Objects and its internal representation in javascript

INTRIDUCTION: Objects are important data types in javascript. Objects are different than primitive datatypes (i.e. number, string, boolean, etc.). Primitive data types contain one value but Objects can hold many values in form of Key: value pair. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

Every object has some property associated with some value. These values can be accessed using these properties associated with them.

var myCar = new Object();

myCar.make = 'Suzuki';

myCar.model = 'Altros';

myCar.year = 1978;

myCar.wheels = 2;

After creating myCar object, the value inside the object can be accessed using keys.

i.e.myCar.year

Output: 1978

These values can be accessed using brackets notation also.

myCar[year]

Output: 1978

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
                //or
objectName["property”]
                //or
objectName[expression]
```

So, conclusion and simple definition for Java Script properties is “Properties are the values associated with a JavaScript object”.

Object methods

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

i.e.,Let’s assume to start the car there will be a mechanical functionality.

```
function(){return ignition.on}
```

And so similar is to stop/brake/headlights on & off, etc.So, conclusion and simple definition for Java Script Object methods is “Methods are actions that can be performed on objects.”

Create JavaScript Object with Object Literal

One of easiest way to create a javascript object is object literal, simply define the property and values inside curly braces as shown below.

```
let bike = {name: 'SuperSport', maker:'Ducati', engine:'937cc'};
```

Create JavaScript Object with Constructor:

Constructor is nothing but a function and with help of new keyword, constructor function

allows to create multiple objects of same flavor as shown below.

```
function Vehicle(name, maker) {
    this.name = name;
    this.maker = maker;
}
let car1 = new Vehicle('Fiesta', 'Ford');
let car2 = new Vehicle('Santa Fe', 'Hyundai')
console.log(car1.name);       //Output: Fiesta
console.log(car2.name);       //Output: Santa Fe
```

Using the JavaScript Keyword new:

The following example also creates a new JavaScript object with four properties:

```
var person = new Object();
person.firstName = "John";
person.lastName = "Doe";
person.age = 50;
person.eyeColor = "blue";
```

Using the Object.create method:

Objects can also be created using the Object.create() method. This method can be very useful, because it allows you to choose the prototype object for the object you want to create, without having to define a constructor function.

animal1.displayType(); // Output:Invertebrates

// Create new animal type called Fishe// Animal properties and method encapsulation.

```
// Animal properties and method encapsulation
var Animal = {
    type: 'Invertebrates', // Default value of properties
    displayType: function() {    // Method which will display type of Animal
        console.log(this.type);
    }
};
// Create new animal type called animal1
var animal1 = Object.create(Animal);
animal1.displayType(); // Output:Invertebrates
// Create new animal type called Fishes
var fish = Object.create(Animal);
fish.type = 'Fishes';
fish.displayType();
// Output:Fishes
```