```javascript
//****************VAR LET CONST**************** */


// Scope
// variable shadowing
// variable declaration
// variable declaration without initialization
// Reinitialisation of variables
// How execution context works?
// Hoisting & Temporal Dead zone

// Three types of Scope

// Global scope
// Functional scope
// Block scope


var a=5; // global scope
{
var a=5
}
console.log(a); // 5 -- a is accessible outside  the block ☑
{
let a=5;
}
console.log(a) //  ReferenceError: a is not defined ✖
{
const a=5;
}
console.log(a); // ReferenceError: a is not defined ✖
{
  let a = 5;
  console.log(a); //5
}
{
  const a = 5;
  console.log(a); //5
}
//VARIABLE SHADOWING
function test() {
    let a = 'Hello';

    if (true) {
        let a = 'Bye';
        console.log(a); //Bye
    }

    console.log(a);//Hello
```

```javascript
}

test();
//Explaination
function test() {
    let a = 'Hello';

    if (true) {
        let a = 'Bye';   // New value assigned
        console.log(a); 1. Bye // a shadows the a defined outside the block
scope and prints "Bye"
    }

    console.log(a);   2.Hello // it will print a which is in the available
scope
}

test();
//Illegal Shadowing
function func() {
    var a = 'Hello';
    let b = 'Roadside coder';

    if (true) {
        let a = 'Hi'; // Legal Shadowing
        var b = 'Bye'; // Illegal Shadowing
        console.log(a); //1.Hi ☑
        console.log(b); //2. SyntaxError: Identifier 'b' has already been
declared ✖
    }
}
test();
//Declaration
var a;
var a; ☑

let a;
let a;//✖ SyntaxError: Identifier 'a' has already been declared

const a;
const a;//   ✖ SyntaxError: Missing initializer in const declaration

let a;
{
let a;
}
//Declaration without initialization
var a; ☑
```

```javascript
let a; ☑
const a; //✖ SyntaxError: Missing initializer in const declaration -- you
need to intialise a const while declaring it
//Re-initialization
var a=5;
a=6; ☑

let a=7;
a=10; ☑

const a=10;
a=12; //✖ TypeError: Assignment to constant variable.

//Hoisting
let a=10;
function multiply(a){
return a*20;
}
let b= multiply(20);
console.log(b);
//example
console.log(count); // undefined
var count=4;
console.log(count);
let count=6; // ReferenceError : cannot access 'count' before initialization
```