



Temporal

The Curious Incident of the Wrong Nighttime

Philip Chimento

 ptomato •  ptomato.name •  mstdn.ca/@ptomato

Igalia, in partnership with Bloomberg

ECMA TC39

- I'm a delegate on the TC39 committee
- TC39 meets 6x / year

TC
39

Bergen

July 2023 TC39 meeting at
the University of Bergen,
Norway

- I stayed there July 10-14



Confirmation

Confirmation dates were **not**
the ones I booked, July 10–14!

HOTEL BOOKING



Confirm Your Booking

July 9 – July 13

Please confirm your booking details.

Confirm Booking



WHAT HAPPENED?

(in the frontend, `TZ=America/Vancouver`)

```
> year = 2023 // from
> month = 7   // date
> day = 10    // picker
```

(in the frontend, `TZ=America/Vancouver`)

```
> year = 2023 // from
> month = 7   // date
> day = 10    // picker
> epochMs = new Date(year, month - 1, day).valueOf()
1688972400000
```

(in the frontend, `TZ=America/Vancouver`)

```
> year = 2023 // from
> month = 7 // date
> day = 10 // picker
> epochMs = new Date(year, month - 1, day).valueOf()
1688972400000
> sendBookingDateToBackend(epochMs)
```

(in the backend, `TZ=Europe/Oslo`)

```
> epochMs = getBookingDateFromFrontend( )
1688972400000
```

(in the backend, `TZ=Europe/Oslo`)

```
> epochMs = getBookingDateFromFrontend( )
1688972400000
> d = new Date(epochMs)
```

(in the backend, `TZ=Europe/Oslo`)

```
> epochMs = getBookingDateFromFrontend( )
1688972400000
> d = new Date(epochMs)
> year = d.getFullYear( )
2023
```

(in the backend, `TZ=Europe/Oslo`)

```
> epochMs = getBookingDateFromFrontend( )
1688972400000
> d = new Date(epochMs)
> year = d.getFullYear( )
2023
> month = d.getMonth( ) + 1
7
```

(in the backend, `TZ=Europe/Oslo`)

```
> epochMs = getBookingDateFromFrontend()
1688972400000
> d = new Date(epochMs)
> year = d.getFullYear()
2023
> month = d.getMonth() + 1
7
> day = d.getDate()
10
```

(in the backend, TZ=Europe/Oslo)

```
> epochMs = getBookingDateFromFrontend()
1688972400000
> d = new Date(epochMs)
> year = d.getFullYear()
2023
> month = d.getMonth() + 1
7
> day = d.getDate()
10
> checkAvailable(year, month, day)
true
```

(in the backend, TZ=Europe/Oslo)

```
> epochMs = getBookingDateFromFrontend()
1688972400000
> d = new Date(epochMs)
> year = d.getFullYear()
2023
> month = d.getMonth() + 1
7
> day = d.getDate()
10
> checkAvailable(year, month, day)
true
> epochMs = d.setHours(0, 0, 0, 0)
1688940000000
> sendConfirmedDateToFrontend(epochMs)
```

(in the frontend again, `TZ=America/Vancouver`)

```
> epochMs = getConfirmedDateFromBackend( )
1688940000000
```

(in the frontend again, `TZ=America/Vancouver`)

```
> epochMs = getConfirmedDateFromBackend( )
1688940000000
> new Date(epochMs).toLocaleDateString( )
'2023-07-09' ← 🥰🥰🥰
```

So did I have a place to sleep on July 13th?

- Yes, I clarified with the hotel 😊
- Could have been worse if I was living east of Norway!

What does this have to do with Temporal?

Temporal is a proposal for enhancing JavaScript with a new, built-in, **strongly typed** date-time library

...and our design goal is to make it difficult to write bugs like this

- "Calendar date with no time" is a different type than "timestamp in milliseconds"!

How would you do this with Temporal?

```
// from date picker:  
> d = Temporal.PlainDate.from({ year: 2023, month: 7, day: 10 })
```

How would you do this with Temporal?

```
// from date picker:  
> d = Temporal.PlainDate.from({ year: 2023, month: 7, day: 10 })  
  
// serialize:  
> s = d.toJSON()  
"2023-07-10"
```

(or `toString()`, depending on your serialization needs)

How would you do this with Temporal?

```
// from date picker:  
> d = Temporal.PlainDate.from({ year: 2023, month: 7, day: 10 })  
  
// serialize:  
> s = d.toJSON()  
"2023-07-10"  
  
// deserialize:  
> d2 = Temporal.PlainDate.from(s)  
> d2.equals(d)  
true  
// No nonsense! True anywhere in the world!
```

What else is good about Temporal?

- Built in to the browser - part of the language!
- Immutable
- Strong support for internationalization

Types: Do more, but also do less

We designed each type so that it's **easy** to do the things you're supposed to do and **difficult** to do the things that you **shouldn't** do



Temporal.Instant

- An exact moment in time*
- No time zone, no summer and winter time
- No calendar
- Data model: elapsed nanoseconds since midnight UTC, Jan. 1, 1970

*disregarding leap seconds, though

Plain types

- ‘Wall’ time
- No time zone, no summer and winter time
- Represent the information you have
- Avoid buggy pattern of filling in 0 for missing info
- Do appropriate calculations based on type

Plain types

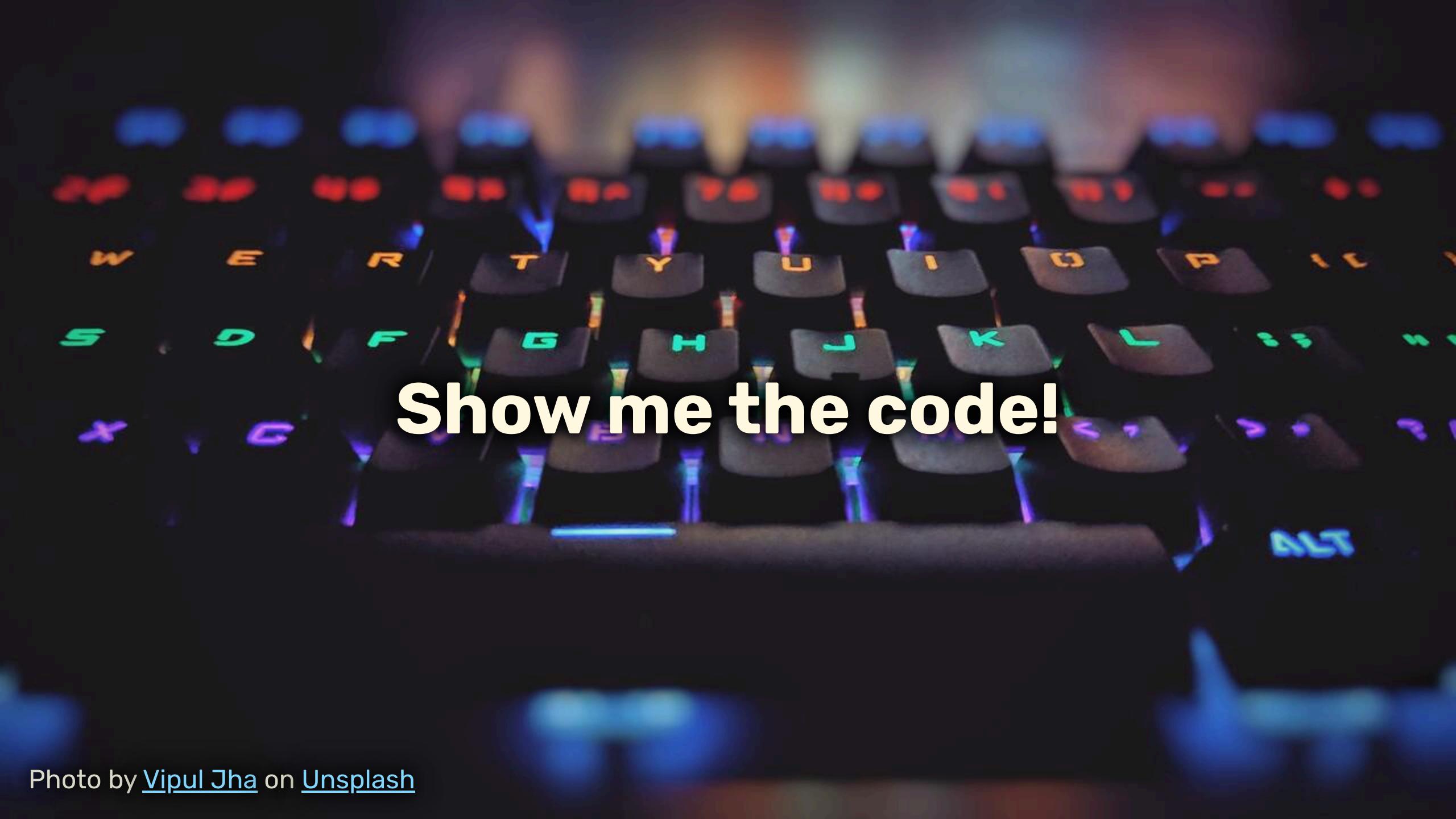
- `Temporal.PlainDate` : “The conference session on June 12th, 2025”
- `Temporal.PlainTime` : “The Q&A is at 17:25”
- `Temporal.PlainYearMonth` : “The June 2025 board meeting”
- `Temporal.PlainMonthDay` : “My birthday is December 15th”

Temporal.ZonedDateTime

- Like `Temporal.Instant`, an exact time
- But also with a calendar and time zone
- Correctly accounts for the time zone's daylight saving rules
- “The conference session is on Thursday, June 12th, 2025, at 4:55 PM, Western European Summer Time”

Temporal.Duration

- returned from other types' `since()` and `until()` methods
- passed to their `add()` and `subtract()` methods



Show me the code!

A worked example

What times are these conference sessions for me?

Asia Pacific
July 23 & 30
11am–3pm
AEST (GMT+10:00)

Europe
July 23 & 30
12pm–4pm
BST (GMT+1:00)

Americas
July 23 & 30
1pm–5pm
EDT (GMT-4:00)

What time is the conference for me?

What we know:

- The two calendar dates that the sessions occur on (July 23 & 30, 2025)
- The start times of the sessions and their time zones (11 AM AEST, noon BST, 1 PM EDT)
- My time zone
- The hours (in my local time) at which I am willing to be at my computer

What time is the conference for me?

- For each session date,
 - For each session,
 - Calculate the exact time of that session start and end on that date;
 - If the local time of the session in my time zone falls within the hours that I'm willing to be at my computer,
 - Print the local time of the session in my time zone.

What time is the conference for me?

```
const dates = [
  Temporal.PlainDate.from('2025-07-23'),
  Temporal.PlainDate.from('2025-07-30')
];
const sessions = [
  { timeZone: 'Australia/Brisbane', hour: 11 },
  { timeZone: 'Europe/London', hour: 12 },
  { timeZone: 'America/New_York', hour: 13 }
];
const sessionLength = Temporal.Duration.from({ hours: 4 });
const myTimeZone = Temporal.Now.timeZoneId();
```

What time is the conference for me?

```
const myCoffeeTime = Temporal.PlainTime.from('08:00');  
const myBedtime = Temporal.PlainTime.from('23:00');  
  
function iCanBeOnlineAt(time) {  
    return Temporal.PlainTime.compare(myCoffeeTime, time) <= 0 &&  
        Temporal.PlainTime.compare(time, myBedtime) <= 0;  
}
```

What time is the conference for me?

```
const formatter = new Intl.DateTimeFormat('en-ca', { dateStyle: 'full', timeStyle: 'short' });
console.log('Put the conference in your calendar:');
dates.forEach((date) => {
  sessions.forEach(({ timeZone, hour }) => {
    const sessionStart = date.toZonedDateTime({ timeZone, plainTime: { hour } });
    const mySessionStart = sessionStart.withTimeZone(myTimeZone);
    const mySessionEnd = mySessionStart.add(sessionLength);
    if (iCanBeOnlineAt(mySessionStart) && iCanBeOnlineAt(mySessionEnd)) {
      console.log(formatter.formatRange(
        mySessionStart.toPlainDateTime(), mySessionEnd.toPlainDateTime())));
    }
  });
});
```

What time is the conference for me?

Put the conference in your calendar:

Tuesday, July 22, 2025, 6:00-10:00 p.m.

Wednesday, July 23, 2025, 10:00 a.m.-2:00 p.m.

Tuesday, July 29, 2025, 6:00-10:00 p.m.

Wednesday, July 30, 2025, 10:00 a.m.-2:00 p.m.

When can I use it?

- <https://caniuse.com/?search=Temporal>
- Firefox unflagged in version 139!
- Safari has some of it available behind a flag in Tech Preview
- Opportunity to get involved shipping in Chrome/V8:
<https://github.com/boa-dev/temporal>

Can I use

Temporal

?

50 results found

Learning more

- MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Temporal
- Temporal Cookbook: <https://tc39.es/proposal-temporal/docs/cookbook.html>

Partnering with Igalia

- **JS:** BigInt, class fields, JSON imports, Temporal, AsyncContext, ...
- **Node.js:** require(esm), compile cache, register hooks, ...
- **Web:** Servo, :has(), CSS grid, MathML, ...
- **And more!** Chromium & WebKit embedding, graphics, RISC-V, ...



igalia
Open Source Consultancy

