JavaScript

1. ES6 features
   * Spread and rest operator
   * Arrow functions
   * Parameter default values
   * Classes
   * Generators
   * Promises etc.
2. how does a function getting executed in JavaScript engine?

Need to explain Call stack, Web API, Queue.

1. How does hosting works in JavaScript
   * Functions and variables are stored in memory for an execution context before we execute our code. This is called hoisting.
   * Functions are stored with a reference to the entire functions, variables with the var keyword with the value of undefined, and variables with the let and const keyword are stored uninitialized
2. What is call bind and apply

* These methods are available in the ‘base constructor function’ from which every function is derived. As we know that on every function execution following things are available in execution scope: This, arguments, Environment variables
* Bind - Return function with new this.
* Call - Call function with new this and arguments.
* Apply - same as call only difference is passing a arguments we need to pass arguments in array for apply.

1. Explain the closure in JavaScript and give once example where we can use it.

* function with lexical scope
* While executing memory will get allocated for the function variable. As soon as the function completes its execution all the variables from that function will be available for the garbage collector.
* If we used outer function variable in inner function then the variable which is used will not be available for garbage collector althose outer function is out of stack.
* The mechanism which holds this kind of variable is called closure.

1. Explain prototypal inheritance in JavaScript.
   * Object is parent object. Like for Array and function and every object is derived from it.
   * Every object contains constructor function.
   * When we crate any object using constructor function, a prototype object gets created with it.
   * Prototype property is refenced to the constructor function.
   * When we create a instance using this object then every instance contains \_\_proto\_\_
   * Property which belongs to the parent prototype property.
   * In this way it will create a prototype chain that is called prototypal inheritance.
   * ES6 – class – it just a syntactical sugar to constructor function.

Angular

1. Why angular becoming so popular and what are the disadvantages of it.

* Angular have many advantages over tradition app
* SPA, CLI, DI, easy development, good architectural style, testing tool, data binding. Lazy loading etc
* Disadvantage – Build size when application grow, pure micro-front end is hard to implement, separate deployment of modules

1. how to improve angular performance.
   * Lazy load modules if possible components,
   * component push strategy,
   * unsubscribe each observable,
   * lightweight dependency injection token,
   * remove unnecessary input outputs,
   * stop calling functions from template expression,
   * use pure pipes
2. Angular life cycle hooks.
   * OnChange
   * OnInit
   * DoCheck
   * NgAfterContentInit
   * ngAfterContentChecked
   * ngAfterViewInit
   * ngAfterViewChecked
   * ngDestroy
3. What is Eager and Lazy loading? how to minimize main module size.
   * Eagerly – load with main.js
   * Lazy – load on demand
   * Good folder architecture, good coding style, do not use shared modules use specific dependency in class, use specific imports from library.
4. What is DI in angular and how does it work.
   * Is a design pattern in which classes ask its dependency from an external source instead of creating it.
   * many ways to define DI -> platform, root, any
   * Injector hierarchy (Injector resolution rules): Null injector <- Platform injector <- Root injector <- Child injector
   * types: module, element injector hierarchy.
5. Different ways of component communication.
   * Input properties
   * Output events
   * Input getter setter
   * Change on OnChanges()
   * After @childView()
   * Via service with observable
   * Via local/template variable
6. How are observables different from promises?

* Promise
* Emits a single value
* Not Lazy Cannot be cancelled
* Observable provides operators like map, forEach, filter, reduce, retry, retryWhen etc.
* Observable
* Emits multiple values over a period of time
* Lazy. An observable is not called until we subscribe to the observable
* Can be cancelled by using the unsubscribe() method

Css

1. What is the Box model in CSS and explain border box.
   * Everything in css has a box around it.
   * Parts of a box , Content box =w, Padding box = p, Border box = b, Margin box =m
   * The box-sizing CSS property sets how the total width and height of an element is calculated.
   * Border box (height = CPB)
   * Content-box (height = C)
2. What are the properties of flexbox?
   * One dimensional layout method for laying out items in row or column.
   * Items will grow and shrink according to available space.
   * Always depends on main and cross axis
   * And this will according to row and column
3. Explain CSS position property?
   * Absolute, Relative, Fixed, Static, Sticky