

JavaScript Objects

A JavaScript object is an entity having state and behavior (properties and method). For example: car, pen, bike, chair, glass, keyboard, monitor etc.

JavaScript is an object-based language. Everything is an object in JavaScript.

JavaScript is template based not class based. Here, we don't create class to get the object. But, we directly create objects.

Creating Objects in JavaScript

There are 3 ways to create objects.

1. By object literal
2. By creating instance of Object directly (using new keyword)
3. By using an object constructor (using new keyword)

1) JavaScript Object by object literal

The syntax of creating object using object literal is given below:

1. `object={property1:value1,property2:value2.....propertyN:valueN}`

As you can see, property and value is separated by : (colon).

Let's see the simple example of creating object in JavaScript.

```
<script>
let emp={id:102,name:"Shyam Kumar",salary:40000}
console.log(emp.id+" "+emp.name+" "+emp.salary);
</script>
```

Output of the above example

```
102 Shyam Kumar 40000
```

2) By creating instance of Object

The syntax of creating object directly is given below:

1. var **objectname**=**new** Object();

Here, **new keyword** is used to create object.

Let's see the example of creating object directly.

```
<script>
var emp=new Object();
emp.id=101;
emp.name="Ravi Malik";
emp.salary=50000;
console.log(emp.id+" "+emp.name+" "+emp.salary);
</script>
```

Output of the above example

101 Ravi 50000

3) By using an Object constructor

Here, you need to create function with arguments. Each argument value can be assigned in the current object by using this keyword

The **this keyword** refers to the current object.

The example of creating object by object constructor is given below.

```
<script>
function emp(id,name,salary){
  this.id=id;
  this.name=name;
  this.salary=salary;
}
let e=new emp(103,"Vimal Jaiswal",30000);

Console.log(e.id+" "+e.name+" "+e.salary);
</script>
```

Output of the above example

103 Vimal Jaiswal 30000

OBJECT METHODS

REFERE CLASSROOM CODE FOR OBJECT METHODS.

//Date object

-Js date object represent a single moment in time in a platform independent format.

-date object contains numbers that represents milliseconds since 1 january 1970

Date object methods

//date object

```
// console.log("date object for regular date");  
var a=new Date();  
// console.log(a);  
//date (year,mon,day,hr,min,sec) 0=jan ,feb=1  
// console.log("manual date");  
// var b=new Date(2022,11,5,2,34,33);  
// console.log(b);  
// // 1sec=1000milisec  
// var c=new Date(1000);  
// console.log(c);  
// //date (date should be in string formate manual input)  
// var d=new Date("25,jan,2022");  
// console.log(d);  
// //get the date  
console.log(a.getDate());
```

```
console.log(a.getMonth()+1);  
console.log(a.getFullYear());  
console.log(a.getTime());  
console.log(a.getHours());  
console.log(a.getDay());  
// //25/12/2022  
console.log(a.getDate()+"/"+a.getMonth()+"/"+a.getFullYear());  
console.log("27"+"/"+"jan"+"/"+"2022");
```

//Math objects

- Math is inbuilt object that has properties and methods for mathematical constant and functions.its not a function object
- math is work with number type

math object methods

```
var a= Math.PI;  
console.log(a);  
document.write(a);  
//round  
var b=Math.round(12.74);//round the nearest int value  
console.log(b);  
//power  
var c= Math.pow(2,3);  
console.log(c);  
//absolute it can conver -value to positive  
var d=Math.abs(-12);  
console.log(d);
```

```
//Square root
var e=Math.sqrt(16);
console.log(e);

//floor give 12.49 or 12.99 it will give the floor value 12
var f=Math.floor(12.99);
console.log(f);

//ceil give 12.49 or 12.99 it will give the next vaule (ceil)
var g=Math.ceil(12.44);
console.log(g);

//max
var h=Math.max(22,33,44,55,66,77,88,11);
console.log(h);

//min
var i=Math.min(22,33,44,55,66,77,88,11);
console.log(i);

//random
var j= Math.random();
console.log(j);
```