

Undefined Academy

Bootcamp de JavaScript Full-stack

Semana 4, Clase 2:

Fechas en JavaScript



Metodología de trabajo

Formato de las sesiones (1h 30m)

- 10 min Revisión de ejercicio anterior
- 60 min Presentación teórica-práctica
- 10 min Preguntas y discusión

Herramientas del Bootcamp

- **Twitch:** <https://undf.sh/en-vivo>
- **Discord:** <https://undf.sh/discord>
- **Notion:** <https://undf.sh/base>

Agenda

- `new Date()`
- `Day.js`
- Temporal API

Revisión de ejercicio anterior



new Date()

new Date()

```
const today = new Date();  
// Muestra la fecha y hora actual del sistema  
console.log(today);
```

```
const birthDate = new Date("1990-05-  
10T08:30:00");  
// Muestra la fecha y hora correspondiente a  
10 de mayo de 1990 a las 8:30 AM  
console.log(birthDate);
```

new Date()

- `getFullYear()` : Devuelve el año de la fecha representada.
- `getMonth()` : Devuelve el mes de la fecha representada (0 para enero, 1 para febrero, etc.).
- `getDate()` : Devuelve el día del mes de la fecha representada (1 para el primer día, 2 para el segundo, etc.).
- `getDay()` : Devuelve el día de la semana de la fecha representada (0 para domingo, 1 para lunes, etc.).
- `getHours()` : Devuelve la hora de la fecha representada (en formato de 24 horas).
- `getMinutes()` : Devuelve los minutos de la fecha representada.
- `getSeconds()` : Devuelve los segundos de la fecha representada.
- `getTime()` : Devuelve el número de milisegundos desde el 1 de enero de 1970 hasta la fecha representada.

Además de estos métodos, también puedes utilizar una serie de métodos para modificar la fecha y hora que representa un objeto `Date`.

`setFullYear()`: Establece el año de la fecha representada.

- `setMonth()`: Establece el mes de la fecha representada.
- `setDate()`: Establece el día del mes de la fecha representada.
- `setHours()`: Establece la hora de la fecha representada.
- `setMinutes()`: Establece los minutos de la fecha representada.
- `setSeconds()`: Establece los segundos de la fecha representada.

Sumando dos fechas

UNIX Timestamp¹

Se trata de un valor entero que representa el número de segundos transcurridos desde el 1 de enero de 1970 a las 00:00:00 UTC

1. También conocido como Epoch time o POSIX time

Sumando dos fechas

```
const today = new Date();  
const TWO_DAYS_IN_MS = 2 * 24 * 60 * 60 *  
1000;  
const twoDaysLater = new  
Date(today.getTime() + TWO_DAYS_IN_MS);  
  
console.log(twoDaysLater);
```

Calculando días entre fechas

```
const yesterday = new Date("2023-03-23T12:00:00");  
const today = new Date("2023-03-24T12:00:00");  
const diffInMilliseconds = today.getTime() -  
yesterday.getTime();  
const diffInDays = Math.floor(diffInMilliseconds  
/ (24 * 60 * 60 * 1000));  
  
console.log(diffInDays);
```

Day.js

¿Por qué usar Day.js?

- Es una librería bastante liviana con soporte de *tree-shaking*.
- Tiene una API intuitiva¹.
- Soporta múltiples idiomas y formatos.
- Permite la manipulación de fechas y horas de manera sencilla.

1. Usa la misma API de Moment la librería más popular de fechas en JavaScript.

Day.js

```
import dayjs from "dayjs"

const today = dayjs();
console.log(today.format());

const birthDate = dayjs("1990-05-10");
console.log(birthDate.format());
```

Day.js

```
import dayjs from "dayjs"

const today = dayjs();
const formatStr = "DD/MM/YYYY HH:mm:ss"
console.log(currentDate.format(formatStr));
```

Lista de los "Parsing tokens"¹

Input	Ejemplo	Descripción
YY	01	Año de dos dígitos
YYYY	2001	Año de cuatro dígitos
M	1-12	Mes, comenzando en 1
MM	01-12	Mes, con 2 dígitos
MMM	Jan-Dec	Nombre abreviado del mes
MMMM	January-December	Nombre completo del mes
D	1-31	Día del mes
DD	01-31	Día del mes, con 2 dígitos

1. <https://day.js.org/docs/en/parse/string-format#list-of-all-available-parsing-tokens>

Day.js

1. Sumar días a una fecha
2. Restar horas a una fecha
3. Obtener la diferencia en días entre dos fechas
4. Obtener el día de la semana de una fecha
5. Redondear una fecha al inicio del día
6. Tiempo relativo

Temporal API

Temporal API

Temporal es una nueva API para trabajar con fechas y horas en JavaScript que está en proceso de estandarización en el W3C¹.

Esta API tiene como objetivo resolver algunas de las limitaciones y problemas de diseño del objeto Date integrado en JavaScript, ofreciendo una sintaxis más intuitiva y consistente, y un conjunto de funcionalidades más completo y preciso.

1. World Wide Web Consortium

#

Temporal - OTHER



A modern API for working with date and time, meant to supersede the original `Date` API.

Current aligned

Usage relative

Date relative

Filtered

All



Chrome	Edge *	Safari	Firefox	Opera	IE
4-110	12-110	3.1-16.2	2-110	10-94	6-10
111	111	16.3	111	95	11
112-114		16.4-TP	112-113		

Temporal.Now

```
console.log('Initialization complete',
Temporal.Now.instant());
// example output:
// Initialization complete 2021-01-
13T20:57:01.500944804Z
```

Temporal.Instant¹

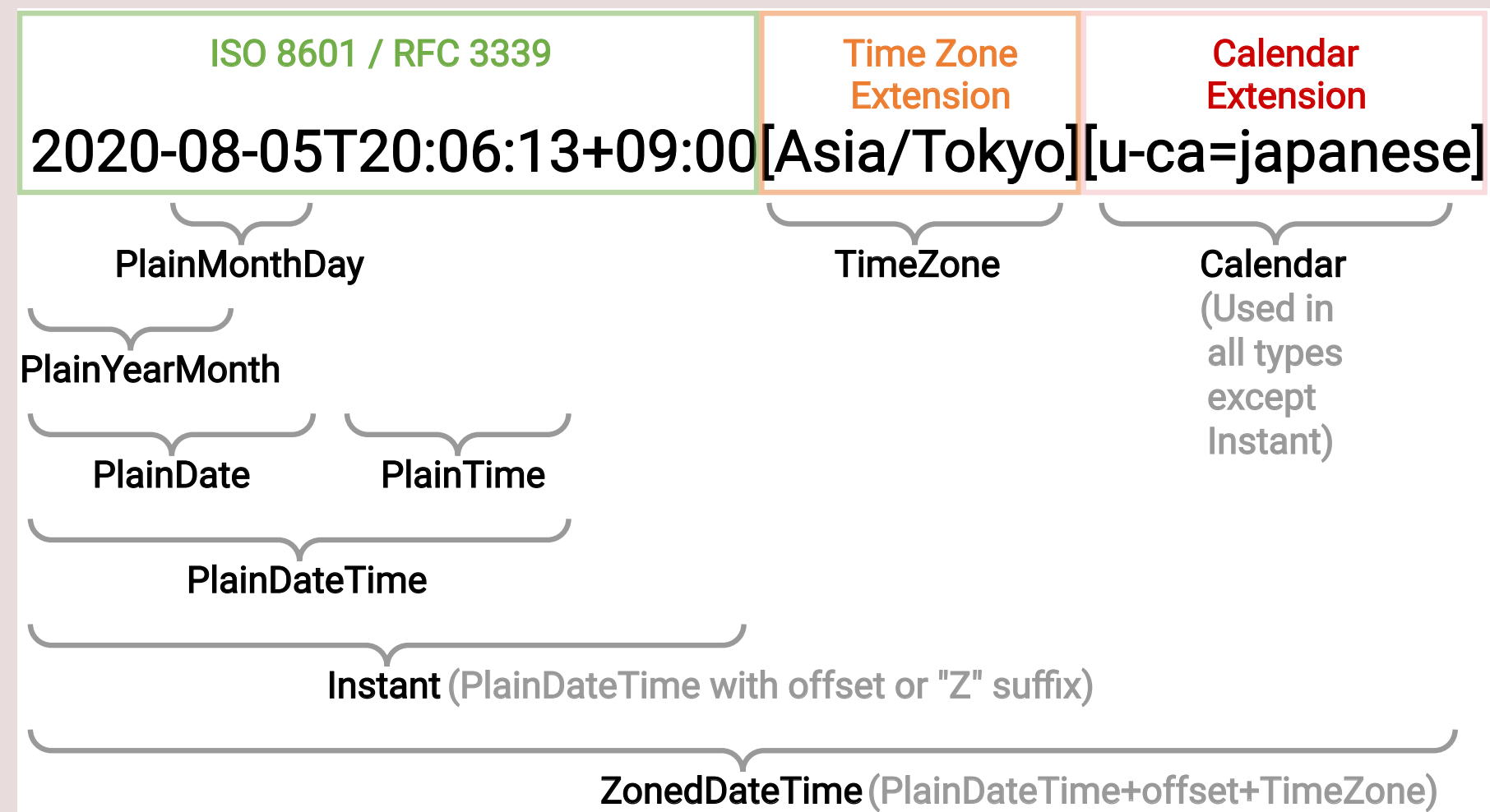
```
const instant = Temporal.Instant.from('1969-07-
20T20:17Z');
instant.toString(); // => '1969-07-20T20:17:00Z'
instant.epochMilliseconds; // => -14182980000
```

Temporal.ZonedDateTime²

```
const zonedDateTime =
Temporal.ZonedDateTime.from({
  timeZone: 'America/Los_Angeles',
  year: 1995,
  month: 12,
  day: 7,
  hour: 3,
  minute: 24,
  second: 30,
  millisecond: 0,
  microsecond: 3,
  nanosecond: 500
}); // => 1995-12-07T03:24:30.0000035-
08:00[America/Los_Angeles]
```

1. Un Temporal.Instant representa un punto fijo en el tiempo (llamado "tiempo exacto"), sin tener en cuenta el calendario o la ubicación, p. 20 de julio de 1969 a las 20:17 UTC.
2. Un Temporal.ZonedDateTime es un objeto de fecha/hora consciente de la zona horaria y del calendario que representa un evento real que sucedió (o sucederá) en un momento exacto particular desde la perspectiva de una región particular de la Tierra, p. 7 de diciembre de 1995 a las 3:24 a. m. en hora del Pacífico de EE. UU. (en calendario gregoriano).

Persistencia de cadena de texto¹



1. Todos los tipos temporales tienen una representación de cadena de texto para persistencia e interoperabilidad.

Recursos

- <https://lenguajejs.com/javascript/fechas/date-fechas-nativas/>
- <https://day.js.org/>
- <https://tc39.es/proposal-temporal/docs/>
- <https://tc39.es/proposal-temporal/docs/cookbook.html>
- <https://www.youtube.com/watch?v=-5wpm-gesOY>

Ejercicio

Crear una función que dada una fecha nos permita "humanizarla" con las siguientes reglas:

1. Si la fecha es menos de un mes, mostrarla de manera humana, por ejemplo `Hace 2 minutos`, `Hace 4 días`, `Hace 2 semanas`.
2. Si la fecha es más de un mes, mostrar mes y día, por ejemplo `Febrero 24`, `Enero 2`.
3. Si la fecha no es el mismo año, mostrar la fecha completa: `Noviembre 20, 2022`.

¿Preguntas?