

Activity 4: Asynchrony

Curs	23/24	Grup	S2P	Data lliurament	3/11 - 23:55
Mòdul	Interfaces Development				
Títol	Working with asynchrony in Javascript				

Tipus de treball	Individual
Guidelines	
<p>Solve these 8 exercises in a single JS file. Keep the main function names (<i>in italics</i>) and include an usage example for them.</p> <p>1. Callback Basics Create a function <i>delayWithCallback()</i> that receives a string message and a callback. The function executes the callback after 2 seconds and returns a String object. The callback prints the message to the console.</p> <p>2. Promise Practice Write a function <i>promiseWithDelay()</i> that returns a promise. The promise should resolve with the value "Success" after a 3-second delay.</p> <p>3. Chaining Promises Create a series of promises that execute one after the other. The first promise should resolve with "First," the second with "Second," and the third with "Third." Hint: <i>.then()</i> method returns a new promise object that can be used for chaining.</p> <p>4. Promise Error Handling Build a promise using the function <i>simulateNetworkRequest()</i> that simulates a network request. It should resolve with "Data received" after 2 seconds, but reject with an error message "Request failed" after 3 seconds.</p> <p>5. Parallel Promises Write a function that takes an array of URLs and returns an array of promises that fetch data from those URLs using <i>fetch()</i>. Use <i>Promise.all</i> to work with the array of promises and wait for all requests to complete. Can use this URLs: <code>const urls = ['https://jsonplaceholder.typicode.com/posts/1', 'https://jsonplaceholder.typicode.com/posts/2', 'https://jsonplaceholder.typicode.com/posts/3'];</code></p>	

6. Async-Await Basics

Create an async function `asyncWithAwait()` that awaits a promise that resolves with "Hello World!" after a 2-second delay. Log the result to the console.

7. Async data request

Implement a function `fetchDataAndPrint` that uses `async/await` to fetch data from a URL and prints the response to the console. Do nothing if the promise is rejected.

8. Async-Await Error Handling

Write an async function `asyncWithErrorHandling` that awaits a promise. The promise should reject with an error message "Promise rejected." Catch the error using a try-catch block and log the error message.

Evaluation and Qualification criteria

Each correct exercise adds 1 point.

This activity corresponds to a 5% of the practical part.