In Playwright, JavaScript/type script is used to write test scripts, and JavaScript/type script has several data types that are essential for interacting with the DOM, manipulating test data, and validating conditions during tests. These data types can be divided into **primitive types** and **reference types/non primitive types**.

**1. Primitive Data Types in JavaScript (Used in Playwright)**

These types store single values and are immutable. They include:

* **String**
  + Represents a sequence of characters enclosed in single or double quotes.
  + Example:

let name = 'Playwright';

let company= “mycompany”

* **Number**
  + Represents both integers and floating-point numbers.
  + Example:

let age = 25;

let price = 12.99;

* **Boolean**
  + Represents true or false values, commonly used in conditional statements.

let isLoggedIn =TRUE;

let isLoggedOut=False;

We are using this data type to enable or disable the settings in playwright.config.js file.

Experiementalstudio: true

* **Undefined**
  + A variable that has been declared but has not been assigned a value.
  + Example:

let value;

console.log(value); // undefined

* **Null**
  + Represents the intentional absence of any value or object.
  + Example:

let user = null;

* **Symbol** (ES6)
  + Represents a unique and immutable value, often used for object property keys.
  + Example:

const uniqueID = Symbol('id');

* **BigInt** (ES11/ES2020)
  + Used to represent very large integers.
  + Example:

const largeNumber = BigInt(1234567890123456789012345678901234567890);

**2. Reference Data Types in JavaScript (Used in Playwright)**

These types store references to memory locations, and they can store multiple values.

* **Object**
  + Represents a collection of key-value pairs. Objects are often used to store and organize related data.
  + Example:

let person = {

name: 'John',

age: 30,

isActive: true

};

let product ={

productName:’Playwright-videos’,

productShortDescription:’Playwright-videos-from-herot0zero’

productDescription:”this is a full version of cypress videos which incluced

1. Introduction to Playwright
2. Installation
3. Setup
4. Folder structure
5. Cypress.config.js file modification
6. Global settings
7. Framework
8. Cloud integration
9. JS basics
10. Database testing and APi testing ”,

productPrice:100,

productStock:0,

productAvailability:false

}

* **Array**
  + An ordered collection of values that can be of any data type.
  + Example:

let fruits = ['apple', 'banana', 'cherry'];

* **Function**
  + A block of code that performs a task and can return a value.
  + Functions are first-class objects in JavaScript, which means they can be assigned to variables, passed as arguments, etc.
  + Example:

function add(a, b) {//supply the input through argument

return a + b;

}//retrun is a keyword return the value

Let x=add(3,4)

* **Date**
  + Used for handling dates and times.
  + Example:

let currentDate = new Date();

* **RegExp**
  + Represents regular expressions used for pattern matching in strings.
  + Example:

let regex = /hello/i; // 'i' is a case-insensitive flag

//attributes selectors - ^ | etc…

**Data Types in Automation testing Context**

In **Automation** testing, you work mainly with the following JavaScript data types:

* **Strings** for interacting with DOM elements
* **Booleans** for assertions and conditional checks.
* **Arrays** for working with multiple DOM elements.
* **Objects** for storing test data or configurations.
* **Functions** for writing Playwright commands or custom commands.