

JavaScript Timeouts and Intervals

20th December 2019 - 7th January 2020

Overview

In this project we will experiment with JavaScript Timeouts and Intervals. The goal is to understand what these are for and improve our basic understanding of JavaScript.

Table of Contents

Project Requirements	1
Risk Management Plan	1
Tasks for the project	2
Chronogram	3
Git Workflow	3
Technologies used	3
File structure	3
Lessons learned	3
Incidents	4

Project Requirements

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

1.1. Create a script that uses setTimeout

First of all you will have to create a script that is responsible for making use of the setTimeout function. This script will display an alert with any text after 5 seconds.

1.2. Create a script that uses setInterval

Secondly, you must create a script that takes care of using the setInterval function. This script will display an alert with any text every 30 seconds.

1.3. Create a local repository

As it could not be otherwise, the first step is to create a Git repository.

1.4. Create a remote repository

Sign up for an online repository service such as Bitbucket and create a new repository. It is important that you develop the operations mentioned above and upload your code to the remote repository.

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 shortcuts would have been taken affecting the robustness of the code	2	3	6	Be more organized with the tasks, set 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: Reading the description of the project to figure out the requirements.
 - Difficulty: 3/10
 - Time estimation: 10 min.
4. Create setTimeout and setInterval methods in the js files:
 - Priority: Medium.
 - Description: Reading the description of the project to figure out the requirements.
 - Difficulty: 3/10
 - Time estimation: 30 min.

Chronogram

Calendar

- **Wednesday:**
 - Create the repo
 - Create the files in VSC

-Create and test setTimeout and setInterval methods.

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:

timeouts-intervals/	
.git/	This folder contains the Git information for the project
js/	This folder contains the JavaScript file for the project
js/setTimeout.js	This file contains the setTimeout method
js/setInterval.js	This file contains the setInterval method
index.html	This is the main HTML for the project
README.md	This contains information on how to use the project.

Lessons learned

- The setTimeout() method calls a function or evaluates an expression after a specified number of milliseconds.
- The setInterval() method calls a function or evaluates an expression at specified intervals (in milliseconds).
- The setInterval() method will continue calling the function until clearInterval() is called, or the window is closed.

Incidents

- At the beginning I faced some difficulties creating the `setTimeout` and `setInterval` methods.