**Objects and its Internal representation in JavaScript**

* Objects, in JavaScript is the most important data-type and forms the building blocks for modern JavaScript
* Objects are more complex and each object may contain any combination of these primitive data-types as well as reference data-types
* Objects are variables too. But objects can contain many values
* Object values are written as name : value pairs (name and value separated by a colon)
* A JavaScript object is a collection of named values
* It is a common practice to declare objects with the const keyword

**Object Properties**

* Properties are the values associated with a JavaScript object
* A JavaScript object is a collection of unordered properties
* The properties of an object define the characteristics of the object

Properties of an object can be accessed with a simple dot-notation:

**objectName.propertyName**

* The **name:values** pairs (in JavaScript objects) are called properties
* One of easiest way to create a JavaScript object is object literal, simply define the property and values inside curly braces

var vehicle = {

Type: “Car”,

Color: “White”,

Drive: “4-Wheeler”

};

* Properties can usually be changed, added, and deleted, but some are read only

The syntax for adding a property to an object is :

* ObjectName.ObjectProperty = propertyValue;

The syntax for deleting a property from an object is:

* delete ObjectName.ObjectProperty;

**Object Methods**

* In JavaScript, the this keyword refers to an object
* JavaScript methods are actions that can be performed on objects.
* An object method is an object property containing a function definition
* Methods are functions stored as object properties

You access an object method with the following syntax:

* objectName.methodName()

**Using the Object.create method**

* Objects can also be created using the Object.create() method
* This method can be very useful, because it allows you to choose the prototype object for the object you want to create, without having to define a constructor