**Scope, Closure, Hoisting in JavaScript**

**Scope**:

-> Scope refers to the visibility and accessibility of variables in JavaScript.

There are two types of scope in JavaScript:

-> **Global scope**

-> **Local scope.**

**Global scope**: Variables declared outside of a function or block are in the global scope and can be accessed from anywhere in the code.

**Local scope**: Variables declared inside of a function or block are in the local scope and can only be accessed from within that function or block.

**Example**:

// Scope in javascript

const globalScope = "i am from global scope";

function exampleFunction (){

const localScope = "i am from local Scope";

console.log(localScope); // i am from local Scope

console.log(globalScope); // i am from global scope

}

exampleFunction()

console.log(globalScope); // logs "I'm a global variable"

console.log(localScope); // ReferenceError: localVar is not defined

// **output**:

// i am from local Scope

// i am from global scope

// i am from global scope

// ReferenceError: localScope is not defined

--------------------------------------------------------------------------------------------

**Closure**:

-> Closure is a feature in JavaScript that allows a function to access and manipulate variables from its parent scope, even after the parent function has returned.

**Example**:

function outerFunc (){

const outerVariable = "i am from outer";

function innerFunc(){

const innerVariable = " i am from inner ";

console.log(outerVariable);

console.log(innerVariable);

}

innerFunc();

}

outerFunc();

**output**:

i am from outer

i am from inner

**Hoisting**:

-> Hoisting is a JavaScript mechanism that moves variable and function declarations to the top of their respective scopes during the compilation phase.

**Example**:

console.log(x); // undefined

var x = 5;

function exampleFunction() {

console.log(y); // undefined

var y = 10;

}

exampleFunction();

**output**:

undefined

undefined

--------------------------------------------------------------------------------------------