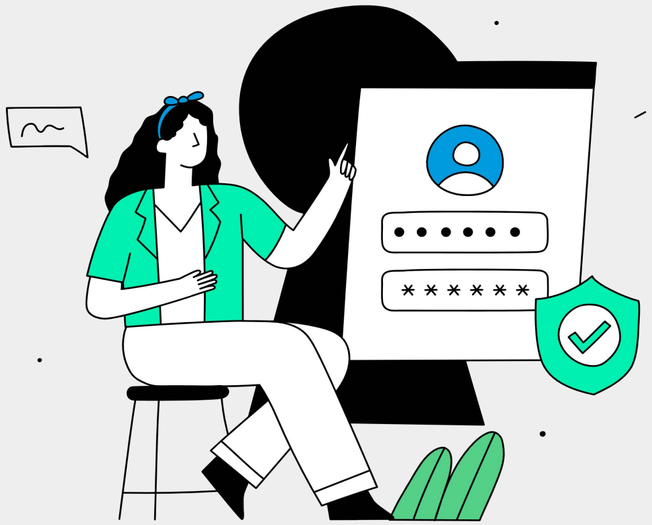
¡Javascript!

guayerd beta hub

guayerd

Agenda del día



02

Asincronía

Definición.

setTimeout.

Promises

Definición.

Estados.

Firma.

Resolve.

Reject.

Then.

Catch.

Finally.

01

Ejercitación

Simular el tiempo de carga de una API.

guayerd

daily

¿Cómo venimos?

¿Algo nos bloquea?

¿Cómo seguimos?

guayerd

Asincronía

A veces hay que esperar...

guayerd

SPRINT 3

Asincronía

Javascript es un lenguaje de un solo subproceso. Sólo puede realizar una tarea a la vez, secuencialmente.

Si hay procedimientos que son difíciles de resolver la aplicación puede bloquearse o demorarse.

guayerd

Javascript

```
console.log("Antes loop")

const loop = () ⇒ {
   Array.from(Array(100000000).keys()).forEach(
      i ⇒ (i / 5) * 4 ** 19000896
   )
}

loop()

console.log("Después loop")
```

Asincronía

Cómo Javascript se ejecuta en el navegador. Javascript puede delegar al navegador las tareas asincrónicas. Cuando estas se resuelvan en un momento posterior, Javascript recibirá el resultado.

guayerd

setTimeout

Este método del objeto window permite retrasar la ejecución de una tarea determinando el tiempo de demora exacto en milisegundos. Esto evitará bloqueos.

```
console.log("Antes loop")

const loop = () ⇒ {
   setTimeout(() ⇒ {
      Array.from(Array(100000000).keys()).forEach(
         i ⇒ (i / 5) * 4 ** 19000896
      )
   }, 0)
}

loop()

console.log("Después loop")
```

guayerd

Qué es una Promesa?


TE PROMETO DARTE EL DINERO.
makeameme.org

guayerd

Promesas en el mundo real

Un familiar te pide 10.000 dólares y se compromete a devolverlo en unos meses.

Somos chevere y colaboramos.

Pasados 2 años, pueden pasar 2 cosas (simplificando).


SMILE
SMILEN'T

Promise

La API Promise permite realizar operaciones asincrónicas con un tiempo de espera indeterminado. Se resolverá o se rechazará cuando termine su trabajo, sea cuando sea.

guayerd

Estados de una promesa

Pending (pendiente)

Este estado ocurre entre que se crea la promesa y se resuelve. Le presto la plata a mi familiar y comienza mi espera a que la devuelva.

Fulfilled (completada)

Cuándo la promesa se resuelve correctamente. Mi familiar me pagó.

Rejected (rechazada)

Cuándo la promesa se rechaza. Mi familiar no me pagó.

guayerd

Crear una promesa

Se instancia con **new** y recibe una función callback como argumento. A la vez, la función callback tiene 2 parámetros llamados **resolve** y **reject**. Resolve y reject son funciones que entregan el resultado de la promesa dependiendo si se completó o rechazó:

guayerd

Promesa completada

```
const myPromise = new Promise((resolve, reject) ⇒ {
   resolve("resuelta")
})
```

guayerd

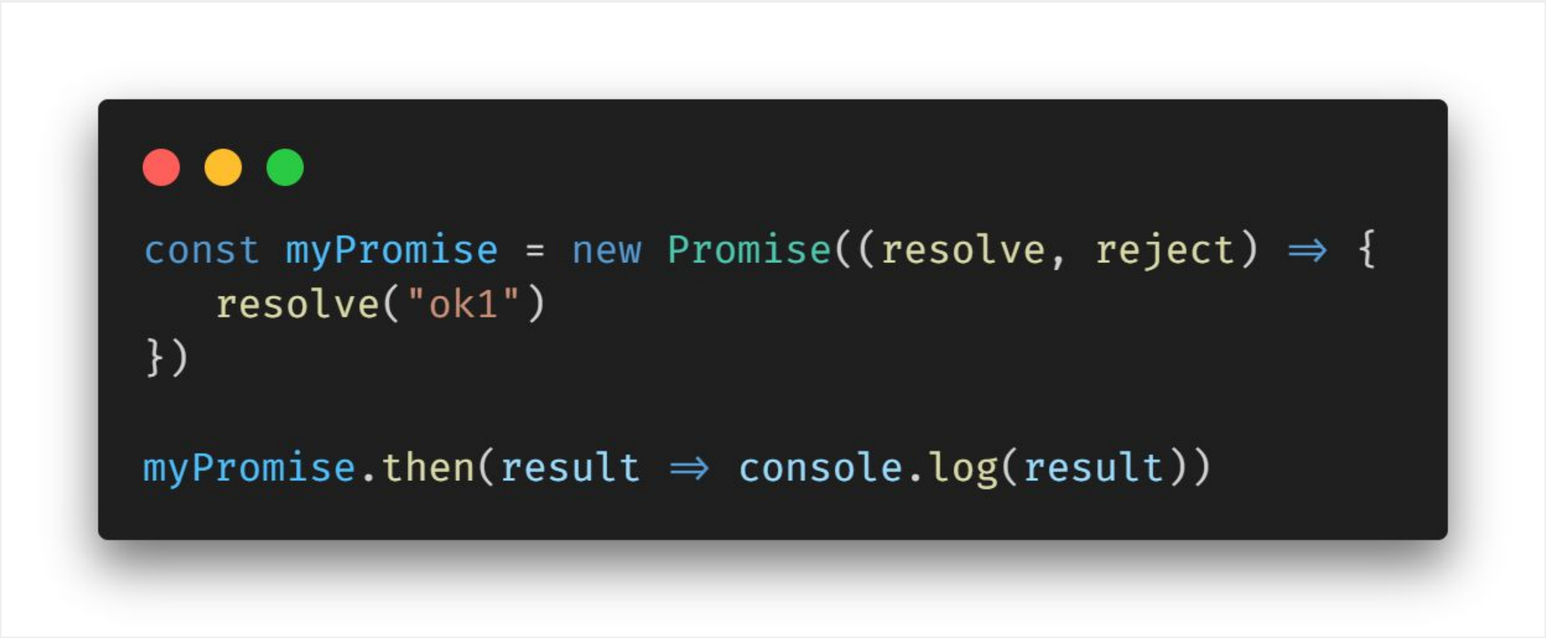
Promesa rechazada

```
const myPromise = new Promise((resolve, reject) ⇒ {
    reject("rechazada")
})
```

guayerd

then

Las promesas no devuelven el resultado por sí mismas. En el caso de completarse (me pagó), debemos encadenar un método then que obtendrá el resultado asincrónico.



```
const myPromise = new Promise((resolve, reject) ⇒ {
   resolve("ok1")
})

myPromise.then(result ⇒ console.log(result))
```

guayerd

catch

En el caso de rechazarse (no me pagó), debemos encadenar un método catch que obtendrá el resultado asincrónico con el error.

```
const myPromise = new Promise((resolve, reject) => {
   const date = new Date()
   const year = date.getFullYear()

   if (year === 2023) {
      resolve("Es 2023")
   } else {
      reject("No es 2024")
   }
})

myPromise.catch(error => console.log({ error }))
```

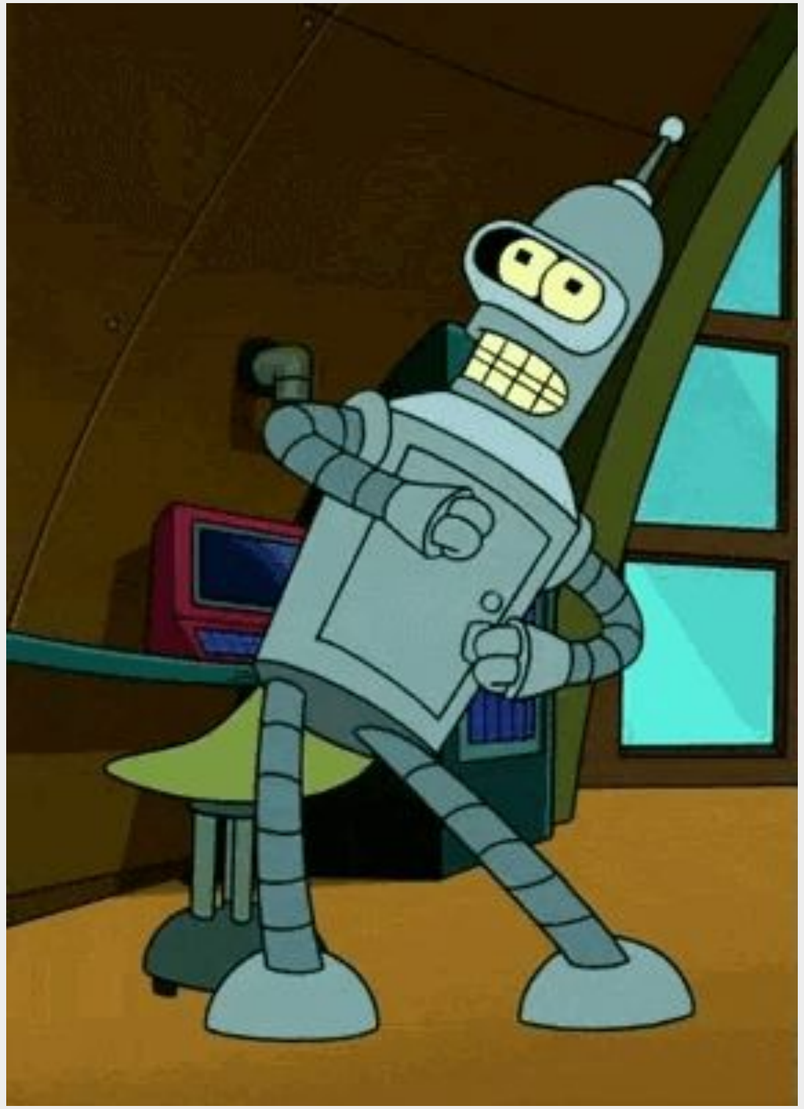
guayerd

finally

Los métodos pueden encadenarse, cómo último podemos poner el método finally, que se ejecutará sin importar si la promesa se completa o rechaza.

```
const myPromise = new Promise((resolve, reject) ⇒
   resolve("ohh dior mio")
)

myPromise
   .then(result ⇒ console.log(1, result))
   .catch(error ⇒ console.log(2, error))
   .finally(() ⇒ console.log("💔 Me amo"))
```



guayerd

¡Manos a la obra!

guayerd

SPRINT 3

Promise

Entregable

En la Home, en dónde iba la grilla de cards poner inicialmente un loader o spinner sin grilla. Crear una promesa, con un setTimeout de 3 segundos dentro, que se resuelva. Al tomar el resultado reemplazar el spinner por la grilla de cards.

guayerd

retro

¿Cómo nos fué?

¿Qué cosas no quedaron claras y necesitamos repasar la próxima?