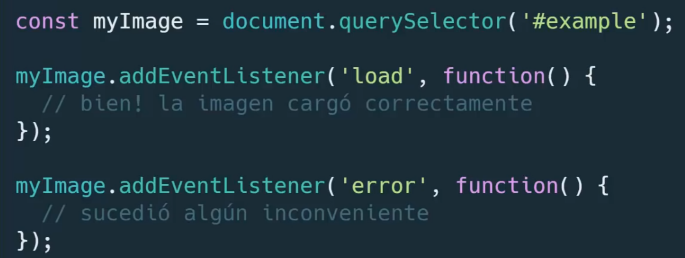
# Promesas (promises)

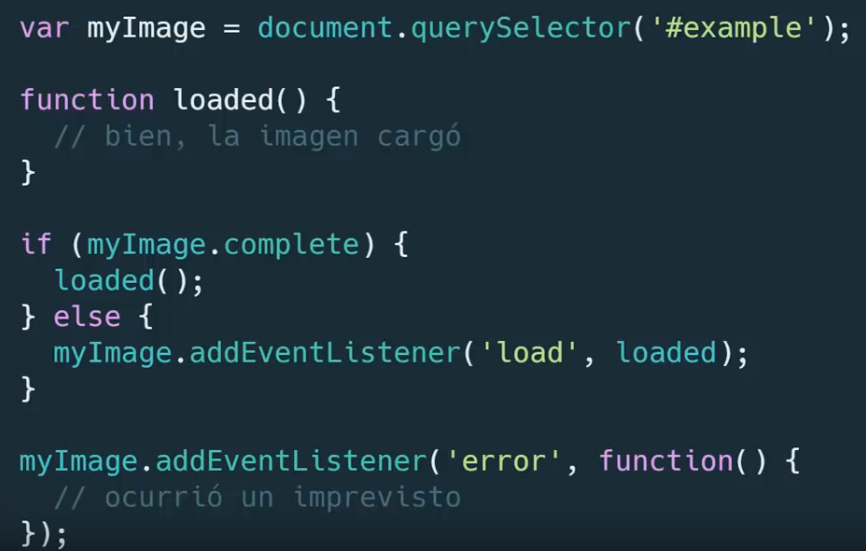
JavaScript es single thread, es decir solo puede tener un hilo de ejecución (en otros lenguajes como Python, C#, etc. es posible hacer varias tareas en simultaneo). Es decir, si tenemos una serie de instrucciones, estas se van a ejecutar uno luego de la otra.

Para solventar esto existen algunos conceptos / técnicas que nos van a permitir obtener un comportamiento similar.

Veamos un ejemplo. Antes de que surgiera el concepto de promesa si queríamos hacer algo cuando una imagen se cargó correctamente haríamos algo parecido a lo siguiente:

En la primera línea nos traemos la referencia del elemento de la imagen, capturamos el evento de *load* si cargo correctamente se va a llamar la función callback (el segundo parámetro de *addEventListener*) y lo mismo hacemos con el evento de *error* si ocurrió algún error.

Si bien este código parece no tener problemas, cuando nosotros registremos el evento *load* probablemente la imagen ya se haya cargado, por lo cual lo correcto seria preguntar por la propiedad *complete* de la imagen y llamar a la función correspondiente, de la siguiente manera.



Este código no presenta problemas, pero si tenemos varias imágenes se volvería muy engorroso, por lo cual utilizar eventos y callbacks no es la mejor opción.

Lo mejor sería tener algo más parecido a esto (expresado en seudocódigo):



Esto es lo que hacen las promesas, pero utilizando una terminología más breve (que va a depender del lenguaje).

## Promesas en JavaScript

Las promesas se incorporan de forma nativa en EcmaScript 6 (si bien anteriormente había librerías de terceros.

Una promesa es un objeto que representa un valor que:

* Puede que esté disponible ahora
* Puede que esté disponible en el futuro
* Puede que nunca lo esté.

Por ejemplo: una petición *get* a un servidor que nos devuelva un JSON, un acceso a base de datos o al localstorage.

Utilizamos las promesas cuando el acceso al recurso puede llegar a demorar un tiempo (lo que provocaría que el proceso se quede esperando una respuesta), es decir **las promesas nos van a permitir manejar flujos de datos asíncronos**

Las promesas nos aseguran que cuando el objeto este disponible se va a ejecutar el código que indiquemos

## Características de las promesas