

# Syntax of Javascript

## 1. Variables

Variables are used to store data. In JavaScript, you declare variables using `var`, `let`, or `const`.

```
let name = "John";           // Variable that can be reassigned
const age = 30;              // Constant variable that cannot be reassigned
var isStudent = true;        // Older way to declare variables, function-scoped
```

Copy

### ▼ Assignment

Create a variable for each of the following: your favorite color, your height in centimeters, and whether you like pizza. Use appropriate variable declarations (`let`, `const`, or `var`). Try logging it using `console.log`

## 2. Data types

```
let number = 42;                // Number
let string = "Hello World";    // String
let isActive = false;          // Boolean
let numbers = [1, 2, 3];        // Array
```

Copy

## 3. Operators

```
let sum = 10 + 5;           // Arithmetic operator
let isEqual = (10 === 10); // Comparison operator
let isTrue = (true && false); // Logical operator
```

Copy

## 4. Functions

```
// Function declaration
function greet(name) {
    return "Hello, " + name;
}

// Function call
let message = greet("John"); // "Hello, John"
```

Copy

### ▼ Assignment #1

Write a function `sum` that finds the sum of two numbers.

Side quest - Try passing in a string instead of a number and see what happens?

### ▼ Assignment #2

Write a function called `canVote` that returns true or false if the `age` of a user is > 18

## 5. If/Else

```
if (age >= 18) {  
    console.log("You are an adult.");  
} else {  
    console.log("You are a minor.");  
}
```

Copy

#### ▼ Assignment

Write an if/else statement that checks if a number is even or odd. If it's even, print "The number is even." Otherwise, print "The number is odd."

## 6. Loops

```
// For loop  
for (let i = 0; i < 5; i++) {  
    console.log(i); // Outputs 0 to 4  
}  
  
// While loop  
let j = 0;  
while (j < 5) {  
    console.log(j); // Outputs 0 to 4  
    j++;  
}
```

Copy

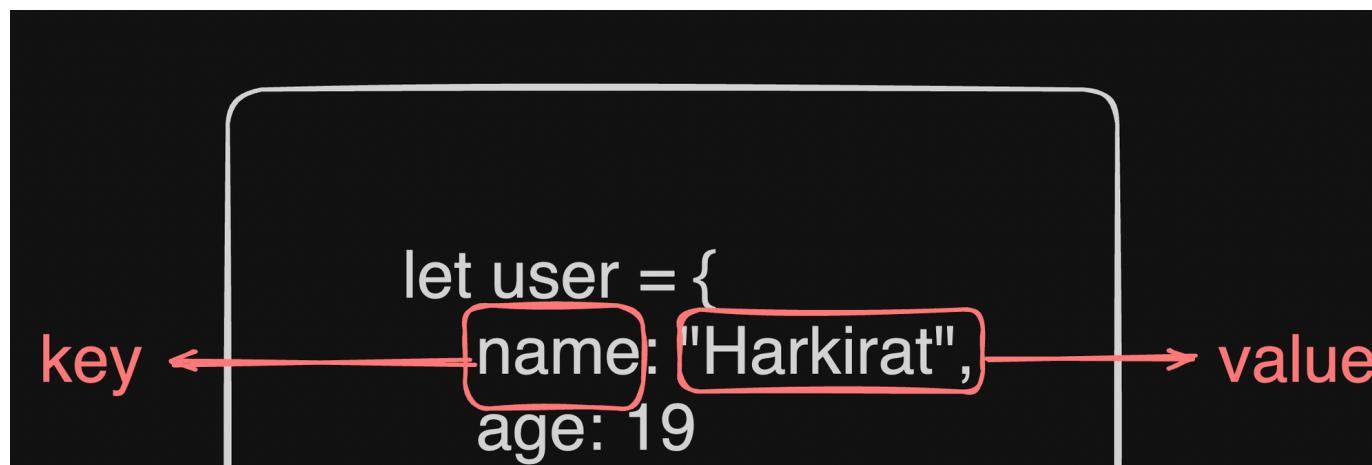
## ▼ Assignment

Write a function called sum that finds the `sum` from 1 to a number

# Complex types

## Objects

An object in JavaScript is a collection of `key-value pairs`, where each `key` is a string and each `value` can be any valid JavaScript data type, including another object.



```
}
```

```
let user = {  
    name: "Harkirat",  
    age: 19  
}  
  
console.log("Harkirats age is " + user.age);
```

Copy

▼ Assignment #1

Write a function that takes a `user` as an input and greets them with their name and age

▼ Assignment #2

Write a function that takes a new object as input which has `name` , `age` and `gender` and greets the user with their gender (Hi `Mr/Mrs/Others` harkirat, your age is 21)

▼ Assignment #3

Also tell the user if they are legal to vote or not

## Arrays

Arrays let you group data together

```
const users = ["harkirat", "raman", "diljeet"];
const totalUsers = users.length;
const firstUser = users[0];
```

Copy

▼ Assignment

Write a function that takes an array of numbers as input, and returns a new array with only even values. Read about [filter](#) in JS

## Array of Objects

We can have more complex objects, for example an array of objects

```
const users = [
    {
        name: "Harkirat",
        age: 21
    },
    {
        name: "raman",
        age: 22
    }
]

const user1 = users[0]
const user1Age = users[0].age
```

Copy

▼ Assignment

Write a function that takes an array of users as inputs and returns only the users who are

more than 18 years old

## Object of Objects

We can have an even more complex object (object of objects)

```
const user1 = {  
    name: "harkirat",  
    age: 19,  
    address: {  
        city: "Delhi",  
        country: "India",  
        address: "1122 DLF"  
    }  
}  
  
const city = user1.address.city;
```

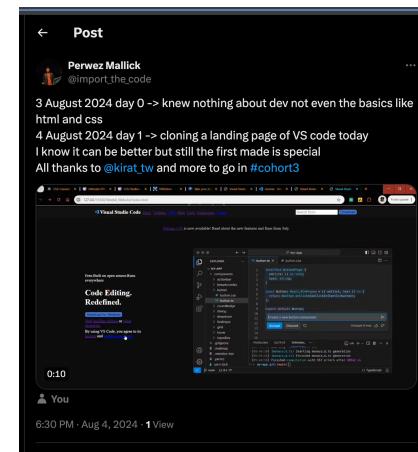
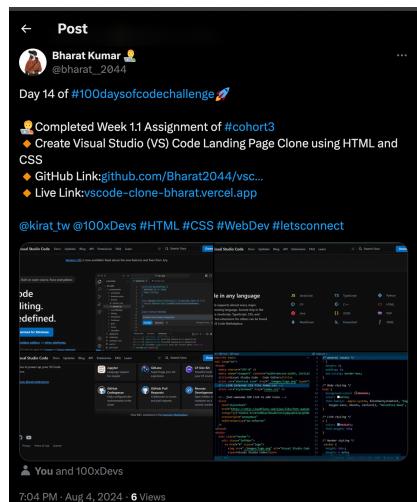
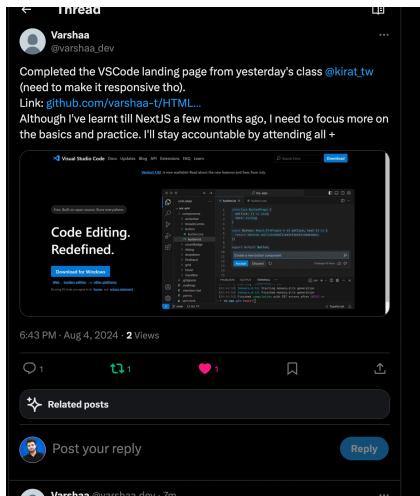
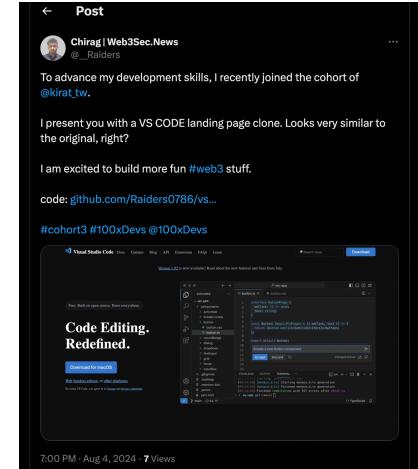
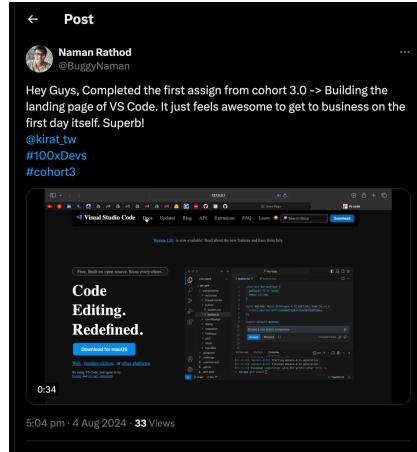
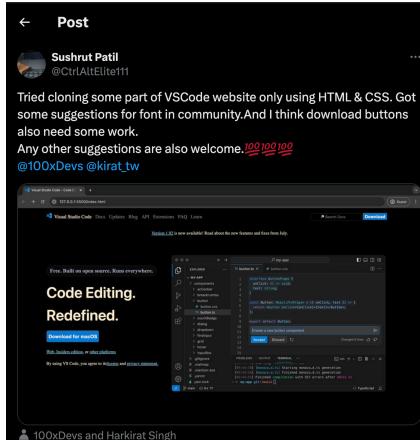
Copy

### ▼ Assignment

Create a function that takes an array of objects as input,  
and returns the users whose age > 18 and are male

# Did you code yesterday?

Did you try coding the VSCode landing page yesterday?



Shoutouts -

1. <https://x.com/CtrlAltElite111/status/1820076637477564416>
2. <https://x.com/BuggyNaman/status/1820060663319769462>
3. [https://x.com/import\\_the\\_code/status/1820082443506114582](https://x.com/import_the_code/status/1820082443506114582)
4. [https://x.com/varshaa\\_dev/status/1820085647190712649](https://x.com/varshaa_dev/status/1820085647190712649)
5. [https://x.com/\\_\\_Raiders/status/1820089916287828123](https://x.com/__Raiders/status/1820089916287828123)
6. [https://x.com/bharat\\_\\_2044/status/1820090993045020979](https://x.com/bharat__2044/status/1820090993045020979)

**Bounty - \$25 to each of you!**

# Javascript - The basics

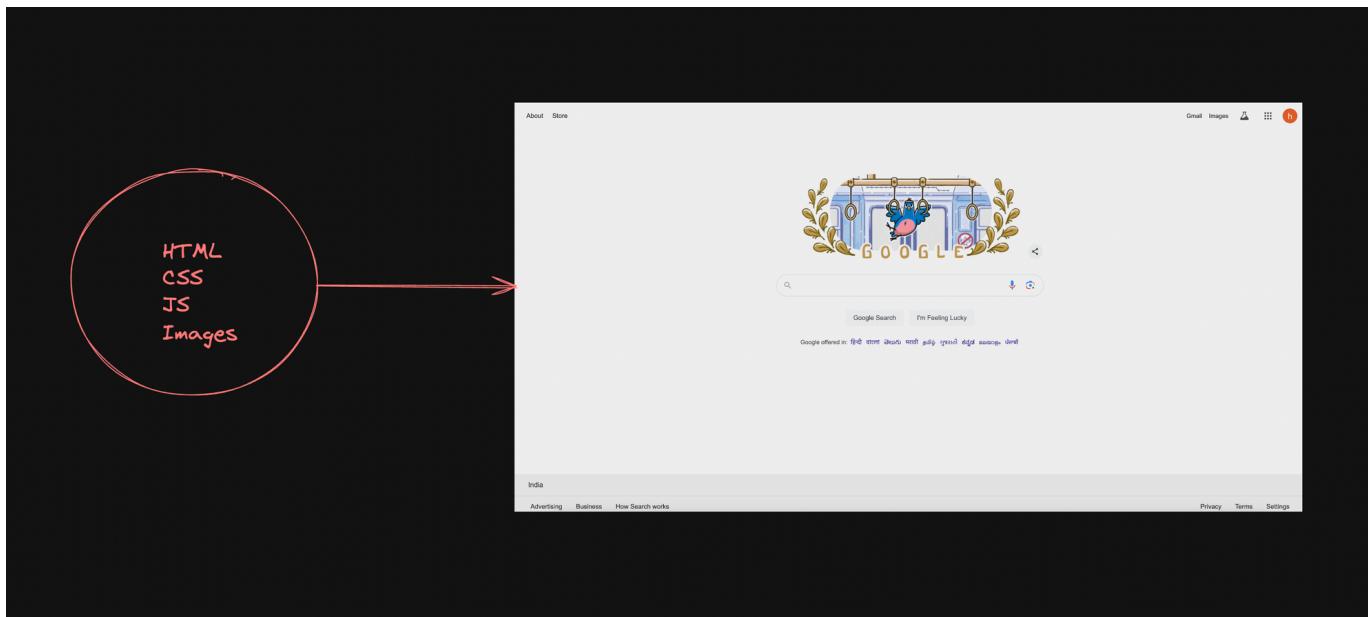
## Web development

Web development involves writing a lot of HTML, CSS and JS code.

Historically (and even today to some extend), browsers could only understand HTML, CSS and JS

Any website that you see, is a bunch of HTML, CSS and JS files along with some assets (images,

videos etc)



## Facts/Callouts

1. React, NextJS are **frameworks**. They compile down to HTML, CSS, JS in the end. That is what your browser understands.
2. When you run your C++ code on **leetcode**, it does not run on your browser/machine. It runs somewhere else. Your browser can't (almost) compile and run C++ code.
3. If someone asks — What all languages can your browser interpret, the answer is HTML, CSS, JS and WebAssembly. It can, technically, run C++/Rust code that is compiled down to Wasm

**Before we proceed, do one of the following -**

1. Create an account on repl.it
2. Install Node.js locally
3. Keep your browser console open for testing locally

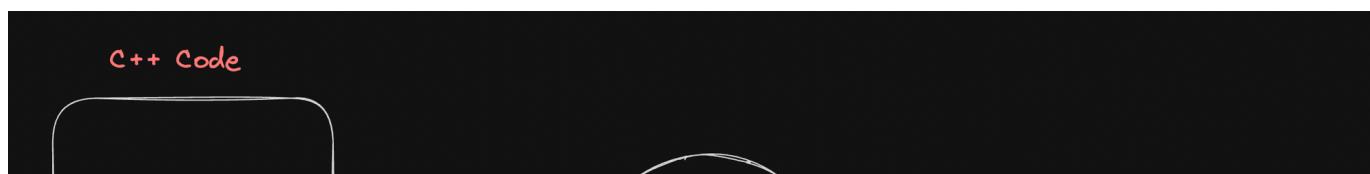
# Properties of JS

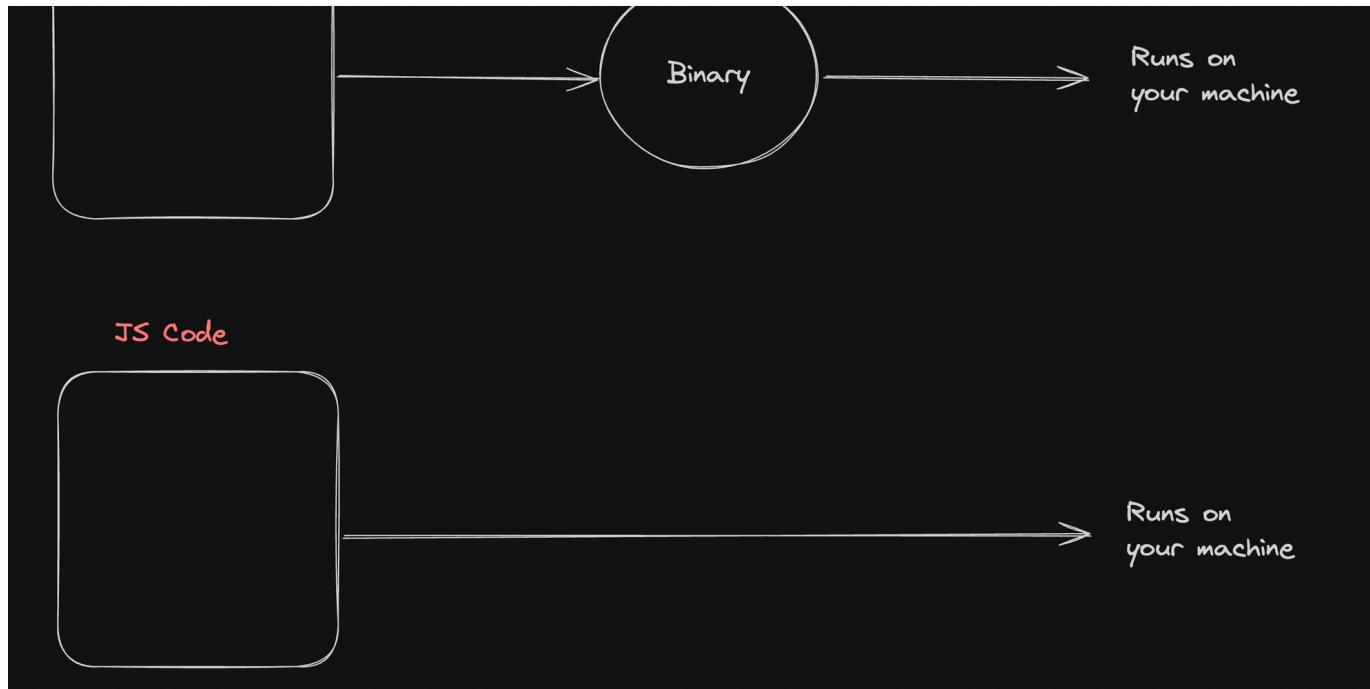
Every language comes with its unique set of features.

Javascript has the following -

## 1. Interpreted

JavaScript is an interpreted language, meaning it's executed line-by-line at runtime by the JavaScript engine in the browser or server environment, rather than being compiled into machine code beforehand.





### Upsides -

1. There is one less step to do before running your code

### Downsides -

1. Performance Overhead:
2. More prone to runtime errors

## 2. Dynamically Typed

Variables in JavaScript are not bound to a specific data type. Types are determined at runtime and can change as the program executes

## C++ Code (won't compile)

```
#include <iostream>

int main() {
    int a = 1;
    a = "hello";
    a = true;
}
```

Copy

## JS Code (will compile)

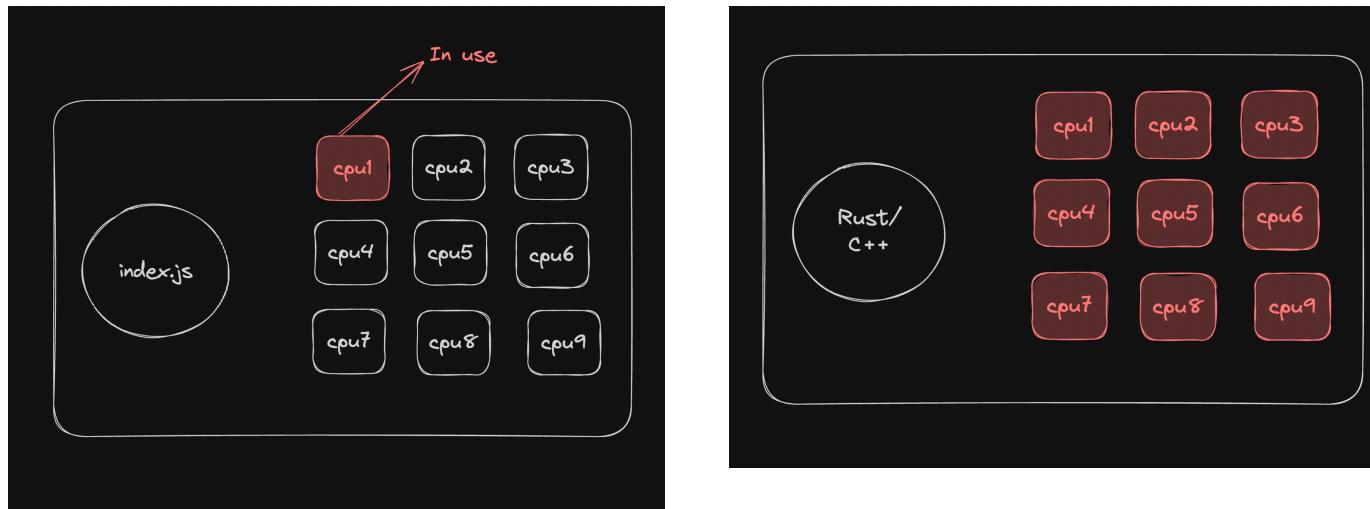
```
var a = 1;
a = "harkirat";
a = true;

console.log(a)
```

Copy

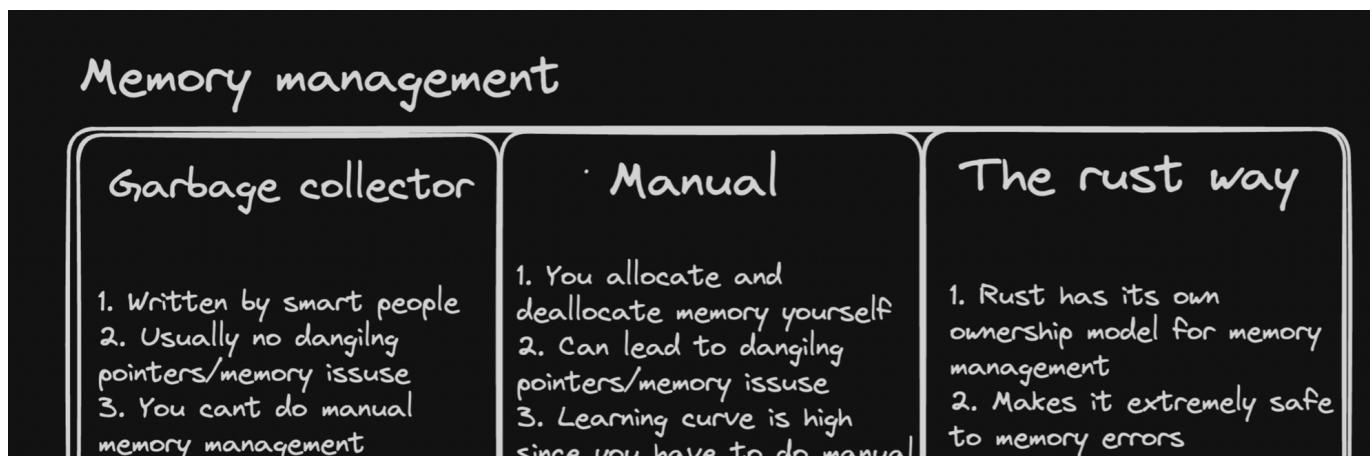
## 3. Single threaded

JavaScript executes code in a single-threaded environment, meaning it processes one task at a time. We will dive deeper into this next week.



## 4. Garbage collected

JavaScript automatically manages memory allocation and deallocation through garbage collection, which helps prevent memory leaks by automatically reclaiming memory used by objects no longer in use.



4. Examples - Java, JS

MM

3. Examples - C

## Conclusion

Is JS a good language?

Yes and no. It is beginner friendly, but has a lot of performance overhead. **Bun** is trying to solve for a lot of this, but there's a long way to go before JS can compete with languages like C++/Rust