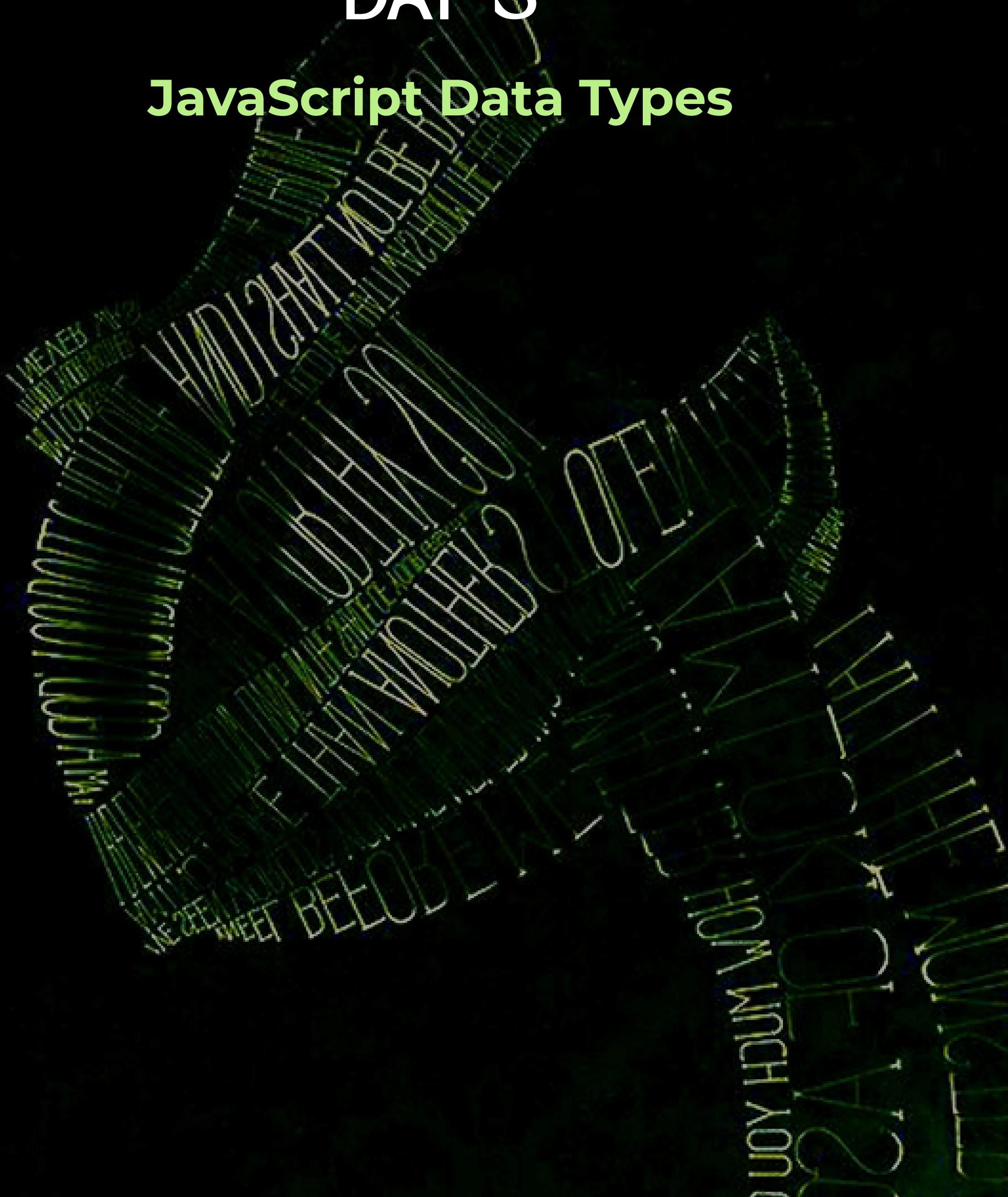
Sheryians
Coding School

Live Cohort

DAY-3

JavaScript Data Types



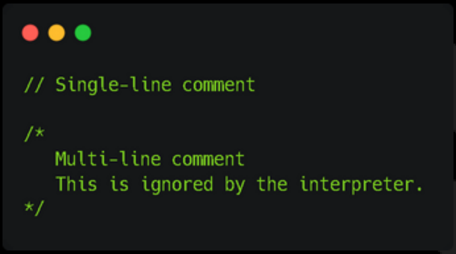
JavaScript Data Types

In JavaScript, **data types** define the type of data that can be stored and manipulated within a program. Understanding data types is essential because they determine how operations are performed on values.

🗒️ 1. Comments in JavaScript

Comments are used to make code readable and explain what it does. They are ignored by the JavaScript engine.

🔹 Types of Comments:

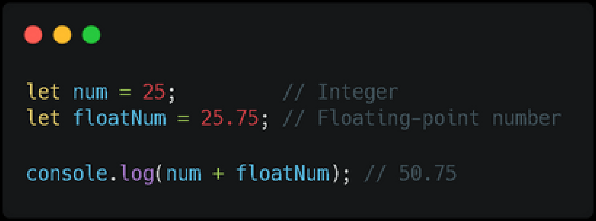


```javascript
// Single-line comment

/*
   Multi-line comment
   This is ignored by the interpreter.
*/
```

🔢 2. Numbers and Floats

JavaScript uses a single type of number: `**number**`. It can represent both integers and floating-point numbers.

🧩 Example:



```javascript
let num = 25;          // Integer
let floatNum = 25.75; // Floating-point number

console.log(num + floatNum); // 50.75
```

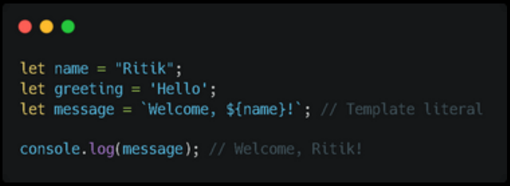
- JavaScript numbers are **64-bit floating-point values** (IEEE 754 format).

JavaScript Data Types

🧵 3. Strings

A **string** is a sequence of characters used to represent text.

🧩 Example:



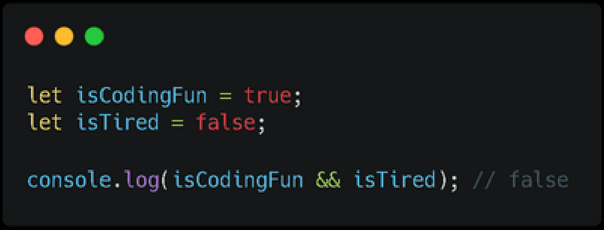
```
let name = "Ritik";
let greeting = 'Hello';
let message = `Welcome, ${name}!`; // Template literal

console.log(message); // Welcome, Ritik!
```

- Strings can be created using single (`'`), double (`"`) or backticks (`` ` ``).

⚙️ 4. Booleans

A **boolean** represents one of two values: `true` or `false`.

🧩 Example:



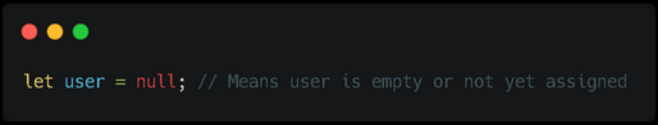
```
let isCodingFun = true;
let isTired = false;

console.log(isCodingFun && isTired); // false
```

- Booleans are often used in conditional statements.

🚫 5. Null

The `` `null` `` type represents an **intentional absence** of any object value.

🧩 Example:



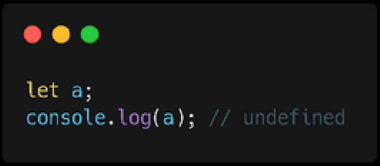
```
let user = null; // Means user is empty or not yet assigned
```

- `` `typeof null` `` returns `` `"object"` `` — this is a historical bug in JavaScript.

🧮 6. Undefined

A variable that has been declared but not assigned a value is **undefined.**

🧩 Example:



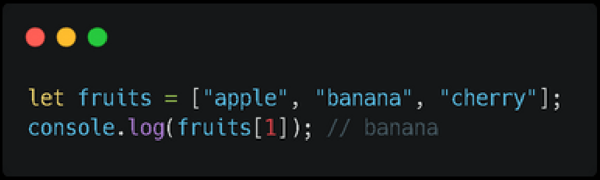
```
let a;
console.log(a); // undefined
```

- `` `undefined` `` means a variable exists but has no value assigned.

🧰 7. Arrays

An **array** is a collection of elements, which can be of any data type.

🧩 Example:



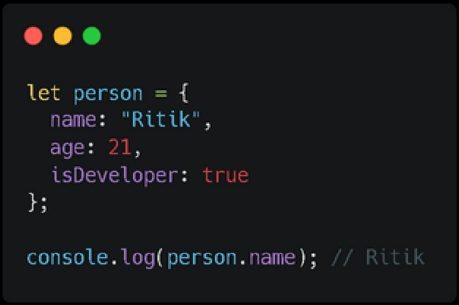
```
let fruits = ["apple", "banana", "cherry"];
console.log(fruits[1]); // banana
```

- Arrays are **objects** in JavaScript. `typeof fruits` returns `"object"`.

🧱 8. Objects

Objects store data in **key-value pairs**. Each property has a name (key) and a value.

🧩 Example:



```
let person = {
  name: "Ritik",
  age: 21,
  isDeveloper: true
};

console.log(person.name); // Ritik
```

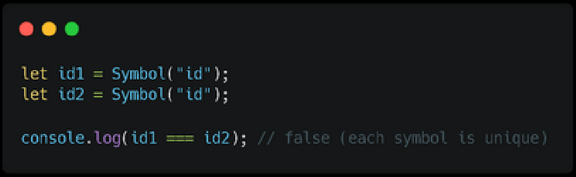
- Objects are the most important data type in JavaScript.

JavaScript Data Types

🟡 9. Symbol

A **Symbol** is a unique and immutable data type introduced in ES6.

🧩 Example:



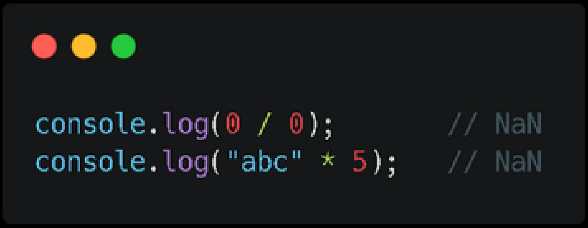
```
let id1 = Symbol("id");
let id2 = Symbol("id");

console.log(id1 === id2); // false (each symbol is unique)
```

- We will learn more about Symbols when we study objects in detail.

❓ 10. NaN (Not-a-Number)

NaN represents a value that is not a valid number.

🧩 Example:



```
console.log(0 / 0);        // NaN
console.log("abc" * 5);    // NaN
```

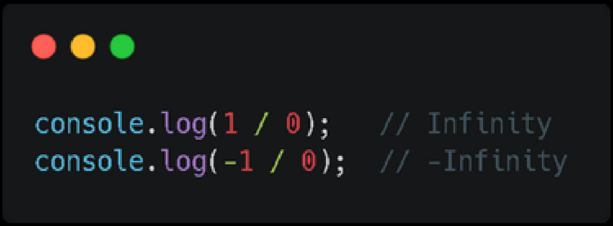
- Use `isNaN()` to check if a value is NaN.

JavaScript Data Types

♾️ 11. Infinity and -Infinity

`Infinity` represents a number too large to be represented in JavaScript.

🧩 Example:



```
console.log(1 / 0);    // Infinity
console.log(-1 / 0);   // -Infinity
```

- You can use it for mathematical comparisons.

🧩 12. Primitive and Reference Data Types

JavaScript data types are divided into two **main categories:**

a) 🧠 Primitive Types (Stored by value)

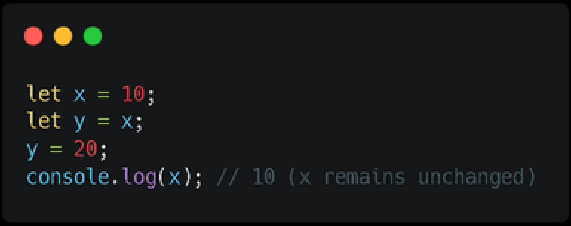
- Number
- String
- Boolean
- Undefined
- Null
- Symbol
- BigInt

- Each primitive value is **immutable** and stored directly in memory.

JavaScript Data Types

a) 🧠 Primitive Types (Stored by value)

🧩 Example:



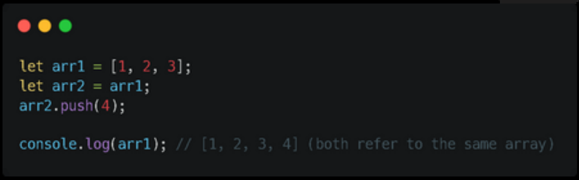
```
let x = 10;
let y = x;
y = 20;
console.log(x); // 10 (x remains unchanged)
```

b) 📦 Reference Types (Stored by reference)

- Object
- Array
- Function

- These are stored as references in memory.

🧩 Example:



```
let arr1 = [1, 2, 3];
let arr2 = arr1;
arr2.push(4);

console.log(arr1); // [1, 2, 3, 4] (both refer to the same array)
```

JavaScript Data Types

✅ Summary Table

Type	Example	Category	Description
Number	`42`, `3.14`	Primitive	Numeric values
String	`'hello'`, `"world"`	Primitive	Text values
Boolean	`true`, `false`	Primitive	Logical values
Null	`null`	Primitive	Empty value
Undefined	`undefined`	Primitive	Uninitialized variable
Symbol	`Symbol("id")`	Primitive	Unique value identifier
Object	`{name: "Ritik"}`	Reference	Collection of data
Array	`[1,2,3]`	Reference	Ordered list
NaN	`NaN`	Primitive	Not a number
Infinity	`Infinity`	Primitive	Beyond numeric limit

🏁 Conclusion

- Understanding data types is the foundation of JavaScript. They help you decide how data should be stored, used, and manipulated efficiently.
- Next, we will dive deeper into **objects and symbols** to understand complex data handling in JavaScript.