



# Date & Time in JavaScript

<https://jit.team>

front.jit #4 (15.11.2022)

# Today's agenda

- Date & Time
- Intl
- Libraries



Tue Jul 21 2020  
GMT+0100 10:01:14

Date & Time

# Date initialization

- `new Date(0);`
- `new Date(24 * 60 * 24);`
- `new Date("October 13, 2014 11:13:00");`
- `new Date("2022-03-25");` // between March 24 and March 25
- `new Date("2022-03");` // between February 28 and March 01
- `new Date("2022");` // between December 31 and January 01
- `new Date("2022-11-15T12:00:00Z");` // UTC
- `new Date("2022-3-25");` // ⚠
-

# Date initialization

- `new Date(year,month)`
- `new Date(year,month,day)`
- `new Date(year,month,day,hours)`
- `new Date(year,month,day,hours,minutes)`
- `new Date(year,month,day,hours,minutes,seconds)`
- `new Date(year,month,day,hours,minutes,seconds,ms)`

```
new Date(2018, 11, 24, 10, 33, 30, 0); // it's December
```

```
new Date(2018, 15, 24, 10, 33, 30) == new Date(2019, 3, 24, 10, 33, 30) // 🍺
```

# Getters

- `getDate()` – getting day number 🤪
- `getMonth()` – get month number (counted from 0) 🤪
- `getDay()` – get weekday number (counted from 0) 🤪
- `getFullYear()` // yyyy
- `getHours()` // (0-23)
- `getMinutes()` // (0-59)
- `getSeconds()` // (0-59)
- `getMilliseconds()` // (0-999)
- `getTime()` // milliseconds since January 1, 1970
- `getUTC...`

## DDD?

Date Driven Development 🤪

# Getters

- `toDateString()` // 'Tue Nov 15 2022'
- `toISOString()` // '2022-11-15T15:42:14.203Z'
- `toJSON()` // '2022-11-15T15:42:14.203Z'
- `toLocaleDateString()` // '15/11/2022'
- `toLocaleTimeString()` // '16:42:14'
- `toString()` // 'Tue Nov 15 2022 16:42:14 GMT+0100 (Central European Standard Time)'
- `getTimeString()` // '16:42:14 GMT+0100 (Central European Standard Time)'
- `toUTCString()` // 'Tue, 15 Nov 2022 15:42:14 GMT'

# Setters

- setDate()
- setFullYear()
- setHours()
- setMilliseconds()
- setMinutes()
- setMonth()
- setSeconds()
- setTime()
  
- setUTC...

**MUTABILITY** 🤮



# Date object is broken

- Javascript was written in 10 days
- It gets event worse:

<https://codesandbox.io/s/date-time-broken-date-object-samples-tw9zyt?file=/src/App.tsx>

- Why not fix the Date object?



# Temporal

- Answer for broken Date object
- More straightforward API to working with both for time, as well as for dates
- Intuitive naming convention („Plain” / „Zoned”)
- Utility methods, e.g. add, subtract, since, until, equals, with, round, compare
- Data types, e.g. Duration, TimeZone, Calendar
  
- ⚠ Experimental – don't use it production!
- [@js-temporal/polyfill](https://github.com/js-temporal/polyfill)

# General advices (front-end vs back-end)

- Backend should store UTC ISO timestamp (but it depends, e.g. birth date, flight times)
- Frontend should store Date object (just format if needed)

```

    / return;
    Class(_json.ClassOpen);
    Open.Show();

    lein écran
    screenEnabled = function () {
    Class(_json.ClassFullscreen);

    le plein écran
    screenDisabled = function () {
    .removeClass(_json.ClassFullscreen);

    e que la viewport est ouverte
    pen = function () {
    on _$elm.hasClass(_json.ClassOpen);

    après la récupération du HTML
    et via ajax
    (response, textStatus) {

```



Intl

7 / 1

# Intl

- ECMAScript Internationalization API
- Basing on from the user is accessing the application
- Currencies and numbers formating

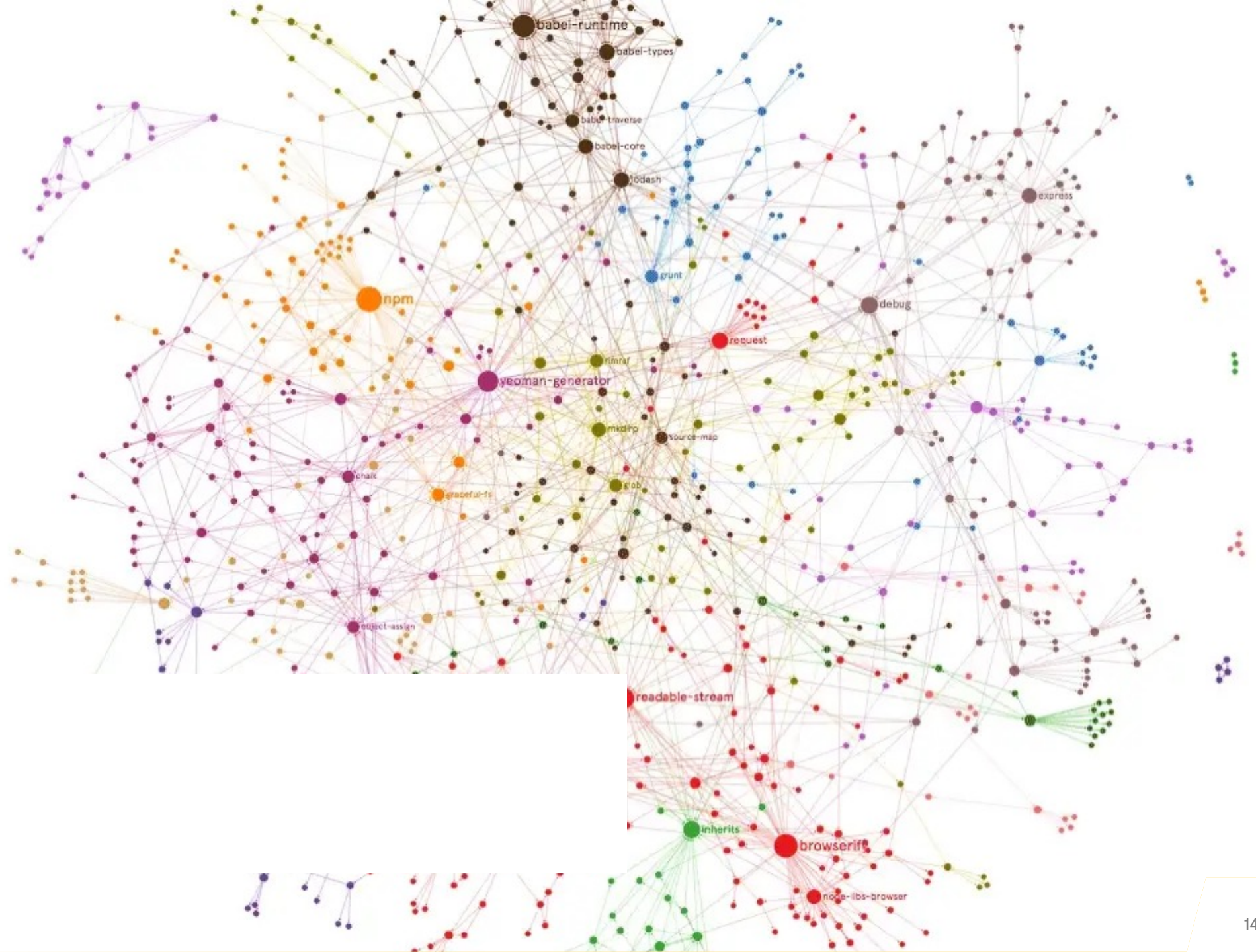
- **Date and time formatting**

<https://codesandbox.io/s/intl-samples-mw982c?file=/src/App.tsx>

- And a lot of more...

<https://egghead.io/blog/understanding-the-intl-javascript-api>





# Libraries

ノ / ト

# Libraries

<u><a href="#">moment</a></u>	<u><a href="#">dayjs</a></u>	<u><a href="#">date-fns</a></u>	<u><a href="#">luxon</a></u>	<u><a href="#">spacetime</a></u>
47k ★	41k ★	30k ★	13k ★	3.7k ★
Biggest bundle (70kb)	2kb	20.4 kb	18 kb	40kb
Mutable, Maintenance mode	Immutable, chainable	Immutable, Typescript support	Immutable, uses Intl API, months are 1-indexed	Immutable, zero deps (no Intl API), Moment-like API
format(), add(), isBefore()	format(), add(), isBefore()	format(), formatDistance(), formatRelative()	isValid(), locale(), tz()	goto(), isAsleep(), hemisphere(),

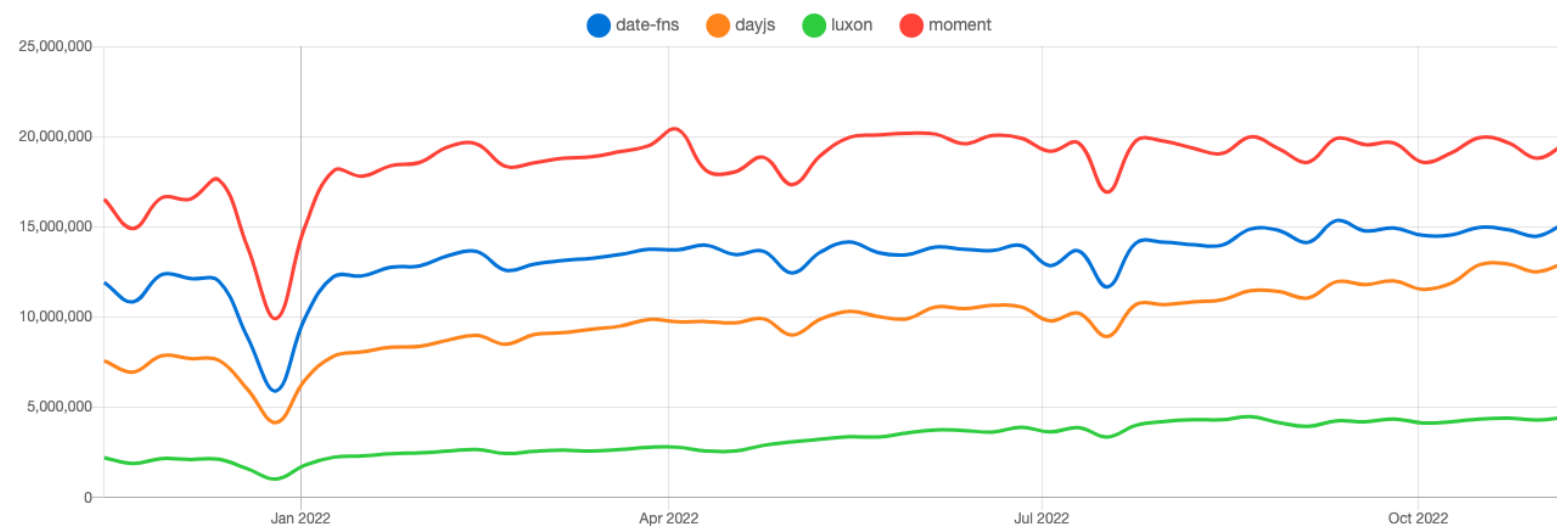
Date Pickers? No libraries at all?

# Libraries













## date-fns vs dayjs vs luxon vs moment

date-fns ×
dayjs ×
luxon ×
moment ×
+ day

Downloads in past 1 Year ▾



### Stats

			Stars	Issues	Version	Updated <sup>?</sup>	Created <sup>?</sup>	Size
<span>■</span>	date-fns	  	30,270	487	2.29.3	2 months ago	8 years ago	minzipped size <span>26.8 KB</span>
<span>■</span>	dayjs	  	41,261	599	1.11.6	25 days ago	5 years ago	minzipped size <span>2.9 KB</span>
<span>■</span>	luxon	  	13,227	125	3.1.0	15 days ago	5 years ago	minzipped size <span>21.3 KB</span>
<span>■</span>	moment	  	47,052	236	2.29.4	4 months ago	11 years ago	minzipped size <span>72.1 KB</span>



# References

## Date & Time:

- [https://www.w3schools.com/jsref/jsref\\_obj\\_date.asp](https://www.w3schools.com/jsref/jsref_obj_date.asp)
- <https://maggiepint.com/2017/04/09/fixing-javascript-date-getting-started/>
- <https://hackernoon.com/dates-in-javascript-are-broken-who-shall-fix-them>
- <https://tc39.es/proposal-temporal/docs/cookbook.html>
- <https://vladmihet.hashnode.dev/temporal-api-javascript-dates-but-better>
- <https://blog.webdevsimplified.com/2022-02/temporal-date-api/>

## Libraries:

- <https://npmtrends.com/date-fns-vs-date.js-vs-day.js-vs-luxon-vs-moment>

## Intl:

- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Intl](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl)
- <https://www.amitmerchant.com/human-readable-date-difference-in-javascript/>
- <https://egghead.io/blog/understanding-the-intl-javascript-api>

# J / 4

Jit Team™  
the human factor of IT



## Date & Time in JavaScript

front.jit #4 (15.11.2022)

<https://jit.team>