**Objects and its internal Representation in js**

Objects, in JavaScript, is it’s most important data-type and forms the building blocks for modern JavaScript. These objects are quite different from JavaScript’s primitive data-types(Number, String, Boolean, null, undefined and symbol) in the sense that while these primitive data-types all store a single value each (depending on their types).

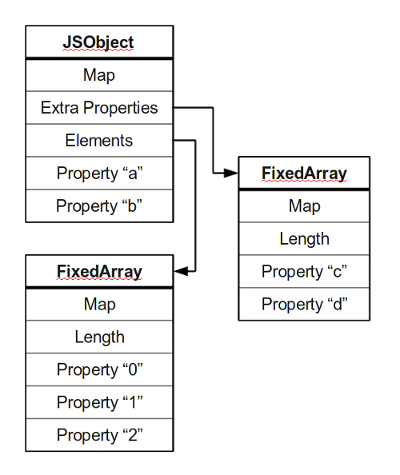
Conceptually, Objects are same in all programming languages i.e they represent real-world things that we want to represent inside our programs with characteristics/properties and methods.

Objects in JavaScript may be 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 and methods, respectively, in the context of an object.  
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 a string (also called a “property name”), and value can be anything.

**For eg:-** let school = {  
name : “NVM School”,  
location : “Delhi”,  
established : “1971”  
}

In the above example “name”, “location”, “established” are all “keys” and “NVM School”, “Delhi” and 1971 are values of these keys respectively.

Each of these keys is referred to as properties of the object. An object in JavaScript may also have a function as a member, in which case it will be known as a method of that object.



The javascript standards allows developers to define objects in a very flexible way, and it is hard to come up with an efficient representation that works for everything. An object is essentially a collection of properties: basically key-value pairs. We can access properties using two different kinds of expressions:

* Obj.prop
* Obj[“prop”]