|  |
| --- |
| 1. What are the possible ways to create objects in JavaScript?](#what-are-the-possible-ways-to-create-objects-in-javascript. |
| There are many ways to create objects in javascript as below,  1. \*Object constructor:\*  The simplest way to create an empty object is using Object constructor. Currently this  approach is  not recommended.  javascript  var object = new Object();  2. \*Object's create method:\*  The create method of Object creates a new object by passing the prototype object as a  parameter  javascript  var object = Object.create(null);  3. \*Object literal syntax:\*  The object literal syntax is equivalent to create method when it passes null as parameter  javascript  var object = {};  4. \*Function constructor:\*  Create any function and apply the new operator to create object instances,  javascript  function Person(name){  var object = {};  object.name=name;  object.age=21;  return object;  }  var object = new Person("Sudheer");  5. \*Function constructor with prototype:\*  This is similar to function constructor but it uses prototype for their properties and methods,  javascript  function Person(){}  Person.prototype.name = "Sudheer";  var object = new Person();  This is equivalent to an instance created with an object create method with a function prototype and then call that function with an instance and parameters as arguments.  javascript  function func {};  new func(x, y, z);  \*\*(OR)\*\*  // Create a new instance using function prototype.  var newInstance = Object.create(func.prototype)  // Call the function  var result = func.call(newInstance, x, y, z),  // If the result is a non-null object then use it otherwise just use the new instance.  console.log(result && typeof result === 'object' ? result : newInstance);  6. \*ES6 Class syntax:\*  ES6 introduces class feature to create the objects  javascript  class Person {  constructor(name) {  this.name = name;  }  }  var object = new Person("Sudheer");  7. \*Singleton pattern:\*  A Singleton is an object which can only be instantiated one time. Repeated calls to its constructor return the same instance and this way one can ensure that they don't accidentally create multiple instances.  javascript  var object = new function(){  this.name = "Sudheer";  } |
|  |
|  |
|  |
|  |
|  |