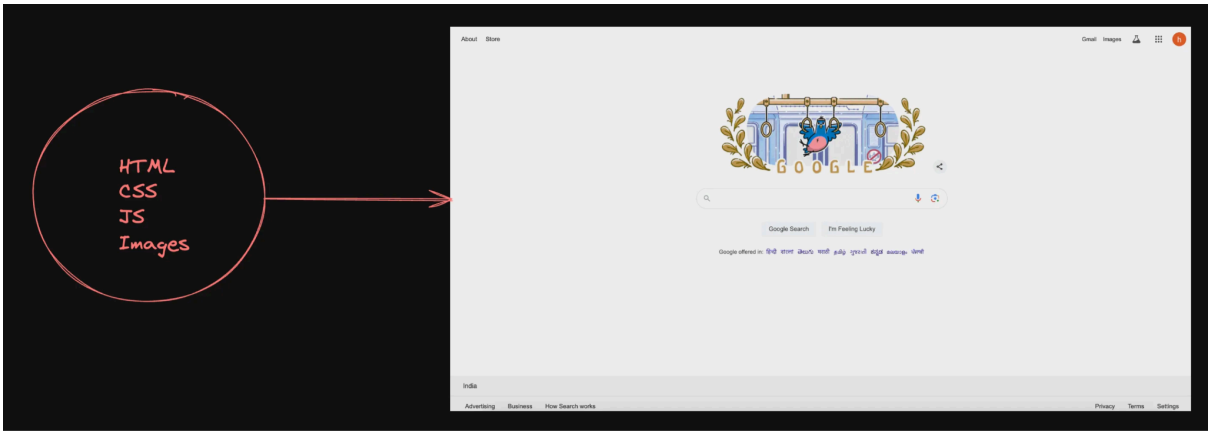
Javascript - The Basics

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:



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.

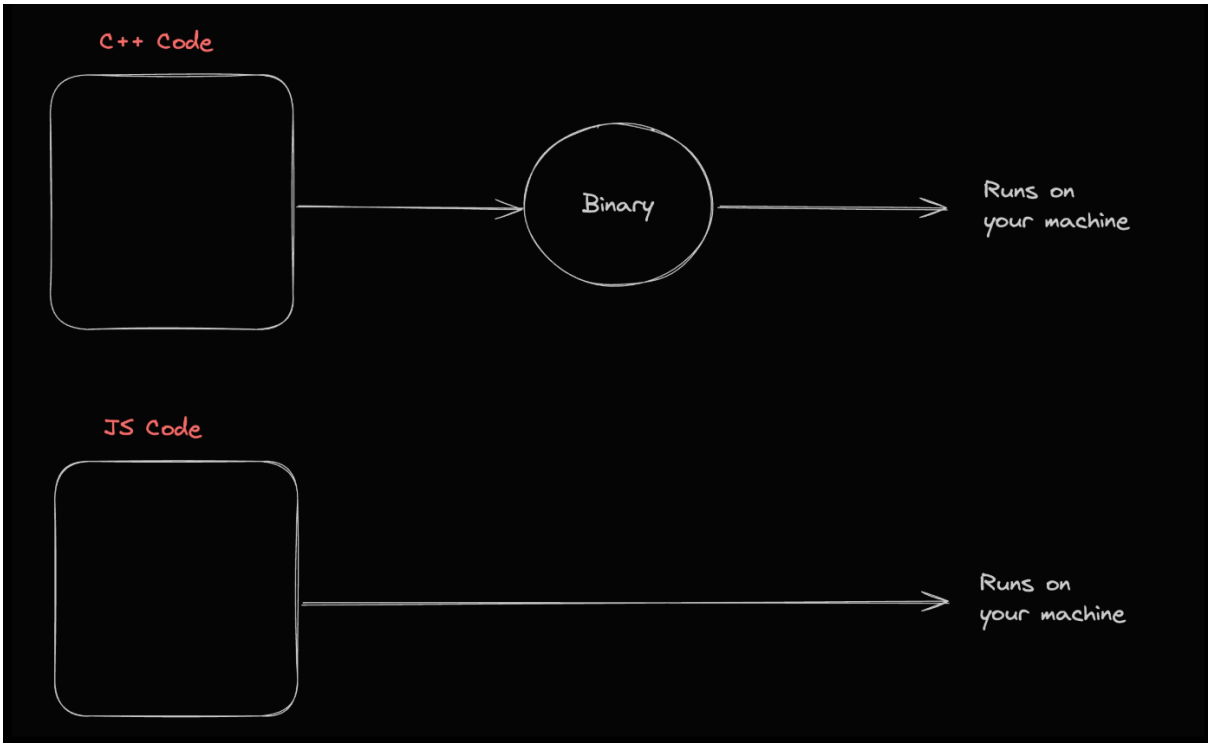
3. If someone asks – What all languages can your browser interpret, the answer is HTML, CSS, JS and WebAssembly. It can, technically, run C++/Rust code that is compiled down to Wasm

Every language comes with it's unique set of features.

Javascript has the following -

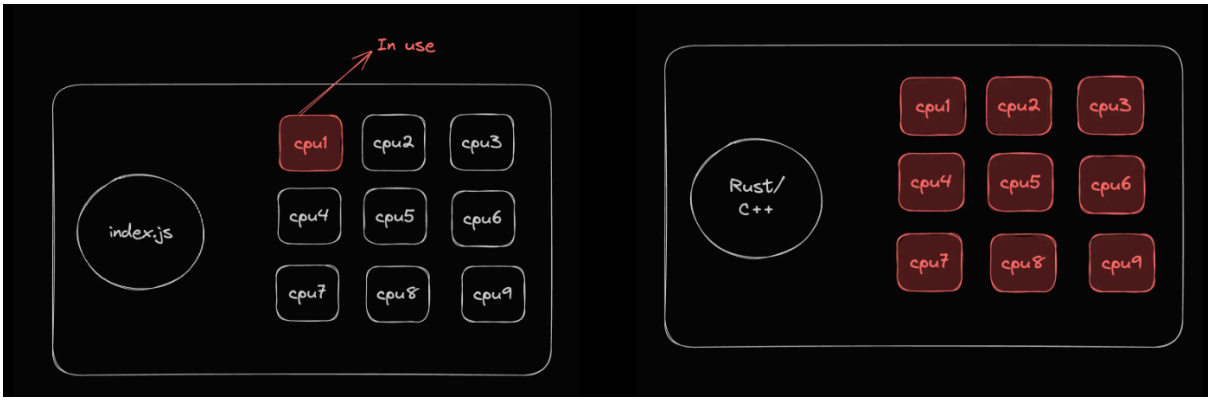
1. Interpreted

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.



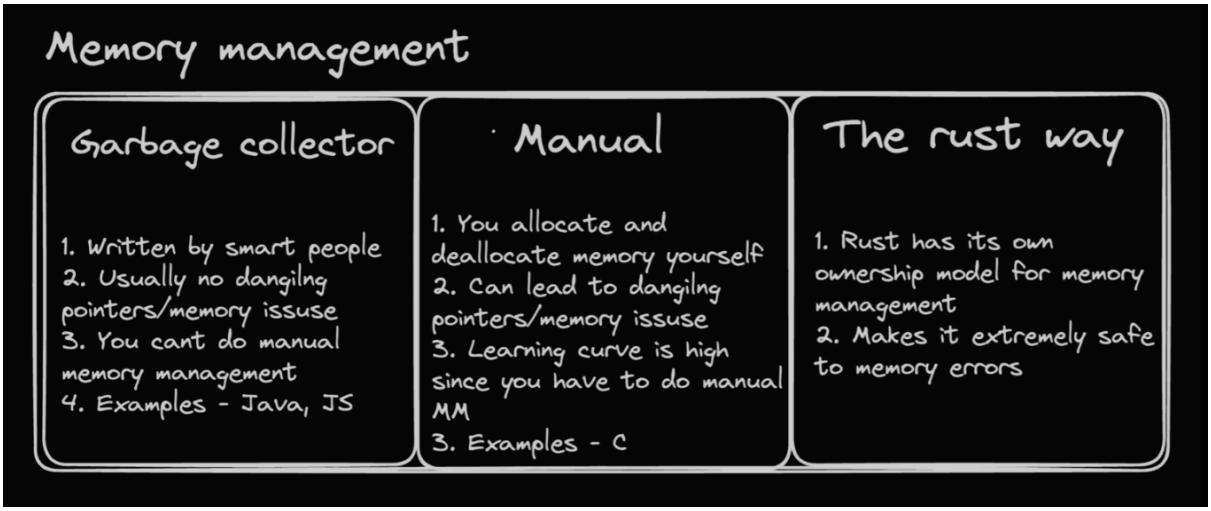
2. Dynamically Typed

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



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.

Syntax of Javascript

2. Data Types

JavaScript supports different types of data:

javascript Copy Edit

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

3. Operators

Operators perform operations on variables and values.

javascript Copy Edit

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

4. Functions

Functions are reusable blocks of code that perform a specific task.

javascript Copy Edit

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

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

5. If/Else

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

▶ Assignment

6. Loops

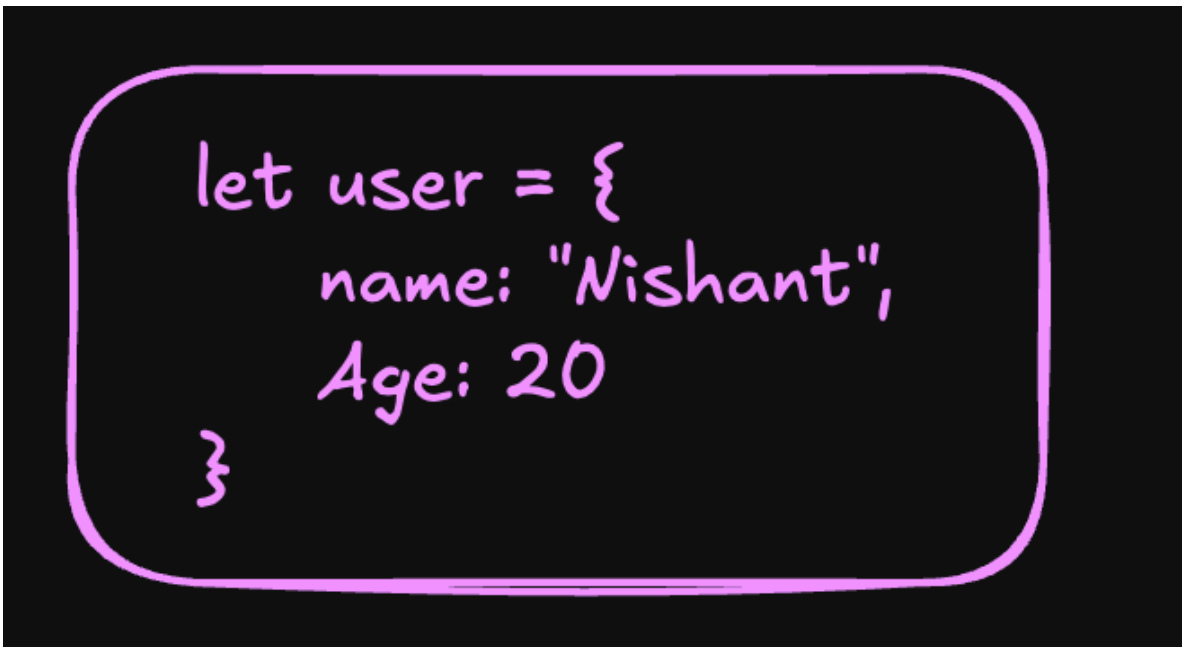
```
// For loop
for (let i = 0; i < 5; i++) {
   console.log(i); // Outputs 0 to 4
}

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

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.



Objects: Storing Data Efficiently

Objects are like mini databases that store related information together!

```
let user = {
    name: "Nishant",
    age: 19
};

console.log("Nishant's age is " + user.age);
```

📦 Arrays

Arrays hold multiple values in an ordered list, accessed by index.

🏗️ Array of Objects

Combines arrays and objects to manage multiple structured data entries.

🌍 Object of Objects

Objects can contain nested objects, making it easier to represent hierarchical data.

```
const users = [
    { name: "Nishant", age: 21 },
    { name: "Raman", age: 22 }
];

console.log(users[0].name);
console.log(users[1].age);
```

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

console.log(user1.address.city);
```

🔥 JavaScript Interview Theory Questions

1️⃣ What is JavaScript?

JavaScript is a high-level, interpreted programming language primarily used for web development. It allows developers to create dynamic and interactive web pages. It follows the ECMAScript standard and runs in the browser as well as on the server (Node.js).

2️⃣ Explain the difference between `var`, `let`, and `const`.

Keyword	Scope	Reassignment	Hoisting
`var`	Function	✅ Yes	✅ Hoisted but undefined
`let`	Block	✅ Yes	❌ Not initialized
`const`	Block	❌ No	❌ Not initialized

3 What are data types in JavaScript?

JavaScript has 8 data types:

1. Primitive: Number, String, Boolean, Undefined, Null, BigInt, Symbol
2. Non-Primitive: Object (includes Arrays, Functions)

4 What is the difference between == and ===?

== (Loose Equality) checks only values. It performs type conversion.

=== (Strict Equality) checks both values and types.

javascript Copy Edit

```javascript
console.log(5 == "5");  // true (Type Coercion)
console.log(5 === "5"); // false (Different types)
```

5 What is Hoisting in JavaScript?

Hoisting moves variable and function declarations to the top of their scope before code execution.

- **Variables using `var` are hoisted** but initialized as `undefined`.
- **Functions are fully hoisted** and can be used before declaration.
- `let` **and** `const` **are hoisted** but remain in a "temporal dead zone" until initialized.

javascript Copy Edit

```javascript
console.log(a); // undefined
var a = 10;

greet(); // Works
function greet() { console.log("Hello!"); }
```

6 What is a Closure in JavaScript?

A closure is a function that remembers variables from its outer scope even after execution.

7️⃣ Explain Event Loop in JavaScript.

JavaScript is single-threaded but uses an event loop to handle asynchronous tasks.

Call Stack executes synchronous code.

1. Web APIs handle async tasks (setTimeout, fetch).

2. Task Queue stores callback functions waiting to be executed.

3. Event Loop moves tasks from the queue to the call stack when it's empty.

8️⃣ What is the difference between null and undefined?

Feature	`null`	`undefined`
Meaning	Absence of value	Variable declared but not assigned
Type	Object	Undefined
Example	`let x = null;`	`let y;`

9️⃣ What is `this` in JavaScript?

`this` refers to the object that is executing the function.

- In **global scope**, `this` refers to `window` (browser) or `global` (Node.js).
- In **object methods**, `this` refers to the object itself.
- In **arrow functions**, `this` is lexically inherited (doesn't change).

javascript Copy Edit

```javascript
const obj = {
    name: "Nishant",
    greet: function() {
        console.log(this.name);
    }
};
obj.greet(); // Nishant
```

1️⃣1️⃣ What are Promises in JavaScript?

Promises handle asynchronous operations and have three states:

Pending – Initial state

Resolved – Success

Rejected – Error