What is object

An object in JavaScript is a non-primitive data type that is used to store multiple values of different primitive data types in the form of key-value pairs.

These objects are quite different from JavaScript’s primitive data types (Number, String, Boolean, null, undefined, and symbol) in the sense that these primitive data types all store a single value each (depending on their types).

 An object is a collection of properties, and a property is an association between a name (or key) and a value. A property’s value can be a function, in which case the property is known as a method.

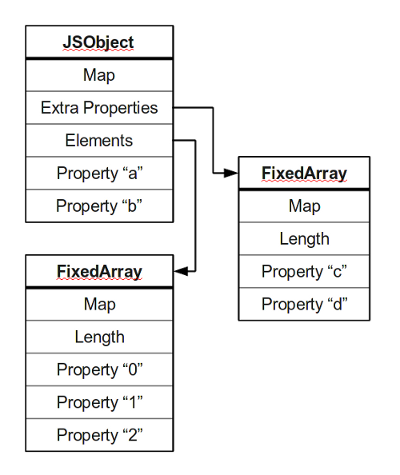
For Example;

|  |  |
| --- | --- |
| INPUT CONTENT | |
| *VarmyCar=  make:  model:   year:*  *};* | {  ‘Ford’,  ‘Mushtang’,  ‘1969’, |

This What we call the object in java script

INTERNAL REPRESENTATION

A simple diagram is probably the best way to give a quick overview of the object representation in Javascript.



Most objects contain all their properties in a single block of memory *(‘a’ and ‘b’)*. All blocks of memory have a pointer to a map, which describes their structure.

Named properties that don’t fit in an object are usually stored in an overflow array *(‘c’ and ‘d’)*.

Numbered properties are stored separately, usually in a contiguous array.

The JavaScript standard 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”]

According to the spec, property names are always strings. If we use a name that is not a string, it is implicitly converted to a string. This may be a little