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

* Objects are important data types in java script. Objects are quite different from primitive data types like numbers, String, Boolean, null, undefined, symbol. Primitive data types can hold only one value but objects can hold multiple data types in key value pair formats.
* Objects are most complex data types each object may contain the combination of primitive data types as well as referenced data types.
* Objects are referenced data types. Variables that are assigned a referenced value is given to the reference or a pointer to that address. That reference or pointer points to that location in memory where the object is stored.
* An object does not store the value it works as for every given value a referenced value is given and it points to the address of that memory location.
* Objects in JavaScript may be defined as an unordered collection of related data of primitive and reference data types in the form of key value pairs. These keys can be variables or methods and are called properties and methods respectively in context of object.

Creating Objects in Java Script

**By Using Object Literal**

object= {property1:value1, property2:value2, property3:value3, ...........propertyN: valueN}

Example

var emp= {

name:"saiteja"

code:1456

isregistered: yes

};

**By Creating Instance of Object directly (Using new keyword):**

var objectname=new Object ();

Example:

var emp = new Object ();

emp.name="saiteja"

emp. code=1456;

emp. isregistered=yes;

By Creating Object with Constructor

function Vehicle (name, maker) {  
this.name = name;  
this. Maker = maker;  
}  
let car1 = new Vehicle (’Fiesta’, 'Ford’);  
let car2 = new Vehicle (’Santa Fe’, 'Hyundai’)  
console.log(car1.name); //Output: Fiesta  
console.log(car2.name); //Output: Santa Fe

**Accessing the Objects**

The syntax for accessing the property of object is using object

1.Dot method => Name.property,

2.Off method => objectName[property]

Example:

Using dot operator

emp.name

Using []

emp["name"]

**Points to Be Remembered**

* Unassigned properties of objects are undefined but not null
* Objects are sometimes called as associative arrays because each property of a string is associated with the string value that can be used to access it. (In the above example we can access the employee properties like name, code, isregistered with a single object).
* An object property name can be any valid Java Script object or anything that can be converted into string like spaces between strings, empty string etc.

myObj.type = 'Dot syntax';  
myObj ['date created'] = 'String with space';  
myObj[str] = 'String value';  
myObj[rand] = 'Random Number';  
myObj[obj] = 'Object';  
myObj [''] = 'Even an empty string’; console.log(myObj);

* We can access properties using a string value that is stored in a variable.

var propertyName = 'make';  
myCar[propertyName] = 'Ford’; propertyName = 'model';  
myCar[propertyName] = 'Mustang';

**Iterating the objects Using For In Loop:**

function showProps (obj, objName) {  
var result = ``;  
for (var i in obj) {  
// obj. hasOwnProperty () is used to filter out properties from the object's prototype chain  
if (obj. hasOwnProperty(i)) {  
result += `${objName}. ${i} = ${obj[i]} \n`;  
}  
}  
return result;  
}

Working

The function call showProps (emp, "myCar") would return the following:

emp.name returns “saiteja"

emp. code returns1456;

emp. isregistered returns yes;