PDP UNIVERSITY

Website development

Lecture 11

Date

Date creates date instance

There are 4 ways to create a date:

```
new Date() // creates current date

new Date(year, month, day, hour, minutes, seconds, milliseconds) //
accepts parameters as numbers

new Date(dateString) // accepts string in different formats. e.g:
'YYYY-MM-DD’

new Date(milliseconds) // accepts milliseconds as number and creates
date starting from 01-01-1970
```

new Date(year, month, day)

Months are counted from 0. January = 0. December = 11. More than 11 goes to the next year

```
const d = new Date(2018, 12, 24); => Thu January 24 2019
```

If you enter day which is more than number of days in a month, it goes to the next month

```
const d = new Date(2018, 5, 35); => Thu Jul 05 2018
```

new Date("dateString")

It can accept different formats

```
const d = new Date("October 13, 2014 11:13:00");
=> Mon Oct 13 2014 11:13:00
```

The most recommended format is from ISO (International Standard Organization): YYYY-MM-DD

```
const d = new Date("2018-03-25"); => Sun March 25 2018
```

new Date(milliseconds)

The count of the date in JS starts from 1st January 1970

When you pass milliseconds it is going to be counter from that date

```
const d = new Date(86400000)
⇒ Fri Jan 02 1970 06:00:00. Because 86400000ms = 1 day
```

Date methods

Component	Get	Set
Year	`getFullYear()`	`setFullYear()`
Month	`getMonth()`	`setMonth()`
Date (of month)	`getDate()`	`setDate()`
Hours	`getHours()`	`setHours()`
Minutes	`getMinutes()`	`setMinutes()`
Seconds	`getSeconds()`	`setSeconds()`
Milliseconds	`getMilliseconds()`	`setMilliseconds()`
Day (of week)	`getDay()`	N/A

toLocale methods

toLocaleString() – makes date more readable

toLocaleDateString() – returns date portion of a date in a more readable representation

toLocaleTimeString() – returns time portion of a date in a more readable representation

Comparing dates

Dates can be compared with >, <, >=, <= signs

E.g.:

```
const d1 = new Date("2018-03-25");
const d2 = new Date("2018-03-26");
console.log(d2 > d1) => true
```

setTimeOut()

setTimeOut is the way to execute some function after specified time:

```
setTimeout(() => {
  console.log("Hello World!");
}, 1000);
```

=> Hello world will be displayed after 1 second (1000ms)

setInterval()

setInterval is the way to execute some function repeatedly with some interval time in between:

```
setInterval(() => {
  console.log("Hello World!");
}, 1000);
```

=> Hello world will be displayed each second

Thank you for your attention