**OBJECT AND IT’S INTERNAL REPRESENTATION IN JAVASCRIPT**

Object is generally defined as real life entities. For example if there is a class Animal , Cat , Dog can be objects and they can have properties like color , age and methods like eat , sleep. Let us see how it is structured and how it is works in javascript.

**Datatype and structure:**

Object is of complex data type. It can have combination of primitive data types (like Number , String , Boolean , null , undefined) / reference data types(points to location in memory where it is stored instead of actual value) in key-value pairs. Key is nothing but properties of object which can have variables or functions.

So properties are nothing but javascript variables related to object as they describe characteristics of object. Property name should be any valid string .So for object dog we can have properties like name , breed , age etc.

**How to access properties of objects?**

1. **Using dot operator**

**Syntax :**

objectname**.**propertyname

**Example:**

dog.name =”Tom”;

dog.breed=”german shepard”;

dog.age; //only declaration

If no values are assigned the default value will be undefined

1. **Using bracket notation.**

As each property is associated with a string value it can be used to access it and this is why objects can also be called as associative arrays.

**Syntax:**

objectname**[**propertyname**]**

**Example:**

dog[‘birthyear’] = 2020;

A property can be accessed using bracket notation only if property name is not a valid javascript identifier like if name starts with number or space or hyphen or when property is dynamically determined.

dog.2-food=”biscuit” // ERROR

dog['1-food']="meat";

dog[""]="No";

//dynamically determining properties

var cat = new Object() ,

str = 'color',

random = Math.random(),

obj = new Object();

cat[str] = "grey";

cat[random] = "random";

cat[obj] = dog;

**OUTPUT:** {color: 'grey', 0.9927202358305334: 'random', [object Object]: {…}}

1. **Using string stored in a variable**

**Syntax:**

var variablename = “propertyname”;

objectname[variablename] = propertyvalue ;

**Example:**

var propertyname = 'age';

cat[propertyname] = 7;

**How to create objects in JavaScript?**

1. **Object literals**

**Syntax:**

Objectname **= {**propertyname1:value , propertyname2:value , …… propertynameN:value**}**

Example:

let dog = {name:"Tom",age:"7"};

1. **Using constructor**

**Syntax:**

function functionname(property1,property2…){

property1 = this.property1;

property2= this.property2;

}

let objectname1 = new functionname(property1 value , property2 value ….);

let objectname2 = new functionname(property1 value , property2 value ….);

.

.

.

**Example:**

function Animal(name,age){

this.name = name;

this.age = age;

}

let dog = new Animal("Tom",7);

let cat = new Animal("Jim",4);

1. **Using new keyword**

**Syntax:**

var objectname = new Object();

objectname.propertyname = value;

**Example:**

var dog = new Object();

dog.name="Tom";

dog.age=7

1. **Using Object.create() method**

**Syntax:**

Variablename = {

//Assigning default values to properties

Propertyname: value;

Propertyname: value

}

Objectname = Object.create(variablename);

//changing default values and assigning new values toproperties

Objectname.proertyname = newvalue;

**Example:**

var Animal = {

name: 'Invertebrates',

age: 0

}

var dog = Object.create(Animal);

dog.name = "Tom";

dog.age = 7