Objects and its internal representation in Javascript

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 .

In JavaScript objects are defined as an unordered collection of related data of primitive or reference types in the form of “key: value” pairs.

These keys can be variables or functions and are called properties or methods in the context of an object.

 Objects are called *associative arrays*, since each property is associated with a string value that can be used to access it.

An object is a standalone entity, with properties and type.

An object can be created with figure brackets {} with an optional list of properties. A property is a “key: value” pair, where a key is the property name value can be anything.

For eg:

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.company = 'Fordi';

myCar.model = 'Aspire';

myCar.year = 2018

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

i.e.

myCar.year

Output: 2018.

**constructor**

The constructor method is a special method for creating and initializing an object instance of that class.

This: It is a temporary keyword pointing to the freshly created object. This is a part of constructor.

For eg:

Class Car{

this.brand = brand;

this.model = Model;

}

Let S1=new car("Ford","abc");

Function Car(brand, model){

this.brand = brand;

this.model= Model;

}

Console.log(S1.brand)