**JAVASCRIPT:**

**Basics:**

**String methods:**

String.length;

indexOf():returns the index of (the position of) the first occurrence of a specified text in a string, second parameter as the starting position for the search

lastIndexOf() method returns the index of the **last** occurrence of a specified text in a string, second parameter as the starting position for the search. searches backwards (from the end to the beginning).

search() method searches a string for a specified value and returns the position of the match

* The search() method cannot take a second start position argument.
* The indexOf() method cannot take powerful search values (regular expressions).
* slice(*start*, *end*) extracts a part of a string and returns the extracted part in a new string. If a parameter is negative, the position is counted from the end of the string.
* substring(*start*, *end*)
* substr(*start*, *length*)

**Object extension:**

***Object.setPrototypeOf();***

It will assign the prototype of one object to the other so that you can access it inside the other object.

let a = {

    x:1

};

let b={

    y:2

};

Object.setPrototypeOf(a,b);

console.log(a.y);

//output:

//2

***Object.assign(target, param1, param2 ,....)***

It will populate the target with the all other parameters

Object.assign is going to walk through the immediate properties of an object and not through the prototype chain.

let a={a :1};

let b={b:2};

let target={};

Object.assign(target,a,b);

console.log(target);

//output:

//{a:1,b:2}

let a={a:1}, b={a:5,b:2}, c={c:20};

Object.setPrototypeOf(b,c);

let target={};

Object.assign(target,a,b);

console.log(target);

//{a:5.b:2}

***Object.defineProperty(object, property name, {value: , enumerable:})***

If enumerable is set to false then added property to defineProperty won’t be included.

By default the properties are enumerable .

***NaN***

let amount=NaN;

console.log(amount === amount);

//false

***Object.is***

let amount=NaN;

console.log(Object.is(amount,amount));

//true

In javascript 0 is identical to -0

let amount=0,total=-0;

console.log(amount===total);

//true

let amount=0,total=-0;

console.log(Object.is(amount,total));

//false

**Explain the difference between classical inheritance and prototypal inheritance.**

**A**

The great thing about JavaScript is the ability to do away with the rigid rules of classical inheritance and let objects inherit properties from other objects.

* **Classical Inheritance:** A constructor function instantiates an instance via the “new” keyword. This new instance inherits properties from a parent class.
* **Prototypal Inheritance**: An instance is created by cloning an existing object that serves as a prototype. This instance—often instantiated using a factory function or “Object.create()”—can benefit from selective inheritance from many different objects.
* **Give an example of a time that you used functional programming in JavaScript.**
* **A**
* Functional programming is one of the key paradigms that makes JavaScript stand out from other languages. Look for examples of functional purity, first-class functions, higher-order functions, or using functions as arguments and values. It’s also a good sign if they have past experience working with functional languages like Lisp, Haskell, Erlang, or Clojure.
* **Give an example of a time that you used Prototypal OO in JavaScript.**
* **A**
* Prototypal OO is the other major programming paradigm that really lets JavaScript shine—objects linked to other objects (OLOO). You’re looking for knowledge of when and where to use prototypes, liberal use of “Object.assign()” or mixins, and a solid grasp of concepts like delegation and concatenative inheritance.
* **What is a RESTful Web Service?**
* **A**
* REST stands for Representational State Transfer, an architectural style that has largely been adopted as a best practice for building web and mobile applications. RESTful services are designed to be lightweight, easy to maintain, and scaleable. They are typically based on the HTTP protocol, make explicit use of HTTP methods (GET, POST, PUT, DELETE), are stateless, use intuitive URIs, and transfer XML/JSON data between the server and the client.

MERN:MongoDb, Express.js , reactjs, nodejs

MEAN:Mongodb,Expressjs,angular,node.js.