

Objects and its Internal Representation in JavaScript:

Object is the most important data type in JavaScript. These objects are different from JavaScript primitive data types. All primitive data (String, integer, Boolean, null, undefined) types store a single value but reference data types (array, object) point the memory location where any combination of primitive data types as well as it also references data types stored in memory. Variables that are assigned a reference value are given a point to that value.

Objects in JavaScript may be defined as an unordered collection 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 of an object.

Properties and method:

A JavaScript object has properties associated with it. A property of an object is basically the same as ordinary JavaScript variables, except for the attachment to objects. The properties of an object define the characteristics of the object. We can access the properties of an object with the help of a dot operator. It is also case sensitive like a JavaScript variable. We can define a property by assigning a value. Like all JavaScript variables, object names are case sensitive. For example, let's create an object named student and give it properties named class, age and blood group.

<pre>Var student = { name: "arun" age:15 class:8 }</pre>	<pre>Var student = new Object (); Student.name = "arun" student.age = "15"; student.class = "8";</pre>
--	---

```
student.bloodGroup = "O+"
```

Unassigned properties of an object are undefined (and not null).

Properties of JavaScript objects can also be accessed or using a square bracket notation. Sometimes an object is called associative arrays, since each property is associated with a string value that can be used to access it. So, for example, you could access the properties of the student object as follows.

```
student["name"] = 'Arun';
```

```
student["age"] = 15;
```

```
student.class = 8;
```

we need to give any valid JavaScript string, or anything that can be converted to a string, including the empty string. However, any property name that is not a valid JavaScript identifier (For example, a property name that has a space or a hyphen, or that starts with a number) can only be accessed using the square bracket notation.

Above we have discussed about the Object and its internal representation in javascript.