6.12 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.

PARTICIPATION ACTIVITY | 6.12.1: Date object constructor. ✓

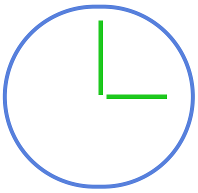
Start □ 2x speed

```
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)
```

Captions ^

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.

Feedback?

PARTICIPATION ACTIVITY | 6.12.2: Date object constructor. ✓

1) The Date constructor must be passed at least one argument.

○ True

● False

✓

Correct

When the Date constructor is passed no arguments, the Date object is initialized to the current date and time.

2) The following code initializes `x` to December 25, 2017.

```
let x = new
Date(2017, 12,
25);
```

○ True

● False

✓

Correct

The month parameter expects a number between 0 and 11. When 12 is specified for the month, the default value of 0 is used, which is January. So `x` is January 25, 2017.

3) The code below displays the same string, regardless of the local time zone.

```
let x = new
Date(2016, 5, 1,
15, 30, 45);
console.log(x);
```

○ True

● False

✓

Correct

The JavaScript interpreter's time zone determines the hour offset from GMT, so a user from California will see "Wed Jun 01 2016 15:30:45 GMT-0700 (Pacific Daylight Time)", and a user from New York will see "Wed Jun 01 2016 15:30:45 GMT-0400 (Eastern Daylight Time)" .

Feedback?

Date methods

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

Table 6.12.1: 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`
		`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`

Method	Description	Example

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`

Feedback?

PARTICIPATION ACTIVITY

6.12.3: Practice with the Date object.

The `notablePeople` map contains a list of some notable individuals and birthdays. Ex: `notablePeople["Elvis Presley"]` contains Elvis' birthday, which is Jan 8, 1935.

1. The first for-in loop displays each person's name and birthday, but the format of the birthday is too wordy. Change the output format to MM/DD/YYYY. Ex: Elvis Presley: 1/8/1935.
2. Add a for-in loop to display each person born before Sonia Sotomayor and the rounded number of days difference in birth day. Ex:
   ```
   Elvis Presley was born 7108 days before Sonia Sotomayor
   Franklin D. Roosevelt was born 26443 days before Sonia Sotomayor
   ...
   ```
   - The getDifferenceInDays() utility function is provided in the code below. When passed a time difference in milliseconds, the function returns the rounded number of days difference. Ex: `getDifferenceInDays(person1Date.getTime() - person2Date.get`
3. Add a for-in loop to display each person born after Sonia Sotomayor and the rounded number of days difference in birth day. Ex:
   ```
   Elon Musk was born 6212 days after Sonia Sotomayor
   Steve Jobs was born 244 days after Sonia Sotomayor
   ...
   ```

```
// Utility function to compute the rounded number of days dif
// time difference in milliseconds
function getDifferenceInDays(timeDifferenceMilliseconds) {
   timeDifferenceMilliseconds = Math.abs(timeDifferenceMillise
   let daysDifference = timeDifferenceMilliseconds / (1000 * 6

   // Return the difference rounded to the nearest whole day
   return Math.round(daysDifference);
}

// Blank line before next section
console.log("");

// Your solution here

```

Run JavaScript

Reset code

Your console output

```
Elvis Presley: Tue Jan 08 1935 00:00:00 GMT-0600 (Central Standard Time)
Sonia Sotomayor: Fri Jun 25 1954 00:00:00 GMT-0500 (Central Daylight Time
Franklin D. Roosevelt: Mon Jan 30 1882 00:00:00 GMT-0550 (Central Standar
Elon Musk: Mon Jun 28 1971 00:00:00 GMT-0500 (Central Daylight Time)
Roger Staubach: Thu Feb 05 1942 00:00:00 GMT-0600 (Central Standard Time)
Steve Jobs: Thu Feb 24 1955 00:00:00 GMT-0600 (Central Standard Time)
Albert Einstein: Fri Mar 14 1879 00:00:00 GMT-0550 (Central Daylight Time
Isaac Asimov: Sat Oct 04 1919 00:00:00 GMT-0500 (Central Daylight Time)
Jada Pinkett Smith: Sat Sep 18 1971 00:00:00 GMT-0500 (Central Daylight T
Grace Hopper: Sun Dec 09 1906 00:00:00 GMT-0600 (Central Standard Time)
```

▼ View solution

Explain

```
--- START FILE: JavaScript ---

const notablePeople = {
   "Elvis Presley":         new Date(1935, 0, 8),
   "Sonia Sotomayor":       new Date(1954, 5, 25),
   "Franklin D. Roosevelt": new Date(1882, 0, 30),
   "Elon Musk":             new Date(1971, 5, 28),
   "Roger Staubach":        new Date(1942, 1, 5),
   "Steve Jobs":            new Date(1955, 1, 24),
   "Albert Einstein":       new Date(1879, 2, 14),
   "Isaac Asimov":           new Date(1919, 9, 4),
   "Jada Pinkett Smith":     new Date(1971, 8, 18),
   "Grace Hopper":           new Date(1906, 11, 9)
};

// Display all names and birthdays
for (let person in notablePeople) {
   let birthday = notablePeople[person];
   console.log(person + ": " + (birthday.getMonth() + 1) +
"/" + birthday.getDate() +
      "/" + birthday.getFullYear());
}

function getDifferenceInDays(timeDifferenceMilliseconds) {
   timeDifferenceMilliseconds =
Math.abs(timeDifferenceMilliseconds);
   let daysDifference = timeDifferenceMilliseconds / (1000 *
60 * 60 * 24);

   // Return the difference rounded to the nearest whole day
   return Math.round(daysDifference);
}

// Blank line before next section
console.log("");

const soniaName = "Sonia Sotomayor";
const soniaTime = notablePeople[soniaName].getTime();

// Display each person born before Sonia Sotomayor
for (let personName in notablePeople) {
   const personDate = notablePeople[personName];
   const timeDifference = personDate.getTime() - soniaTime;
   if (timeDifference < 0) {
      console.log(personName + " was born " +
         getDifferenceInDays(timeDifference) + " days before
" + soniaName);
   }
}

// Blank line before next section
console.log("");

// Display each person born after Sonia Sotomayor
for (let personName in notablePeople) {
   const personDate = notablePeople[personName];
   const timeDifference = personDate.getTime() - soniaTime;
   if (timeDifference > 0) {
      console.log(personName + " was born " +
         getDifferenceInDays(timeDifference) + " days after
" + soniaName);
   }
}


--- END FILE: JavaScript ---
```

Feedback?

CHALLENGE ACTIVITY 6.12.1: Using Date methods. ✓

530096.4000608.qx3zqy7

Jump to level 1

✓ 1 ✓ 2 ✓ 3

Update the variable updatedOn to the month March using Date methods.

```
let updatedOn = new Date(2010, 3, 21);

/* Your solution goes here */
updatedOn.setMonth(2);

```

1 2 3

Check Next

✓

✓ Checking the date stored in updatedOn

Yours 3/21/2010

View your last submission ^

```
/* Your solution goes here */
latestNews.setDate(15);
```

Feedback?

Exploring further:

- Date object (MDN)

How was this section? 👍 👎 Provide section feedback