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

In JavaScript, objects are collections of key-value pairs, where keys are strings (or symbols) and values can be of any data type, including other objects. Objects are used to represent real-world entities, data structures, and more complex data types.

These objects are quite different from JavaScript’s primitive data-types (Number, String, Boolean, null, undefined and symbol) in the sense that while these primitive data-types all store a single value each (depending on their types).

Objects are more complex and each object may contain any combination of these primitive data-types as well as reference data-types.

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

Let’s have an **Example** of my favourite merc car and list out its properties (Features):

1. Make: Mercedes
2. Model: C-Class
3. Colour: White
4. Fuel: Diesel
5. Weight: 850kg
6. Mileage: 8Kmpl
7. Rating: 4.5

Taking the above as reference, I'll stress up on objects, Object properties and Methods.

**1)Objects:**

The following code assigns a simple value (Mercedes) to a variable named car:

var car = "Mercedes";

Objects are variables too. But objects can contain many values. The following code assigns many values (Mercedes, C-class, White and so on) to a variable named Car:

var car = {Make: “Mercedes”, Model: “C-Class”, Colour: “White”, Fuel: Diesel, Weight: “850kg”, Mileage: “8Kmpl”, Rating: 4.5};

The values are written as name: value pairs (name and value separated by a colon).

**Syntax:**

var <object-name> = {key1: value1, key2: value2,... keyN: valueN};

**2)Object Properties:**

The name: values pairs (in JavaScript objects) are called **properties**

var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel, Weight: “850kg”,Mileage: “8Kmpl”, Rating: 4.5};

From the above snippet, let’s have a look what falls under property and property value:



The object properties can be different primitive values, other objects and functions.

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;

**The syntax to access a property from an object is:**

objectName.property        // Car.Make

or

objectName["property”]    // Car["Make"]

**3) Object Methods:**

An object method is an object property containing a function definition.

i.e.,

Let’s assume to start the car there will be a mechanical functionality.

function(){return ignition.on}