**Objects in JavaScript:**

At its core, an object in JavaScript is a collection of key-value pairs, where each key is a string (or symbol) and each value can be of any data type, including other objects.

Objects in JavaScript provide a powerful way to structure and organize data, allowing developers to create complex data structures with ease.

**Internal Representation of Objects:**

Internally, JavaScript engines use various techniques to represent objects efficiently. One of the key concepts is the use of a data structure known as a hash table or a dictionary. This data structure enables fast access and retrieval of values based on their keys.

**Here's a simplified overview of how objects are internally represented**:

**Object Initialization:**

When an object is created, the JavaScript engine allocates memory for it and sets up a data structure to store its properties.

**Example:**

// Object Initialization

let person = {

name: "John",

age: 30,

profession: "Developer"

};

**Property Access:**

Accessing a property of an object involves a process known as property lookup. The engine uses the property name to compute a hash, allowing it to quickly locate the corresponding value in the internal representation.

**Example:**

// Property Access

console.log(person.name); // Output: John

**Dynamic Nature:**

JavaScript objects are dynamic, meaning properties can be added or removed at runtime. This dynamic nature is achieved through the flexibility of hash tables.

**Example:**

// Adding a new property dynamically

person.location = "Chennai";

console.log(person.location); // **Output:** Chennai

// Removing a property dynamically

delete person.age;

console.log(person.age); // **Output:** undefined

**Object Methods:**

Objects in JavaScript can also contain methods, which are functions associated with the object. These methods are treated as values and can be called just like any other function.

**Example:**

// Object Method

var car = {

brand: "Toyota",

model: "Camry",

start: function() {

console.log("Engine started!");

}

};

car.start(); **// Output:** Engine started!