Aridany de la Fe Sarmiento

API

Application Programming Interface: Conjunto de reglas que permiten la transferencia de datos entre dos webs, aplicaciones, etc. Consumo del backend mediante el frontend.

¿CON QUÉ LLAMAMOS A LA API?:

Callback: Funciones que se pasan por parámetros a otras funciones.

example.js

```
function action() {
  console.log("He ejecutado la función");
}
setTimeout(action, 2000);
```

Promesas: Tipo especial de objetos que permiten realizar tareas asíncronas, permite manejar una secuencialidad de acciones para poder consumir APIS.

example.js

```
let promise = new Promise(function(resolve, reject) {
  // la función se ejecuta automáticamente cuando se construye la promesa

  // después de 1 segundo, indica que la tarea está hecha con el resultado "hecho"
  setTimeout(() => resolve("hecho"), 1000);
});

```


🔗 **Sergie Code**
🔗 **Promesas en JavaScript: By Manz**
🔗 **Developer Mozilla**
🔗 **Chalk**
🔗 **Javascript.info**

Fetch(): Es una de las formas que tenemos para realizar peticiones http (url) asíncronas. Siempre te devuelve una promesa.

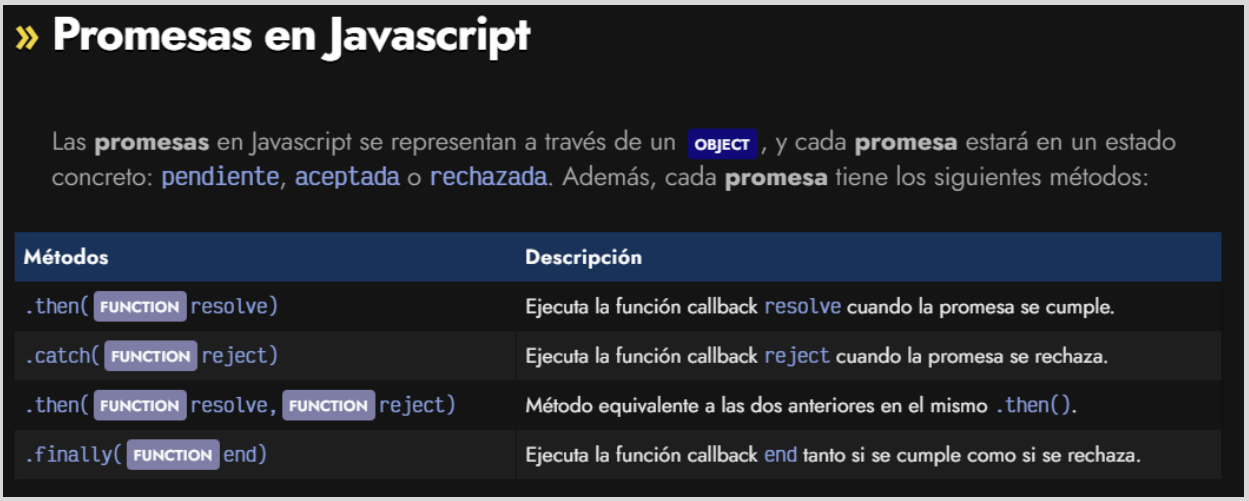
example.js

```
fetch("/robots.txt").then(function(response) {
  /* Código a realizar cuando se cumpla la promesa */
});
```

Then(): La función then() sirve para resolver las promesas generadas por Fetch() cuando se cumplen.

Catch(): Ejecuta la función callback cuando la promesa es rechazada, retorna una Promise y solo se ejecuta en los casos en los que la promesa se marca como Reject.

Finally(): Ejecuta la función sin importar cual sea el resultado (Rechazo o éxito).



🔗 **Sergie Code**
🔗 **Promesas en JavaScript: By Manz**
🔗 **Developer Mozilla**
🔗 **Chalk**
🔗 **Javascript.info**

Aridany de la Fe Sarmiento

ASINCRONÍA

Decimos que nuestro código es asíncrono cuando tiene un retardo en la respuesta. Se usa cuando llamamos a la API y queremos que la respuesta sea enviada en "x" cantidad de tiempo.

Async: Función que detecta que esa parte del código es asíncrona.

Await: Espera a la respuesta de la función async y la ejecuta.

A TENER EN CUENTA:

JSON: JavaScript Object Notation. Formato ligero de intercambio de datos entre servidor / cliente o distintas partes de la web.
{"name":"John"}

🔗 **Sergie Code**
🔗 **Promesas en JavaScript: By Manz**
🔗 **Developer Mozilla**
🔗 **Chalk**
🔗 **Javascript.info**