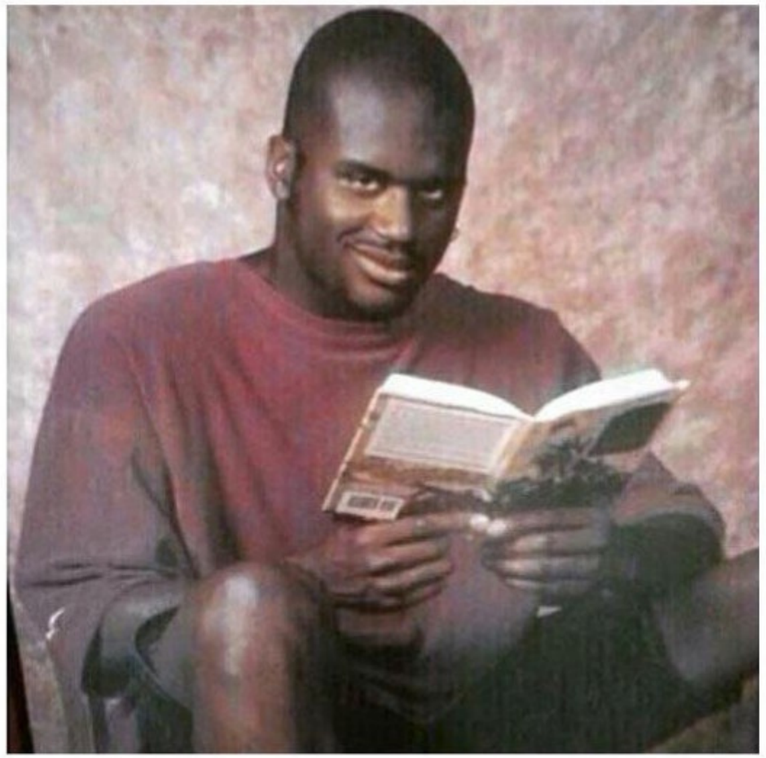
DOCUMENT OBJECT MODEL (DOM)

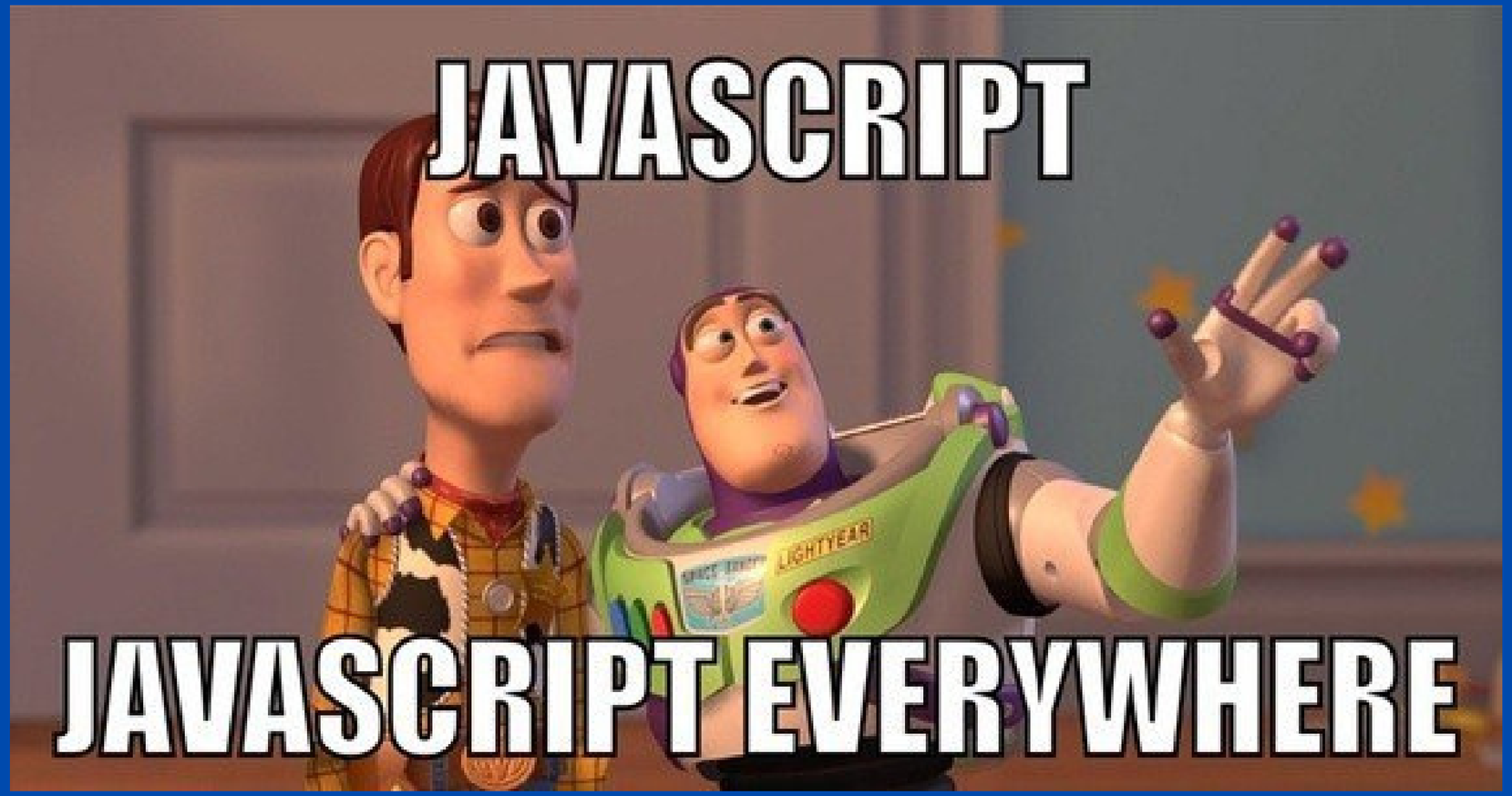
When a web page is loaded, the browser creates a Document Object Model of the page.
The HTML DOM model is constructed as a tree of Objects:




```
<!DOCTYPE html>
<html>
  <head>
    <title>My title</title>
  </head>
  <body>
    <h1>My header</h1>
    <a href="#">My link</a>
  </body>
</html>
```

Now I am a JavaScript Developer




JAVASCRIPT
JAVASCRIPT EVERYWHERE