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

**Object:**

[**In JavaScript, objects are the most important data type and form the building blocks for modern JavaScript**](https://medium.com/analytics-vidhya/objects-and-its-internal-representation-in-javascript-c5b576b20a99)**.**

**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.**

**// Example of an object in JavaScript**  
const person = {  
firstName: ‘arun’,  
lastName: ‘prakash’,  
age: 30,  
address: {  
city: ‘chennai’,  
country: ‘india’  
}  
};

**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 store the value.**

**Internal Representation of Objects:**

Understanding how JavaScript internally represents objects is crucial for writing efficient and performing code.

**Object Properties**:

Each object in JavaScript is internally represented as a collection of properties. Properties consist of a key (or name) and a value. These properties can be data properties (containing a value) or accessor properties (containing a getter or setter function).

objectName.propertyName

**Hidden Classes:**

JavaScript engines use a concept called “hidden classes” to optimize property access and assignment. When an object is created, the engine assigns it a hidden class based on its structure. Objects with the same structure share the same hidden class, which improves performance by allowing the engine to predict the object’s layout in memory.

**Prototypes and Inheritance:**

Objects in JavaScript can be linked to other objects through prototypes. When a property is not found in an object, the JavaScript engine looks up the prototype chain until it finds the property or reaches the end of the chain (usually with the ‘Object’ prototype). This mechanism enables inheritance and the creation of reusable object structures.

**Creating Objects In JavaScript** :

Create JavaScript Object with Object Literal

One of easiest way to create a javascript object is object literal, simply define the property and values inside curly braces.

Create JavaScript Object with Constructor

Constructor is nothing but a function and with help of new keyword, constructor function allows to create multiple objects of same flavor.