

2. Write a blog about objects and its internal representation in Javascript

Introduction: JavaScript, the language of the web, is known for its versatility and dynamic nature. At the heart of its flexibility lies a fundamental data structure – objects. Objects in JavaScript are powerful entities that allow developers to model real-world entities, organize data, and facilitate complex interactions.

The Basics of Objects in Javascript.

In JavaScript, objects are collections of key-value pairs, where each key is a string (or a symbol) and each value can be of any data type, including other objects. This flexible structure makes objects an essential part of the language's expressive power.

// Example of a simple object

```
let car = {  
  make: 'Toyota',  
  model: 'Camry',  
  year: 2022,  
  isAutomatic : true,  
  features: ['power windows', 'air conditioning', 'rearview camera']  
};
```

- **Internal Representation:**

1.Hash Table:

A hash table is a data structure that maps keys to values. JavaScript engines use a hash function to convert the keys into an index in the underlying array. This index is then used to store and retrieve the corresponding value.

2.Hidden Classes and Shapes:

JavaScript engines optimize object property access by using hidden classes and shapes. When an object is created, the engine assigns it a hidden class based on its structure. If two objects have the same structure, they share the same hidden class, which improves performance.

- **Object Prototype and Inheritance:**

JavaScript supports prototype-based inheritance, allowing objects to inherit properties and methods from other objects. Objects in JavaScript have an internal property called `[[Prototype]]` that references another object.

- **Dynamic Nature and Mutable:**

One of JavaScript's defining features is its dynamic and mutable nature. Objects can be modified at runtime by adding, updating, or deleting properties.

```
// Adding a new property
```

```
car.color = 'blue';
```

```
// Updating a property
```

```
car.year = 2023;
```

```
// Deleting a property
```

```
delete car.make;
```

- **Conclusion:**

Understanding the internal representation of objects in JavaScript provides insights into the language's performance characteristics and helps developers write more efficient code. Objects, with their dynamic nature and versatile representation, play a crucial role in shaping the language's expressive power. Happy coding!