**OBJECTS AND ITS INTERNAL REPRESENTATION IN JAVASCRIPT:**

Objects is one of the most important data types in Javascript . It consists of key value pairs which could represent any real world entities.They form the basis for the modern Javascript like interacting with the data that are fetched either from database or an third party API .Its different from normal JavaScript primitive data types like number,boolean,float,strings etc.

Objects are a collection of key value pairs which could consists of various data types and each object can be represented as the following

{

Key1 : value1,

Key2 : value2,

...

Keyn : valuen.

}

Values of an property in an JavaScript can be of any primitive data types like number,string,boolean etc or complex data types like arrays,functions or even an object itself.

{

Key1:value1,

Key2:[value1,value2,value3,…,valuen],

Key3 : function{function definition},

Key4:{key5:value5,key6:value6,…keyn:valuen}

}

Each property of an Object is a key value pair where key is basically string and value is of any data types.Objects in javascript could be created in two ways

1. Object literal
2. Object constructor

Through Object literals,javascript objects could be created using flower braces and initializing a set of properties to it.For e.g.

Var human={

“name”:”Ram”,

“age”:20,

“school”:”ABC school”

}

Properties within the javaScript objects are accessed through keys and these keys could be accessed in the following ways.

Object.key

Object[“key”]

The second approach is the mostly used one and its also useful in accessing the values which has keys with spaces (like Object[“hello world”]).

The second approach is the Object constructor .An constructor in programming languages is basically a special function or method that helps in creation of objects and initializing its properties with initial values;

Eg:

Var sam = new Object();

sam.color:”brown”

sam.breed:”Labrador”

sam.owner:”Alice”

The new keyword is used during the creation of objects in Javascript.The usage of new keyword before any function signature during the initialization makes it as an constructor.

function Food\_menu(name, maker) {  
this.name = name;  
this.chef = chef;  
}

let car1 = new Food\_menu(‘Chicken Biriyani’, ‘ramya’);  
let car2 = new Food\_menu(‘Fried rice’, ‘priya’)