

wawiwa

Await

Async + await

Async y await hacen que las promesas sean más fáciles de escribir:

- async hace que una función retorne una Promise
- await hace que la función espere a la Promise

Mira el ejemplo:

https://www.w3schools.com/js/tryit.asp?filename=tryjs_async2

Await

El operador “await” es usado para esperar a una Promise.

Solamente puede ser usado dentro de una función async con código Javascript regular; sin embargo, puede ser usado por sí misma en módulos de Javascript.

La expresión await cause que la ejecución de la función async pause hasta que la Promise sea resuelta (ya sea cumplida o rechazada), y reanuda la ejecución de la función async después de la resolución. Cuando se reanuda, el valor de la expresión await es el valor de la promesa cumplida (fulfilled Promise).

Si la Promise es rechazada, la expresión await lanza el valor rechazado.

Si el valor de la expresión después del operador await no es una Promise, es convertido a una promesa resuelta.

Un “await” divide el flujo de la ejecución, permitiendo al llamador de la función async que reanude la ejecución. Después An await splits execution flow, allowing the caller of the async function to resume execution. Después de que el `await` retrase la continuación de la función asíncrona, la ejecución de las declaraciones posteriores continúa. Si este “await” es la última expresión ejecutada por su función, la ejecución continúa retornando al llamador de esta función una promesa pendiente “pending Promise” para la finalización de la función “await” y la reanudación de la ejecución del llamador.

Await

Esperando a que una promesa sea resuelta.

Si una Promise es pasada a una expresión “await”, espera a que la Promise sea resultada y retorne el valor del resultado.

```
function resolveAfter2Seconds(x) {  
  return new Promise(resolve => {  
    setTimeout(() => resolve(x), 2000);  
  });  
}  
  
async function f1() {  
  const x = await resolveAfter2Seconds(10);  
  console.log(x); // 10  
}  
  
f1();
```

Await

```
const delay = seconds => {  
  return new Promise(  
    resolve => setTimeout(resolve, seconds * 1000)  
  )};  
  
const countToFive = async() => {  
  console.log("one");  
  await delay(1);  
  console.log("Two");  
  await delay(1);  
  console.log("three");  
  await delay(1);  
};  
  
countToFive();
```

5

A nighttime photograph of a city skyline. The central focus is a tall, modern skyscraper with a grid-like facade, illuminated with blue and white lights. To its right, another building with a curved, glass facade is visible. In the foreground, a red speech bubble with the word 'wawiwa' in white lowercase letters is positioned. The background shows other city buildings and streetlights, creating a vibrant urban scene.

wawiwa

¿Preguntas?