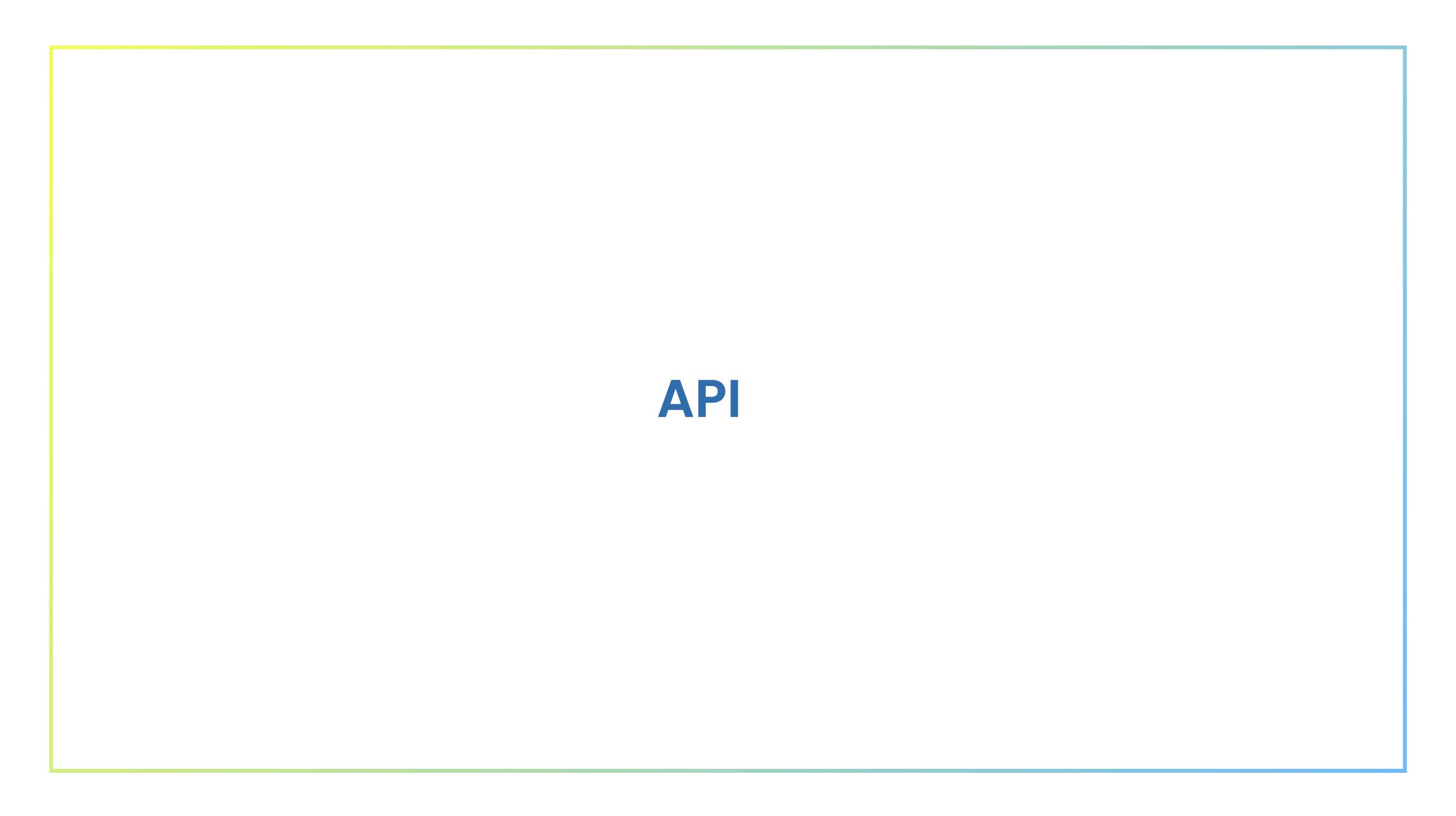


Índice

API

Fetch

Async y Await



API

https://jsonplaceholder.typicode.com/

race

myfakeapi.com

API

```
Listar => GET /todos/1
 Crear => POST /todos
Modificar => PUT /todos/1
Borrar => DELETE /todos/1
```



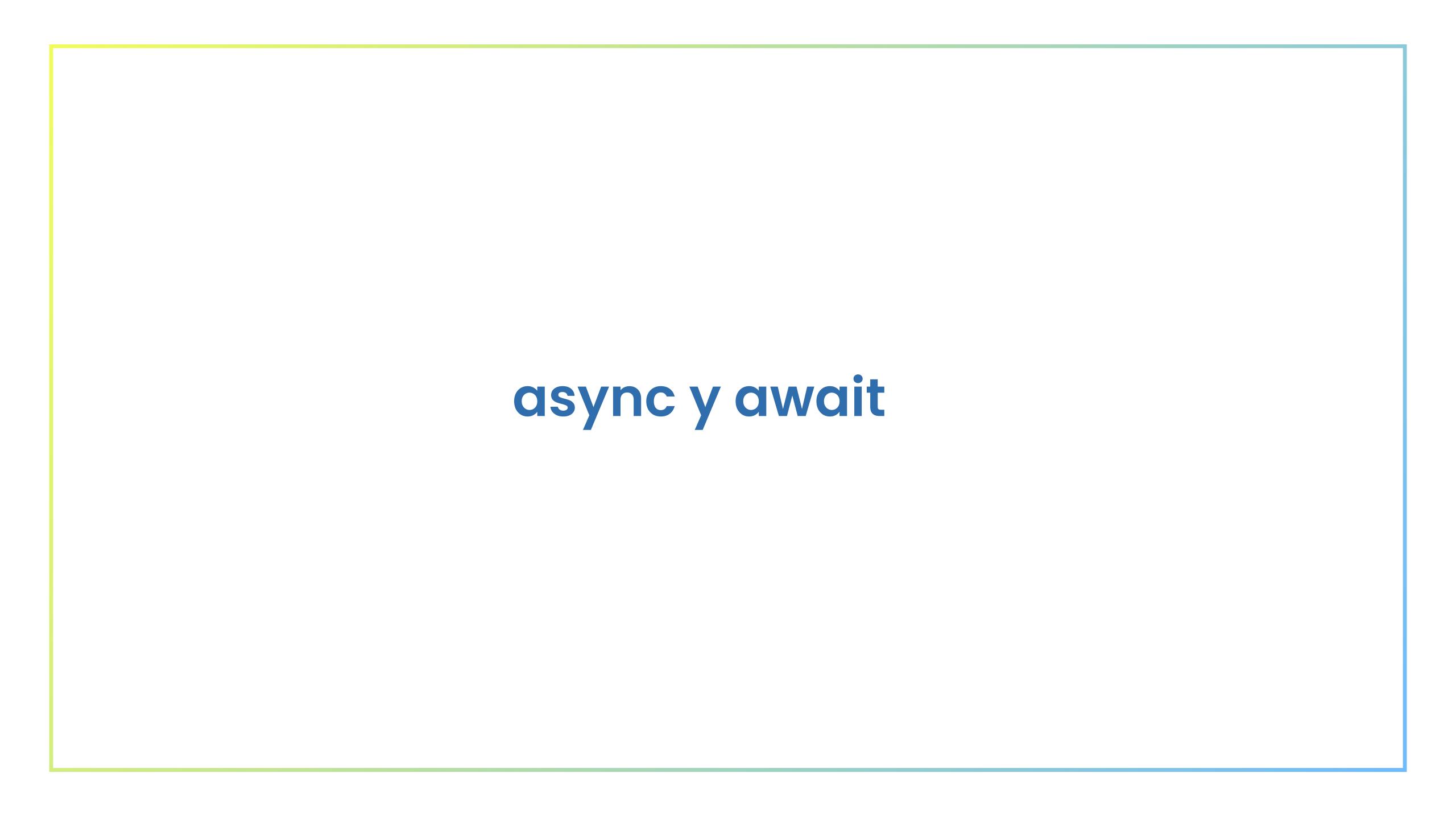
Fetch será utilizado para pedir o enviar datos a un servidor desde nuestro código javascript

Fetch siempre devuelve una promesa resuelta (cuidado con esto)

Fetch necesita dos parámetros url y un objeto con opciones que veremos posteriormente

Podemos usar then, catch y todo lo aprendido con las promesas

Hagamos nuestro primer fetch



La declaración de <u>función async define una función</u> <u>asíncrona que devuelve un objeto, lo</u> <u>cual permite a un programa correr</u> <u>una función sin congelar todo la</u> compilación

Dada que la finalidad de las funciones async/await es simplificar el comportamiento del uso síncrono de promesas, se hace más fácil escribir mesas.

Nos evita tener que encadenar con .then si no que usamos un sistema más tradicional y lógico ante nuestros ojos

"Bloqueamos" hasta que son resueltas para continuar dentro de una función

Lo que hace await es detener la ejecución y no continuar. Se espera a que se resuelva la promesa, y hasta que no lo haga, no continua. A diferencia del .then(), aquí tenemos un código queante.

Sobre todo viene a resolver el callback hell

Probemos

C Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez