**ASSIGNMENT 2**

DATA TYPES AND VARIABLES

1. What are the different Data types used in JavaScript variables?

String: Used to represent text data, enclosed in single or double quotes.

Number: Used to represent numerical values, including integers and floating numbers.

Boolean: Used to represent logical values, either true or false.

Undefined: Represents a variable that has been declared but not yet assigned a value.

1. Explain the difference between var, let, and const in JavaScript
2. Var: can be reassigned and redeclared within their scope.
3. Let: can be reassigned but not declared within the same scope.
4. Const: once assigned a value, the value cannot be changed throughout its scope.
5. Why does JavaScript allow assigning different data types to the same variable?

JavaScript is a dynamically typed language, meaning you can assign different data types to the same variable throughout its lifetime.

1. How does JavaScript handle variables declared but not initialized? Illustrate with an example from the code.

If a variable is declared using var or let but not assigned a value, it will have the undefined value.

Let y;

Console.log(y) //output undefined

1. Discuss the significance of variable names in programming and how they are used in JavaScript.

Meaningful variable names make the code easier to understand and debug.

Some common naming convention include:

Camel case: myVariableName

Pascal case: MyVariableName

Underscore case: my\_varible\_name

NUMERIC DATA TYPE

1. What are the various numeric data types used in JavaScript?
2. Number: This data type represents both integers and floating-point numbers.
3. BigInt: It is used for working with very large integers that might overflow the number type.
4. Explain the difference between integers, doubles, and infinity in JavaScript with examples.
5. Integers: It is whole numbers without decimal points e.g., 10. JavaScript converts integers to floating-point numbers for calculation using Number. isInteger () to check if a number is an integer.
6. Doubles: It is floating-point numbers with decimal points e.g., 3.14
7. Infinity: A special value representing positive or negative infinity. E.g., 1/0 or -1/0
8. How does JavaScript handle arithmetic operations involve different numeric data types?

* If on operand is a double, the results will be a double.
* If both operands are integers, the results will be an integer unless the operation results in a value outside the range of integers.

STRING DATA TYPE

1. How are string represented in JavaScript?

Strings are represented as sequence of characters enclosed within either single quotes (‘) or double quotes (“”)

1. Discuss the difference between declaring strings with single quote (‘) and double quotes (“”) in JavaScript.

You choose one style and use it consistently throughout your code for better readability.

You need to quote a character for escaping special characters.

1. Explain why characters are automatically treated as string in JavaScript.

To make it easier to work with text data in JavaScript.

BOOLEAN AND UNDEFINED DATA TYPES

1. Explain the purpose of Boolean variables in Java Script.

Are used to represent logical values either true or false

1. Discuss the concept of undefined in JavaScript variables and provide examples for each.

The undefined value indicates that a variable has been declared but not yet assigned a value.

e.g., let isLoggedIn;

console.log(isLoggedIn)

1. How are Boolean variables useful in conditional statement and control flow in JavaScript?

They are used in conditional statements like if, else if, and else to make decisions based on logical conditions.

NULL DATA TYPE

1. Describe the significance of the null value in JavaScript.

Null value represents the absence of an object reference.

1. Differentiate between null and undefined in JavaScript.

* Null: Indicates the absence of an object. It is used to represent the intentional absence of a value.
* Undefined: Indicates the variable that has been declared but not yet assigned a value.

1. Use code illustration to provide an example.

let car = null; // Intentionally set to null

let color; // Declared but not assigned, value is undefined

console.log(car); // Output: null

console.log(color); // Output: undefined

OBJECT DATA TYPE

1. Explain how objects are represented in JavaScript.

Objects are collection of key-value pairs, where each key is a property name and each value can be any data type.

1. Discuss the structure and purpose of the country Info object in the provided code.

let countryInfo = {

name: "Kenya",

capital: "Nairobi",

population: 55.36 million,

currency: "Kenyan Shilling",

languages: ["English", "Kiswahili"],

landmarks: ["Mount Kenya", "Masai Mara National Park"]

};

Object has property like name, capital, population, currency and language. Each property has a corresponding value which can be a string, number, array.

1. How can objects be nested within other objects in JavaScript.

Objects can be nested within other objects to create more complex data structures.

ARRAY DATA TYPE

1. Describe the purpose and structure of arrays in JavaScript.

Arrays are used to store collection of elements of different data types, making them versatile for representing various kinds of data.

1. Provide examples from the code demonstrating arrays containing different data types.

let numbers = [1, 2, 3, 4, 5]; // Array of numbers

let fruits = ["apple", "banana", "orange"]; // Array of strings

let mixedArray = [10, "hello", true, null]; // Array containing mixed data types

1. Discuss the concept of “arrays of arrays” and its significance.

Arrays can contain other arrays, creating multi-dimensional arrays. It is useful for representing tabular data or complex structures.

VARIABLES NAMING CONVENTIONS

1. What are the conventions for naming variables in JavaScript?

* Camel case: Words are separated by a lowercase letter, with first word starting in lowercase.

E.g., myVaribleName

* Underscore Case: Words are separated by underscores. E.g., my\_variable\_name.

1. Discuss the importance of choosing meaningful and descriptive variable names.

Readability: Clear and descriptive variable names an make your code easier to understand and follow.

Consistency: Adhering to consistent naming conventions improve code consistency and readability.

1. Identify any variable naming conventions followed in the provided code.

CONSTANTS IN JAVASCRIPT

1. Explain the use of const keyword in JavaScript.

Const keyword is used to declare variables whose values cannot be reassigned after they are initialized.

1. Discuss why reassigning a value to a constant variable result in an error.

When declare a variable using const, it creates a read-only reference to the value. Attempting to reassign a const variable will result in a TypeError.

1. Provide examples from the code demonstrating the declaration and use of constants. const PI = 3.14159;

// This will throw an error:

PI = 3.14;

const myArray = [1, 2, 3];

// This will throw an error:

myArray = [4, 5, 6];

// However, you can modify elements of an array declared with const:

myArray[0] = 10; // This is allowed