



JavaScript OO
Interview
Questions

JavaScript OO Interview Questions

JavaScript Object Oriented Programming Interview Questions

Left Blank

Q1	What makes Javascript an Object Oriented?
Answer	<ul style="list-style-type: none"> • Dynamic Dispatch Which Code to execute determined at time of execution. Javascript used prototype chain • Polymorphism Like object can be treated similarly • Encapsulation Like data and functionality is bundled • Inheritance Object can be specialized
Q2	What is Object Oriented Programming ?
Answer	OOP is a way of logically organizing our code in a way that makes sense with respect to the problem we are trying to solve.
Q3	What is Classes?
Answer	<ul style="list-style-type: none"> • Classes are abstract representation of things. • It has 'A' relation with thing. • Classes are like template • Example:- A Car but not a specific car.
Q4	What are Object or Instance?
Answer	<ul style="list-style-type: none"> • Object is concrete thing. • They inherited not only their super class but all their ancestor. • Example My pet
Q5	How Javascript Differs from Other Object Oriented languages?
Answer	<ul style="list-style-type: none"> • There is no formal concept of a class • Classes are function that create Object • javascript provides additional mechanisms for code re-use • Javascript is prototypal • Methods can be invoked as if they belong to other object • Function can be invoked as methods
Q6	What is Simplest way of defining Object in javascript?
Answer	<ul style="list-style-type: none"> • Simplest way of defining object in javascript is map or container. • Curly braces are used to define container. Example : var I = {} <p>This is simplest object without property.</p>

Q7	How to Add Property in Object?
Answer	<p>There 2 common way for adding property.</p> <ul style="list-style-type: none"> • Add inside the curly braces or container Example :- var I = {name:"Avinash"} • Add with dots to object Example: var I = {}; I.name = "Avinash"
Q8	How to delete Property in Object?
Answer	<p>Delete and Object Property.</p> <p>Example delete I.name;</p>
Q9	How access property in Javascript Object?
Answer	<p>There commonly 2 ways of accessing property</p> <ul style="list-style-type: none"> • Dot notation Example I.name • Square bracket notation is also called computed member access. Example I['name']
Q10	What will have if object name start with number?
Answer	<ul style="list-style-type: none"> • Javascript will not able to process request. Example I.2name ='Avinash2' won't work. • if we want make it work we have define as string literal. Example I['2name'] = 'Avinash2'
Q11	How to add property with space and access them?
Answer	<p>We have define as string literal We cannot access using dot notation We have use Square bracket notation to access property with spaces</p> <p>Example I.['Birth place'] = 'India'</p>
Q12	How to access property as variable name?
Answer	<p>Using Square bracket.</p> <p>Example :-</p>

	<pre>var I = { name : 'Avinash'; }</pre> <p>name need to accessed as variable.</p> <pre>var propertyasVariable = 'name';</pre> <p>when we access I ['propertyasVariable'] the out will be "Avinash".</p>
Q13	What are primitive type?
Answer	<ul style="list-style-type: none"> • Undefined • Null • Boolean • String • Number
Q14	Can Be create object out of Primitive type?
Answer	<ul style="list-style-type: none"> • Yes, but we should not do that. It can be created without new keyword.
Q15	What is typeof operator used for?
Answer	<p>typeof used for finding what type object or primitive is.</p> <p>Is it string, number etc.</p>
Q16	Is function are Object in javascript?
Answer	<ul style="list-style-type: none"> • Yes example :- function helloworld() {return "Hello world"} • If use command instanceof helloworld instanceof Function Output would be :- true • helloworld instanceof Object Output would be :- true
Q17	Can function be chained in Javascript?
Answer	<ul style="list-style-type: none"> • Yes, function can be chained
Q18	How to invoke a function in object?
Answer	<ul style="list-style-type: none"> • By adding round bracket after function name (). <p>example :-</p> <pre>Var data = { totalpopulation : function () {return 1}; }</pre> <p>If we call using function "data.totalpopulation" Output :- [FUNCTION]</p> <ul style="list-style-type: none"> • If We add rounded bracket ().

	<p>"data.totalpopulation"</p> <p>Output :- 1;</p> <ul style="list-style-type: none"> Function can be invoked using square bracket also. <pre>data['totalpopulation']();</pre>
Q19	What is inner function in javascript?
Answer	<ul style="list-style-type: none"> Function defined inside function is called inner function. Example <pre>Var data = { rateofgrowth = 2; totalpopulation : function () {return 1}; populationMutipltedBygrowth: function() { function innerfunction(){ return this.rateofgrowth * this. Totalpopulation()} } }</pre>
Q20	Explain callback in javascript?
Answer	<ul style="list-style-type: none"> Call back calling back function or method for operation purposes. SetTimeout is used normally to create callback. <p>Example :-</p> <pre>var persondetail = { personalData : [{name:'Avinash', country:'India'}], addNewPersonaldetail: function() { this. personalData.push({name:'Avinash', country:'USA'})} }</pre> <ul style="list-style-type: none"> SetTimeout(function() { persondetail. addNewPersonaldetail()},2000); <p>The set time out will execute the function in 2 sec and add it to personalData array.</p>
Q21	What the ways to Call function?
Answer	<ul style="list-style-type: none"> Call and Apply the other 2 way of calling function. Example :- <pre>Function sqsum(a,b){ Return Math.pow(a,2) + Math.pow(b,2); }</pre>

	<ul style="list-style-type: none"> • <code>Sqsum.call(null,2,3);</code> output :- 13 • <code>Sqsum.apply(null,[2,3]);</code> output :- 13
Q22	What is Bind in javascript?
Answer	<ul style="list-style-type: none"> • Binding is other way dealing with method when we cannot call against object. • When bind it used object wedded with method. • Syntax: <code>object.method.bind(parameter)</code>
Q23	What scope does javascript have?
Answer	<ul style="list-style-type: none"> • It has function scope. example <pre>function TestScope() { if(true) { var foo ='1'; } console.log(foo) }</pre> <ul style="list-style-type: none"> • In Language such as C# foo will fail and give compile time error but in Javascript it will work. • Javascript manages function level scope know as hoisting.
Q24	How hoisting works?
Answer	<ul style="list-style-type: none"> • In javascript variable declaration is kept on top of function. <p>Example</p> <pre>function TestScope() { if(true) { var foo ='1'; } console.log(foo) }</pre> <ul style="list-style-type: none"> • How java script see its. <pre>function TestScope() {</pre>

	<pre> Var foo; if(true) { foo ='1'; } console.log(foo) </pre>
Q25	What s “let” variable introduced in ES6?
Answer	<ul style="list-style-type: none"> • Let variable is replacement of var keyword. • It declares variable at block level rather than function level
Q26	What if Var or let keyword is not defined before variable?
Answer	<ul style="list-style-type: none"> • Javascript thinks that we want declare it as global variable. • This not a good idea to use. • It can be prevent by using ‘use strict’
Q27	What is closure in javascript?
Answer	<ul style="list-style-type: none"> • In javascript we don’t have private, public etc to define access level of function and method. • Closure is issued to hide function method and other thing in javascript. <p>Example</p> <ul style="list-style-type: none"> • <pre> function getcounter() { var count = 0; return function(arg) { if(arg === 'r' {count = 0;} else (console.log ('function' +++ count) } } </pre> <p>In above example it shows closure only exposes what it want to exposes.</p>
Q28	What is new keyword in javascript?
Answer	<ul style="list-style-type: none"> • New operator is used to create instance of object. <p>Example :-</p> <pre> function Car() { this.carStore = 'Maruti Gallrey'; this.items =[]; } Var car = new Car(); </pre>
Q29	What will happen if we don’t use new operator while initializing the object?
Answer	<ul style="list-style-type: none"> • Var which is init becomes undefined.

	<ul style="list-style-type: none"> • The property in object becomes global. • If eliminate new operator, function will execute in global scope.
Q30	What is prototype?
Answer	<ul style="list-style-type: none"> • It means when function is used in object constructor. • Allows shared functionality • Make inheritance possible. • Every function have property prototype • Defined once and used by other instances
Q31	Example of prototype?
Answer	<pre>function Car() { this.carStore = 'Maruti Gallery'; this.items =[]; }</pre> <p>If Initialize object</p> <pre>Var car1 = new Car();</pre> <pre>Var car2 = new Car()</pre> <pre>car1.prototype.empty = function() {this.items[]}</pre> <p>The above function will available in both car1 and car2 even it was declared on car1.</p> <p>Other way declaration is</p> <pre>Car.prototype.empty = function() {this.items[]}</pre>
Q32	How to Prototype or Inheritance chain is setup in javascript?
Answer	<ul style="list-style-type: none"> • By using prototype and object create. • Example <pre>function A() {}; function B() {}; function C(){} A.prototype = Object.create(B.prototype){}; C.prototype = Object.create(A.prototype){};</pre>
Q33	What does slice do if no argument is passed?
Answer	<ul style="list-style-type: none"> • It will create shallow copy of the object upon which it had been called.
Q34	What is Syntactic Sugar In Harmony ES6?
Answer	<ul style="list-style-type: none"> • The way be define class in javascript

	<p>have changed.</p> <ul style="list-style-type: none"> • Rather then using prototype we can use class keyword. • Rather the attaching method and function to class we can create them as we do in normal C# or java.
Q35	What is duck typing?
Answer	<ul style="list-style-type: none"> • Idea is if walks like duck, and quacks like duck it is duck. • It is not good way finding type of object. • Example :- There are 4 ways for finding if it array Instanceof Array Array.isArray() Length push
Q36	What is keyword extend?
Answer	<ul style="list-style-type: none"> • Extend is usually a high level function that copies the prototype of new subclass that you want to extend from the base class.

}