

IMY 220

Practical 4:

ES6

Due: Friday 26 August @ 09:30

The submission instructions are available on ClickUP. Any deviation from these instructions will cause a 10% deduction from your mark.

Instructions

- For this practical, you will need to update the JavaScript from last week to use ES6 as well as add another function.
- Download *index.html* and *script.js* from ClickUP. These files contain the functionality for last week's practical. You are not allowed to edit *index.html*, except for putting your name and surname in the "author" meta tag. All your code for this practical must be written inside *script.js*. Don't just copy the functions and expect to receive marks, you will have to make the correct changes to convert the functions to ES6.

Section 1: ES6

Rewrite all the code inside *script.js* to use recommended ES6 syntax. This includes:

- Using an ES6 **class** to define *FactorialChecker*
- Using **arrow functions**
- Using appropriate variable declarations, i.e., **changing it from the *var* keyword**
- For the function *PigLatinEncrypt()* you are **not allowed to write your own loops**. You will have to make use of the correct array functions in order to get the functionality to work (you are **not** allowed to use the *forEach* array function for obvious reasons). You can still make use of the same methods (such as using regular expressions), but you will have to change them slightly so that they make use of **array functions**. If you do not make use of array functions, you will receive lower marks compared to what you would get if you made use of array functions.

HINT: Look into array functions such as *map*, *include* and *match*. These are all discussed in the notes and will be helpful for this practical

Furthermore, you must redefine how the functionality is printed on the page. Each result must be printed in a **descriptive string** which must look as follows:

Generate Factorial

Factorial number

Submit

The factorial value of 6 is 720

Factorial List

Factorial value

Submit

The values that make up the factorial 720: 1,2,3,4,5,6

Pig Latin Encryption

Enter a word to convert it to pig latin

Convert

The result when converting apple to pig latin is: appleway

Note that you must print the results to the page using **ES6 template strings**.

Section 2: Unique letters checker

Write a **function** called **checkUniqueLetters** that **counts unique letters in a word** (i.e., only counts a letter once if it repeats) and returns that value. For example, the word "ensemble" contains 6 unique characters ("e" gets repeated 3 times so we only count it once). The trick with this is that you are **not allowed to use loops** (for loops etc.) and you must make use of **array functions** (you are **not** allowed to use the *forEach* array function for obvious reasons).

You must use the **Array.reduce** function to check for this. However, if you are unable to complete this functionality using *Array.reduce*, you should use another method but you will not receive the same marks that you would have if you used the *array.reduce* method.

HINT: the **Array.includes()** array function returns true if an array contains one/more of an element and will help you in this practical.

This section must also be done with ES6 syntax where appropriate and must also be printed to the page in a **descriptive string**, for example:

Count unique letters

count how many unique letters are present in the string

Submit

The word ensemble has 6 unique letters

Count unique letters

count how many unique letters are present in the string

noodlenoodle

Submit

The word noodlenoodle has 5 unique letters

You do not have to cater for spaces, we will only test one word at a time.

Additional Information

- Refer to the slides as well as MDN website (<https://developer.mozilla.org/en-US/>) for help

Submit only the following file(s) according to the submission instructions.

- *index.html*
- *script.js*