Introduction to Data Types

Data Types

We had a brief discussion about the seven types of data types in the previous module. **Data Type** is the type of data a variable stores. In JavaScript, a variable can store different types of data.

```
Example : let temp = 10 ;                    // Number
          let temp2 =  “Coding Ninjas” // String
```

- ➔ JavaScript evaluates data from left to the right.

```
Case 1 :  let temp = 5 + 4 + “Coding Ninjas” ;
          temp becomes “9CodingNinjas”

Case 2 :  let temp = “Coding Ninjas”  +  5 + 4  ;
          temp becomes “CodingNinjas54”
```

Dynamically Typed Language

Javascript is a dynamic typing language, unlike Java/C++.
When you declare a variable, you do not need to specify what type this variable is. Javascript engine infers what type of this variable is based on the value assigned to at the run time.

A variable can be assigned any type of value, and you need not mention the type.

```
Example : let temp = 2 ;
          let temp = true ;
```

In the above example, the temp variable is storing type number in case 1 and type boolean in the second case, and we didn’t mention it in the JavaScript code.

typeof Operator

The typeof operator is a unary operator placed before its single operand, which can be of any type. Its value is a string indicating the data type of the operand. The typeof operator evaluates to "number", "string", or "boolean" if its operand is a number, string, or boolean value and returns true or false based on the evaluation.

Type	String Returned by typeof
Number	"number"
String	"string"
Boolean	"boolean"
object	"object"
Function	"function"
Undefined	"undefined"
Null	"object"

There are 6 primitive and 1 non-primitive data types:

1. **Number:** It represents variables whose value is either an integer or float. Other than integer and float numbers, it has three symbolic values: **+Infinity**, **-Infinity**, and **NaN**. The number lies between **+Infinity** and **-Infinity**.

```
Example : let temp = 10 ;
          typeof(temp) ;  // Number

          let temp = 10.54 ;
          typeof(temp) ;  // Number

          let temp = NaN ;
          typeof(temp) ;  // Number
```

★ **In JavaScript, NaN is also of type Number.**

2. **String:** It represents textual data. The string contains elements that can be accessed using the index. The first element has an index of type., "hello", "1234", "12here". You can access each string element like - str="HelloWorld", then str[1] will output 'e' on the console.

```
Example : typeof(5) ;  // Number
          typeof(“5”) ; // String
```

3. **Undefined**: 'undefined' is the value assigned to the variable that has not yet been assigned any value. We can also explicitly assign an 'undefined' value to a variable, but that does not make any sense due to its meaning.

```
Example: var a ;
Defines a variable that has not been assigned any value.
console.log(a) ; // Output - 'undefined'
```

4. **Boolean:** Booleans can only have two values: true or false. It is often used for conditional testing.

```
Example : let x = 2 , y = “2” ,  z = 3 ;
          console.log(x==y) ;  // true
          console.log(x===y) ;  //false
          console.log(x==z) ;   // false
```

5. **null:** 'null' is the value that represents a reference that points to a non-existent object or address. This means that there is an absence of a value. The data type for the null value is "Object".

6. **NaN**: 'NaN' means **Not-A-Numbe**r. So, if any expression fails to return a number, then 'NaN' is printed on the console

```
typeof(NaN) ; // Number
```

```
Example : (12 - "abc") ;
cannot be evaluated to a number, so 'NaN' is printed on the console.
```