# JavaScript Date Object

The **JavaScript date** object can be used to get year, month and day. You can display a timer on the webpage by the help of JavaScript date object.

You can use different Date constructors to create date object. It provides methods to get and set day, month, year, hour, minute and seconds.

## Constructor

You can use 4 variant of Date constructor to create date object.

1. Date()
2. Date(milliseconds)
3. Date(dateString)
4. Date(year, month, day, hours, minutes, seconds, milliseconds)

## JavaScript Date Methods

| Methods | Description |
|---|---|
| getDate() | It returns the integer value between 1 and 31 that represents the day for the specified date on the basis of local time. |
| getDay() | It returns the integer value between 0 and 6 that represents the day of the week on the basis of local time. |
| getFullYears() | It returns the integer value that represents the year on the basis of local time. |
| getHours() | It returns the integer value between 0 and 23 that represents the hours on the basis of local time. |
| getMilliseconds() | It returns the integer value between 0 and 999 that represents the milliseconds on the basis of local time. |
| getMinutes() | It returns the integer value between 0 and 59 that represents the minutes on the basis of local time. |
| getMonth() | It returns the integer value between 0 and 11 that represents the month on the basis of local time. |
| getSeconds() | It returns the integer value between 0 and 60 that represents the seconds on the basis of local time. |
| getUTCDate() | It returns the integer value between 1 and 31 that represents the day for the specified date on the basis of universal time. |
| getUTCDay() | It returns the integer value between 0 and 6 that represents the day of the week on the basis of universal time. |
| getUTCFullYears() | It returns the integer value that represents the year on the basis of universal time. |
| getUTCHours() | It returns the integer value between 0 and 23 that represents the hours on the basis of universal time. |
| getUTCMinutes() | It returns the integer value between 0 and 59 that represents the minutes on the basis of universal time. |
| getUTCMonth() | It returns the integer value between 0 and 11 that represents the month on the basis of universal time. |

# JavaScript Math

The **JavaScript math** object provides several constants and methods to perform mathematical operation. Unlike date object, it doesn't have constructors.

Synatax

- Math.pow(base value,power)
- Math.sqrt(value)
- Math.max(v1,v2,v3,vn)
- Math.min(v1,v2,v3,vn)
- Math.abs(negative vaue)
- Math.floor(value)
- Math.ceil(value)
- Math.random( )
- Math.round(value)
- Math.PI( )

## OBJECT DESTRUCTURING

JavaScript Object Destructuring is an expression which allows us to access the data from objects like arrays, objects, maps and assign it to new variables. Through this object destructuring, we can create variables easily from the object's properties

Syntax:

Variable{key1:variable1,key2:variable2,keyn:variablen}

Ex-

```
let obj ={
```

```
        name:`ganesh`,
        age:25,
        status:true,
    }

Console.log(obj);
```

Destructuring

```
let {name:objName,age:objAge,status:objSatus}=obj;
```

## ARRAY DESTRUCTURING

Destructuring means to break down a complex structure into simpler parts. With the syntax of destructuring, you can extract smaller fragments from objects and arrays. It can be used for assignments and declaration of a variable.

Syntax-

```
let [variable1,variable2,variablen]=array refVariable;

let array=[10,20,30,40]

console.log(array)
```

Destructuring

```
let[value1,value2,value3,value4]=array
console.log(value1+value2+value3+value4)//100
```

# JavaScript Array map() method

The JavaScript array map() method calls the specified function for every array element and returns the new array. This method doesn't change the original array.

## Syntax

The map() method is represented by the following syntax:

1. array.map(callback(currentvalue,index,arr),thisArg)

## Parameter

**callback** - It represents the function that produces the new array.

**currentvalue** - The current element of array.

**index** - It is optional. The index of current element.

**arr** - It is optional. The array on which map() method operated.

**thisArg** - It is optional. The value to use as this while executing callback.

## Return

A new array whose each element generate from the result of a callback function.

## JavaScript Array map() method example

```
let  fName=[“ajay”,”atul”,”jay”];
let fullNAme=fName.map((cValue)=>{
        return cValue+ “ patil”
```

})

Console.log(fullName);

[“ajay patil ”,”atul patil”, “ jay patil” ]

# JavaScript Array filter() method

The JavaScript array filter() method filter and extract the element of an array that satisfying the provided condition. It doesn't change the original array.

## Syntax

The filter() method is represented by the following syntax:

1. array.filter(callback(currentvalue,index,arr),thisArg)

## Parameter

**callback** - It represents the function that test the condition.

**currentvalue** - The current element of array.

**index** - It is optional. The index of current element.

**arr** - It is optional. The array on which filter() operated.

**thisArg** - It is optional. The value to use as this while executing callback.

## Return

A new array containing the filtered elements.

## JavaScript Array filter() method example

```
let array=[10,20,30,40,50,60,70]
```

```
   let arrfil= array.filter((cValue)=>{
       if(cValue>10 && cValue<60){
          return cValue;
}
})
Console.log(arrfil)//  [20,30,40,50]
```