OBJECT IN JAVASCRIPT

Like integers, string, array and object is also a data type. Object is a collection of variables.

NOTE: The object can store values of any data type.

As the variables are attached to an object they are thus called properties. In an object we can also store function. The function in an object is referred to as method.

To access property of an object:

Object_name['prop'];

Object_name.prop;

Creating an object

1. Object Initializer.

It starts and ends with curly braces {}. Within the curly braces we define all the properties and methods along with their values. Each property separated using comma and each property and its value is separated using colon.

Example:

```
let dog ={

    breed: 'Golden retriever',
    height: '4ft',
    age: 2,
};
console.log(dog.breed); // Golden retriever
```

Dot (.) operator

In an object the dot operator is used to access the property or method of an object. It helps to assign a new value to an existing property. It also helps to define a new property.

Ex:

```
dog.breed = 'Husky';//Golden retriever to Husky.
dog.weight=’32kg’;  //new property.
```

Square [] brackets

Square bracket is used to access the property, assign a new value to an existing property and also define a new property like the dot operator.

```
Console.log(dog['breed']);
Console.log (dog['height']);
```

Both the dot operator and square brackets helps us to access the properties of the object. Also both allow us to edit an existing property and allow us to define a new property. But using square bracket we can’t access the method of an object or call a function. For that dot operator is used.

Properties and methods can’t be accessed directly, we must access them with the object. Even inside the object we can’t access properties directly.

2. new object ()

```
let dog = new Object();
dog.breed = 'Golden retriever';
dog.height = '4ft';
dog.age = 2;

```

let dog = new Object();
has no property in it.
Once we created the object we can then add the property using dot operator.

3. Function constructor

```
    function car(make,model,year){
```

```
    this.make=make;
    this.model=model;
    this.year=year;
    this.display=function(){
        console.log(this.make+thiis.model+this.year);
    };
}
let car1 = new car(“hyundai”,”i20”,2014);
```

this.make=make; which means we are expecting an object here and we have assigned the make property of the object with the value that the function receives similarly the (this.model=model and this.year=year)

first we see how we can call this function

(car(“Hyundai”,”i20”,2014); when we call this function by passing 3 arguments. The function will be called and three arguments will get the respective values.

But the moment the code encounters “this” it will throw an error as there is no object. So now the solution to this error is by calling the function by using the new keyword.(let car1 = new car(“Hyundai”.”i20”,2014).

This will assign the respective values to the 3 arguments ie: (make=”Hyundai”, model=”i20”, year=2014)

We have called the function using ‘new’ keyword. This creates a new object. So in our code when it encounters ‘this’ keyword now it will refer to this new object.

3. Class constructor

```
    class car{

    constructor(make,model,year){
        this.make=make;
        this.model=model;
        this.year=year;
    }
    display(){
        console.log(this.make+this.model+this.year);
    };
```

```
};
```

The class constructor syntax is similar to the function constructor syntax

Arguments are in the function named constructor further the function constructor is inside the class

Here we may have a doubt, why we need classes to create object, If the syntax is similar to that of a function constructor?

The reasons are,

It offers convenient syntax.

It allows us to declare thee function outside the constructor.

NOTE: we are no longer using the function keyword along with the display method.

Rest of the things remain the same. Like the way the object are created and the way properties and the methods are accessed.

For-in-loop

It is exclusively used for objects in JavaScript. It iterates through all the properties in an object.

Creating an object using classes

```
class car{
    constructor(make,model,year){
        this.make=make;
        this.model=model;
        this.year=year;
    }
    display(){
        console.log(this.make+this.model+this.year);
    };
};
let car1 = new car("Hyundai","i20",2014);
for(let prop in car1){
    console.log(prop+":"+car1[prop]+<br/>);
}
//output: make:Hyundai
```

```
        Model:i20
        Year:2014
```

To create object we will use classes and we can define class and the constructor a function with arguments and we will then add all the properties and outside the constructor function we will define the function and then we will define the object. And lastly we will use the for-in-loop.

Here we are iterating through the car1object and printing the value and its properties.

And to concardinate the string wee use(+)operator.