

### 5.1.1 new Date()

You can create a date object without passing any arguments to the

`new Date()` constructor function.

For example,

JAVASCRIPT

```
1 let now = new Date();
2
3 console.log(now); // Tue Feb 02 2021 19:10:29 GMT+0530 (India Standard Time) { }
4 console.log(typeof(now)); // object
```

Here,

`new Date()` creates a new date object with the current date and local time.

#### Note

1. Coordinated Universal Time (UTC) - It is the global standard time defined by the World Time Standard. (This time is historically known as Greenwich Mean Time, as UTC lies along the meridian that includes London and nearby Greenwich in the United Kingdom.)
2. Local Time - The user's device provides the local time.

### 5.1.2 new Date(milliseconds)

The

`Date` object contains a number that represents milliseconds since 1 January 1970 UTC.

The

`new Date(milliseconds)` creates a new date object by adding the milliseconds to zero time.

For example,

JAVASCRIPT

```
1 let time1 = new Date(0);
2 console.log(time1); // Thu Jan 01 1970 05:30:00 GMT+0530 (India Standard Time) { }
3
4 // 100000000000 milliseconds from 1 Jan 1970 UTC
5 let time2 = new Date(100000000000);
6 console.log(time2); // Sat Mar 03 1973 15:16:40 GMT+0530 (India Standard Time) { }
```



#### Note

1000 milliseconds is equal to 1 second.

### 5.1.3 new Date(date string)

The

`new Date(date string)` creates a new date object from a date string.

#### Syntax:

`new Date(datestring);`

JAVASCRIPT

```
1 let date = new Date("2021-01-28");
2
3 console.log(date); // Thu Jan 28 2021 05:30:00 GMT+0530 (India Standard Time) { }
```



You can also pass only the year and month or only the year. For example,

```
1 let date = new Date("2020-08");
2 console.log(date); // Sat Aug 01 2020 05:30:00 GMT+0530 (India Standard Time) { }
3
4 let date1 = new Date("2020");
5 console.log(date1); // Wed Jan 01 2020 05:30:00 GMT+0530 (India Standard Time) { }
```

### Short date format

```
1 // short date format "MM/DD/YYYY"
2 let date = new Date("03/25/2015");
3 console.log(date); // Wed Mar 25 2015 00:00:00 GMT+0530 (India Standard Time) { }
```

### Long date format

```
1 // long date format "MMM DD YYYY"
2 let date1 = new Date("Jul 1 2021");
3 console.log(date1); // Thu Jul 01 2021 00:00:00 GMT+0530 (India Standard Time) { }
```

Month and Day can be in any order

```
1 let date2 = new Date("1 Jul 2021");
2 console.log(date2); // Thu Jul 01 2021 00:00:00 GMT+0530 (India Standard Time) { }
```

The month can be full or abbreviated. Also, month names are case insensitive.

```
1 let date3 = new Date("July 1 2021");
2 console.log(date3); // Thu Jul 01 2021 00:00:00 GMT+0530 (India Standard Time) { }
3
```

```
4 // commas are ignored
5 let date4 = new Date("JULY, 1, 2021");
6 console.log(date4); // Thu Jul 01 2021 00:00:00 GMT+0530 (India Standard Time) { }
```

#### 5.1.4 new Date(year, month, day, hours, minutes, seconds, milliseconds)

It creates a new date object by passing a specific date and time.

For example,

JAVASCRIPT

```
1 let time1 = new Date(2021, 1, 20, 4, 12, 11, 0);
2 console.log(time1); // Sat Feb 20 2021 04:12:11 GMT+0530 (India Standard Time) { }
```

Here, months are counted from 0 to 11. January is 0 and December is 11.

The passed argument has a specific order.

If four numbers are passed, it represents the year, month, day and hours.

For example,

JAVASCRIPT

```
1 let time1 = new Date(2021, 1, 20, 4);
2 console.log(time1); // Sat Feb 20 2021 04:00:00 GMT+0530 (India Standard Time) { }
```

Similarly, if two arguments are passed, it represents year and month.

For example,

JAVASCRIPT

```
1 let time1 = new Date(2020, 1);
2 console.log(time1); // Sat Feb 20 2021 04:00:00 GMT+0530 (India Standard Time) { }
```



## ⚠ Warning

If you pass only one argument, it is treated as milliseconds. Hence, you have to pass two arguments to use this date format.

### 5.2 AutoCorrection in Date Object

When you assign out of range values in the Date object, it auto-corrects itself.

For example,

JAVASCRIPT

```
1 let date = new Date(2008, 0, 33);
2 // Jan does not have 33 days
3 console.log(date); // Sat Feb 02 2008 00:00:00 GMT+0530 (India Standard Time) { }
```

33 days are auto corrected to 31 (jan) + 2 days in feb.

### 5.3 Instance Methods

There are methods to access and set values like a year, month, etc. in the Date Object.

Method	Description
now()	Returns the numeric value corresponding to the current time (the number of milliseconds passed since January 1, 1970, 00:00:00 UTC)
getFullYear()	Gets the year according to local time
getMonth()	Gets the month, from 0 to 11 according to local time
getDate()	Gets the day of the month (1–31) according to local time
getDay()	Gets the day of the week (0-6) according to local time
getHours()	Gets the hour from 0 to 23 according to local time

Method	Description
getMinutes	Gets the minute from 0 to 59 according to local time
getUTCDate()	Gets the day of the month (1–31) according to universal time
setFullYear()	Sets the full year according to local time
setMonth()	Sets the month according to local time
setDate()	Sets the day of the month according to local time
setUTCDate()	Sets the day of the month according to universal time

JAVASCRIPT

```

1 let date1 = new Date(1947, 7, 15, 1, 3, 15, 0);
2
3 console.log(date1.getFullYear()); // 1947
4 console.log(date1.getMonth()); // 7

```

### 5.3.1 Setting Date Values

JAVASCRIPT

```

1 let date1 = new Date(1947, 7, 15);
2 date1.setYear(2021);
3 date1.setDate(1);
4
5 console.log(date1); // Sun Aug 01 2021 00:00:00 GMT+0530 (India Standard Time) { }

```



Try out the JS function properties and Date Methods in the JavaScript Code Playground.