FLIPPED CLASS: TEST

El objetivo de la flipped classroom es que la teoría se vea en casa, y los ejercicios se hagan en el aula. Siguiendo este procedimiento, las lecturas/actividades propuestas son:

Título	Descripción	Enlace
Breve introducción	Introducción a los test	Sección 1 de este documento
Comienza con Jest	Introducción a jest, cómo se instala y de qué se trata. Nos fijaremos en la estructura con npm y sólo se llegará a “Running from command line”.	https://jestjs.io/docs/getting-started
Avanza con Jest	A la hora de testear, se definen las diferentes herramientas para comprobar las soluciones.	https://jestjs.io/docs/using-matchers
Mocks	Introducción a cómo moquear funciones, muy útil algunas veces.	https://jestjs.io/docs/mock-functions
Introducción a la cobertura de código	A la hora de realizar proyectos, debemos asegurar una buena cobertura de nuestro código.	https://www.atlassian.com/es/continuous-delivery/software-testing/code-coverage
Coverage con Jest	Una explicación de cómo se debe estructurar y lanzar los test con jest para obtener bien el coverage.	https://www.valentinog.com/blog/jest-coverage/
Setup-teardown a test	OPCIONAL: explicación sobre cómo preparar el entorno y la información para varios test a la vez.	https://jestjs.io/docs/setup-teardown
Test código asíncrono.	OPCIONAL: cómo manejar el código asíncrono. Especial mención al async/await, que es el más utilizado.	https://jestjs.io/docs/asynchronous

1. **Notas sobre test cases**

Como ya hemos visto repetidas veces, testear en programación es una parte muy importante de nuestro propio código. No daremos por bueno ninguna solución que no tenga al menos el 80% del coverage de test.

Para ello, en este capítulo, vamos a hablar sobre Jest, un framework muy utilizado en Javascript para realizar testing.

NOTA

A la hora de testear una función, tenemos que fijarnos en que nuestros test hagan una serie de comprobaciones muy importantes:

- Titula el test case como se debe para saber qué estás testeando
- Comprueba y piensa sobre las precondiciones (cómo llega el código a ese programa) -> Given, When, Then
- Mantén el test sencillo, si es muy difícil de hacer, replantéate factorizar la función
- Incluye el resultado esperado
- Incluye casos que sean extremos

Ejercicios

1. Recorre todos los ejercicios hechos hasta ahora y haz sus test unitarios y comprueba su coverage.
2. En esta otra página también tienes ejercicios que puedes añadir jest: https://github.com/dentednerd/jest-katas . Por favor, lee con atención los ejercicios y mira los archivos que hay en la kata que se dan como propuesta antes de hacer nada.

BIBLIOGRAFÍA

Atlassian. (s. f.). *¿Qué es la cobertura de código?* | *Atlassian*.

https://www.atlassian.com/es/continuous-delivery/software-testing/code-coverage

Configuring code coverage in Jest, the right way. (2021, 3 noviembre).

https://www.valentinog.com/blog/jest-coverage/

Jest. (s. f.). https://jestjs.io/es-ES