Alfonso De La Guardia **alfonsodelag1@gmail.com**

JavaScript Callbacks

20th December 2019 - 7th January 2020

Overview

The main objectives of this pill are to understand what a callback function is and when to use it, and to improve our knowledge in JavaScript

Table of Contents

Project Requirements

For this project, it is imperative to do the following:

- You must use GIT
- Create a clear and orderly directory structure
- Both the code and the comments must be written in English
- Use the camelCase code style to define variables and functions
- In the case of using HTML, never use online styles
- In the case of using different programming languages always define the implementation in separate terms
- Remember that it is important to divide the tasks into several sub-tasks so that in this way you can associate each particular step of the construction with a specific commit
- You should try as much as possible that the commits and the planned tasks are the same
- Delete files that are not used or are not necessary to evaluate the project

Risk Management Plan

Every project has risks. This risks must be taken into account to improve the workflow of the project. I've listed the risks for the project, along with the impact they might have, and the priority of them.

iD	Risk	Consequence	Prob. (1-5)	imp. (1-5)	Pri. (1-25)	Mitigation approach
1	Breaking my computer	Can't do anything	1	5	5	Keep my repo work to date, look for other computers
2	Getting sick	Wouldn't be as productive	2	3	6	Eat and sleep well
3	Not concentrating enough	Won't com	2	5	10	Focus on the Minimum Viable Product.
4	Unrealistic deadlines	Deadlines wouldn't be met + development	2	3	6	Be more organized with the tasks, set

		shortcuts would have been taken affecting the robustness of the code				new deadlines.
5	Being unfocused	Loss of control over the development flow of the project	4	5	20	Focus on finishing the most important tasks only.

Tasks for the project

Defining this part is crucial to the development of the project. It is important to make a good analysis of the situation to organize the project in a good way.

1. Reading the description
    - Priority: Medium.
    - Description: Reading the description of the project to figure out the requirements and tasks to do.
    - Difficulty: 2/10
    - Time estimation: 20 min

2. Create repo
    - Priority: Medium.
    - Description: Creating the repo for the project.
    - Difficulty: 2/10
    - Time estimation: 5 min.

3. Create js and html files
    - Priority: Medium.
    - Description: Creating the Js and HTML files to work with the project
    - Difficulty: 3/10
    - Time estimation: 10 min.

4. Investigate about Callbacks and create functions
    - Priority: High.
    - Description: Reading the description of the project to figure out the requirements.
    - Difficulty: 8/10
    - Time estimation: 2 hours.

Chronogram

ID #	Wednesday	Thursday
1	X	
2	X	
3		X
4		X

Calendar

- **Wednesday:**

  -Reading the project description

  -Creating the repo for the project

- **Thursday:**

  -Creating the html and js files.

  -Investigating about callbacks and creating examples.

Git Workflow

For this project, all commits we'll be pushed directly to the Master branch. All commits will use a descriptive message, so that myself or other users can easily go to the Git version that they need to. This is very important for working in teams as it increases communication and efficiency between all members.

Technologies used

For this project, we will use the following technologies:

- HTML and JS
- Visual Studio Code
- Git

File structure

The files will be organized in the following way:

callbacks/	
.git/	This folder contains the Git information for the project
callbacks.js/	This file contains the callback function examples
index.html	This is the main HTML for the project
README.md	This contains information on how to use the project.
Styles.css	This file contains the css code.

Lessons learned

- A Callback, also known as a "call-after"function, is any executable **code that is passed as an argument to other code** that is expected to call back (execute) the argument at a given time. This execution may be immediate as in a synchronous callback, or it might happen at a later time as in an asynchronous callback.

Incidents

- A lot of difficulty for understanding the concept of callbacks.
- Wasting time for not being focused enough.