Second Guided Exercise

UC3M Computer Science and Engineering

Prepared by:

Group 12
100429678 Sergio Caño Amor
100451207 Paloma Núñez Reyes

25/02/2022

INDEX

1. Introduction 2

2. Coding Regulation.....2

3. Screenshots.....4

Introduction

This document gives coding conventions for the Python code comprising the standard in the main Python distribution to make sure that following pair programming techniques we can work in a coordinated way.

Coding Regulation

- *Step 1. The standard for the Organization of Files*

Source Code Files:

1. To track the copyright legend, when an algorithm is developed the corresponding member should include a commented line, explaining the algorithm, including its name and date of execution. Is also possible to include the corresponding changes, date, and name of the person who has developed it inside the README.txt file before exporting it to GitHub, besides it is a must to add at least next to the algorithm a brief explanation to ease the work.
2. The version control of the source files will be managed through the GitLab service.
3. We have established the following regular expression where we allow the name of the file to be in capital letters, starting with UC3M, underscores and the following can have words in capital and lowercase letters up to 20 characters.

module-rgx=UC3M_[A-Z][a-z]{3,20}$

Class Files:

1. It is compulsory to use independent files for each class made. For those files, the corresponding name will be UC3M_CLASSES, where the name corresponds to the given name to the created class.
2. Each class file should contain a header to provide information about the class, and in the case to be used, import in the corresponding file.

- *Step 2. The standard for Names and Variables*

Classes and Members of Classes:

1. The name of classes should follow the convention: NAME, where name is the corresponding label chosen of the class and it must be in uppercase, the interfaces and data type names must be described in lowercase and all of them must have a meaningful meaning.
2. For the naming of the files, methods, properties, and constants, their label must be meaningful and in lower case and words separated by underscores to improve readability. In the case of constants is preferable for them to be in uppercase, to make sure that their value is not changed among the code.

Visibility:

1. Variables with static values must be established using properties to make them private in order for them to not be able to be changed by any external users. Other variables which are not static can be left public. In case of doubt, choose non-public as is easier to make it public later than to make a public attribute non-public.
2. We will use four spaces per indentation level, and the four-space rule is optional for continuation lines. Moreover, the programmer must include only one statement per line. Limit all lines to a maximum of 79 characters. Imports should usually be on separate lines.
3. Global variables will not be, in general, allowed, and we must declare variables inside the corresponding functions. “For” loops require an index and that index must be initialized just before the loop is developed.
4. The “this” keyword must be added when referencing class attributes in a method to set the difference between them and local variables.

- *Step 3. Standard for Methods*

Method structure:

1. There is not a maximum size of the number of methods inside the source. The developer shall give comments about the description of the method.

Indentation and braces:

1. Four spaces per indentation level.
2. Continuation lines should align wrapped elements either vertically using Python's implicit line joining inside parentheses, brackets, and braces, or using a ***hanging indent***[#fn-hi]_. When using the hanging indent, there should not be arguments on the first line and further indentation should be used to clearly distinguish itself as a continuation line.

Method definitions:

1. In case the parameters of methods do not fit in the same line of declaration, one argument per line makes for great readability.
2. The developer must include a commented description of the method implemented just before the corresponding code.

- *Step 4. The standard for exception handling*

1. Regarding the detection of errors inside the input parameters, the developer should check the following: if the number of requested parameters is equal to the number of inputs introduced and if the data type of each parameter matches the data types asked for inside the method. For each of these cases, the developer should throw a specific and informative exception for the user to handle the error.
2. Regarding exception handling, the developer should handle these exceptions inside the component itself rather than handling them in the main program. This would improve the readability and comprehension of the method, as it will include all its different parts together.
3. Regarding the methods with a high probability of throwing an exception, the developer shall pay special attention to the input checking convention established before and the overall method performance. That is, each of the operations inside the method that are thought of throwing an exception shall be covered, for example, when checking if a certain variable is correctly defined (checking the data type) or when there is some possibility to perform certain illegal instructions (dividing by zero).

- *Step 5: Other standards*

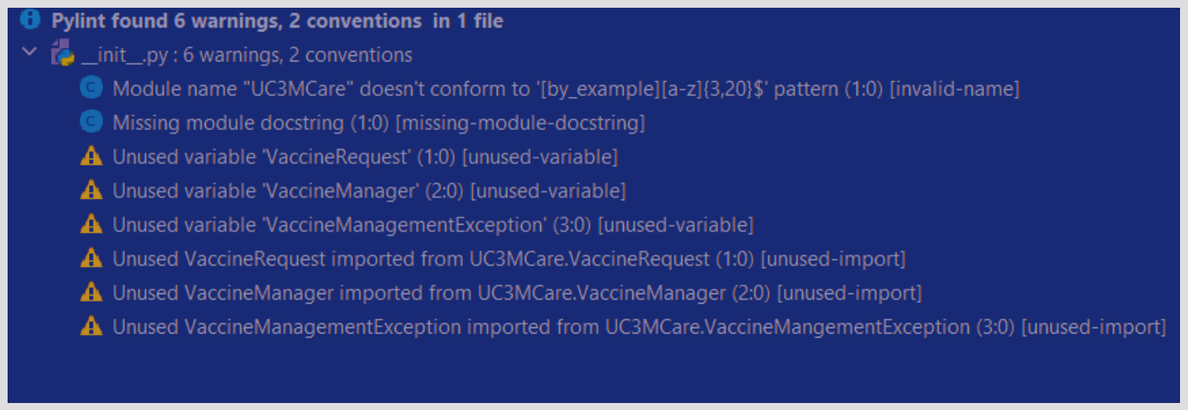
1. Other standards should be taken into account by the developer, such as parameter splitting or parenthesis placement. Regarding the previously mentioned, parameters shall be organized according to their importance inside the function they are referring to. That is, the first parameter will be the most relevant one and the last one will be less important (such as a temporal variable). It is also important for the developer to place the parenthesis for the function he/she is referring to as close as possible.
2. Regarding loop implementation, the developer shall follow the conventional principles established, paying special attention to the following ones:

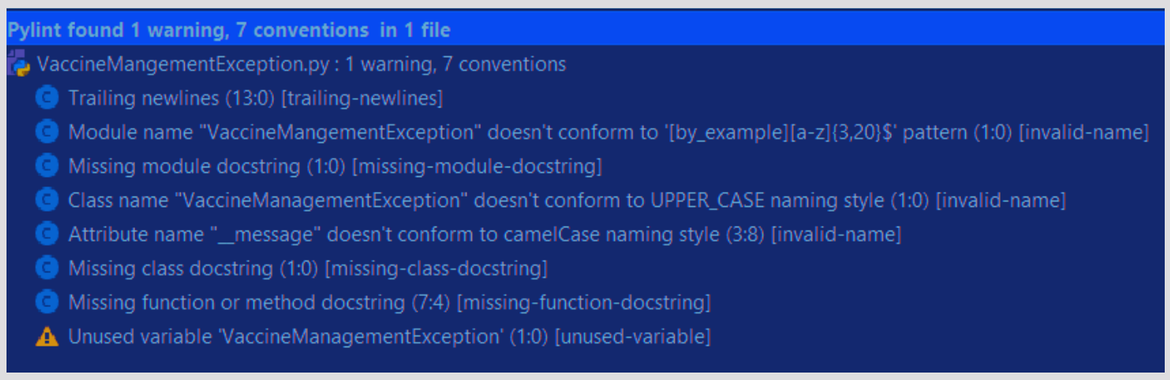
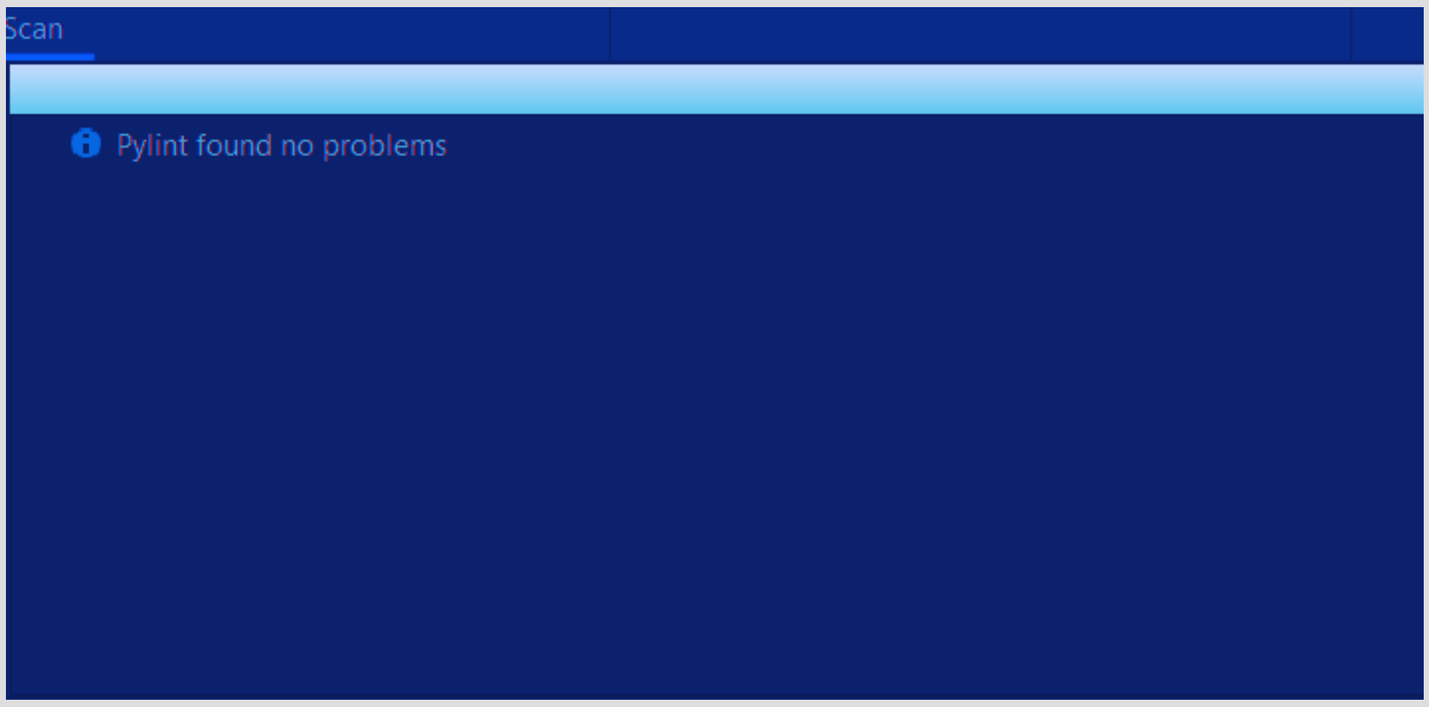
For loops: the developer shall take advantage of the way python defines this type of loop, that is, using iterable objects such as lists or ranges.

While loops: the developer shall place the variable increment/decrement, in case there is one, always at the end of the loop. This option is preferred to increase the readability and comprehension of the loop.

Screenshots

BEFORE





AFTER