**Objects & Internal representation**

In JavaScript, Objects is most important data-type and forms the building blocks for modern JavaScript. These objects are quite different from JavaScript’s primitive data-types.

Objects are more complex and each object may contain any combination of these primitive data-types as well as reference data-types.

An object, is a reference data type. Variables that are assigned a reference value are given a reference or a pointer to that value.

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

A property is a “key: value” pair, where a key is a string (also called a “property name”), and value can be anything.

For example

let college = {  
name : “Sastra”,  
location : “Tanjore”,  
established : “1971”  
}

In the above example “name”, “location”, “established” are all “keys” and “Sastra”, “Tanjore” and 1971 are values of these keys.

Bracket notation:

let college = {  
“name” : “Sastra”

}

console.log(college[“name”]);

Dot notation:

let college = {  
“name” : “Sastra”

}

console.log(college.name);

Creating Objects  
var obj = {  
member1 : value1,  
member2 : value2,  
};

Unlike the dot notation, the bracket keyword works with any string combination, including, but not limited to multi-word strings.

For example:  
somePerson.first name // invalid  
somePerson[“first name”] // valid.