OBJECTS and its Internal Representation in JS

INTRODUCTION :

* Javascript, a versatile programming language, relies heavily on objects to model and manipulate data. Objects serve as a fundamental building block, allowing developers to organize and structure their code effectively. In this blog post, we’ll delve into the internal representation of objects in javascript, exploring how they are created, accessed and manipulated.

Basics of objects :

* In Javascript, an object is a complex data type that allows you to store and organize data in key-value pairs. You can create objects using objects literals or constructor functions.

Lets consider a simple example:

// Object literal

const person = {

name: ‘Vijay’ ,

age: 25,

profession: ‘Developer’

};

Here, ‘person’ is an object with properties such as ‘name’, ‘age’ and ‘profession’ . Understanding how these objects are represented internally is crucial for efficient programming.

Objects properties and methods :

* Objects in javascript can have both properties and methods. Properties store data, while methods are functions associated with the object. To access properties, you can use dot notation or square brackets:

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

console.log(person [ ‘age’ ] ); // Output : 25

Object internal representation :

* Internally, javascript engines use various mechanisms to represent objects efficiently. One common approach is a hash table or dictionary-like structure, where each property is stored as a key-value pair. This allows for fast access to properties but may result in memory overhead.

Object prototypes and inhertiance :

* Javascript, supports prototype-based inheritance, where objects can inherit properties and methods from other objects. Understanding prototype chains is crucial for comprehending how objects share and inherit.

Dynamic Nature of objects :

* One of Javascript’s strength is its dynamic nature, allowing you to add or remove properties from objects at runtime.

Conclusion :

* Understanding the internal representation of objects in Javascrpit is vital for writing efficient and maintainable code. Objects,with their properties,methods and dynamic nature, form a cornerstone of the languages flexibility.

BLOG BY,

NITHISHKUMAR. B