How to create objects in JavaScript?

There are 3 ways to create object in JavaScript.

1. By object literal

emp={id:102,name:"Shyam Kumar",salary:40000}

1. By creating instance of Object

var emp=new Object();

emp.id=101;

 emp.name="Ravi Malik";

emp.salary=50000;

1. By Object Constructor

function emp(id,name,salary){

**this**.id=id;

this.name=name;

this.salary=salary;

}

### How to create array in JavaScript?

There are 3 ways to create array in JavaScript.

1. By array literal

var emp=["Sonoo","Vimal","Ratan"];

1. By creating instance of Array

var emp = new Array();

emp[0] = "Arun";

emp[1] = "Varun";

emp[2] = "John";

1. By using an Array constructor

var emp=new Array("Jai","Vijay","Smith");

### Difference between Client side JavaScript and Server side JavaScript?

**Client side JavaScript** comprises the basic language and predefined objects which are relevant to running java script in a browser. The client side JavaScript is embedded directly by in the HTML pages. This script is interpreted by the browser at run time.

**Server side JavaScript** also resembles like client side java script. It has relevant java script which is to run in a server. The server side JavaScript are deployed only after compilation.

### How to handle exceptions in JavaScript?

By the help of try/catch block, we can handle exceptions in JavaScript. JavaScript supports try, catch, finally and throw keywords for exception handling

**What are JavaScript types?**

Following are the JavaScript types:

* Number
* String
* Boolean
* Function
* Object
* Null
* Undefined

**Which company developed JavaScript?**

* Netscape is the software company who developed JavaScript

**What is the ‘Strict’ mode in JavaScript and how can it be enabled?**

* Strict Mode adds certain compulsions to JavaScript. Under the strict mode, JavaScript shows errors for a piece of codes, which did not show an error before, but might be problematic and potentially unsafe. Strict mode also solves some mistakes that hamper the JavaScript engines to work efficiently.
* Strict mode can be enabled by adding the string literal “use strict” above the file. This can be illustrated by the given

**What is the difference between .call() and .apply()?**

The function .call() and .apply() are very similar in their usage except a little difference. .call() is used when the number of the function’s arguments are known to the programmer, as they have to be mentioned as arguments in the call statement. On the other hand, .apply() is used when the number is not known. The function .apply() expects the argument to be an array.

The basic difference between .call() and .apply() is in the way arguments are passed to the function. Their usage can be illustrated by the given example.



|  |  |
| --- | --- |
|  | var someObject = {    myProperty : 'Foo',    myMethod : function(prefix, postfix) {    alert(prefix + this.myProperty + postfix);    }    };    someObject.myMethod('<', '>'); // alerts '<Foo>'    var someOtherObject  = {    myProperty : 'Bar'    };    someObject.myMethod.call(someOtherObject, '<', '>'); // alerts '<Bar>'    someObject.myMethod.apply(someOtherObject, ['<', '>']); // alerts '<Bar>' |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | **Write about the errors shown in JavaScript?**  JavaScript gives a message if it encounters an error. The recognized errors are –   * Load-time errors: The errors shown at the time of the page loading are counted under Load-time errors. These errors are encountered by the use of improper syntax, and thus are detected while the page is getting loaded. * Run-time errors: This is the error that comes up while the program is running. It is caused by illegal operations, for example, division of a number by zero, or trying to access a non-existent area of the memory. * Logic errors: It is caused by the use of syntactically correct code, which does not fulfill the required task. For example, an infinite loop.  **What is the use of DOM?** DOM is also known as Document Object Model which is used to develop a model with documents or web pages containing objects like elements, links, etc. These objects can be manipulated or certain actions like add, delete or change of an element can be performed using this document object model. Through this change in attributes can be done to get all the list of all the elements in the document. The DOM model responds to API calls that result in documented level of DOM recommendation. It is used to support additional behavior on the web page and use of API give an extensible advantage over other models existing. DOM codes are reused to meet the requirement of the real world and to make all the program interoperable.  var z = function foo(){};  typeof z= function  typeof foo= **undefined**;  function foo(){};  typeof foo= **function**;  What is closure?  Closures are created whenever a variable that is defined outside the current scope is accessed from within some inner scope.  Give an example of closure?  Following example shows how the variable counter is visible within the create, increment, and print functions, but not outside of them −  function create() {  var counter = 0;  return {  increment: function() {  counter++;  },    print: function() {  console.log(counter);  }  }  }  var c = create();  c.increment();  c.print(); // ==> 1 |
|  |  |
|  |  |
|  |  |
|  |  |