1. Give an example where call apply bind is. required?

**Bind =>** The bind() method is reminiscent of call() and apply(). But instead of executing a function immediately, bind() returns a functionthat can be executed later on.

**Example for bind**: - let car = {

brand:"Toyota",

price:300000,

details:function(ownername,color){

return {

brand:this.brand,

price:this.price,

ownername:ownername,

color:color

}

}

}

let avinash\_car=car.details.bind(car)

let avinash=avinash\_car("avinash","red")

let yogesh\_car=car.details.bind(car)

let yogesh=yogesh\_car("yogesh","yellow")

console.log(avinash,yogesh)

**Call** => The Call() method invokes a function with a specified context. In other words, you can tie a function into an object as if it belonged to the object.

**Apply** => The apply() method does the exact same as call(). The difference is that call() accepts an argument list, but apply() accepts an **array** of arguments.

**Example for call and apply**: - let car = {

brand:"Toyota",

price: 300000,

}

function details(ownername,color){

return {

brand:this.brand,

price:this.price,

ownername:ownername,

color:color

}

}

let avinash\_car=details.call(car,"avinash","red")

// OR

let avinash\_car=details.apply(car,["avinash","red"])

console.log(avinash\_car)

1. What is the difference between readFile and readFileSync?

readFileSync() is synchronous and blocks execution until finished. These return their results as return values. readFile() are asynchronous and return immediately while they function in the background. You pass a callback function which gets called when they finish.

**Exampel**:- var fs = require('fs');

fs.readFile(filename, "utf8", function(err, data) {

if (err) throw err;

console.log(data);

});

1. What does process in node.js mean?

Node.js provides the facility to get process information such as process id, architecture, platform, version, release, uptime, upu usage etc. It can also be used to kill process, set uid, set groups, unmask etc.

The process is a global object, an instance of EventEmitter, can be accessed from anywhere.

1. Explain what node.js is?

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

1. What is the difference of JS from browser to JS on node.js
2. Write three different ways to reverse a string in Javascript? a. using inbuilt method, b. iteratively, c. recursively

let str = "abcd"

//inbuilt way

let arr=str.split('')

let ans = arr.reverse()

console.log(ans.join(''))

// inbuilt way

let bag=""

for(let i=str.length-1;i>=0;i--){

bag+=str[i]

}

console.log(bag)

// recursive way

console.log(check(str,str.length-1,''))

function check(str,index,bag){

if(index<0){

return bag

}

bag+=str[index]

return check(str,index-1,bag)

}

1. Write a program to check two objects are equal ( deep equal )

function object\_equal(obj1,obj2){

let keys1=Object.keys(obj1)

let keys2=Object.keys(obj2)

if(keys1.length!==keys2.length){

return false

}

for(let i=0;i<keys1.length;i++){

if(keys1[i]!==keys2[i]){

return false

}

}

let values1 = Object.values(obj1)

let values2 = Object.values(obj2)

for(let i=0;i<values1.length;i++){

if(values1[i]!==values2[i]){

return false

}

}

return true

}

let obj1={name:"Avinash",age:22}

let obj2={name:"Avinash",age:22}

console.log(object\_equal(obj1,obj2))

1. What is shallow equal?

Shallow equal is a library which is used to compare the two objects or two arrays