# INTERNET NECHNOLOGY

**[BCAC501]**

## TAMRALIPTA INSTITUTE OF MANAGEMENT & TECHNOLOGY(212)

**Affiliated to**

**MAULANA ABUL KALAM AZAD UNIVERSITYOF TECHNOLOGY,WB**

**Submitted by**

NAME: Anurag Dolui

DEPARTMENT: BCA

REGISTRATION NO.: 222121010007

YEAR: 3RD

ROLL NO.: 21201222024

SEMESTER: 5TH

REPORT ON: “Data Types In Java Script”

# Data Types in JavaScript

This report provides a comprehensive overview of data types in JavaScript, exploring both primitive and non-primitive types. JavaScript is a dynamically typed language, meaning the data type of a variable is not explicitly defined at compile time. The type is determined at runtime based on the value assigned to the variable. The report delves into the core data types, including numbers, strings, booleans, null, undefined, and symbols. It also covers the non-primitive object type and its properties. Understanding data types is fundamental to writing effective JavaScript code, allowing you to work with and manipulate data in a variety of ways.

## Primitive Data Types

JavaScript provides several primitive data types that represent fundamentalvalues. These types are immutable, meaning their values cannot be changeddirectly. Primitive data types include numbers, strings, booleans, null, undefined, and symbols. Let's examine each of these types in more detail.

## Number

The Number data type represents numerical values in JavaScript. Numbers can be integers, decimals, or exponents. JavaScript uses the IEEE-754 standard for representing numbers, which provides support for both floating-point and integer values. This data type offers various mathematical operations and methods for performing calculations, comparisons, and rounding.

## String

Strings represent textual data in JavaScript. They are sequences of characters enclosed within single (' ') or double (" ") quotes. Strings are immutable, meaning their content cannot be altered directly. You can manipulate and work with strings using methods like concatenation, substring, and search, allowing you to process and extract information from text.

## Boolean

The Boolean data type represents truth values in JavaScript. It can only have two possible states: `true` or `false`. Boolean values are essential for conditional statements and logical operations, where you need to evaluate expressions and make decisions based on their truthiness. Boolean operations involve comparisons, logical operators, and other logical expressions.

## Undefined

The `undefined` data type represents a variable that has been declared but has not been assigned a value. When you declare a variable without assigning a value, its default value is `undefined`. This signifies that the variable exists, but its content is not defined. It is important to differentiate between `undefined` and `null`, which indicates the intentional absence of a value.

## Null

The `null` data type represents the intentional absence of a value. It is a special value indicating that a variable has no meaningful value assigned to it.

`null` is often used to represent the absence of an object or the lack of data. You can assign `null` to variables to explicitly indicate that they are intentionally empty or that no value is currently available.

## Symbol

Symbols are a relatively new addition to JavaScript, introduced in ECMAScript 6 (ES6). They are unique and immutable values used to representidentifiers for properties or keys within objects. Symbols are created using the
`Symbol()` function, and they help prevent naming conflicts betweenproperties when working with complex object structures.

## Non-Primitive Data Types

In contrast to primitive types, non-primitive data types are complex data structures that can store collections of data or have their values modified. The most prominent non-primitive data type in JavaScript is the Object type.

## Object

Objects are versatile data structures that can represent real-world entities or concepts. Objects consist of key-value pairs, where each key is a unique identifier for a property, and each value can be a primitive data type or another object. Objects are mutable, meaning their properties and values can be modified after they are created. Object properties can be accessed using dot notation or bracket notation.