**EXSM 3936: JavaScript II - API / Async Assignment**

Deadline: Sunday Jun 5, 2022 at 11:59 PM

<https://classroom.github.com/a/OLtg1b_x>

Introduction

This assignment is a graded project that is meant to test your knowledge and skills learned over this module. Follow the instructions below to complete the task.

Instructions

You are hired by a group of Pokemon enthusiasts to create a basic website where they load information about any Pokemon onto the page by simply searching by name. The information comes from a Pokemon API which they provide you with. Read the following hints carefully and ask for clarification if needed.

1. Go to <https://pokeapi.co/>
2. In the search box, enter the following: pokemon/pikachu
3. Note the format in which the API returns the information regarding Pikachu
4. Write a basic search page that takes input from the user and passes the search term into the API
5. Use axios or fetch to retrieve information from the API
6. Implement error handling such that even a failed API link will not crash your webpage
7. Unpack the following information from the JSON package received from the API:
   1. Species Name
   2. Type Name(s)
   3. Weight
   4. Height
   5. Base Experience
8. Include a finally success message displayed via console log
9. Display the information obtained from the API onto your search page using DOM manipulation
10. All code should pass the [HTML Validator](https://validator.w3.org/)
11. All code should pass the [CSS Validator](https://jigsaw.w3.org/css-validator/)

Please note that your solution is expected to work with all Pokemon, including those with multiple types.

Criteria

| Requirement | Marks Available | Mark(s) Awarded | Comments |
| --- | --- | --- | --- |
| Web page retrieves correct information from API when search term is entered. | 3 |  |  |
| Webpage does not crash when no information is retrieved from the API (error handling). | 1 |  |  |
| Only select information needed is extracted and displayed. | 1 |  |  |
| A success message is implemented using the finally keyword. | 1 |  |  |
| Information displayed is created with DOM. | 1 |  |  |
| When a new search is done, information displayed is updated properly. | 1 |  |  |
| All code passes the [HTML Validator](https://validator.w3.org/). | 1 |  |  |
| All code passes the [CSS Validator](https://jigsaw.w3.org/css-validator/). | 1 |  |  |
| Deductions: |  |  |  |
| README.md file is not present in the docs directory, or does not contain a reasonable minimum amount of information. | -1 |  |  |
| Version control is not used appropriately (-1 for vague or nonsensical commit messages, up to -2 for insufficient commits, -3 for a single commit regardless of message). | -3 |  |  |
| Whitespace is not managed (tab stops / formatting). | -1 |  |  |
| Variable, class, constant, property and/or method names do not meet [the code style requirements](https://docs.google.com/document/d/12iEtYtYiKwRAjhLayI-JVNVqXkud3X_PtOE9D0sU9Cw/edit) (-0.5 each declaration). | -3 |  |  |
| Missing semicolons (-0.5 each). | -2 |  |  |
| Unhandled exception(s) generated (-1 each scenario that generates one). | -3 |  |  |
| Whitespace isn’t stripped from inputs. | -1 |  |  |

**Total: /10**