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

⦁ Objects And Its Internal Representation In JavaScriptObjects, in JavaScript, is it’s most important data-type and forms the building blocks for modern JavaScript. 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 singlevalue 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.Loosely speaking, objects in JavaScript may be defined as an unordered collection of related data, of primitive or reference types, in the form of “key: value” pairs. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

⦁ For Eg. If your object is a student, it will have properties like name, age, address, id, etc and methods like4updateAddress,4updateNam, etc.

**Objects are Variables**

JavaScript variables can contain single values:

**Example**

let person = "John Doe";

JavaScript variables can also contain many values.

Objects are variables too. But objects can contain many values.

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

**Example**

let person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

A JavaScript object is a collection of named values

It is a common practice to declare objects with the const keyword

**Example**

const person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

**Creating a JavaScript Object**

With JavaScript, you can define and create your own objects.

There are different ways to create new objects:

⦁ Create a single object, using an object literal.

⦁ Create a single object, with the keyword new.

⦁ Define an object constructor, and then create objects of the constructed type.

⦁ Create an object using Object.create().

**Using the JavaScript Keyword new**

The following example create a new JavaScript object using new Object(), and then adds 4 properties:

**Example**

const person = new Object();

person.firstName = "John";

person.lastName = "Doe";

person.age = 50;

person.eyeColor = "blue";

**JavaScript Objects are Mutable**

Objects are mutable: They are addressed by reference, not by value.

If person is an object, the following statement will not create a copy of person:

const x = person; // Will not create a copy of person.

The object x is not a copy of person. It is person. Both x and person are the same object.

Any changes to x will also change person, because x and person are the same object.

**Example**

const person = {

firstName:"John",

lastName:"Doe",

age:50, eyeColor:"blue"

}

const x = person;

x.age = 10; // Will change both x.age and person.age