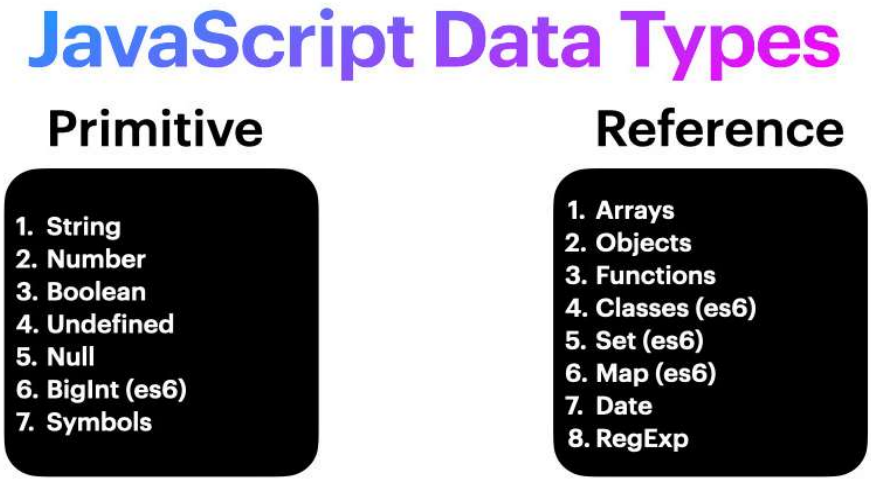
# Documentation on data types In JavaScript

## Data types in JavaScript:

**Data types:**

- Data types are values stored in variables to inform the Interpreter of what type of data we are handling.
  - There are two types of data types:
    - Primitive data types.
    - Non-primitive data types.

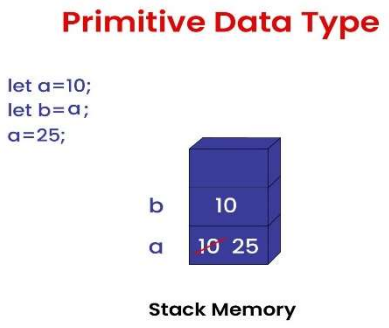


## Primitive and non-primitive types:

| Data Types | | | |
|---|---|---|---|
| Primitive | | Reference | |
| **null** | let example = null; | **Object** | let example = {hello: "world!"}; |
| **undefined** | let example = undefined; | **Function** | let example = () => 2 + 2; |
| **Boolean** | let example = true; | | |
| **Number** | let example = 33; | | |
| **String** | let example = "hello"; | | |
| **BigInt** | let example = 33n; | | |
| **Symbol** | let example = Symbol("hello"); | | |

### ❖ Primitive data types:

Primitive data types are the most basic types of data that represent a simple value, they are immutable, meaning once a value, is assigned it can not be changed.




### 1. Number:

Number data type in Javascript can be used to hold decimal values as well as values without decimals.

**Example:**

```
var x = 250;
var y = 40.5;
console.log("Value of x=" + x);
console.log("Value of y=" + y);
```

**output:**

```
Value of x=250
Value of y=40.5
```

2. **Boolean:**
   The Boolean data type can accept only two values there are true and false.
   **Example:**

```
var x;
console.log(x); // Outputs: undefined
```

3. **Undefined:**
   This means that a variable has been declared but has not been assigned a value, or it has been explicitly set to the value 'undefined'.
   **Example:**

```
let x;
console.log(x); // Outputs: undefined
```

**output:**

4. **String:**
   The string data type in JavScript represents a sequences of characters that are surrounded by single or double quotes.

   **Example:**

```
Var str = 'Hello All';
var str1 = "Welcome to my new house";
console.log("Value of str=" + str);
console.log("Value of str1=" + str1);
```

   **output:**

```
Value of str=Hello All
Value of str1=Welcome to my new house
```

5. **BigInt:**
   BigInt data type can represent numbers greater than $2^{54}$ -1 which helps to perform operations on large numbers. The numbers especially by writing ‘n’ at the end of the value.
   **Example:**

```
var a = 1234222222222222222222222222222222222n
console.log(a)
```

   **output:**

```
1234222222222222222222222222222222222n
```

6. **Null:**
   This data type can hold only one possible value that is null.
   **Example:**

```
var x = null;
  console.log("Value of x=" + x);
```

**output:**

```
Value of x=null
```

7. **Symbol:**

   Symbol data type is used to create objects which will always be unique. These objects can be created using symbol constructor.

   **Example:**

```
var sym = Symbol("Hello")
console.log(typeof(sym));
console.log(sym);
```

   **output:**

```
symbol
Symbol(Hello)
```

❖ **Non-primitive data types:**

Non-primitive data types represent more complex structures and can hold values or more complex data collections. They are mutable, meaning their contents can be changed.