Assignment 1.

Question 1: Explain what JavaScript is and its role in web development.

Answer : JavaScript is a versatile and widely-used programming language that is primarily utilized in web development. It is a high-level, interpreted scripting language that enables developers to create dynamic and interactive content within web browsers. JavaScript plays a crucial role in enhancing the functionality and user experience of websites.

It’s role in web development :-

1. Client-Side Scripting
2. Dynamic content and Interactivity
3. Event Handling
4. Frameworks and libraries
5. AJAX (Asynchronous Javascript and XML)

Question 2: Explain the key differences between JavaScript and HTML. Provide examples of situations where you would use each.

Answer. JavaScript and HTML are both essential components of web development, but they serve different purposes and are used in distinct ways. JavaScript is used for creating dynamic and interactive behavior on web pages, while HTML is used for structuring and presenting content. They often work together, with HTML providing the structure and content of a webpage, and JavaScript adding functionality and interactivity.

Question 3: List and describe the five primitive data types in JavaScript.

Answer: Javascript has five primitive data types :

1) String : It represents sequences of characters and are used to store and manipulate text. Eg:- let myString = 'Hello, World!';

2) Number : It represent sequences of characters and are used to store and manipulate text . Eg:- let myNumber = 42; let myFloat = 3.14;

3) Boolean : Boolean represents truth values and have only two possible values : ‘true’ or ‘false’. Eg:- let isTrue = true; let isFalse = false;

4) Undefined : It indicates that a variable has been declared but has not been assigned a value.Eg:- let myUndefinedVariable; console.log(myUndefinedVariable);

5) Null : It represents the intentional absence of any object value. It is often used to signify that a variable has no value. Eg:- let myNullVariable = null;

Question 4: What is the purpose of declaring variables in JavaScript, and how do you declare them using the 'let' keyword?

Answer : In JavaScript, variables are used to store and manage data. Declaring variables is essential for storing values, manipulating data, and maintaining state in a program.

We can declare variables using let keyword :-

Syntax – let message = 'Hello, World!';

Here ,

* **‘let’** is the keyword used for variable declaration.
* **‘message'** is the name you can give to the variable.
* **‘=’** is the assignment operators used to assign a value to the variable.
* The right side of the assignment (**'Hello, World!'**) is the initial value assigned to the variable.

Question 5: Explain the importance of comments in JavaScript and provide examples of single-line and multi-line comments.

Answer:

Comments in JavaScript are non-executable lines of text that you include in your code to provide explanations, notes, or reminders. They are not processed by the JavaScript interpreter and do not affect the execution of the program. Comments are crucial for enhancing code readability, maintaining codebase documentation, and facilitating collaboration among developers.

Question 6: Explain the importance of choosing meaningful and descriptive variable names in JavaScript. Provide an example where using a clear identifier improves code readability.

Answer:

Choosing meaningful and descriptive variable names is a crucial aspect of writing maintainable and readable JavaScript code. Clear identifiers enhance code readability, reduce the likelihood of errors, and make it easier for other developers (or even yourself) to understand the purpose and functionality of your code.

Example:

// Without meaningful names

let a = 5;

let b = 8;

let c = a + b;

// With meaningful names

let width = 5;

let height = 8;

let area = width + height;

ASSIGNMENT 2.

Q1. Explain the role of operators in JavaScript. Why are they essential in programming?

Answer:

Operators in JavaScript are symbols or keywords that perform operations on operands—values or variables.

The reason behind they are essential due to :-

1. Arithmetic Operations : It operates basic arithmatic operations such as addition (‘+’), subtraction (‘-’), multiplication (‘\*’), division (‘/’) , and modulus (‘%’).
2. Comparison Operations : Comparison operators (**==**, **===**, **!=**, **!==**, **<**, **>**, **<=**, **>=**) are used to compare values and determine the relationship between them.
3. Logical Operations : Logical operators (**&&**, **||**, **!**) perform logical operations, such as AND, OR, and NOT.
4. Assignment Operators : Assignment operators (**=**, **+=**, **-=**, **\*=**, **/=**) are used to assign values to variables.
5. Increament and decreament Operator : Increment (**++**) and decrement (**--**) operators increase or decrease the value of a variable by 1, respectively.
6. Type Conversion : Operators can be involved in type conversion, allowing you to convert data from one type to another.
7. String Concatenation : The **+** operator is used for string concatenation, allowing you to combine multiple strings into a single string.

Q2. Describe the categorization of operators in JavaScript based on their functionality. Provide examples for each category.

Answer :

Operators in JavaScript can be categorized based on their functionality into several groups. Here's an overview of the main categories:-

1. Arithmatic Operations :

Perform basic arithmetic operations on numeric values.

Example:

let addition = 5 + 3; // Addition

llet subtraction = 8 - 2; // Subtraction

let multiplication = 4 \* 6; // Multiplication

let division = 10 / 2; // Division

let modulus = 15 % 7; // Modulus (remainder after division)

1. Comparison Operators :

Compare values and return a Boolean result.

Examples:

let isEqual = (10 === 10); // Equal to

let isNotEqual = (5 !== '5'); // Not equal to (type-safe)

let isGreaterThan = (8 > 5); // Greater than

let isLessThan = (3 < 7); // Less than

1. Logical Operators :

Perform logical operations on Boolean values.

Example:

let andOperation = true && false; // Logical AND

let orOperation = true || false; // Logical OR

let notOperation = !true; // Logical NOT

1. Assignment Operators :

Assign values to variables and perform related operations.

Examples:

let x = 5; // Assignment

x += 3; // Addition assignment (x = x + 3)

x -= 2; // Subtraction assignment (x = x - 2)

1. Unary Operators :

Operate on a single operand.

Examples:

let negation = -5; // Unary negation

let increment = ++x; // Prefix increment

let decrement = y--; // Postfix decrement

1. Ternary Operators :

Provides a concise way to express a conditional (if-else) statement.

Example:

let result = (age >= 18) ? 'Adult' : 'Minor';

1. String Operators :

Concatenate strings or perform operations related to strings.

Example:

let greeting = 'Hello, ' + 'World!'; // String concatenation

1. Typeof Operater :

Returns a string indicating the data type of a variable or expression.

Example:

let dataType = typeof 42; // Returns 'number'

Q3. Differentiate between unary, binary, and ternary operators in JavaScript. Give examples of each.

Ans. In JS , operators are categorized based on the number of operands they operate on.

**Unary Operators:**

* **Definition:** Unary operators work with a single operand.
* **Example:**

let x = 5;

let y = -x; // Unary negation, negates the value of x

* Common unary operators include:
  + Unary Negation (**-**): Negates the value.
  + Unary Plus (**+**): Converts an operand to a number, but is rarely used.

**Binary Operators:**

* **Definition:** Binary operators work with two operands.
* **Example:**

let a = 10;

let b = 20;

let sum = a + b; // Binary addition, adds the values of a and b

Common binary operators include:

* Arithmetic Operators (**+**, **-**, **\***, **/**, **%**): Perform basic mathematical operations.
* Comparison Operators (**==**, **===**, **!=**, **!==**, **<**, **>**, **<=**, **>=**): Compare two values.
* Logical Operators (**&&**, **||**): Perform logical AND and OR operations.

**Ternary Operator:**

* **Definition:** The ternary operator is a shorthand for an if-else statement. It works with three operands.
* **Example:**

let age = 20;

let status = (age >= 18) ? 'Adult' : 'Minor'; // Ternary operator

Q4. Discuss the precedence and associativity of operators in JavaScript. Why is understanding these concepts important?

Answer :

Operators : Operator precedence determines the order in which operators are evaluated when more than one operator appears in an expression. Operators with higher precedence are evaluated first. For example, in the expression **a + b \* c**, the multiplication (**\***) has a higher precedence than addition (**+**), so **b \* c** is evaluated first.

Associativity : Associativity refers to the order in which operators of the same precedence are evaluated. It can be left-to-right or right-to-left. For example, in the expression **a = b = c**, the assignment operator (**=**) has right-to-left associativity, so **b = c** is evaluated first, followed by **a = b**.

Understanding operator precedence and associativity is fundamental for writing correct, readable, and maintainable JavaScript code. It ensures that expressions are evaluated in the expected order, leading to code that behaves predictably.

Que.05 Write a JavaScript program that calculates the simple interest using the formula Simple interest = (principal \* rate \* time) / 100.

Answer:

function calculateSimpleInterest(principal , rate , time) {

var interest = (principal\*rate\*time)/100;

return interest;

}

var principalAmount = 1000;

var annualInterestRate = 5;

var tiemInYears = 2;

var simpleInterest = calculateSimpleInterest(principalAmount, annualInterestRate,timeInYears);

console.log("Principal: $"+ principalAmount);

console.log("Annual Interest Rate:"+ annualInterestRate + "%");

console.log("Time (in years): "+ timeInYears);

console.log("Simple Interest: $"+ simpleInterest);

Que.06 Write a Javascript program to calculate the Body Mass Index (BMI) using the formula BMI = weight (kg)/height \* height.

Answer: // Function to calculate BMI

function calculateBMI(weight, height) {

// BMI formula: weight (kg) / (height (m) \* height (m))

var bmi = weight / (height \* height);

return bmi;

}

// Example usage

var weightInKg = 70; // Replace with your weight in kilograms

var heightInMeters = 1.75; // Replace with your height in meters

// Calculate BMI

var bmiResult = calculateBMI(weightInKg, heightInMeters);

// Display the result

console.log("Weight: " + weightInKg + " kg");

console.log("Height: " + heightInMeters + " m");

console.log("BMI: " + bmiResult.toFixed(2)); // Display BMI with two decimal places

Que.07 Write a program in JavaScript to calculate the area of a circle given its radius value of 10. Use appropriate arithmetic operators.

Answer: // Function to calculate the area of a circle

function calculateCircleArea(radius) {

// Area formula: π \* radius^2

var area = Math.PI \* Math.pow(radius, 2);

return area;

}

// Example usage

var radius = 10; // Replace with the radius of the circle

// Calculate the area of the circle

var circleArea = calculateCircleArea(radius);

// Display the result

console.log("Radius: " + radius);

console.log("Area of the circle: " + circleArea.toFixed(2)); // Display area with two decimal places