

# JavaScript Primitive Data Types

**Instructor:** Pramod Kumar Jena

---

## Topics to Cover:

1. `null`
  2. `undefined`
  3. `number`
  4. `bigInt`
  5. `symbol`
  6. `NaN`
  7. `boolean`
  8. `string`
- 

## 1. Null

- **Definition:** `null` is a primitive data type in JavaScript. It represents the intentional absence of any object value.
- **Use Case:** `null` is used when a variable should explicitly have no value.

```
let user = null; // No user logged in yet
console.log(user); // null
```

```
user = "Pramod";
console.log(user); // "Pramod"
```

- **Real-life example:** You can set a `currentUser` as `null` when no one is logged into the system, and later update it when a user logs in.
- 

## 2. Undefined

- **Definition:** `undefined` is a primitive data type used to denote a variable that has been declared but not assigned any value yet.
- **Use Case:** It commonly appears when a variable is declared but not initialized.

```
let dataFetched; // variable is declared but not initialized
console.log(dataFetched); // undefined
```

- **Real-life example:** When fetching data from an API, the variable `dataFetched` remains undefined until the data arrives.
- 

### 3. Number

- **Definition:** `number` represents both integers and floating-point numbers in JavaScript.
- **Use Case:** It is used for any numeric calculations.

```
let age = 25;
console.log(age); // 25
```

```
let pi = 3.1416;
console.log(pi); // 3.1416
```

#### Mathematical Operations Task:

```
let num1 = 100;
let num2 = 200;

let addition_result = num1 + num2; // 300
let subtraction_result = num2 - num1; // 100
let multiplication_result = num1 * num2; // 20000
let division_result = num2 / num1; // 2
```

**Task-1:** Do the above task and upload it on GitHub, then share the link in the group.

---

### 4. BigInt

- **Definition:** `BigInt` is used to represent numbers larger than the maximum `number` type in JavaScript.
- **Use Case:** For scenarios where you need to handle very large integers, such as astronomical numbers.

```
let largeNumber = 1234567890123456789012345678901234567890n;  
console.log(largeNumber); // BigInt value
```

- **Real-life example:** You might use `BigInt` to store the number of stars in the universe.
- 

## 5. Symbol

- **Definition:** `Symbol` creates a unique and immutable identifier, mainly useful in object properties.
- **Use Case:** Prevent property name conflicts when working with objects.

```
let symbol1 = Symbol("test");  
let symbol2 = Symbol("test");  
console.log(symbol1 === symbol2); // false
```

- **Real-life example:** Adding unique metadata to objects without causing property name collisions.

```
const car = {  
  brand: "Tesla"  
};  
const metadata = Symbol("metadata");  
car[metadata] = { year: 2024 };  
console.log(car); // "Tesla", with hidden metadata using Symbol
```

---

## 6. NaN (Not a Number)

- **Definition:** `NaN` is used to represent the result of a mathematical operation that cannot be performed.

```
let invalidNumber = 10 / "Pramod";  
console.log(invalidNumber); // NaN
```

---

## 7. Boolean

- **Definition:** `boolean` represents a logical value: `true` or `false`.
- **Use Case:** It is essential for decision-making in programming.

```
let isLoggedIn = true;  
console.log(isLoggedIn); // true  
console.log(isLoggedIn ? "User is logged in" : "User is not logged  
in"); // User is logged in
```

- **Real-life example:** Using `boolean` to check if a user is logged in or not.
- 

## 8. String

- **Definition:** `string` represents a sequence of characters. It can be created using single quotes, double quotes, or backticks (template literals).
- **Use Case:** It is used to handle text data.

```
let str1 = 'Hello, World!';  
let str2 = "Hello, World!";  
let str3 = `Hello, World!`; // template literal  
console.log(str1, str2, str3);
```

### Real-life Example: Personal Details

```
let name = "Pramod";  
let age = 25;  
let address = "Bangalore";  
let contact = "1234567890";
```

```
console.log(`My name is ${name}, my age is ${age}, my address is  
${address}, and my contact no is ${contact}`);
```

### **String Manipulation:**

- Finding the length of a string:

```
let name_string = "Pablo";  
console.log(name_string.length); // 5
```

- Replacing characters in a string:

```
let url = "pramodjena.github.io";  
console.log(url.replace(".", "-")); // "pramodjena-github.io"
```

- Checking for a substring:

```
let email = "pramod@gmail.com";  
console.log(email.includes(".com")); // true
```

### **Type Coercion:**

```
let score1 = "100";  
let score2 = 200;  
let coercedAddition = score1 + score2;  
console.log(coercedAddition); // "100200" (due to coercion)
```