Lecture

Date & Time

Topics

Number

- **ParseInt**
- **IsNaN**

Math

- **Floor**
- **Ceil**
- **Abs**
- **Round**
- **Random**

Date Object

Date Object

JavaScript has a built-in Date object that stores the date and time and provides methods for handling them.

To create a new instance of the Date object, use the new keyword:

```
const date = new Date();
```

The Date object contains a Number that represents milliseconds passed since the Epoch, that is 1 January 1970.

```
// Mon Aug 10 2019 12:58:21 GMT-0400 (Eastern Daylight Time)
```

Data Object

You can pass a date string to the Date constructor to create an object for the specified date:

```
const date = new Date('Jul 12 2011');
```

To get the current year, use the getFullYear() instance method of the Date object. The getFullYear() method returns the year of the specified date in the Date constructor:

```
const currentYear = date.getFullYear();
console.log(currentYear); //2020
```

Data Object cont..

Getting the date this way is very flexible. All of the examples below return valid Date objects:

```
new Date('2019-06') // June 1st, 2019 00:00:00
new Date('2019-06-16') // June 16th, 2019
new Date('2019') // January 1st, 2019 00:00:00
new Date('JUNE 16, 2019')
new Date('6/23/2019')
```

Get a Date and Time with individual values

The syntax is Date(year, month, day, hour, minute, second, millisecond).

Note that the months are zero-indexed, beginning with January at 0 and ending with December at 11.

```
const specifiedDate = new Date(2019, 4, 29, 15, 0, 0, 0);

// Wed May 29 2019 15:00:00 GMT-0400 (Eastern Daylight Time)
```

EPOCH TIME

Unix time (also known as Epoch time, POSIX time, seconds since the Epoch, or UNIX Epoch time) is a system for describing a point in time.

It is the number of seconds that have elapsed since the Unix epoch, minus leap seconds; the Unix epoch is 00:00:00 UTC on 1 January 1970.

The Unix Epoch is important because it's what JavaScript, Python, PHP, and other languages and systems use internally to calculate the current time.

Get a Date and Time from a timestamp

This represents the time at Thursday, January 1st, 1970 (UTC), or the Unix Epoch time.

```
const unixEpoch = new Date(0);
```

new Date(ms) returns the date of the epoch plus the number of milliseconds you pass in. In a day there's 86,400,000 milliseconds so:

```
const dayAfterEpoch = new Date(86400000);
```

Date Object Methods

Date Object Methods

Similarly, there are methods for getting the current day of the month and the current month:

```
const today = date.getDate();
const currentMonth = date.getMonth() + 1;
```

The getDate() method returns the current day of the month (1-31).

The getMonth() method returns the month of the specified date.

One point to note about the getMonth() method is that it returns 0-indexed values (0-11) where 0 is for January and 11 for December. Hence the addition of 1 to normalize the month's value.

Often you will not need the entire date, but just part of it like the day, week or month. Fortunately there are a number of methods to do just that:

```
const birthday = new Date('6/13/2018 06:27:39');

birthday.getMonth() // 5 (0 is January)
birthday.getDate() // 13
birthday.getDay() // 3 (0 is Sunday)
birthday.getFullYear() // 2018
birthday.getTime() // 1528838859000 (milliseconds since the Unix Epoch)
birthday.getHours() // 6
birthday.getMinutes() // 27
birthday.getSeconds() // 39
birthday.getTimezoneOffset() // -540 (time zone offset in minutes based on
```

Date Object Methods cont..

Date Parse

You can also use the Date.parse() method to return the number of milliseconds since the epoch (January 1st, 1970):

```
Date.parse('1970-01-02') // 86400000
Date.parse('6/16/2019') // 1560610800000
```

Date now

now() is a static method of the Date object. It returns the value in milliseconds that represents the time elapsed since the Epoch.

You can pass in the milliseconds returned from the now() method into the Date constructor to instantiate a new Date object:

```
const timeElapsed = Date.now();
const today = new Date(timeElapsed);
```

Formatting The Date

Date Formatting

You can format the date into multiple formats (GMT, ISO, and so on) using the methods of the Date object.

The toDateString() method returns the date in a human readable format:

```
today.toDateString(); // "Sun Jun 14 2020"
```

The toISOString() method returns the date which follows the ISO 8601 Extended Format:

```
today.toISOString(); // "2020-06-13T18:30:00.000Z"
```

Date Formatting cont..

The toUTCString() method returns the date in UTC timezone format:

```
today.toUTCString(); // "Sat, 13 Jun 2020 18:30:00 GMT"
```

The toLocaleDateString() method returns the date in a locality-sensitive format:

```
today.toLocaleDateString(); // "6/14/2020"
```

Video

Time Zones

What are Time Zones?

A time zone is a region that follows a uniform local time which is legally stated by the country.

It’s common for many countries to have its unique time zone, and some large countries, such as the USA or Canada, even have multiple time zones.

Interestingly, even though China is large enough to have multi time zones, she uses only one time zone. This sometimes results in such an awkward situation where the sun rises around 10:00 AM in the western part of China

Greenwhich Mean Time - GMT

The Korean local time is normally GMT +09:00. GMT is an abbreviation for Greenwich Mean Time, which is the clock time at the Royal Observatory in Greenwich, U.K. located at longitude 0.

The GMT system began spreading in Feb. 5, 1925 and became the world time standard until Jan. 1, 1972.

Universal Time Coordinate - UTC

Many consider GMT and UTC the same thing, and the two are used interchangeably in many cases, but they are actually different. UTC was established in 1972 to compensate for the slowing problem of the Earth's rotation.

This time system is based on International Atomic Time, which uses the cesium atomic frequency to set the time standard. In other words, UTC is the more accurate replacement system of GMT.

Although the actual time difference between the two is tiny, UTC is whatsoever the more accurate choice for software developers.

Time Zone Offsets

The difference of time between UTC standard time and the local time is called “offset”, which is expressed in this way: +09:00, -03:00, etc.

It’s common that countries name their time zone using their own unique names. For example, the time zone of Korea is called KST (Korea Standard Time), and has a certain offset value which is expressed as KST = UTC+09:00.

However, the +09:00 offset is also used by not only Korea but also Japan, Indonesia, and many others

Setting a Time Zone

When passing a date string without setting a time zone, JavaScript assumes the date/time are in UTC before converting it to your browser's time zone:

```
const exactBirthdate = new Date('6/13/2018 06:27:00');

console.log(exactBirthdate) // Wed Jun 13 2018 06:27:00 GMT+0900 (Korean Standard Time)
```

This can lead to errors where the date returned is off by many hours. To avoid this, pass in a time zone along with the string:

```
const exactBirthdate = new Date('6/13/2018 06:27:00 GMT-1000');

console.log(exactBirthdate) // Thu Jun 14 2018 01:27:00 GMT+0900 (Korean Standard Time)
```

You can also pass some, but not all, time zone codes:

```
const exactBirthdate = new Date('6/13/2018 06:27:00 PDT');

console.log(exactBirthdate) // Thu Jun 14 2018 01:27:00 GMT+0900 (Korean Standard Time)
```

Setting a Time Zone cont..

Moment

Moment.js

Make Working with Dates Easier with Moment.js

Getting dates and times right is no small task. Every country seems to have a different way of formatting dates, and accounting for different time zones and daylight savings/summer time takes, well, a whole lot of time. That's where Moment.js shines – it makes parsing, formatting, and displaying dates a breeze.

To start using Moment.js, install it through a package manager like npm, or add it to your site through a CDN. See the Moment.js documentation for more details.