



Live Cohort

Notes Day 15



Day 15 : Primitive Data Types in JavaScript

Primitive data types are immutable and stored directly in memory.

1. Number (Integer & Floating-point)

```
let num = 42;  
let floatNum = 3.14;
```

2. String (Text enclosed in quotes)

```
let str = "Hello World!";
```

3. Boolean (Represents true or false)

```
let isAvailable = true;
```

4. Null (Represents an empty or non-existent value)

```
let emptyValue = null;
```

5. Undefined (Declared but not assigned a value)

```
let notDefined;
```

6. Symbol (Unique and immutable value)

```
let sym = Symbol("unique");
```

7. BigInt (Handles large numbers beyond Number.MAX_SAFE_INTEGER)

```
let bigInt = 123456789012345678901234567890n;
```

Day 15 : Reference (Relative) Data Types in JavaScript

Reference data types are stored in memory by reference and can be modified.

1. Object (Collection of key-value pairs)

```
let person = { name: "John", age: 30 };
```

2. Array (Ordered list of values)

```
let fruits = ["Apple", "Banana", "Cherry"];
```

Day 15 : JavaScript Data Types

JavaScript has 8 data types, categorized into Primitive and Reference types.

Primitive Data Types:

- Number → let num = 10;
- String → let text = "Hello";
- Boolean → let isActive = false;
- Null → let data = null;
- Undefined → let notDefined;
- Symbol → let sym = Symbol("id");

Reference Data Types:

- Array → let list = [1, 2, 3];
- Object → let obj = { key: "value" };

Example:

```
let num = 10; // number
let text = "Hello"; // string
let isActive = false; // boolean
let data = null; // null
let list = [1, 2, 3]; // array
let obj = { key: "value" }; // object
let sym = Symbol("id"); // symbol
let notDefined; // undefined
```

Day 15 : Some Important Values in JavaScript

1. **undefined** (Variable declared but not assigned a value)

2. **null** (Intentional absence of value)

3. **NaN ("Not-a-Number" - Invalid mathematical operations)**

4. **Infinity (Represents an infinite value)**

Example:

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

let price = null; // null (No price assigned yet)
console.log(price); // null

let result = "hello" / 2; // NaN (Invalid operation)
console.log(result); // NaN

let infiniteNumber = 10 / 0; // Infinity
console.log(infiniteNumber); // Infinity
```

Day 15 : Basic Operators in JavaScript

1. Arithmetic Operators (+, -, *, /, %, ++, --)

Example:

```
let a = 10;
let b = 5;

console.log(a + b); // 15  (Addition)
console.log(a - b); // 5   (Subtraction)
console.log(a * b); // 50  (Multiplication)
console.log(a / b); // 2   (Division)
console.log(a % b); // 0   (Modulus – Remainder)

a++;
console.log(a); // 11  (Increment)

b--;
console.log(b); // 4   (Decrement)
```

2 . Assignment Operators (=, +=, -=, *=, /=, %=)

Example:

```
let x = 10;

x += 5; // Equivalent to x = x + 5
console.log(x); // 15

x -= 3; // Equivalent to x = x - 3
console.log(x); // 12

x *= 2; // Equivalent to x = x * 2
console.log(x); // 24

x /= 4; // Equivalent to x = x / 4
console.log(x); // 6

x %= 5; // Equivalent to x = x % 5
console.log(x); // 1
```

3 . Comparison Operators (==, ===, !=, !==, >, <, >=, <=)

Example:

```
console.log(5 == "5"); // true (loose equality, type conversion happens)
console.log(5 === "5"); // false (strict equality, no type conversion)

console.log(10 != "10"); // false (loose inequality)
console.log(10 !== "10"); // true (strict inequality)

console.log(8 > 5); // true
console.log(8 < 5); // false
console.log(8 >= 8); // true
console.log(8 <= 10); // true
```

4 . Logical Operators (&&, ||, !)

Example:

```
console.log(true && false); // false (Both conditions must be true)
console.log(true || false); // true (At least one condition must be true)
console.log(!true); // false (Negates the value)
```

Day 15 : Variable Hoisting in JavaScript

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

✓ Using var (Hoisted but Undefined)

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

✗ Using let and const (Hoisted but Not Initialized)

```
console.log(b); // ReferenceError: Cannot access 'b' before initialization  
let b = 10;
```

Key Takeaway:

- var is hoisted with an initial value of undefined.
- let and const are hoisted but remain in the Temporal Dead Zone until assigned.

Day 15 : Condition Operators in JavaScript

1 . if-else Statement

- Used for basic conditional checks.

```
let age = 18;
if (age >= 18) {
  console.log("Adult");
} else {
  console.log("Minor");
}
```

2 . Ternary Operator

- A shorthand way of writing if-else statements.

```
let status = age >= 18 ? "Adult" : "Minor";
console.log(status); // Output: Adult
```

Key Takeaway:

- if-else gives more flexibility for multiple conditions and is easy to read.
- The ternary operator is great for simple conditions in a single line.

Day 15 : Loops in JavaScript

1 . for Loop

- Used when the number of iterations is known beforehand.

```
for (let i = 0; i < 5; i++) {  
    console.log(i); // Output: 0, 1, 2, 3, 4  
}
```

2 . while Loop

- Used when the loop should run as long as a condition is true.

```
let i = 0;  
while (i < 5) {  
    console.log(i); // Output: 0, 1, 2, 3, 4  
    i++;  
}
```

3 . do...while Loop

- Runs the block of code once, then checks the condition.

```
let j = 0;  
do {  
    console.log(j); // Output: 0, 1, 2, 3, 4  
    j++;  
} while (j < 5);
```

Assignments:

- 1 . Age Category Message – Ask the user for their age. If they are under 18, print “You are a minor.” If they are between 18 and 60, print “You are an adult.” If they are above 60, print “You are a senior citizen.”
- 2 . Even or Odd Sum – Take two numbers from the user using `prompt()`. If the sum of both numbers is even, print “Even Sum”; otherwise, print “Odd Sum.”
- 3 . Character Case Checker – Ask the user for a single character. Check if it's uppercase, lowercase, or neither (not a letter).
- 4 . Largest of Three Numbers – Take three numbers as input and print the largest number among them without using `Math.max()`.
- 5 . Leap Year Checker – Ask the user for a year and determine if it's a leap year or not.
- 6 . Simple Calculator – Ask the user for two numbers and an operator (+, -, *, /). Perform the operation and display the result.

7 . Positive, Negative, or Zero – Take a number input and check if it is positive, negative, or zero.

8 . User Greeting – Ask for the user's name and time (24-hour format). Greet them accordingly:

5 AM–12 PM: "Good Morning, [Name]!"

12 PM–5 PM: "Good Afternoon, [Name]!"

5 PM–9 PM: "Good Evening, [Name]!"

9 PM–5 AM: "Good Night, [Name]!"

9 . Traffic Light System – Ask the user for a traffic light color (red, yellow, green). Print appropriate messages:

Red: "Stop!"

Yellow: "Get Ready!"

Green: "Go!"

10 . Multiplication Table – Ask the user for a number and print its multiplication table up to 10.

11 . 11. Grade Calculator – Ask the user for their marks (0-100). Assign grades based on the range:

90-100: A

80-89: B

70-79: C

60-69: D

Below 60: F

12 . Simple Login System – Set a predefined username and password. Ask the user to enter their credentials using `prompt()`. If correct, print “Login Successful”; otherwise, print “Incorrect username or password.”

- 13 . Swapping Without Third Variable – Take two numbers from the user and swap their values without using a third variable.
- 14 . FizzBuzz (Multiple of Both) – Ask the user for a number. If it's a multiple of both 3 and 5, print “FizzBuzz”; if only 3, print “Fizz”; if only 5, print “Buzz”; otherwise, print the number itself.
- 15 . Number Reversal – Take a three-digit number from the user and print its reverse. (Example: 123 → 321).
- 16 . Sum of Digits – Take a number from the user and print the sum of its digits. (Example: 123 → $1+2+3 = 6$).
- 17 . Palindrome Checker – Ask the user for a word. Check if it reads the same forward and backward. Print “Palindrome” or “Not a Palindrome.”
- 18 . Reverse Without String Methods – Ask the user for a number and reverse it without using `.split()`, `.reverse()`, or `.join()`.
- 19 . Find Second Largest – Take three numbers as input and find the second largest number (without using `sort()` or `Math.max()`).
- 20 . Find First Non-Repeating Character – Ask the user for a word and find the first character that does not repeat. Example: hello → h (since e, l, and o repeat).

- 21 . Even Digit Counter – Take a number from the user and count how many even digits it has.
- 22 . Nested Condition Challenge – Ask the user for their age and salary. Print a message based on conditions:
- If age is below 18, print “Not eligible”
 - If age is 18 or more but salary is less than ₹20,000, print “Low Salary”
 - If salary is ₹50,000 or more, print “High Salary”
 - . Otherwise, print “Medium Salary”
- 23 . Toggle Case – Ask the user for a word and toggle the case of every character. Example: HeLLo → hElLo.
- 24 . Find the Missing Number in a Sequence – Take a list of consecutive numbers (except one missing) and find the missing number. Example: 1, 2, 3, 5 → Missing number is 4.
- 25 . Convert Number to Words – Take a single-digit number and print it in words (Example: 1 → One, 2 → Two).