**JavaScript**

* Const A = new String (“Gaurav”). // Browzer
* Math.random()

**Array**

* Const A12 = [23, 34, [45,67], 76, [12,17,91], 3]. //Dynamic Array in JS
* Const A = new Array(1,2,3,4)
* A.push(34).
* A.pop()
* A.unshift(9)
* Console.log(A.indexof(2)). // etc
* Slice and Splice
* Let A = [1,2,3,4,5,6,7,8,9]
* Let B = A.slice(1,3). Vs let C = A.splice(1,3)

**Concat vs spread Operator**

* Const A = [1,2,3]
* Const B = [4,5,6]
* Const C = A.concat(B)
* Const D = […A, …B]
* Const E = A12.flat(infinity)
* Array.isArray(“Gaurav”). // false
* .from(“Gaurav”). // will convert into array

**forEach, filter, map, reduce**

**Object**

* Const A = { 1: “Gaurav”, roll : “007”, “email”: [gaurav@mail.com](mailto:gaurav@mail.com)”}
* Console.log(A.age)
* Console.log(A[“roll”])
* Console.log(A.email)
* Console.log(A[“email”])
* A.roll = “001”. // overrite
* Object.freeze(A).
* A.roll = “003” // It will not work

**Assign vs spread Operator**

* Const C = Object.assign({ }, A, B) // A and B will be merged in C
* Const D = { …A , …B } // It will do the same
* Array of Objects
* Object Destructuring
* Const ABC = { A: “ghg”, B : “hgjgj”, C : “fhgfhgj”}
* Const {A, B , C } = ABC. //Destructured
* Console.log(A). // Directly we can access
* Discuss Something About JSON

**Function Execution (Browser and Node environment)**

Variable Environment(Memory) and Thread of Execution(Code)

**Function**

* Const function add(num1, num2){} //Anonymous
* Const add = function(num1, num2){} //Named
* Const add = ()=> {} //Arrow
* Discuss **Hoisting**
* Const function A(…num1){

Console.log(num1)

} //Rest Operator

Function reference

Function call

A(20);

A(20,50,90);

* Const function A(num1, num2, …num3, num4){

Console.log(num1)

}

A(10,20,30,40,50,60,70,80)

**Arrow Function**

* Const A = () = > { return }. // return is neccessary
* Const A = () = > a+b
* Const A = () = > (a+b)
* Const A = () = > { name: “MERN”} // can not return
* Const A = () = > ({ name: “MERN”}) // correct syntax
* **Immediately Invoked Function Expression**

**(() = > {}) () ;**

**((name) = > {}) (“Raj”) ;**

Discuss **this. //Both for Browser and Node**

* **This for Arrow function**