JavaScript Callbacks

Alfonso De La Guardia

What is a Callback?

Simply put: A callback is a function that is to be executed after another function has finished executing — hence the name ‘call back’.

More complexly put: In JavaScript, functions are objects. Because of this, functions can take functions as arguments, and can be returned by other functions. Functions that do this are called higher-order functions. Any function that is passed as an argument is called a callback function.

How the pill was organized?

- Organization was based on the tasks of the project.

- The first phase were to create a Git Repo and create the html and js files and start the documentation.

- The second phase was to copy the functions provided and experiment.

- The third phase was to find examples of Callbacks and develop the presentation.

Explain the knowledge learned

- Callbacks are used for asynchronous programming.

- Callbacks can be used for when we want a function to work after another.

- Callbacks are a way to make sure that certain code can't execute until another has already finished execution.

What difficulties have arisen during the pill?

- At first, understanding the basic concepts about Callback.

- Understanding how to use callbacks in code.

- Finding the 5 real examples of Callbacks.

When do I recommend the use of Callbacks?

- It’s good for working with Objects. It makes it easier.

- When you want to work with JavaScript dynamically. When you want your params to be entered by the user.

What are the advantages and disadvantages of Callbacks?

Advantages:

- One advantage is that you can call a function as many times as required.
- Another advantage is that it lets you activate functions in different timings.

Disadvantages:

- One of the main disadvantages is that you can fall into CallBack hell.
- It makes for too much code.

In which cases is not recommended to use them?

- When you already have too many callbacks, you can get into callback hell.

- CallBack It consists of multiple nested callbacks which makes code hard to read and debug. It is understandable how one might unknowingly get caught in Callback Hell while dealing with asynchronous logic.

- If you are not expecting your application logic to get too complex, a few callbacks seem harmless. But once your project requirements start to swell, you will quickly find yourself piling layers of nested callbacks.

THANKS!