### Blog about the object and its internal representation in javascript :

The object:

An object in JavaScript is a data type that can take in collections of key-value pairs. A major difference between an object and other data types such as strings and numbers in JavaScript is that an objects can store different types of data as its values .

Here are some of the key properties of objects in JavaScript:

* + Objects are unordered collections of data.
  + Objects are mutable, meaning that their properties can be changed after they are created.
  + Objects can contain any type of data, including other objects.
  + Objects can have methods, which are functions that are attached to the object.
  + Objects can be created using object literals or constructors.

Information about objects and their internal representation in JavaScript.

* Objects are a fundamental data type in JavaScript. They are used to store collections of data and functions. Objects are created using curly braces {} and can contain any number of properties. Properties are key-value pairs, where the key is a string and the value can be any data type, including other objects.
* Objects are stored in memory as dictionaries. A dictionary is a data structure that maps keys to values. In JavaScript, the keys are strings and the values can be any data type.
* The internal representation of an object is a pointer to a dictionary. The pointer is stored in the object's header. The header also contains the object's size and a flag that indicates whether the object is sealed or extensible.
* A sealed object cannot be modified, while an extensible object can be modified by adding new properties.
* When you create a new object, the JavaScript engine allocates a new dictionary and stores a pointer to it in the object's header. The object's size is set to the number of properties in the dictionary.
* When you add a new property to an object, the JavaScript engine checks to see if the object is sealed. If the object is not sealed, the engine adds the new property to the dictionary. If the object is sealed, the engine throws an error.
* You can remove a property from an object using the delete operator. When you delete a property, the JavaScript engine removes the property from the dictionary. If the property does not exist in the dictionary, the engine does nothing.
* You can also iterate over the properties of an object using a for...in loop. The for...in loop iterates over the keys in the dictionary.