

# JavaScript Objects

JavaScript is an Object Oriented Programming (OOP) language. A programming language can be called object-oriented if it provides four basic capabilities to developers.

- **Encapsulation** – the capability to store related information, whether data or methods, together in an object.
- **Aggregation** – the capability to store one object inside another object.
- **Inheritance** – the capability of a class to rely upon another class (or number of classes) for some of its properties and methods.
- **Polymorphism** – the capability to write one function or method that works in a variety of different ways.

Objects are composed of attributes. If an attribute contains a function, it is considered to be a method of the object, otherwise the attribute is considered a property.

A JavaScript object is an entity having state and behavior (properties and method). For example: car, pen, bike, chair, glass, keyboard, monitor etc.

JavaScript is an object-based language. Everything is an object in JavaScript.

JavaScript is template based not class based. Here, we don't create class to get the object. But, we directly create objects.

# Creating a JavaScript Object

With JavaScript, you can define and create your own objects.

There are different ways to create new objects:

- Define and create a single object, using an object literal.
- Define and create a single object, with the keyword new.
- Define an object constructor, and then create objects of the constructed type.

## JavaScript Object by object literal

The syntax of creating object using object literal is given below:

object={property1:value1,property2:value2.....propertyN:valueN}

As you can see, property and value is separated by : (colon).

### Example

Example	Output
<pre>&lt;script&gt; emp={ id:001,name:"ABC",salary:20000} document.write(emp.id+" "+emp.name+" "+ emp.salary); &lt;/script&gt;</pre>	001 ABC 20000

# JavaScript Object Constructors

Here, you need to create function with arguments. Each argument value can be assigned in the current object by using this keyword.

this keyword refers to the current object.

## Example

Example	Output
<pre>&lt;script&gt; function emp(id,name,salary){ this.id=id; this.name=name; this.salary=salary; } e=new emp(001,"ABC",20000); document.write(e.id+" "+e.name+" "+e.salary); &lt;/script&gt;</pre>	001 ABC 20000