

Blog

Home (<https://Mindmajix.Com/>) / JavaScript (<https://Mindmajix.Com/Javascript/>) / JavaScript Interview Questions

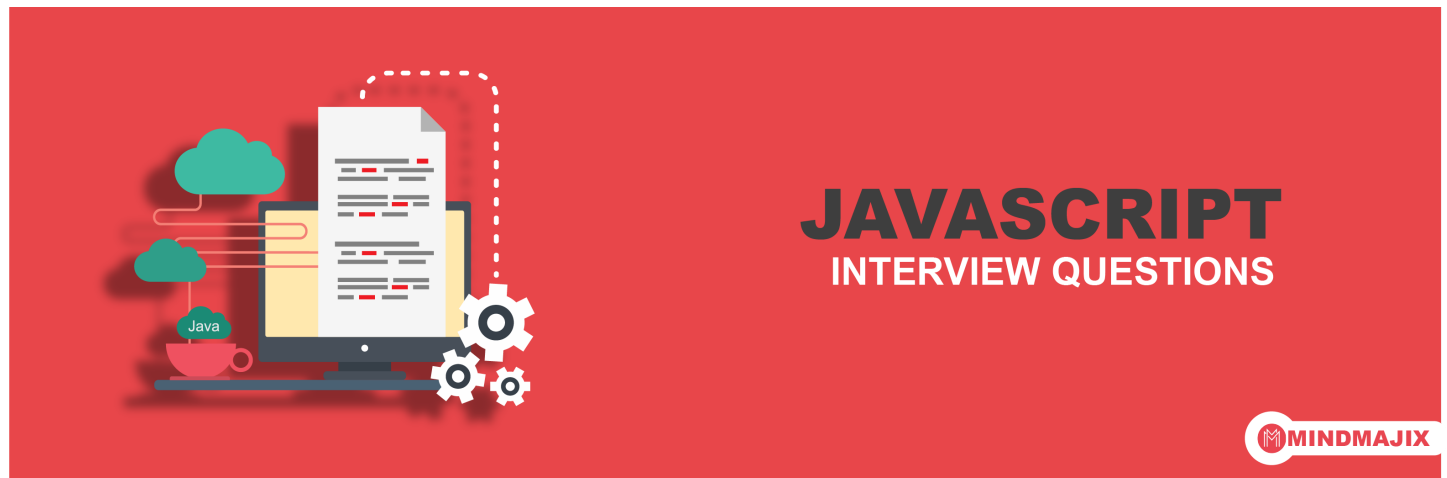
JavaScript Interview Questions

★★★★☆ (4.0) | 70381 Ratings

Bookmark

Email

70401



If you're looking for JavaScript Interview Questions for Experienced or Freshers, you are at right place. There are lot of opportunities from many reputed companies in the world. According to research JavaScript has a market share of about 54.87%. So, You still have opportunity to move ahead in your career in JavaScript Development. Mindmajix offers Advanced JavaScript Interview Questions 2019 that helps you in cracking your interview & acquire dream career as JavaScript Developer.

Are you interested in taking up for JavaScript Course? Enroll for Free Demo on JavaScript Online Training (<https://mindmajix.com/javascript-training>).

Q. Javascript Vs AngularJS

Javascript Vs AngularJS	
JavaScript	Angular JS
Programming language	Framework used to create mobile and web applications
It supports loops, conditions and exceptions	Doesnot support loops, conditions and exceptions
Filters are not supported	Supports filters
Helps in building frameworks	Built by using Javascript

Q. What are the differences between JavaScript and Java?

Call us on

Drop us a Query

The main difference between Java and JavaScript is that the former is a complete type of programming language while the latter is a programming language which is coded. The latter one can be introduced and put into HTML pages. Other than this, JavaScript and Java are designed for completely different purposes. The programming language like the C++ and C are all part of Java which in turn is actually an object-oriented programming structure. On the other hand, JavaScript is a client-oriented scripting programming language.

Q. Name the different types of JavaScript data?

JavaScript data are of the following types -

1. String
2. Function
3. Boolean
4. Object
5. Number
6. Undefined

Q. What is the definition of global variables? In what way, these variables are declared?

Global variable is a special kind of variable in JavaScript. This kind of variable is something that is easy to use and also available across the entire length of the JavaScript code. Mainly, the var keyword is used whether to declare a global or local variable.

Q. Name the problems that are associated with the use of global variables?

Even though global variables are easy to use, these have some shortfalls. While using this type of variables, the problem of clashing of the variable names of different global and local scope occurs. The code that is often relied on the global variable also gets difficult to be tested and debugged.

Q. What is the definition of prompt box?

The Input provided by any user in the JavaScript is entered with the help of a prompt box. While putting forward the data or the input, the prompt box allows the user to do so with the help of a text box. To include the number, label box is also used.

Q. What are the disadvantages of using inner HTML in JavaScript?

The use of inner HTML in JavaScript has the following disadvantages -

1. The process of using inner HTML is much slower than the rest of the variables as its content is slowly built into different elements and takes time to get re-parsed.
2. While using the inner HTML, the content gets replaced in JavaScript.
3. Appending to inner HTML can't be used properly.
4. Using inner HTML can also break the document of the JavaScript. Since no validation is required by it, any type of valid inner HTML can be used. Even broken HTML can also be used and this can cause problems.
5. The old content also gets easily replaced.

Q. Name the different type of groups of data types that are used in JavaScript and define them?

There are two basic groups of data types -

1. **Reference type** - These are complex types of data which can mainly include dates and strings.
2. **Primitive type** - These are types of data includes number data.

Q. Mention the advantages of using JavaScript programming language?

The different kinds of advantages of using JavaScript are -

☎ Call us on

✉ Drop us a Query

1. JavaScript provides the users to get immediate feedback. They can check the data which they entered without waiting for the page to be reloaded.
2. The server interacting is lesser and the load on the server is also much lesser. JavaScript helps in saving the traffic on the server.
3. Users can use various items to provide a rich interface to their own sites like they can use sliders and various drag-and-drop components.
4. JavaScript also increased interactivity between various users.

Q. Mention of disadvantages of using JavaScript programming language?

JavaScript has the following loopholes -

For the security of the JavaScript, the client side of the programming language does not allow the files to be read or to be written.

It also doesn't provide multiprocessor and multithreading capabilities.

Networking applications can't be provided by JavaScript as there is no such support that is available.

Q. JavaScript is a case sensitive language - Explain?

JavaScript programming language is case sensitive. This basically means those different types of variables, language keywords, functions and various identities, all these should be consistently used with the help of capitalized letters.

Q. Mention the different types of functions are supported by JavaScript? Define each of them?

There are two types of functions which are supported by JavaScript. They are -

1. **Anonymous function** - This type of function generally has no name and this is the difference between it and a normal function.
2. **Named function** - On the other hand, this function is something which is named properly and specifically.

Q. What are the different scopes of a variable? Define local variable?

The scope of a variable is generally defined as the region which acts as the place of the program.

Local variables are those types of variable which are visible across the specifically defined function. The parameter used is also local to the function.

Q. What is the meaning of the word 'callback'?

Callback is a typical function of the JavaScript which can be passed as an option or argument of JavaScript. Sometimes, callbacks can also be termed as simple events. Users are given calls to react to different kind of triggered situations.

Q. Mention the different types of rules which are used to name the various conventions in JavaScript?

The JavaScript reserved keywords as variable names should not be used while naming the convention. These keywords mainly appear in another section. Like the break variable or the Boolean ones are not valid ones for naming conventions.

Numerical data should not be used to start the variable name of the JavaScript. Letters and underscore should be used.

These variables are also case sensitive and hence there is a big difference between the words Name and name.

Q. Define the mechanism of the typeof operator?

The typeof operator is basically just the unary operator which is used before the single operand. The value indicates the operand's data type. The type includes Boolean, number, and string.

Q. How to create a class?

JavaScript does not have a class definition. To mimic classes in JavaScript functions can be used to declare a class.

Example:

☎ Call us on

✉ Drop us a Query

Let's create a student class in JavaScript which takes two parameters name and roll as property. The code will look like below,

```
Function Student (name ,roll){
  this.name = name;
  this.roll = roll;
}
```

Q. How to create an object?

An object in JavaScript can be created using two ways:

New Key word:

To create a student object from the above student class we can call the Student function using new keyword.

```
var student1 = new Student('santosh',2)
```

Anonymous Object:

Anonymous objects can be created using pair of curly braces containing property name and value pairs.

```
Var rose = { 'color': 'red' }
```

Q.How to open URL in new tab in javascript?

I think constant does not exist in javascript. But you can follow same type convention to declare constant.

```
var CONSTANT_NAME = "constant value";
```

Q. Can Cookie be accessed with the help of JavaScript?

JavaScript consists of a document object and its properties can be used to manipulate different kinds of cookies. It can modify and delete cookies. It can even create and read them.

Q. What are the meanings of undefined variables and undeclared variables?

Undefined variables are the types of variables which get declared in the program. However, the value of undefined variables is not provided in JavaScript programming language. The value gets returned if one tries to read the undefined value.

Undeclared variables are the types of variables which do not even exist in a program and are also not declared properly. The value also can't be read properly when tried to be read and hence a runtime error is shown.

Q. Define the various types of errors which occur in JavaScript programming language?

JavaScript programming language has three kinds of errors. They are as follows-

Run time errors - This type of error is the outcome of the misuse of the use of command within the HTML language.

Load time errors - Load time errors are basically syntax errors which are improper and arise when a web page is tried to be loaded. This type of error is generated dynamically.

Logical errors - A function often has a different operation and this type of error arises when the logic of the function is badly performed.

Q. Define the types of functional components which are available in JavaScript programming language?

There are two types of functional components that are available in JavaScript. These are -

1. Nested functions - As the name suggests, nested functions are the ones which are included in other functions. They are basically called each time when the invoked functions are provided.



2. First class functions - These are the types of functions which are used as first-class objects. Such functions can be used as arguments and opinions against other types of functions. The data structures can store the values of such functional components and the variables can also be stored properly in the data structures.

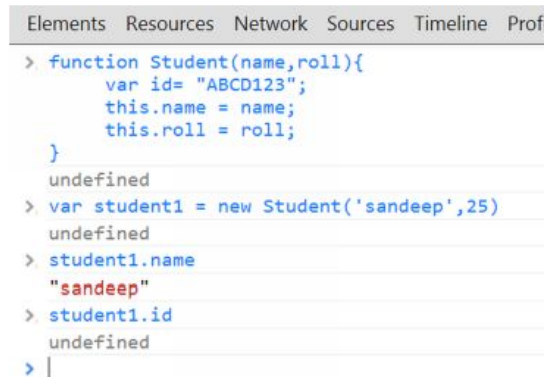
Check Out JavaScript Tutorials (<https://mindmajix.com/javascript>)

Q. How to declare a private and a public member?

Private members are declared using var keyword and constructor function.

```
Function Student (name, roll){
  var id= ABCD123;
  this.name = name;
  this.roll = roll;
}
```

When a Student object will be created the properties name and roll will be accessible using dot operator but id will not be accessible as it behaves as a private member and return undefined on call.



```
Elements Resources Network Sources Timeline Profil
> function Student(name,roll){
    var id= "ABCD123";
    this.name = name;
    this.roll = roll;
  }
undefined
> var student1 = new Student('sandeep',25)
undefined
> student1.name
"sandeep"
> student1.id
undefined
> |
```

The above chrome console is showing a student1 object is created. name property is accessible as it is showing sandeep on student1.name call. So name is a public property for the student object. But id property is not accessible and returned undefined on student1.id call. This shows id is a private property in student1 object.

Q. What is prototype property?

By Using Prototype we can add new members to an existing object. Every JavaScript object has this property internally. Initially it is an empty object.

Example:

```
function Student (name, roll){
  this.name = name;
  this.roll = roll;
}
```

```
var student1 = new Student('sangeeta',30);
```

```
Student.prototype.mark = 100;
```

Checkout the below chrome console for the use of Prototype.

Initially the student1 object has only two properties name and roll. By using prototype a new property mark has been added to student object with a value of 100. Now the console shows that the mark property is also added to the existing student1 object.

Q. How to convert a string to lowercase?

```
var str='This is testing String';
str = str.toLowerCase();
console.log(str);
```

Call us on

Drop us a Query

Q. Define Screen Objects. What is its proper?

Screen Objects are basically the objects which are used to read a different type of information from the screen of the client. The properties of the screen objects are as follows -

The height of the client screen is provided by it.

The images which appear on the client screen is also provided by it.

The width of the screen is also available.

Often the height of the client screen included the task bar and Screen Objects give the total height.

The width of the screen also includes task bar and the total width is available to the users.

Q. How can one handle exceptions in JavaScript?

The operators used to handle exceptions in JavaScript programming language are try, catch and construct. These are the latest versions.

Q. List the purpose of onError event handler?

The error handling in JavaScript is done mostly by onError event handler. It was basically JavaScript's feature. During the occurrence of an exception, the error event gets on the window object. To know the nature of the error, it provided three important messages like -

1. URL - This indicates the place of the file containing the error.

2. Error message - The error is provided and shown by this message to the user.

3. Line number - This is the line number of the shown URL. The particular URL is shown where there is an error.

Q. Differentiate between Jscript and JavaScript?

JScript is provided and developed by Microsoft whereas JavaScript is provided by Netscape.

Q. What is constructor property?

Constructor property of an object maintains a reference to its creator function.

Example:

Let us checkout an example by creating a student object and calling the constructor property on it.

```
function Student( name, mark){
  this. name = name; this. mark =mark;
}
var student 1 = new Student ( 'sandeep', 123);
console.log (student1.constructor);
function Student(name, mark){
  this.name = name;
  this.mark = mark;
}
var student1 = new Student('sandeep' ,123);
console.log(student1.constructor);
```

Checkout the following screen shot for above code in chrome console. The console log is printing the referenced function by student1 object.



Elements	Resources	Network	Sources	Timeline	Profiles
<pre> > function Student(name,roll){ this.name = name; this.roll = roll; } undefined > var student1 = new Student('sangeeta',30) undefined > student1 Student {name: "sangeeta", roll: 30} > Student.prototype.mark = 100 100 > student1 Student {name: "sangeeta", roll: 30, mark: 100} > </pre>					

Q.What Is Scope In JavaScript?

The scope determines the accessibility of variables, objects, and functions in particular part of your code.

In JavaScript, the scope is of two types.

1. Global Scope
2. Local Scope

Q. How to call other class methods?

Using call () and apply () method we can use methods from different context to the current context. It is really helpful in reusability of code and context binding.

call (): It is used to calls a function with a given this value and arguments provided individually.

apply (): It is used to call a function with a given this value and arguments provided as an array.

Below code has two function getTypeOfNumber () and getTypeOfAllNumber. The details pf these functions are below.

getTypeOfNumber: This method takes single number as parameter and return the type either Even or Odd.

getTypeOfAllNumber: This method takes array of numbers as parameter and returns the types in an array with Even or Odd.

```
var MyNumber = {
```

getTypeOfNumber:

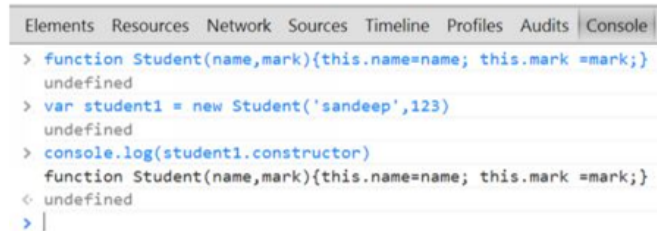
```
function(number){
  var type = (number % 2 === 0)? "Even": "Odd";
  return type;
}
```

getTpeOfAll Number:

```
function O{
  var result = [],i=0;
  for (; i < arguments. length; i++){
    var type = MyNumber.GetTypeOfNumber. call(null,arguments[i]);
    result.push(type);
  }
  return result;
}
var typeOfNumber = MvNumber.getTypeOfNumber. call ( null, 21.)
console.log(typeOfNumber)
var tpeOfAfInumber = MvNumber.getTypeOfAllNumber. apply(null [2,4,5,78,21])
console.log(typeOfAllNumber)
```

Below screenshot shows output of the above code Firebug console.

Drop us a Query



```

> function Student(name,mark){this.name=name; this.mark =mark;}
undefined
> var student1 = new Student('sandeep',123)
undefined
> console.log(student1.constructor)
function Student(name,mark){this.name=name; this.mark =mark;}
< undefined
> |

```

Q. How does one read a cookie in JavaScript?

JavaScript is easily used to read a cookie. The cookie is basically just the value of another object which is the `document.cookie`. This string can be used properly to access various cookies. The same string keeps accounts of the name list and writes different cookies in different ways like by using semicolons. Generally, the value of the cookie is just the string value.

Q. In what ways are Cookies deleted by using JavaScript?

Deleting a cookie refers to not being able to read a cookie. This happens if the expiration date of the cookie is set to a specific time in the past. In this way, the deleted cookie is not being able to be viewed by any user.

Q. How many types of objects are there in JavaScript? Define them?

There are two types of objects in JavaScript -

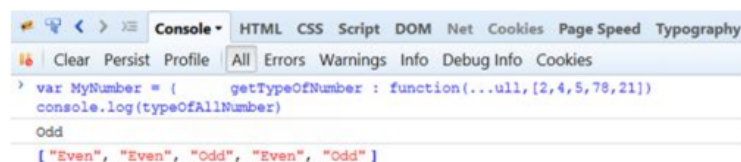
1. Date object - This type of object is built within the JavaScript programming language. The date objects are created with the help of `new Date`. It can be operated with the help of a bunch of methods once it is created. The methods allow the inclusion of the year, the month, day, hour and even minutes, seconds and millisecond of the date object. These are set with the help of local standard time of universal time.

2. Number object - This type of number object also includes the dates as it solely numerical dates are represented by it like integers and fractions. The literals of the numbers get converted to the number class automatically.

Q. Explain method overriding with an Example?

We can override any inbuilt method of JavaScript by declaring its definition again. The existing definition is accessible or override by the Prototype property. Consider the below example, `Split ()` class. But we have overridden its definition using its prototype property.

Below screen shows the inbuilt behavior of `split ()` method. It has divided the string into an array of element.

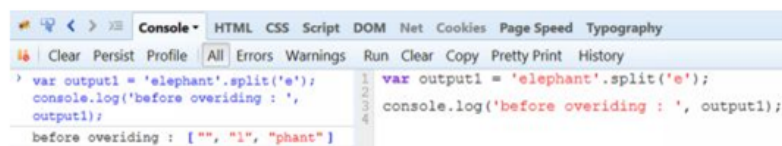


```

> var MyNumber = {   getNumberOf : function (...ull,[2,4,5,78,21])
  console.log(typeofAllNumber)
  Odd
  ["Even", "Even", "Odd", "Even", "Odd"]

```

Below screenshot shows the new overridden definition of `split ()` method. It is normally returns string **"I am overridden"**.



```

> var output1 = 'elephant'.split('e');
  console.log('before overriding : ',
  output1);
  before overriding : [ "", "l", "phant" ]

```

Q. What is the meaning of continuing and break statements?

Continue statements are the ones which continue from the next loop with the next set of statements. On the other hand, break statements are the ones that start from the current existing loop.

Q. Mention the process by which an Object can be created in JavaScript?

☎ Call us on

✉ Drop us a Query

The concept of Object is very well supported by the JavaScript. An object literal is used to create an object in the JavaScript.

Q. Define a name function. What is its meaning?

A named function is a type of function which has a specific name when defined properly in the JavaScript.

Q. What is the different type of features of JavaScript?

JavaScript has five types of features -

It is an open platform.

It is a lightweight programming language which is easily interpreted.

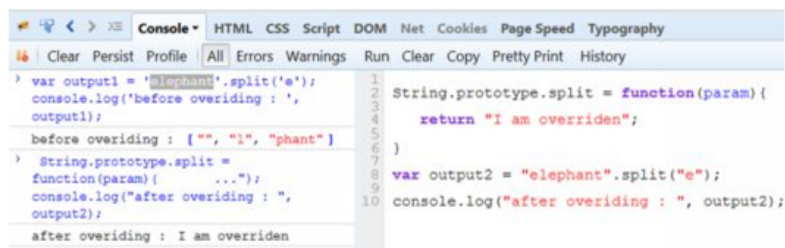
It is a programming language which is well integrated and complements well with Java.

It is designed in such a way which is supported for creating every type of network-centric application.

JavaScript is also integrated well with HTML and also compliments with it.

Q. How to inherit form a class?

Inheritance can be achieved in JavaScript using Prototype property.



We need to follow 2 steps to create an inheritance.

Step1:

Child class prototype should point to parent class object.

.prototype = new ();

Step2:

Reset the child class prototype constructor to point self. .prototype .

constructor = student;

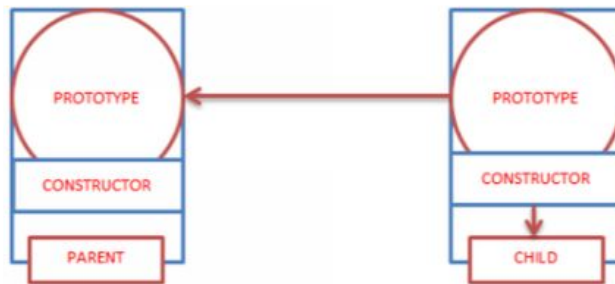
Example:

Below example shows **ScienceStudent** class as a child class of Student class. As the method showMyType () is available for Science Student object.

```
Function Student (name){
  this.name=name;
}
Student.prototype.sayMyType = function(){
  console.log ("I am student type")
}
function Science Student (name){
}
Science Student.prototype = new Student();
ScienceStudent.prototype.constructor = Science Student;
var student2 = new Science Student(sangeeta);
console.log(student2 .sayMyType ());
```

Check out the below screen shot for the output of the above code in the developer console.





To test the type of an object belongs to a class or not **instanceOf** can be used. This returns a Boolean value, **TRUE** for an object belongs to a class or else **FALSE**. Check the below screen shot for the test of student2 object using **instance Of**.

```

Elements Resources Network Sources Timeline Profiles Audits
> function Student(name){
    this.name=name;
}
Student.prototype.sayMyType = function(){
    console.log("I am student type")
}
function ScienceStudent(name){
}
ScienceStudent.prototype = new Student();
ScienceStudent.prototype.constructor = ScienceStudent;
var student2 = new ScienceStudent('sangeeta');
console.log(student2.sayMyType());
I am student type

```

Q. What is Object.create () method do?

ECMAScript 5.1 has this method in its specification. This method can be used to create a new object with given prototype object and properties. The syntax of this method is listed below.

Object. Create (proto [, properties Object])

Example:

Below code has a Student class functioned with name property and the prototype object has a method getStudentName () which return the name of the student. A new student1 object has been creates using Object.Create() method by passing the prototype object of Student with value of the name as sandeep. Then the getStudentName() method is called on student1 object and logged in the console.

```

function Student(name) {
    this.name = name;
}
Student.prototype = {
    getStudentName: function() {
        return "Name of Student is : " + this.name;
    }
};
var student 1 = Object .create(Student .prototype);
student1.name = "Sandeep";
console.log(student1.getStudentName () );

```

The following screenshot shows the output of the above code in the developer console.

```

Elements Resources Network Sources Timelin
> student2 instanceof Student
true
> student2 instanceof ScienceStudent
true
> |

```

Elements	Resources	Network	Sources	Timeline
<pre> > student2 instanceof Student true > student2 instanceof ScienceStudent true > </pre>				

Q. Write a polyfill for Object.create() method if it is not present in the browser?

ECMAScript 5.1 has this method in its specification. If the browser is old then Object.create () method is not present. To resolve this we need to write a polyfill. Below code shows the polyfill for Object. Create () method.

```

//check if create method is present inside Object
if (typeof Object.create != 'function') {
  //define the create method
  Object.create = (function() {
    var Object = function() {};
    return function (prototype) {
      if (arguments.length > 1) {
        throw Error ('Second argument not supported');
      }
      if (arguments.length > 1){
        throw Error ('Second argument not supported');
      }
      if (typeof prototype != 'object') {
        throw TypeError('Argument must be an object');
      }
      Object.prototype = prototype;
      var result = new Object();
      Object.prototype = null;
      return result;
    }
  })();
}

```

The above code checks if the create () method is already present inside the Object using if condition and comparing its type to function. If this condition true it means the create () method is not present. Then the polyfill code block gets executed and assigns the empty object to Object. Create property.

Q. How are JavaScript and ECMA Script related?

ECMA Script are like rules and guideline while Javascript is a scripting language used for web development.

Q. What is the purpose of Object.defineProperty () method?

ECMAScript 5.1 provides Object.defineProperty () method to create new properties to a defined object. It provides many configuration options to initialize these members. Below code shows the use of this method.

```

function Student (name) {
  this.name = name;
}
var student 1 = Object .create(Student tvp,
properties ={
  "subject": {
    value: "Computer",
    writable: true,
    enumerable :true
  },
  "marks": {
    value: 0,
    writable: false,
    enumerable: true
  }
});

```

Call us on

Drop us a Query

```
Object. Define Properties (student1, properties);
student1.name = "Sandeep";
student1.subject = "Mathematics";
student1.marks=75;
console.log (student1);
```

In the above code a student1 object created using Object.create() method. Then some new properties like subject and marks are added to this object. The enumerable option decided whether the property can be enumerated as own property. Writable property decides whether the property is modifiable or not. The value property takes the default value of the property.

Below screenshot shows the output of the above code in the developer console.

Explore JavaScript Sample Resumes! Download & Edit, Get Noticed by Top Employers!

Download Now! (<https://mindmajix.com/javascript-sample-resumes>)

Subscribe For Free Demo

Phone *

E-mail Address *

SUBSCRIBE

Free Demo for Corporate & Online Trainings.



About The Author

Ravindra Savaram is a Content Lead at Mindmajix.com (<https://mindmajix.com>). His passion lies in writing articles on the most popular IT platforms including Machine learning, DevOps, Data Science, Artificial Intelligence, RPA, Deep Learning, and so on. You can stay up to date on all these technologies by following him on LinkedIn (<https://www.linkedin.com/in/savaram-ravindra-48064641/>) and Twitter (<https://twitter.com/s11ravindra>).



(<https://www.dmca.com/Protection/Status.aspx?ID=5b159fca-d195-4289-83e3-b69a0818b38d&refurl=https://mindmajix.com/javascript-interview-questions>)

Social Share



(<https://www.facebook.com/mindmajix/>) (<https://twitter.com/mindmajix>) (<https://plus.google.com/+mindmajix>) (<https://www.linkedin.com/company/mindmajix>)
<https://www.facebook.com/mindmajix/> (<https://www.facebook.com/mindmajix/>) (<https://twitter.com/mindmajix>) (<https://plus.google.com/+mindmajix>) (<https://www.linkedin.com/company/mindmajix>)
<https://www.facebook.com/mindmajix/> (<https://www.facebook.com/mindmajix/>) (<https://twitter.com/mindmajix>) (<https://plus.google.com/+mindmajix>) (<https://www.linkedin.com/company/mindmajix>)

PREVIOUS (<https://mindmajix.com/bitcoin-that-works-on-blockchain-interview-questions&summary=&source=>)

NEXT (<https://mindmajix.com/big-data-greenplum-dba-interview-questions&summary=&source=>)

Categories

- ▶ CompTIA (<https://mindmajix.com/comptia/>)
- ▶ SAS (<https://mindmajix.com/sas/>)

Call us on

Drop us a Query

- ▶ Peoplesoft CRM (<https://mindmajix.com/peoplesoft-crm/>)
- ▶ Business Analyst (<https://mindmajix.com/business-analyst/>)
- ▶ Tibco BW (<https://mindmajix.com/tibco-bw/>)
- ▶ SAP Hybris (<https://mindmajix.com/sap-hybris/>)
- ▶ AngularJS (<https://mindmajix.com/angularjs/>)
- ▶ Ruby on Rails (<https://mindmajix.com/ruby-on-rails/>)
- ▶ TIBCO SPOTFIRE (<https://mindmajix.com/tibco-spotfire/>)
- ▶ PeopleSoft HRMS (<https://mindmajix.com/peoplesoft-hrms/>)

[VIEW MORE \(HTTPS://MINDMAJIX.COM/BLOG-CATEGORY\)](https://mindmajix.com/blog-category)

Popular Courses in 2019

Azure Training (<https://mindmajix.com/microsoft-azure-training>)

★★★★★

© 2178 Learners

OpenStack Training (<https://mindmajix.com/openstack-training>)

★★★★★

© 1112 Learners

JENKINS Training (<https://mindmajix.com/jenkins-training>)

★★★★☆

© 921 Learners



Majix

Mindmajix - Online global training platform connecting individuals with the best trainers around the globe. With the diverse range of courses, Training Materials, Resume formats and On Job Support, we have it all covered to get into IT Career. Instructor Led Training - Made easy.

[f](https://www.facebook.com/MindMajixTechnologies) (<https://www.facebook.com/MindMajixTechnologies>) [t](https://twitter.com/mindmajix) (<https://twitter.com/mindmajix>) [g+](https://plus.google.com/+MindmajixTechnologies) (<https://plus.google.com/+MindmajixTechnologies>)

[in](https://www.linkedin.com/company/mindmajix-technologies-pvt-ltd-) (<https://www.linkedin.com/company/mindmajix-technologies-pvt-ltd->) [yt](https://www.youtube.com/channel/UcKemMaRnFPINLHZ0zOYfpA) (<https://www.youtube.com/channel/UcKemMaRnFPINLHZ0zOYfpA>)

Company

[Home \(https://mindmajix.com/\)](https://mindmajix.com/)

[About Us \(https://mindmajix.com/about\)](https://mindmajix.com/about)

[Courses \(https://mindmajix.com/all-courses\)](https://mindmajix.com/all-courses)

[Sample Resumes \(https://mindmajix.com/sample-resumes\)](https://mindmajix.com/sample-resumes)

[Call us on Blog \(https://mindmajix.com/blog\)](https://mindmajix.com/blog)

[✉ Drop us a Query](#)

Contact Us (<https://mindmajix.com/contact>)

Reviews (<https://mindmajix.com/reviews>)

Write for us (<https://mindmajix.com/write-for-us>)

Trending Courses

Tableau Training (<https://mindmajix.com/tableau-training>)

Oracle DBA training (<https://mindmajix.com/oracle-dba-training>)

Qlikview Training (<https://mindmajix.com/qlikview-training>)

Docker Training (<https://mindmajix.com/docker-training>)

JBoss Training (<https://mindmajix.com/jboss-training>)

Informatica Training (<https://mindmajix.com/informatica-training>)

Cassandra Training (<https://mindmajix.com/apache-cassandra-training>)

Blockchain Training (<https://mindmajix.com/blockchain-training>)

Contact info

📍 244 Fifth Avenue, Suite 1222 New York(NY) United States (US) - 10001

📞 USA : +1 917 456 8403

✉ info@mindmajix.com (<mailto:info@mindmajix.com>)

📍 #811, 10th A Main, Suite No.506 1st Floor, Indira Nagar Bangalore, India - 560038

📞 India : +91 905 240 3388

✉ info@mindmajix.com (<mailto:info@mindmajix.com>)

Copyright © 2019 Mindmajix Technologies Inc. All Rights Reserved

[Privacy Policy \(https://mindmajix.com/terms-of-service-and-privacy-policy\)](https://mindmajix.com/terms-of-service-and-privacy-policy)

[Terms of use \(https://mindmajix.com/terms-of-service-and-privacy-policy\)](https://mindmajix.com/terms-of-service-and-privacy-policy)

[Refund Policy \(https://mindmajix.com/terms-of-service-and-privacy-policy\)](https://mindmajix.com/terms-of-service-and-privacy-policy)

📞 Call us on

✉ Drop us a Query



📞 Call us on

✉ Drop us a Query