Date Object

Date object

Date object

A ***Date*** object represents a single moment in time, based on the number of milliseconds since the Unix Epoch (January 1, 1970 UTC). UTC (Coordinated Universal Time), also known as GMT (Greenwich Mean Time), is a 24-hour time standard. The Date object is created with the new operator and a constructor. A ***constructor*** is a function that creates an instance of an object.

Date object constructor.

```
let currDateTime = new Date();
console.log(currDateTime);

let oneSecPastEpoch = new Date(1000);
console.log(oneSecPastEpoch);

// Feb 22, 1732
let georgeBirthday = new Date(1732, 1, 22);
console.log(georgeBirthday);

// Oct 21, 2035 at 7:28:00
let theFuture = new Date(2035, 9, 21, 7, 28, 0);
console.log(theFuture);
```

```
Thu Apr 18 2019 15:26:13 GMT-0500 (Central Daylight Time)
Wed Dec 31 1969 18:00:01 GMT-0600 (Central Standard Time)
Fri Feb 22 1732 00:00:00 GMT-0600 (Central Standard Time)
Sun Oct 21 2035 07:28:00 GMT-0500 (Central Daylight Time)
```

Explanation :

1. Initialize the variable currDateTime to the current date and time using the Date constructor.
2. Display the currDateTime variable, which is in the local time zone. Central Daylight Time is 5 hours before Greenwich Mean Time (GMT).
3. Initialize the variable oneSecPastEpoch to 1000 milliseconds past Jan 1, 1970 using the Date constructor.
4. Central Standard Time is 6 hours before GMT. Daylight time (called Daylight Saving Time) is one hour different than standard time because clocks are turned forward one hour.

5. Initialize the variable georgeBirthday to Feb 22, 1732. The month parameter ranges from 0-11, so 1 = Feb.
6. georgeBirthday falls on a Friday and is 6 hours before GMT.
7. Initialize the variable theFuture to Oct 21, 2035 at 7:28:00. theFuture date falls on a Sunday.

Date methods

The Date object provides a number of methods to get and set Date properties.

Date object getter and setter methods.

Method	Description	Example
getDate() ***setDate()***	Gets or sets the day relative to the current set month	let day = new Date(2016, 0, 30); day.getDate(); *// 30* day.setDate(21); *// 30 -> 21*
getDay()	Returns the day of the week (0-6)	let day = new Date(2016, 0, 30); day.getDay(); *// 6 = Saturday*
getFullYear() ***setFullYear()***	Gets or sets the 4 digit year	let day = new Date(2016, 0, 30); day.getFullYear(); *// 2016* day.setFullYear(2017); *// 2016 -> 2017*
getHours() ***setHours()***	Gets or sets the hour (0-23)	let day = new Date(2016, 0, 30, 5, 0); day.getHours(); *// 5* day.setHours(2); *// 5 -> 2*
getMilliseconds() ***setMilliseconds()***	Gets or sets the milliseconds (0-999)	let day = new Date(2016, 0, 1, 5, 20, 10, 250); day.getMilliseconds(); *// 250* day.setMilliseconds(500); *// 250 -> 500*
getMinutes() ***setMinutes()***	Gets or sets the minutes (0-59)	let day = new Date(2016, 0, 30, 5, 20); day.getMinutes(); *// 20*

Method	Description	Example
		day.setMinutes(35); *// 20 -> 35*
getMonth() ***setMonth()***	Gets or sets the month (0-11)	let day = new Date(2016, 0, 30, 5, 20); day.getMonth(); *// 0* day.setMonth(3); *// 0 (Jan) -> 3 (Apr)*
getSeconds() ***setSeconds()***	Gets or sets the seconds (0-59)	let day = new Date(2016, 0, 1, 5, 20, 10, 250); day.getSeconds(); *// 10* day.setSeconds(45); *// 10 -> 45*
getTime() ***setTime()***	Gets or sets the number of milliseconds since Jan 1, 1970, 00:00:00 UTC	let day = new Date(2016, 0, 30, 5, 20); day.getTime(); *// 1454152800000* day.setTime(1454153700000); *// Sat Jan 30 2016 05:35:00 GMT-0600*