

Node JS: Runtime Environment, where code can be executed

- JS: i) reads code
ii) executes code

```
console.log(a);  
var a = 10;  
console.log(a);
```

var stores in global scope
output: undefined
 10

```
console.log(a)  
let a = 10;  
console.log(a)
```

let stores variable in
Script (local scope)
output: error

Temporal dead zone: It is the time period in which the variables that have been declared using "let" and "const" are not accessible and they are present as undefined.

Scan & execute

interpreter ?

```
var a = 10  
var b = 20  
function a () {  
    ↴  
    function b () {  
        }  
    }  
}
```

a
b
Function {}
c

d
e
b()

b

* null (datatype)
defined

"Use strict": used to avoid mistakes
non strict check strict check
===== =====

* template literal : (Javascript Interpolation)

let nameOfVariable = "Kunawat";

console.log(`Moshit \${nameOfVariable}`);

console.log(type of nameOfVariable);
 literal
 string

* Array is an object in Javascript
because they have properties & methods
(similar to objects)

* conditions

- if else
- switch case
- Ternary

Ternary operator: condition ? True : False

var hello = "hey";

(hello == "hey") ? console.log(hello):
console.log("no");

Primitive: Number, String, Boolean, null, undefined (single value)

Nonprimitive: Array & Object (multiple values)

Errors: Reference error
Syntax error
Type of error

* Array: under and values (fixed under)

let name = ["Mudit", "Mayank"];

let names = new Array("Ruchil", "Kareena");

Object: Key and value (key can be changed)

* Array:

* Math

+ Date

* loops: while, do while, for

① for of and for in
Array Object

```
const number = [1, 2, 3, 4, 5, 6]
for (const num of number) {
    console.log(num);
}
```

```
const number = { 1: 1, 2: 2, 3: 3, 4: 4, 5: 5 }
for (const num in number) {
    console.log(number[num]);
}
```

to get key value

* Functions : Normal Function

Function with return type & Parameter

Anonymous Function

Expression Function

Self Invoking Function

* no arrow functions in objects

```
* let userdetails = {
```

name: "Mohit",

last name: "Kumarant",

courseList: [],

website: "Mohit.me",

```
buyCourse: function (courseName) {
```

this.courseList.push(courseName);

}, userdetails.courseList.push(courseName); error

}

```
userdetails.buyCourse("FS TS");
```

```
console.log(userdetails);
```

userdetails not
in functions

```
* let userdetails = {  
    name: "Mohit",  
    get name: "Kumarant",  
    courselist: [],  
    website: "Mohit.me",  
    buyCourse: function (courseName) {  
        this.courselist.push(courseName);  
    },  
    getCourseCount: function () {  
        return `${this.firstname} is enrolled  
        in ${this.courselist.length}`;  
    },  
    userdetails.buyCourse("FS TS"),  
    console.log(userdetails());
```

* Dribble.com for Practise