**JavaScript Objects and its internal representations.**

**Objects:**

* In JavaScript, an object is like a container that holds related data and functions, known as properties and methods, respectively.
* They can store multiple values as a collection of key-value pairs.
* You can create objects in JavaScript using two main methods:

1. object literals.
2. constructors.

* Object Literals: This is the simplest way to create an object. You define the object’s properties and methods right when you create it.
* For ex:
* const car = {  
   make: 'Toyota',  
   model: 'Camry',  
   year: 2022,  
  };
* Constructors: Constructors are like templates for creating objects. You define a constructor function and then create new objects using the `new` keyword.
* function Car(make, model, year) {  
   this.make = make;  
   this.model = model;  
   this.year = year;  
   }  
  const myCar = new Car('Honda', 'Civic', 2023);
* You can access the properties of an object using dot notation (e.g., objectName.propertyName)
* or bracket notation (e.g., objectName["propertyName"]).

**Internal Representation:**

* Internally, JavaScript engines represent objects in a way that optimizes performance.
* Properties are stored in a structure that allows for efficient access and modification.
* JavaScript objects are dynamic, meaning you can add or delete properties at runtime.
* This flexibility allows for powerful patterns but also requires careful management of memory and performance.
* Some engines use concepts like hidden classes and inline caching to speed up property access.