

# LECTURE 2 — JAVASCRIPT **DATA TYPES**

---

## JavaScript Fundamentals — Data Types (Primitive)

 First Principles • Real-Life Mapping • Zero Confusion

---

## FIRST PRINCIPLE — “DATA TYPE KYU CHAHIYE?”

### Simple Soch (Real-Life Example)

Socho tum ek **almirah** ho:

- Kapde alag jagah
- Paise alag jagah
- Documents alag jagah

#### Reason:

Har cheez ka **type alag hota hai**, isliye use handle karne ka tareeqa bhi alag hota hai.

### Same rule programming me apply hota hai

#### Data Type decide karta hai:

- Variable ke andar **kis type ka data store hogा**
- Us data ke sath **kaunse operations allowed honge**

---

### Real-Life Mapping (Must Understand)

Real Life	JavaScript Data Type
-----------	----------------------

Instagram comment **String**

Like count **Number**

Block status **Boolean**

❖ Isliye data types JavaScript ka foundation hote hain

---

## ■ EXAMPLE — DATA TYPE IN ACTION

```
let comment = "Nice pic!";  
  
console.log(comment); // Nice pic!  
  
console.log(typeof comment); // string
```

### ■ HIGHLIGHT

`typeof` operator batata hai variable ka data type kya hai.

---

## ■ TYPES OF DATA IN JAVASCRIPT

◆ JavaScript me data 2 categories me hota hai

### 1 Primitive Data Types

- Simple values
- Immutable (change hone par nayi value banti hai)

### 2 Non-Primitive Data Types

- Complex structures (Array, Object, Function)

### ■ IMPORTANT

👉 Is lecture me sirf Primitive Data Types cover ho rahe hain

---

## ■ ■ ■ PRIMITIVE DATA TYPES ■ ■ ■

### ① NUMBER

◆ Kya Represent Karta Hai?

- Integers (10, 20, -5)
- Decimals (10.5, 5000.75)

## ◆ Use Cases

- Age
- Balance
- Likes

```
let balance = 5000.75;

console.log(balance);          // 5000.75

console.log(typeof balance); // number
```

### ■ NOTE

JavaScript me **integer aur decimal dono ka type = number** hota hai.

---

## ② STRING

### ◆ Kya Represent Karta Hai?

Textual data jo likha jata hai:

- 'single quotes'
- "double quotes"
- `backticks`

### ◆ Use Cases

- Username
- Comments
- Messages

```
let username = "Harshal";

console.log(username);

console.log(typeof username); // string
```

```
let post = "Rohit is a bad boy";  
console.log(post);  
console.log(typeof post); // string
```

## ■ HIGHLIGHT

Quotes ke andar jo bhi hota hai → **STRING**

---

## ③ BOOLEAN

### ◆ Kya Represent Karta Hai?

Sirf 2 values:

- `true`
- `false`

### ◆ Use Cases

- Login status
- Block status
- Admin check

```
let isBlocked = true;  
  
console.log(isBlocked); // true  
  
console.log(typeof isBlocked); // boolean
```

## ■ REAL-LIFE ANALOGY

Switch ON/OFF → Boolean true/false

---

## ④ UNDEFINED

### ◆ Meaning

- Variable declare kiya
- Lekin value assign nahi ki

### ◆ Use Cases

- Input field empty
- Data abhi receive nahi hua

```
let searchInput;

console.log(searchInput);           // undefined

console.log(typeof searchInput); // undefined
```

### ■ IMPORTANT

Undefined ka matlab: “**Value abhi di hi nahi gayi**”

---

## ⑤ NULL

### ◆ Meaning

- Intentionally empty value
- Developer khud bol raha hai: “yahan kuch nahi hai”

### ◆ Real-Life Case

Server down hai  
Isliye **0 (zero)** bhejna galat hogा  
**👉 null bhejna sahi hota hai**

```
let result = null;

console.log(result);           // null

console.log(typeof result); // object
```

---

## ⚠ BIGGEST JS BUG — `typeof null`

### 🔴 Weird Behavior

```
typeof null // "object"
```

### 🧠 Reason (First Principle)

- JavaScript ka **old bug**
- Pehle type **memory address pattern** se detect hota tha
- `null` ka pattern **object jaisa lagta tha**

### ✖ Agar aaj fix kar diya:

- Purana code toot jaayega

### 🟩 Conclusion

```
typeof null === "object"
```

👉 Bug hai, par backward compatibility ke liye rakha gaya hai

---

## ⑥ BIGINT

### ♦ Purpose

Bahut **large integers** ko store karne ke liye

### ◆ Use Cases

- Aadhaar number
- Cryptographic keys

```
let bigValue = 123456789123456789123456789n;  
  
console.log(bigValue);  
  
console.log(typeof bigValue); // bigint
```

### 🟡 HIGHLIGHT

`n` suffix → JavaScript ko batata hai ye **BigInt** hai

## ⑦ SYMBOL

### ◆ Meaning

- Unique
- Immutable identifier

### ◆ Use Cases

- Object keys
- Hidden / private values

```
let userId = Symbol("id");

console.log(userId);

console.log(typeof userId); // symbol
```

### ■ REAL-LIFE ANALOGY

Aadhaar number → unique  
Symbol bhi waise hi **unique identity** deta hai

## ■ ■ ■ SUMMARY TABLE ■ ■ ■

Data Type	Meaning	Example	typeof
Number	Integer / Decimal	<code>let x = 45.6</code>	<code>"number"</code>
String	Text	<code>"Raj"</code>	<code>"string"</code>
Boolean	true / false	<code>false</code>	<code>"boolean"</code>
Undefined	Value not assigned	<code>let y;</code>	<code>"undefined"</code>

Null      Intentional empty      `null`      "object"

BigInt      Large integers      `123n`      "bigint"

Symbol      Unique identifier      `Symbol()`      "symbol"

---

## ■ ■ ■ FINAL POWER SUMMARY ■ ■ ■

### ■ ONE-GLANCE REVISION

- Data types decide **data + operations**
- JavaScript me **2 categories** hote hain
- Primitive = simple & immutable
- `typeof` type batata hai
- `null` ka "object" ek **JS bug** hai
- BigInt = large numbers
- Symbol = unique identity