Welcome to Module 3 of the IOD Software Engineering course!

This is a handy guide to help students make sure their lab work is easily understood and completed. Make sure to ask your trainers if you have any questions about how to complete these exercises.

Trainers will release answers once everyone has had a chance to complete them, but often there are many possible answers so don’t worry if yours are different. The extensions are there for students who have extra time after completing the main exercises - they are not required.

| **JS Fundamentals** | **JSFundamentals\_LabExercises.pdf** |
| --- | --- |

**Instructions**: Download the **JSFundamentals\_LabExercises.pdf** file from this folder and complete the 10 exercises within it. Include all the answers in a single .js file, individually commented to indicate each of the 10 exercises. Use Node.js to run and test your code, using the Terminal in VS Code.

Exercises 1-4 require some explanations as well as code in the answers - include this in the comments of your JS file.

All of the information you need to complete the exercises should be in the JS Fundamentals module material. See slides 1-29 for exercises 1-4, slides 30-36 for exercise 5, slides 37-51 for exercises 6-9, and slides 52-54 for exercise 10. The links below may help as well.

**Goal**: To practise using the fundamental elements of Javascript, including primitive and complex data types, conditionals, functions and loops.

**Submission**: Create a new folder for **Module 3** that you can include in a GitHub repository. You may wish to re-use the same private repo as your lab exercises from other modules and include a new folder in there, or create a separate private repo. Group all Module 3 exercises in the same folder, with separate files for each of the 3 sections, and commit your changes once complete. Ensure your trainers have access.

**Extension**: Try the JS Challenge Rush at <https://www.jschallenger.com/games/rush/>

**Resources & Extra Learning**:

* <https://www.jschallenger.com/javascript-basics>
* <https://web.dev/learn/javascript/data-types>
* <https://web.dev/learn/javascript/comparison>
* <https://web.dev/learn/javascript/functions>

| **JS Intermediate** | **JSIntermediate\_LabExercises.pdf** |
| --- | --- |

**Instructions**: Download the **JSIntermediate\_LabExercises.pdf** file from this folder and complete the 10 exercises within it. Include all the answers in a single .js file, individually commented to indicate each of the 10 exercises. Use Node.js to run and test your code, using the Terminal in VS Code.

All of the information you need to complete the exercises should be in the Intermediate Javascript module material. See slides 1-20 for exercises 1-2, slides 20-45 for exercises 3-4 & 7, slide 13 for exercise 5, slides 46-56 for exercises 6 & 8-9, and slides 67-71 for exercise 10. The links below may help as well.

**Goal**: To practise using some intermediate aspects of Javascript, including an in-depth look at data types including built-in prototype methods, especially for Strings and Arrays, and special types like Map, Set and Date.

**Submission**: Once complete, commit your JS file containing all intermediate exercise answers to the Github repository containing your Module 3 answers. Ensure your trainers have access.

**Extension**: Copy this starter code:<https://github.com/jobatkinIOD/SEModule1/blob/main/script.js> and follow the instructions in the comments to practise using JS functions, arrays and objects. Also try the practice questions at <https://www.geeksforgeeks.org/practice-javascript-online/>.

**Resources & Extra Learning**:

* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Map>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Set>

| **JS Advanced** | **JSAdvanced\_LabExercises.pdf** |
| --- | --- |

**Instructions**: Download the **JSAdvanced\_LabExercises.pdf** file from this folder and complete the 10 exercises within it. Include all the answers in a single .js file (or multiple files if clearly named), individually commented to indicate each of the 10 exercises. Use Node.js to run and test your code, using the Terminal in VS Code.

All of the information you need to complete the exercises should be in the Advanced Javascript module material. See slides 1-10 for exercise 1, slides 11-20 for exercises 2-4, slides 21-29 for exercise 5, 31-36 for exercise 6, slides 37-47 for exercise 7, all slides up to 51 for exercise 8, and slides 52-62 for exercises 9 & 10. The links below may help as well.

**Goal**: To practise using some advanced aspects of Javascript, including closures, decorators, applying context, asynchronous coding with timeouts and promises, prototypes, classes and error handling.

**Submission**: Once complete, commit your JS file/s containing all advanced exercise answers to the Github repository containing your Module 3 answers. Ensure your trainers have access.

**Extension**: Follow the instructions in **Module 3 - Advanced JS Extra Exercises - eComm 1-8 v2.pdf** (in this folder on Google Drive).

**Resources & Extra Learning**:

* <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object_prototypes>
* <https://developer.mozilla.org/en-US/docs/Web/API/setTimeout> and <https://developer.mozilla.org/en-US/docs/Web/API/setInterval>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Function>
* <https://javascript.info/call-apply-decorators>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes>
* <https://www.freecodecamp.org/news/javascript-classes-how-they-work-with-use-case/>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Using_promises>
* <https://www.joshwcomeau.com/javascript/promises/>
* <https://javascript.info/try-catch>