**Objects and its internal representation in JavaScript:**

JavaScript is an object-oriented programming language, so everything in JavaScript is an object.

In JavaScript, an object is a standalone entity, with properties and type.

JavaScript is template based and we can create objects without the need of having a class. JavaScript doesn’t require any class to be defined for objects.

There are different ways to create new objects in JavaScript:

* Create a single object, using an object literal.
* Create a single object, with the keyword new.
* Define an object constructor, and then create objects of the constructed type.
* Create an object using Object.create().

**Using Object literal**: We can use the object literal to create and define a JavaScript object. In this method, an object can be created with **keys and associated values**.

Example:

Var student = {

name: “Kenny”,

age: 25,

branch: “Computer Science”,

course: “FSD Tamil”

};

**Using keyword new**: We can use the **new keyword** to create and define an object. We can easily add properties to JavaScript objects.

Example:

   var student = new Object();

student.name = “Kenny”;

      student.age = 25;

      student.branch = “Computer Science”;

student.course = “FSD Tamil”;

**Using object constructor**: We can also use an **object constructor** to initialize a JavaScript object. This method is also commonly known as object prototyping. The constructor takes in a few parameters and using those parameters, we define the value for each property in an object.

    Example:

function student(name, age, branch, course) {

        this.name = name;

        this.age = age;

        this.branch = branch;

this.course = course;

      }

      var student = student (“Kenny”, 25, “Computer Science”, “FSD Tamil”);

**Using Object.create()**: This method creates a new object using the **prototype** of the given object. The **create()** method, being a static method, is called using the Object class name.

* **Object.create()** returns a new object with the specified prototype object and properties.
* **prototype**: It is the prototype object from which a new object has to be created.
* **propertiesObject**: It is an optional parameter. It specifies the enumerable properties to be added to the newly created object.

**Syntax**: Object.create(prototype, propertiesObject)

**JavaScript Object Properties:** A JavaScript object is basically a collection of unordered properties. Values associated with a JavaScript object are called its properties. Properties can usually be added, updated, and deleted, excluding read-only properties.

**JavaScript Object Methods:** Actions that can be performed on a JavaScript object are called methods.

**JavaScript Object Accessors:** Getters and setters allow the defining of object accessors. We first create an object and add a few properties to that object.

* we can use a **getter ()** function to access the properties of an object, this helps in the implementation of abstraction in a JavaScript application.
* We can also set the value of a property of an object using **setter ()** function, this provides more control as to what values can be assigned to that particular property.