

## ASYNCHRONOUS JAVASCRIPT AND NETWORKING

Title	Descripción	Enlace
Timeout y setInterval	Explicación de cómo se trabaja con timeout y setInterval en javascript para planificar tareas.	<a href="https://es.javascript.info/settimeout-setinterval">https://es.javascript.info/settimeout-setinterval</a>
Promises	Explicación guiada con ejercicios sobre cómo se crean promesas y cómo se utilizan.	<a href="https://www.codecademy.com/courses/learn-intermediate-javascript/lessons/promises/exercises/introduction">https://www.codecademy.com/courses/learn-intermediate-javascript/lessons/promises/exercises/introduction</a>
Async and await	Explicación guiada con ejercicios sobre cómo hacer el mismo sistema de promesas, pero con async y await.	<a href="https://www.codecademy.com/courses/learn-intermediate-javascript/lessons/async-await/exercises/introduction">https://www.codecademy.com/courses/learn-intermediate-javascript/lessons/async-await/exercises/introduction</a>
Fetch	Explicación sobre cómo se trabaja con otras páginas web o servidores mediante la instrucción fetch.	<a href="https://es.javascript.info/fetch">https://es.javascript.info/fetch</a>
<b>OPCIONAL:</b> promises II	Explicación extra sobre las promesas, de otra manera a la anterior.	<a href="https://es.javascript.info/promise-basics">https://es.javascript.info/promise-basics</a>
<b>OPCIONAL:</b> promises encadenadas	Explicación sobre cómo encadenar promesas, es interesante, pero viendo cómo se trabaja con async y await no es necesario.	<a href="https://es.javascript.info/promise-chaining">https://es.javascript.info/promise-chaining</a>
<b>OPCIONAL:</b> async and await	Otra explicación extra sobre cómo se trabaja con async y await.	<a href="https://es.javascript.info/async-await">https://es.javascript.info/async-await</a>
<b>OPCIONAL:</b> fetch abort	Para profundizar, cómo abortar una petición fetch si está tardando mucho o en otros casos.	<a href="https://es.javascript.info/fetch-abort">https://es.javascript.info/fetch-abort</a>
<b>OPCIONAL:</b> query params	Para profundizar, más explicación sobre cómo se pueden montar las peticiones fetch con parámetros de servidor.	<a href="https://es.javascript.info/url">https://es.javascript.info/url</a>
<b>OPCIONAL:</b> fetch con ejercicios	Para profundizar, una explicación sobre cómo usar fetch pero a la vez haciendo ejercicios.	<a href="https://www.codecademy.com/courses/learn-intermediate-javascript/lessons/js-requests-with-fetch-api/exercises/js-requests-with-fetch-api-intro">https://www.codecademy.com/courses/learn-intermediate-javascript/lessons/js-requests-with-fetch-api/exercises/js-requests-with-fetch-api-intro</a>
Introducción al DOM	Pequeña introducción a especificaciones propias de los navegadores.	<a href="https://es.javascript.info/browser-environment">https://es.javascript.info/browser-environment</a>
Introducción a los eventos DOM	Introducción a los diferentes eventos que se producen en un navegador cuando se carga una	<a href="https://es.javascript.info/onload-onDOMContentLoaded">https://es.javascript.info/onload-onDOMContentLoaded</a>

	página. Es interesante al menos conocer los que hay y cómo se utilizan.	
Eventos	Introducción a los eventos del navegador y al DOM. <b>Interesante lectura aunque sea diagonal, saber que existe.</b>	<a href="https://es.javascript.info/introduction-browser-events">https://es.javascript.info/introduction-browser-events</a>

## EJERCICIOS

### Asíncronia

1. Crea un intervalo que vaya imprimiendo un 1 cada segundo por pantalla o por consola si es un programa que no utiliza web.
2. Crea una función que cuando el usuario clique el ratón se imprima un 1 (en este caso sí que tendrás que utilizar web).
3. Crea una función que cuando el usuario clique el ratón espere un segundo e imprima un 1 (en este caso sí que tendrás que utilizar web)
4. Crea un *timer* que cada dos segundos vaya mostrando por pantalla una lista de animales. Puedes hacerlo por web o por pantalla con console.log. Si quieres, incluso puedes hacer que esa lista de animales sea aleatoria.
5. Recibe un valor desde el usuario que marque el número de segundos que hay que esperar y haz una cuenta atrás y avisa cuando haya acabado.
6. Crea una función asíncrona que descargue la página principal de Google.
7. Crea una función que sea asíncrona y que permita introducir una dirección y te devuelva su geocoding con este sistema de Google: <https://developers.google.com/maps/documentation/geocoding/overview>
8. Crea un cronómetro en una página web (muy simple, sólo minutos y segundos)
9. Sobre el ejercicio anterior, haz que con el clic del ratón se pueda parar o continuar el tiempo. (No hace falta controlar el momento en el que se para el tiempo a la hora de volver a empezar, es decir, se presupone que al parar el cronómetro empezaremos desde el segundo que esté contando un segundo entero)
10. Crea un juego que cumpla los siguientes requisitos:
  - User story: A user can enter a number
  - User story: The system picks a random number from 1 to 6
  - User story: If the user's number is equal to a random number, give the user 2 points
  - User story: If the user's number is different than the random number by 1, give the user 1 point. Otherwise, give the user 0 points.
  - User story: Show the result during a second or two and then start again.
  - User story: The user can play the game as long as they want to
11. ¿Qué es HTTP? ¿Que son los errores http? ¿Cuántos hay? ¿Cuáles son los más importantes? Indaga sobre el protocolo y empápate bien de él, es muy frecuente usarlo.

12. Realiza una función que haga un fetch de la página de Google y gestione diferentes errores (prueba a desconectarte de internet cuando lanzas la aplicación, por ejemplo). ¡Ojo! Es muy probable que tengas que salvar un problema de seguridad al intentar conectarte a la página. Hazlo de la manera que se te ocurra.
13. Pide información con fetch a la url: <https://jsonplaceholder.typicode.com/posts/1>. Loguea el status de la petición e imprime por pantalla el contenido del artículo que has recibido.
14. Modifica el ejercicio anterior o crea uno nuevo que permita cambiar el número del artículo que se recibe.
15. Modifica el ejercicio anterior y recibe la lista de todos los artículos. ¿Cuántos hay? ¿Podrías listar los títulos? ¿Y hacer una tabla con los títulos y los contenidos?
16. Realiza los ejercicios de esta página web: <https://education.launchcode.org/js-independent-track/chapters/fetch-json/exercises.html#fetch>
17. Realiza los requisitos de esta página web (sin mirar la solución): <https://education.launchcode.org/js-independent-track/chapters/fetch-json/studio.html#bonus-missions>

## Networking

18. ¿Qué es HTTP? ¿Que son los errores http? ¿Cuántos hay? ¿Cuáles son los más importantes? Indaga sobre el protocolo y empápate bien de él, es muy frecuente usarlo.
19. Realiza una función que haga un fetch de la página de Google y gestione diferentes errores (prueba a desconectarte de internet cuando lanzas la aplicación, por ejemplo). ¡Ojo! Es muy probable que tengas que salvar un problema de seguridad al intentar conectarte a la página. Hazlo de la manera que se te ocurra.
20. Establece una comunicación con esta página web e imprime su contenido (<wss://socketsbay.com/wss/v2/2/demo/>). Prueba a enviarle un mensaje y de nuevo printea su respuesta. Investiga en su página web (<https://socketsbay.com/test-websockets>) sobre cómo hacerlo y cómo funciona la demo.
21. Establece una comunicación con un servidor que permita websocket y printea por consola todos los eventos mencionados en la teoría: connecting, connected, closing y closed.
22. Crea un servidor en javascript que sea capaz de abrir una conexión en tu ordenador local y que conteste siempre la misma información a cualquier página que intente acceder a él. Impleméntalo con protocolo http y con websockets. Asimismo, permite que loguee todo lo que va recibiendo.
23. Pide información con fetch a la url: <https://jsonplaceholder.typicode.com/posts/1>. Loguea el status de la petición e imprime por pantalla el contenido del artículo que has recibido.
24. Modifica el ejercicio anterior o crea uno nuevo que permita cambiar el número del artículo que se recibe.
25. Modifica el ejercicio anterior y recibe la lista de todos los artículos. ¿Cuántos hay? ¿Podrías listar los títulos? ¿Y hacer una tabla con los títulos y los contenidos?

26. Realiza los ejercicios de esta página web: <https://education.launchcode.org/js-independent-track/chapters/fetch-json/exercises.html#fetch>
27. Realiza los requisitos de esta página web (sin mirar la solución): <https://education.launchcode.org/js-independent-track/chapters/fetch-json/studio.html#bonus-missions>

## **BIBLIOGRAFÍA**

*El tutorial de JavaScript Moderno.* (s. f.-b). <https://es.javascript.info/>

*Learn Intermediate JavaScript | Codecademy.* (s. f.). Codecademy.  
<https://www.codecademy.com/courses/learn-intermediate-javascript/>