**Object representation in Java script**

“A JavaScript object is a collection of named values having state and behaviour (properties and method)”.

For example: Person, car, pen, bike, Personal Computer, Washing Machine etc.

In JavaScript, almost "everything" is an object.

* Booleans can be objects (if defined with the new keyword)
* Numbers can be objects (if defined with the new keyword)
* Strings can be objects (if defined with the new keyword)
* Dates are always objects
* Maths are always objects
* Regular expressions are always objects
* Arrays are always objects
* Functions are always objects
* Objects are always objects

All JavaScript values, except primitives, are objects.

Take the case of cars.

All cars have the same properties, but the property values differ from car to car. All cars have the same methods, but the methods are performed at different times.

Let’s have an example of merc car and list out its properties (Features):

1. Make: Mercedes
2. Model: C-Class
3. Color: White
4. Fuel: Diesel
5. Weight: 850kg
6. Mileage: 8Kmpl
7. Rating: 4.5

**Objects are variables**

The following code assigns a **simple value** (Mercedes) to a **variable** named car:

var car = "Mercedes";

Objects are variables too. But objects can contain many values.

The following code assigns **many values** (Mercedes, C-class, White and soo on) to a **variable** named Car:

var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel, Weight: “850kg”, Mileage: “8Kmpl”, Rating: 4.5};

The values are written as **name : value** pairs (name and value separated by a colon).

Syntax:

var <object-name> = {key1: value1, key2: value2,... keyN: valueN};

**Object Properties**

The name : values pairs (in JavaScript objects) are called **properties**.

var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel, Weight: “850kg”,Mileage: “8Kmpl”, Rating: 4.5};

From the above snippet, let’s have a look what falls under property and property value:



The object properties can be different primitive values, other objects and functions.

.So, simple definition for Java Script properties is “Properties are the values associated with javaScript object”.

**Object methods**

Methods are **actions** that can be performed on objects.

Object properties can be primitive values, other objects, and functions.

An **object method** is an object property containing a **function definition**.

|  |  |
| --- | --- |
| **Property** | **Value** |
| firstName | John |
| lastName | Doe |
| age | 50 |
| eyeColor | blue |
| fullName | function() {return this.firstName + " " + this.lastName;} |

JavaScript objects are containers for named values, called properties and methods.

**Create Java script object**

There are different ways to create new objects:

* Create a single object, using an object literal.
* Create a single object, with the keyword new.
* Define an object constructor, and then create objects of the constructed type.
* Create an object using Object.create().

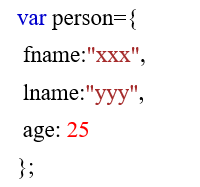
## By object literal:

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

https://miro.medium.com/max/655/1*BPfZK2NMr7-nsIXYyXNFew.png

Property and value is separated by colon(:).

**Example:**



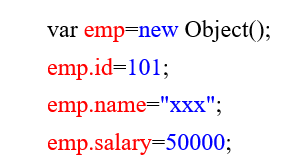
## By creating instance of Object directly (using new keyword):

The syntax of creating object directly is given below:

https://miro.medium.com/max/330/1*Cl7DNbx5EPkMmmxMgekUNQ.png

Here, **new keyword** is used to create object.

**Example:**



Properties can usually be changed, added, and deleted, but some are read only

**The syntax for adding a property to an object is :**

ObjectName.ObjectProperty = propertyValue;

**The syntax for deleting a property from an object is:**

delete ObjectName.ObjectProperty;

**The syntax to access a property from an object is:**

objectName.property        // Car.Make

//or

objectName["property”]    // Car["Make"]

//or

objectName[expression]   // x = "Make"; Car[x]