**4. Control Flow**

**Conditional statement**:

In JavaScript we have two type of Conditional Statement

1. if-else
2. switch-case

**if-else**:

The if statement executes a statement if a specified condition is truthy. If the condition is falsy, another statement can be executed.

**Example**:

function testNum(a) {

if (a > 0) {

return "positive";

} else {

return "NOT positive";

}

}

console.log(testNum(-5));*//"NOT positive"*

**switch-case**:

The switch statement evaluates an expression, matching the expression's value to a case clause, and executes statements associated with that case, as well as statements in cases that follow the matching case.

**Example**:

var expr = "java";

switch (expr) {

case "java":

console.log("Programming Language");

break;

case "HTML":

console.log("Markup Language");

break;

case "javascript":

console.log("scripting language");

*// expected output: "Mangoes and papayas are $2.79 a pound."*

break;

default:

console.log("None of the above");

}

*//output is "Programming Language"*

**Loop**:

Loops offer a quick and easy way to do something repeatedly. There are many different kinds of loops, but they all essentially do the same thing: they repeat an action some number of times.

The statements for loops provided in JavaScript are

1. for loop
2. while loop
3. do...while loop
4. for...in loop
5. for...of loop

**for loop**:

A for loop repeats until a specified condition evaluates to false. The JavaScript for loop is similar to the Java and C for loop.

for ([initialExpression]; [condition]; [incrementExpression]){

statement;

}

**Example**:

for (let i = 1; i <= 5; i++) {

console.log(i);

}

*/\**

*1*

*2*

*3*

*4*

*5*

*\*/*

**while loop**:

A while statement executes its statements as long as a specified condition evaluates to true. A while statement looks as follows:

while (condition) {

statement;

}

**Example**:

let i = 5;

while (i >= 1) {

console.log(i);

i--;

}

*/\**

*5*

*4*

*3*

*2*

*1*

*\*/*

**do-while loop**:

The do-while loop is always executed once before the condition is checked (and then again until the while condition returns false). A do...while statement looks as follows:

do {

statement;

}

while (condition);

**Example:**

let i = 1;

do {

console.log(i);

i++;

} while (i <= 5);

*/\**

*1*

*2*

*3*

*4*

*5*

*\*/*

**for...in loop**:

The for...in statement iterates a specified variable over all the enumerable properties of an object. For each distinct property, JavaScript executes the specified statements. A for...in statement looks as follows.

for (variable in object) {

statements;

}

**Example**:

const person = {

name: "Ruhul",

age: 20

};

for (let key in person) {

console.log(key, person[key]);

}

*/\**

*name Ruhul*

*age 20*

*\*/*

**for...of loop**:

The for...of statement creates a loop Iterating over inerrable objects (including Array, Map, Set, arguments object and so on), invoking a custom iteration hook with statements to be executed for the value of each distinct property.

for (variable of object) {

statement;

}

**Example:**

const colors = ["red", "green", "blue"];

for (let color of colors) {

console.log(color);

}

*/\**

*red*

*green*

*blue*

*\*/*

**Break and Continue**:

JavaScript provides full control to handle loops and switch statements. There may be a situation when you need to come out of a loop without reaching its bottom. There may also be a situation when you want to skip a part of your code block and start the next iteration of the loop.

To handle all such situations, JavaScript provides break and continue statements. These statements are used to immediately come out of any loop or to start the next iteration of any loop respectively.

The break statement "jumps out" of a loop. The break statement breaks the loop and continues executing the code after the loop (if any).

**Example**:

for (i = 0; i < 10; i++) {

if (i === 5) {

break;

}

console.log("The number is: " + i);

}

*/\**

*The number is: 0*

*The number is: 1*

*The number is: 2*

*The number is: 3*

*The number is: 4*

*\*/*

The continue statement tells the interpreter to immediately start the next iteration of the loop and skip the remaining code block. When a continue statement is encountered, the program flow moves to the loop check expression immediately and if the condition remains true, then it starts the next iteration, otherwise the control comes out of the loop.

**Example**:

for (i = 0; i < 5; i++) {

if (i === 5) {

continue;

}

console.log("The number is: " + i);

}

*/\**

*The number is: 0*

*The number is: 1*

*The number is: 2*

*The number is: 3*

*The number is: 4*

*\*/*

**Exercise**

**Exercise-1**:

Write a function that takes two parameter and return the max number of two umber.

**Solution**:

function findMax(number1, number2) {

if (number1 == number2) return "two number are equal";

return number1 > number2 ? number1 : number2;

}

const number = findMax(20, 10);

console.log(number); *//20*

const num = findMax(20, 20);

console.log(num); *//two number are equal*

**Exercise-2**:

Create a function that return true if an image is landscape and false if Portrait.

**Solution**:

function isLandscape(width, height) {

*//return true if width > height*

*//return false if width < height*

return width > height;

}

console.log(isLandscape(300, 200)); *//true*

console.log(isLandscape(100, 200)); *//false*

**Exercise-3**:

FizzBuzz program. Write a program that take a parameter and if

Number Divisible by 3 => Fizz

Number Divisible by 5 => Buzz

Number Divisible by both 3 and 5 => FizzBuzz

Number Not Divisible by both 3 and 5 => return number

Number Not a Number => Not a Number

**Solution**:

*// Number Divisible by 3 => Fizz*

*// Number Divisible by 5 => Buzz*

*// Number Divisible by both 3 and 5 => FizzBuzz*

*// Number Not Divisible by both 3 and 5 => return number*

*// Number Not a Number => Not a Number*

function fizzBuzz(input) {

if (typeof input != "number") {

return "Not a Number";

}

if (input % 3 == 0 && input % 5 == 0) {

return "FizzBuzz";

}

if (input % 3 == 0) {

return "Fizz";

}

if (input % 5 == 0) {

return "Buzz";

}

return input;

}

console.log(fizzBuzz("12")); *//Not a Number*

console.log(fizzBuzz(15)); *//FizzBuzz*

console.log(fizzBuzz(9)); *//Fizz*

console.log(fizzBuzz(25)); *//Buzz*

console.log(fizzBuzz(11)); *//11*

**Exersise-4**:

Write a function that calculate the speed of vehicles and display the following output.

Speed Limit = 70 => ok

After 70 for every 5 => 1 point

Math.floor(1.3) => 1

12 points => License Suspended

**Solution**:

*//Speed Limit = 70 => ok*

*//After 70 for every 5 => 1 point*

*//Math.floor(1.3) => 1*

*//12 points => License Suspended*

function checkSpeed(speed) {

const speedLimit = 70;

const kmPerPoint = 5;

if (speed < speedLimit + kmPerPoint) {

console.log("Ok");

} else {

const points = *Math*.floor((speed - speedLimit) / kmPerPoint);

if (points >= 12) {

console.log("License Suspended");

} else {

console.log("Points: " + points);

}

}

}

checkSpeed(74); *//Ok*

checkSpeed(80); *//Points: 2*

checkSpeed(129); *//Points: 11*

checkSpeed(130); *//License Suspended*

**Exersise-5**:

Write a function called “showNumbers” that takes a parameter called limit. The function print even or odd number from 0 to limit.

**Solution**:

function showNumbers(limit) {

for (let number = 0; number <= limit; number++) {

if (number % 2 == 0) {

console.log(number + ' "Even"');

} else {

console.log(number + ' "Odd"');

}

}

}

showNumbers(10);

or

function showNumbers(limit) {

for (let number = 0; number <= limit; number++) {

const message = number % 2 == 0 ? "Even" : "ODD";

console.log(number + ' "' + message + '"');

}

}

showNumbers(10);

**Exersise-6**:

Creates a function called “countTruthy“ that takes an array and returns the number of truthy element present in that array.

**Solution**:

function countTruthy(array) {

let count = 0;

for (let value of array) {

*//execute when value is truthy*

if (value) {

count++;

}

}

return count;

}

const myArray = [1, 2, 3, 4, 5, "", undefined, null];

console.log(countTruthy(myArray)); *//5*

const myArray1 = [1, 2, 3, 4, 5, "ruhul"];

console.log(countTruthy(myArray1)); *//6*

**Exersise-7**:

Write a function that takes an array and returns the number of properties in the array that are typed string.

**Solution**:

function showProperties(object) {

for (let key in object) {

if (typeof object[key] == "string") {

console.log(key, object[key]);

}

}

}

movei = {

title: "a",

releaseYear: 2018,

rating: 4.5,

director: "b"

};

showProperties(movei);

*/\**

*title a*

*director b*

*\*/*

**Exersise-8**:

Write a function that take a number and calculate the sum of Multiples 3 and 5.

Number 10

Multiples of 3: 3, 6, 9

Multiples of 5: 5, 10

Sum => 33

**Solution**:

function calculateSum(limit) {

sum = 0;

for (let i = 1; i <= limit; i++) {

if (i % 3 == 0 || i % 5 == 0) {

sum += i;

}

}

return sum;

}

console.log(calculateSum(10)); *//33*

**Exersise-9**:

Write a function that takes an array of marks and calculate the following result.

1 – 59 : F

60 – 69 : D

70 – 79 : C

80 – 89 : B

90 – 100 : A

**Solution**:

function calculateGrade(array) {

const avarage = calculateAverage(array);

if (avarage < 60) return "F";

if (avarage < 70) return "D";

if (avarage < 80) return "C";

if (avarage < 90) return "B";

return "A";

}

function calculateAverage(array) {

let sum = 0;

for (let value of array) {

sum += value;

}

return sum / array.length;

}

const numbers = [50, 70, 90];

console.log(calculateGrade(numbers)); *//C*

**Exersise-10**:

Write a function that print a pattern of stars.

**Solution**:

function showStars(row) {

for (let i = 1; i <= row; i++) {

let pattern = "";

for (let j = 0; j < i; j++) {

pattern += "\*";

}

console.log(pattern);

}

}

showStars(5);

*/\**

*\**

*\*\**

*\*\*\**

*\*\*\*\**

*\*\*\*\*\**

*\*/*

**Exersise-11**:

Write a program that print a range of prime number.

**Solution**:

*//prime number*

function showPrime(limit) {

for (let number = 2; number <= limit; number++) {

let isPrime = true;

for (let factor = 2; factor < number; factor++) {

if (number % factor === 0) {

isPrime = false;

break;

}

}

if (isPrime) {

console.log(number);

}

}

}

showPrime(10);

*/\**

*2*

*3*

*4*

*5*

*7*

*\*/*