**Objects and its Internal representation in javascript**

* Loosely speaking, objects in JavaScript may be defined as an unordered collection of related data, of primitive or reference types.
* Object are the most important data types in javascript. Object are different than primitive data types i.e. Boolean, number, string, etc.
* The primitive data type contains one value but object can hold many values in form of “key: value” pair.
* These keys can be variables or functions and are called properties and methods in the context of an object.
* Every object has some property associated with some values. These values can be accessed using these properties associated with them.
* Properties are the values associated with the javascript object.
* The syntax for adding a property to 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]**

var mycar = newobject();

mypc.make = ‘DELL’;

mypc.model = ‘inspiron 15’;

mypc.year = 2022;

mypc.price = 35000;

After creating mypc object, the values inside the object can be accessed by calling the keys which we declared.

i.e. mypc.year

**Output:** 2022

These values can be accessed using brackets notations also;

i.e. mypc[year]

**Output:** 2022

**Object Methods:**

* An object method is an object property containing a function definition.
* Let us assume to start the pc there will be a mechanical functionality.

**Function() {**

**return pc.on**

**}**

* And so similarly is to access pc like applications open close and then using the web browser, games, video player, music player, etc.
* “Methods are the actions that can be performed an objects”
* One of the easiest way to create a javascript object is **object** **literal**, simply define the property and the values in the curly braces as :

**let pc = {name: ‘macbook’, market: ‘apple’, processor: ‘core i7’};**

* **Constructor** is nothing but a function and with help of keyword, constructor function allows to create multiple objects of same flavour

**Function pc(name, market)**

**{**

**this.name = name;**

**this.market = market;**

**}**

**let pc1 = New pc(‘HP’, ‘Dell’);**

**let pc2 = New pc(‘Sony’, Acer’);**

**console.log(pc1.name);**

**console.log(pc2.name);**

**OUTPUT:**

**HP**

**Sony**

* Example for create a new object with four properties:

**var person = new Object();**

**person.firstName = “vishnu”;**

**person.lastName = “Gowtham”;**

**person.age = 25;**

**person.hairColor = “black”;**

* 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

// 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