26 Apr 2015

Object Oriented JavaScript Interview Questions and Answers

What is JavaScript?

The JavaScript is most popular scripting languages and it developed by Netscape and used to develop the client side web applications.

Most popular 17 key questions for Interviews as given below.

1). What is Object in JavaScript?
2). What is the Prototype object in JavaScript and how it is used?
3). What is "this"? What is its value?
4). Explain why "self" is needed instead of "this".
5). What is a Closure and why are they so useful to us?
6). Explain how to write class methods vs. instance methods.
7). Can you explain the difference between == and ===?
8). Can you explain the difference between call and apply?
9). Explain why asynchronous code is important in JavaScript
10). Can you please tell me a story about a JavaScript performance problems?
11). Tell me your JavaScript Naming Convention?
12). How do you define a class and its constructor?
13). What is Hoisted in JavaScript?
14). What is function overloading in JavaScript?
15). What is scope variable in JavaScript?
16). What is associative arrays in JavaScript?
17). How to achieve inheritance in JavaScript?

Is JavaScript case sensitive?

Yes! JavaScript is a case sensitive because a function str is not equal to Str.

What is the Type of JavaScript?

There are different of Type as given below.

1. String,
2. Number,
3. Boolean,
4. Function,
5. Object,
6. Null,
7. Undefined etc

What types of Boolean operators in JavaScript?

There are three types of Boolean operators as given below.

1. AND (&&) operator,
2. OR (||) operator and
3. NOT (!) Operator

What is the difference between “==” and “===”?

The double equal “==” is an auto-type conversion. It’s used to check the value and type of its variable.

The three equal “===” is not auto-type conversion. It’s used to check only value of its variable.

The real example as given below:

```
if(1 == ”1”) // Its returns true because it's an auto-type conversion and it check value and type both.

if(1 === ”1”) // Its returns false because it's not auto-type conversion and it check only value not  type.

If(1=== parseInt(”1”)) // Its returns true.
```

What is an Object?

The object is a collection of properties and the each property associated with the name-value pairs. The object can contain any data types (numbers, arrays, object etc.)

BLOG ARCHIVE

▼ 2015 (160)
► Sep 2015 (1)
► Aug 2015 (18)

POPULAR POSTS

Angularjs Do
Implementati
Hello everyor
going to shar
interesting dc
and its live demo of Angula

- ► Jul 2015 (26)
- ► Jun 2015 (31)
- ► May 2015 (16)
- ▼ Apr 2015 (20)
  - ► Apr 29 (1)
  - ► Apr 28 (1)
  - ▼ Apr 26 (1)
    - Object Oriented JavaScript Interview Questions and...
  - ► Apr 24 (1)
  - ► Apr 21 (1)
  - ► Apr 15 (1)
  - ► Apr 09 (12)
  - ► Apr 07 (1)
  - ► Apr 01 (1)
- ► Mar 2015 (14)
- ► Feb 2015 (23)
- ► Jan 2015 (11)
- ► 2014 (183)
- ► 2013 (62)
- ► 2012 (11)

The example as given below

```
Var myObject= {empId : “001”, empCode :”X0091”};
```

In the above example, here are two properties one is empId and other is empCode and its values are “001” and “X0091”.

The properties name can me string or number. If a property name is number i.e.

```
Var numObject= {1 : “001”, 2 :”X0091”};

Console.log(numObject.1); //This line throw an error.

Console.log(numObject[“1”]); // will access to this line not get any error.
```

As per my thought, the number property name should be avoided.

Types of creating an object

1. Object literals
2. Object constructor

Object Literals: This is the most common way to create the object with object literal and the example as given below.

The empty object initialized using object literal i.e.

```
var emptyObj= {};
```

This is an object with 4 items using object literal i.e.

```
var emptyObj ={
  empId:”Red”,
  empCode: “X0091”,
 empDetail : function(){
    alert(“Hi”);
};
};
```

Object Constructor: The second way to create object using object constructor and the constructor is a function used to initialize new object.

The example as given below

```
Var obj = new Object();
Obj.empId=”001”;
Obj.empCode=”X0091”;
Obj.empAddressDetai = function(){
       Console.log(“Hi, I Anil”);


};
```

What is scope variable in JavaScript?

The scope is set of objects, variables and function and the JavaScript can have global scope variable and local scope variable.

The global scope is a window object and its used out of function and within the functions.
The local scope is a function object and its used within the functions.

The example for global scope variable

```
  var gblVar = "Anil Singh";

  function getDetail() {
      console.log(gblVar);
  }
```

and other example for global

```
  function demoTest() {
      x = 15;
  };
  console.log(x); //out put is 15
```

The example for local scope variable

docume...

ng-if else in A

Hello everyon
going to shar
for angularjs i
expression. T
provide the if(...

Date Calendar in ASP.NET image icon

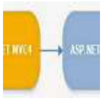
Hello everyone, In this post the code-sample for date c calender textbox in asp.net

Angularjs Inte and Answers

1. What is Ar
angularjs is a
|| MV*] fram
maintain by Google. The m
creating...

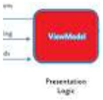
Object Orient
Interview Que
Answers

What is Java
JavaScript is
scripting languages and it d
Netscape and used to devel



Difference Be
MVC 4, MVC

ASP.NET MV
introduced a
new features
Razor . MVC 3 introduced
is ...



knockout js in
and answers

1. What is Kn
Knockout is a
library and wi
knockout JS we can create
...

Single and M
asp.net

Code sample
multiple file u
with c# I'm g
code sample for single and

Export ng-grid data to Excel format in angularjs

In the 1st Step, we write a p
exporting ng-grid data to CS
formats. In this plug-In, we

DropDownList in ASP.NET

Hello everyone, In this post,
to code-sample for bind cou
dropdownlist in asp.net usin

G+1 152

FOLLOWERS

```
function getDetail() {
    var gblVar = "Anil Singh";
    console.log(gblVar);
}
```

and other example for local

```
function demoText() {
    var x = 15;
};
console.log(x); //undefined
```

What is associative arrays in JavaScript?

What is array? Array is a collection of index items and it is a number indexes.

Some of programming language support array as named indexes and the JavaScript not support the array as name indexes and its provide only number indexes but provide this feature using the associative array.

The array with name indexes are called associative array and the associative array is provide a ways to store the information.

The number index array example as given below

```
var users = new Object();
users["Name1"] = "Anil 1";
users["Name2"] = "Anil 2";
users["Age"] = 33;
alert(Object.keys(users).length); //output is 3.
var length = Object.keys(users).length;  // 3
```

The name index array example as given below

```
var users = [];
users["Name1"] = "Anil 1";
users["Name2"] = "Anil 2";
users["Age"] = 33;
var length = users.length;         // users.length will return 0
var detail = users[0];             // users[0] will return undefined
```

Where to use the associate array?

I am going to explain the associate array over the database table columns. A table have some columns and its type. i.e.

The empName as text type, empAge as number type and enpDOB as date type.

If we need to find the type of a column that time we can create the associate array. i.e.

```
var empDetailType = new Array();

empDetailType["empName"] = "ANil";
empDetailType["empAge"] = 30;
empDetailType["empDOB"] = "10/03/1984";

console.log("Find the emp age type :" + empDetailType["empAge"]);
```

How to achieve inheritance in JavaScript?

In the JavaScript, we can implement the inheritance using the some alternative ways and we cannot define a class keyword but we create a constructor function and using new keyword achieve it.

The some alternative ways as given below.

1. Pseudo classical inheritance
2. Prototype inheritance

Pseudo classical inheritance is the most popular way. In this way, create a constructor function using the new operator and add the members function with the help for constructor function prototype.

The **prototype based** programming is a technique of object oriented programming. In this mechanism we can reuse the exiting objects as prototypes. The prototype inheritance also know as prototypal,
classless or instance based inheritances.

The Inheritance example for **prototype based** as given below

```
 // Create a helper function.
  if (typeof Object.create !== 'function') {

      Object.create = function (obj) {
          function fun() { };
          fun.prototype = obj;
```

Join this site
with Google Friend Connect

Members (41) More »

Already a member? Sign in

BLOG LIST

ASP.Net C# WCF SQL
KnockoutJs AngularJ
ASP.NET Membership Lo

Code Geek
How to Leverage Mongol
AngularJS

AngularJsHub
AngularJs Forms Validati

Upcoming New Techn
Upcoming Technology Th
The World

AllInOne

Online Core Java Tuto
Servlet Tutorial

oops concepts in c#

GA

POLL

This site is useful for

□ Yes

□ No

Vote

View Results P

```
            return new fun();
        };
    }

    //This is a parent class.
    var parent = {
        sayHi: function () {
            alert('Hi, I am parent!');
        },
        sayHiToWalk: function () {
            alert('Hi, I am parent! and going to walk!');
        }
    };

    //This is child class and the parent class is inherited in the child class.
    var child = Object.create(parent);

    child.sayHi = function () {
        alert('Hi, I am a child!');
    };

<button type="submit" onclick="child.sayHi()"> click to oops</button>
```

The output is : Hi, I am a child!

What is typeof operator?

The **typeof** operator is used to find the type of variables.

The example as given below.

```
typeof "Anil Singh"              // Returns string
typeof 3.33                 // Returns number
typeof true                 // Returns Boolean
typeof { name: 'Anil', age: 30 } // Returns object

typeof [10, 20, 30, 40]           // Returns object
```

What is public, private and static variables in JavaScript?

I am going to explain like strongly type object oriented language (OOPs) like(C#,C++ and java etc.).

Fist I am creating a conductor class and trying to achieve to declare the public, private and static variables and detail as given below.

```
function myEmpConsepts() { // This myEmpConsepts is a constructor  function.

    var empId = "00201"; //This is a private variable.
    this.empName = "Anil Singh"; //This is a public variable.
    this.getEmpSalary = function () {  //This is a public method
        console.log("The getEmpSalary method is a public method")
    }
}

//This is a instace method and its call at the only one time when the call is instanciate.
myEmpConsepts.prototype.empPublicDetail = function () {
    console.log("I am calling public vaiable in the istance method :" + this.empName);
}

//This is a static vaiable and its shared by all instace.
myEmpConsepts.empStaticVaiable = "Department";

var instanciateToClass = new myEmpConsepts();
```

Who to create the namespace in JavaScript?

Please see the below example for how to create the namespace in JavaScript.

```
//Create the namespace.
var nameSpace = nameSpace || {};

nameSpace.myEmpConsepts = function () {

    var empId = "00201"; //This is a private variable.
    this.empName = "Anil Singh"; //This is a public variable.
    //This is public function
    this.getEmp = function () {
        return "Anil Singh"
    }
    //This is private function
    var getEmp = function () {
          return "Anil Singh"
    }

    return {
        getEmp: getEmp,// work as public
        getEmp: getEmp // work as public
    }
}();
```

How to add/remove properties to object in run-time in JavaScript?

I am going to explain by example for add and remove properties from JavaScript objects as give below.

This example for delete property.

```
//This is the JSON object.
var objectJSON = {
    id: 1,
    name: "Anil Singh",
    dept: "IT"
};

//This is the process to delete
delete objectJSON.dept;

//Delete property by the array collection
MyArrayColection.prototype.remove = function (index) {
    this.splice(index, 3);
}
```

This example for add property

```
//This is used to add the property.
objectJSON.age = 30;

console.log(objectJSON.age); //The result is 30;


//This is the JSON object.
var objectJSON = {
    id: 1,
    name: "Anil Singh",
    dept: "IT",
    age :30

};
```

How to extending built-in objects in JavaScript?

JavaScript support built-in objects which use to develop the flexibility of JavaScript. The built in object are date, string, math, array and object. It's very similar to other languages and its available in the window content and wok independently when brewers are loaded.

Example as give below

```
var date = new Date(); //This is date built-in object.
var math = Math.abs(10); // this is math built-in object.

var string = "my string" // this is string built-in object.
```

Why never use new Array in JavaScript?

We have some fundamental issues with new Array() the example in detail for array constructor function as given below.

When array have more the one integer?

```
var newArray = new Array(10, 20, 30, 40, 50);

console.log(newArray[0]); //returns 10.
console.log(newArray.length); //returns 5.
```

When array have only one integer?

```
var newArray = new Array(10);

console.log(newArray[0]); //returns undefined
console.log(newArray.length); //returns 10 because its has an error "array index out of bound";
```

```
//This is the fundamental deference to need to avoid the new array();
```

What is eval() and floor() functions in JavaScript?

The **eval()** function used in execute an argument as expression or we can say that evaluate a string as expression and it used to parse the JSON.

The example over eval() function as given below.

```
var x = 14;
eval('x + 10'); //The output is 24.
```

Another over eval() function example

```
eval('var myEval = 10');
console.log(myEval); // The output is 10.
```

The **floor()** function is a static method of Math and we can write as Math.floor() and used to round the number of downwards. i.e.

```
Math.floor(1.6);//The output is 1.
```

What is join() and isNaN() functions in JavaScript?

The is **join()** function used to join the separator in the array.

Syntax: myArray.join(mySeparator);

The example as given below.

```
var alfabets = ["A", "B", "C", "D"];

//Join without seperator
var result1 = alfabets.join();//The output is A B C D.

//Join with seperator.
var result2 = alfabets.join(','); //The output is A, B, C, D.
```

The **isNaN()** function is used to check the value is not-a-number.

The example as given below

```
var var1 = isNaN(-1.23);//The output is false.
var var2 = isNaN(3);//The output is false.
var var3 = isNaN(0);//The output is false.

var var3 = isNaN("10/03/1984"); //The output is true.
```

What is closure in JavaScript?

While you create the JavaScript function within another function and the inner function freely access all the variable of outer function. i.e.

```
function ourterFun(i) {
    var var1 = 3;

    function innerFun(j) {
        console.log(i + j + (++var1)); // It will return the 16.
    }
    innerFun(10);
}

ourterFun(2); // Pass an argument 2
```

The output will get 16 because *innerFun()* function can access to the argument "i" & variable "*var1*" but both are define in the *outerFun()* function that is closure.

That means simply accessing variable outside of your scope create a closure.

```
// OR Other WAYS

function ourterFun(i) {
    var var1 = 3;

    return function (j) {
        console.log(i + j + (++var1)); // It will return the 16.
    }
}

var innerFun = ourterFun(2); // innerFun() function is now a closure.
innerFun(10);
```

What is JavaScript Hoisted?

In the JavaScript, the variables can be used before declared, this kinds of mechanism is called Hoisted. It's a default behavior of JavaScript.

You can easily understanding in the below example in detail.

```
//The variable declaration look like.
var emp;

//The variable initialization look like.
emp = "Anil Singh";


var emp; //The declaration of emp is hoisted but the value of emp is  undefined.
emp = 10; //The Assignment still occurs where we intended (The value of emp is 10)

function getEmp() {
    var emp; //The declaration of a different variable name emp is hoisted but the value of emp is  undefined.

    console.log(emp); //The output is undefined
    emp = 20; //The assignment values is 20.
    console.log(emp); //The output is 20.
}

getEmp();
console.log(emp); //The variable named emp in the outer scope still contains 10.
```

What is function overloading in JavaScript?

There is no real function overloading in JavaScript and it allows to pass any number of parameters of any type.

You have to check inside the function how many arguments have been passed and what is the type arguments using *typeof*.

The example for function overloading not supporting in JavaScript as give below.

```
function sum(a, b) {
    alert(a + b);
}

function sum(c) {
    alert(c);
}

sum(3);//The output is 3.
sum(2, 4);//The output is 2.
```

In the JavaScript, when we write more than one functions with same name that time JavaScript consider the last define function and override the previous functions. You can see the above example output for the same.

We can achieve using the several different techniques as give below

1. You can check the declared argument name value is undefined.
2. We can check the total arguments with *arguments.length*.
3. Checking the type of passing arguments.
4. Using number of arguments
5. Using optional arguments like x=x || 'default'
6. Using different name in the first place
7. We can use the arguments array to access any given argument by using *arguments[i]*

What is prototype in JavaScript?

The prototype is a fundamental concept of JavaScript and its must to known JavaScript developers.

All the JavaScript objects has an object and its property called prototype and its used to add and the custom functions and property.

The example without prototype as given below.

```
var employee = function () {
    //This is a constructor  function.
}

//Crate the instance of above constructor  function and assign in a variable
var empInstance = new employee();
empInstance.deportment = "IT";

console.log(empInstance.deportment);//The output of above is IT.
```

The example with prototype as given below.

```
var employee = function () { //This is a constructor  function.}

employee.prototype.deportment = "IT";//Now, for every instance employee will have a deportment.

//Crate the instance of above constructor function and assign in a variable
var empInstance = new employee();
empInstance.deportment = "HR";

console.log(empInstance.deportment);//The output of above is IT not HR.
```

Posted by www.code-sample.com at Sunday, April 26, 2015

G+1 +89 Recommend this on Google

Labels: javascript interview questions answers, Object Oriented JavaScript Interview Questions Answers

55 comments

Add a comment as Jayakumar Srinivasan

Top comments

Anil Singh 1 month ago - Blogger (Debate)

A glance of Object Oriented JavaScript concepts and questions answers http://goo.gl/AHE3Kc

1

ante kotarac shared this 3 months ago - Programming (Discussion)

1

Sivabalan S via Google+ 1 month ago - Shared publicly

> **Anil Singh** originally shared this
> ask interview questions after interview in JavaScript

1 · Reply

Anil Singh 1 month ago - HTML5 CSS3 JavaScript (► Javascript jQuery)

A glance of Object Oriented JavaScript concepts and questions answers http://goo.gl/AHE3Kc

1

David Rivera via Google+ 3 months ago - Shared publicly

> **Anil Singh** originally shared this
> Advance Object Oriented JavaScript http://www.code-sample.com/2015/04/javascript-interview-questions-answers.html

1 · Reply

Kingsten Banh via Google+ 3 months ago - Shared publicly

Anil Singh originally shared this
Advance Object Oriented JavaScript http://www.code-sample.com/2015/04/javascript-interview-questions-answers.html

1 · Reply

Anil Singh 1 month ago - C# Developers (Help)
A glance of Object Oriented JavaScript concepts and questions answers http://goo.gl/AHE3Kc

1

Anil Singh 1 month ago - Educational Technology (Discussion)
A glance of Object Oriented JavaScript concepts and questions answers http://goo.gl/AHE3Kc

1

Anil Singh 1 month ago - JavaScript (Pure JS)
A glance of Object Oriented JavaScript concepts and questions answers http://goo.gl/AHE3Kc

1

Anil Singh 1 month ago - Web Development (Discussion)
A glance of Object Oriented JavaScript concepts and questions answers http://goo.gl/AHE3Kc

1

Anil Singh 1 month ago - Quality Guest blogging & posting (Guest posting opportunity)
A glance of Object Oriented JavaScript concepts and questions answers http://goo.gl/AHE3Kc

1

Anil Singh 1 month ago - Blogging (Business & Work blogs)
A glance of Object Oriented JavaScript concepts and questions answers http://goo.gl/AHE3Kc

1

Anil Singh 1 month ago - Programming (General Discussion)
A glance of Object Oriented JavaScript concepts and questions answers http://goo.gl/AHE3Kc

1

Anil Singh 1 month ago - Interesting Technology (Discussion)
A glance of Object Oriented JavaScript concepts and questions answers http://goo.gl/AHE3Kc

1

Anil Singh via Google+ 4 months ago - Shared publicly
JavaScript Interview Questions Answers
What
is JavaScript? The JavaScript is most popular scripting
languages and it developed by Netscape and used to develop the client side web
applications. Is JavaScript case sensitive? Yes! JavaScript is a case sensitive because a
function str is not equal ...

1 · Reply

Pradeep Rajpoot 1 week ago - Shared publicly
Its very useful

1 · Reply

Muhammad nasir 3 weeks ago - Foundations of Programming and Design Patterns (JavaScript)
http://www.code-sample.com/2015/04/javascript-interview-questions-answers.html

+1 1

Nj Al 4 weeks ago - Shared publicly
so many incorrect answers

1 · Reply

1 · Reply

Msc Web Solutions via Google+ 4 weeks ago (edited) - Shared publicly

#JavaScript #OOP
via **+Anil Singh**

1

Anil Singh 2 months ago - Promote Your Blog (Promote Personal Blog)

Object Oriented JavaScript Concepts
http://www.code-sample.com/2015/04/javascript-interview-questions-answers.html

1

Show more

Newer Post Home Older Post

code sample