To print :- console.log(variable);

Variable

1) Global variable — c = 20;
2) Function variable — var d = 30;
3) Block Scope. — let e = 50;

It is only limited to a block.
It can be accessed inside the function
It can be accessed from anywhere.

example :)

```
c = 20;   [Global variable]
function fun(){
    let a = 5;   [function scope]
    if (a === 5){
        var f = 10;   [scope - BLOCK]
        console.log("inside", f);
    }
    console.log("outside", f);
}
console.log(f);
```

Output :-

Inside 10
f NOt availble
f NOt availble

Geobal variable can be intialised anywhere. not perticularely above the function. you define geobal variable inside a for-loop also.

To find squart root. JS has a inbuild function for that.

⇒ ~~math.sqrt~~ `Math.sqrt(number);`

In JS you can return anything, you don't have to define it while making the function.

FUNCTIONS

e.g.)
```
function squart_root(n){
    console.log("In first sqrt fn");
    return;
}
```

The above type of functions will be by-default move to the top while ~~compiling~~ compiling. and all other statements will move to buttom.

This concept is knowns as HOISTING.

e.g.)
```
var squart_root = function(){
    console.log("In second sqrt fn");
    return
}
```

This type of function will not move to the top.

If you want to call this type of function then call them from bottom.

ARRAYS

```
let arr = ["Apple", "mongo", "Guava"];
console.log(arr);
```

output :- [Apple, mongo, Guava]

→ To access length :-

① arr["length"];

② ~~arr~~ arr.length;

both above function will return length of arr.

→ To access single element :-

arr[1];

output :- mongo.

→ To add a element in the end.

arr.push("Banana");

output :- [Apple, mongo, Guava, Banana].

→ To remove last element present.

```
arr.pop()
```

output :- [Apple, mongo, Gauva].

→ To find index of perticular element present in array.

```
arr.indexOf("mongo")
```

output :- 1.

→ To remove the element from the front.

```
arr.shift();
```

output :- ['mongo, Gauva].

→ To add something in the front of array.

```
arr.unshift("Kiwi");
```

output :- [Kiwi, mongo, Gauva].

Conclusion :-

array name ←

```
a.push("Banana");   // Insert at back
a.pop();            // remove from back
a.shift();          // remove from front
a.unshift("Kiwi");  // insert at front
a.indexOf("Kiwi");  // finds you the index
```