

Promesas

Una promesa es un objeto que representa el resultado de una operación asíncrona.

Una promesa es algo que en un futuro se cumplirá o se rechazará.

Al contrario de los callbacks, las promesas mantienen la legibilidad del código.

Estados de las promesas

- Pendiente
- Aceptada
- Rechazada

| Código | Qué hace y cuándo lo hace? |
|------------|---|
| Productor | Va producir un resultado.....pero ese resultado va tomar tiempo |
| Consumidor | Va a consumir el resultado del Productor....cuando este listo |

Promesas

Producir una promesa

```
let miPromesa = new Promise(function(resolve, reject) {  
  // Código que va producir una respuesta y esta respuesta tardará  
  // dentro de esta funcion tendremos que llamar a resolve o reject  
  setTimeout(() => resolve("exito"), 1000); // Promesa resuelta, llamamos a resolve()  
  
  setTimeout(() => reject(new Error("Error")), 1000); // Promesa rechazada llamamos a reject()  
});
```

| Propiedades | new Promise() | resolve(value) | reject(error) |
|-------------|---------------|----------------|---------------|
| state: | pending | fulfilled | rejected |
| result: | undefined | value | error |

| Métodos | Descripción |
|-----------------------|---|
| .then(resolve) | Ejecuta el callback resolve cuando la promesa se cumple |
| .catch(reject) | Ejecuta el callback reject cuando la promesa se rechaza |
| .then(resolve,reject) | Los dos anteriores en el mismo .then() |
| .finally(end) | Ejecuta el callback end independientemente de si fue cumplida o rechazada |

Promesas

Consumir una promesa

El método `.then()` de la promesa tiene dos parámetros que son funciones una para manejar la aceptación y en otra se maneja el rechazo.

```
miPromesa.then(  
  function(result) { console.log(result) },  
  function(error) { console.log(error.message) }  
);
```

Podemos simplificar y usar funciones flecha y el código queda mas limpio:

```
promesa.then(  
  (result) => { console.log(result) },  
  (error)  => { console.log(error.message) }  
);
```

Como en este caso se recibe solo un parámetro y la función tiene solo una instrucción los paréntesis y corchetes de la función flecha son opcionales.

```
promesa.then(  
  result => console.log(result),  
  error  => console.log(error.message)  
);
```

Promesas

Funcionamiento de la promesa

1. Al llamar a `new Promise()` se corre automáticamente el código
2. La función que se le pasa a `new Promise()` es la que va producir algo
3. `resolve` y `reject` son callbacks
4. Dentro del código hay que llamar a `resolve()` si se tuvo éxito o a `reject()` para rechazar la promesa

Código que produce una promesa

```
let miPromesa = new Promise(function(resolve, reject) {  
  //Código que va producir una respuesta en el futuro  
  
  resolve("exito");    // Promesa resuelta, llamamos a resolve()  
  
  reject("Error");     // Promesa rechazada llamamos a reject()  
  
});
```

Código que consume la promesa

```
miPromesa.then(  
  function(result) { console.log(result) },  
  function(error) { console.log(error.message) }  
);
```