**=>Write a blog about objects and its internal representation in Javascript:**

* Objects are the most important data-type and form the building blocks for modern JavaScript. These objects are quite different from JavaScript’s primitive data-types(Number, String, Boolean, null, undefined and symbol).
* An object is a reference data type. Variables that are assigned a reference value are given a reference or a pointer to that value. That reference or pointer points to the location in memory where the object is stored. The variables don’t actually store the value.
* The internal representation of an object in JavaScript is complex and varies based on the engine that is executing the code. However, in general, an object is represented as a collection of properties, where each property is a key-value pair. Properties can be data properties, which contain a value, or accessor properties, which contain a getter and/or a setter method.
* In addition to properties, an object also has a prototype, which is a reference to another object that provides a set of default properties and methods that the object can inherit. The prototype chain is a mechanism by which an object can inherit properties and methods from its prototype and its prototype's prototype.
* The internal representation of an object also includes several hidden properties, such as [[Class]], [[Prototype]], and [[Extensible]], that are used by the JavaScript engine to manage the object and provide additional functionality.
* The internal representation of an object in JavaScript depends on how it is created. There are two main ways to create objects in JavaScript:

**Object literals:** This is the most common way to create objects in JavaScript. It involves defining an object using curly braces {}, and specifying its properties and their corresponding values using the syntax key: value

**Object constructor:** Another way to create objects in JavaScript is by using the Object constructor function.