

Laboratory Report



Ollscoil
Teicneolaíochta
an Atlantaigh

Atlantic
Technological
University

Lecturer:	John O'Raw
Report Title:	Python Report on Walkthroughs 1 - 6
File Name:	L00177769.pdf
Submit to:	Blackboard in PDF format only
Date Submitted:	9/10/2022

Student Name:	Martina Atkinson
Student Number:	L00177769
Programme of Study:	PGDip in Cloud Computing
Module:	Infrastructure as Code (2022/23)

Contents

Description	3
Aims	3
Methodology.....	3
Results and Testing	3
Conclusion.....	4

Description

This report documents the work completed in weeks 1 and 2 of the Infrastructure as Code module which focused on learning the basics of the Python programming language through a series of technical walkthroughs. No previous programming knowledge was assumed, and the module aimed to teach students the basic skills required for scripting rather than full-scale programming. The report will examine the learning experience by looking at the learning environment and focusing on course content in terms of understanding, difficulties encountered and learning outcomes achieved.

Aims

The following aims were identified to examine the student's experience in the first two weeks of the Python course,

1. Learning environment and suitability to student needs
2. Review of the course content and difficulties encountered.
3. Based on these experiences examine the learning success

Methodology

The course material was available via six walkthrough documents which were designed by the lecturer to bring the student through the technical tasks [1]. Students could choose to use their own personal laptop/computer or a VDI provided by ATU. Instructions were provided on installing the recommended software, i.e. Python 3.0 (or higher), Notepad++ and Visual Studio Code. The software could be downloaded from the ATU website, and all software came preinstalled for those using the VDI.

Students were given 3 weeks to complete the walkthroughs and produce a report on their experiences.

Results and Testing

To acquire practical skills in the Python language, it was a requirement to type up all the sample code and complete exercises for each walkthrough. Exercise folders were created for the 6 walkthroughs and submitted with a report at the end.

- Walkthrough 1 – The most challenging task was to create own script design in Budget.py
- Walkthrough 2 – Hello.World.py was created to show an understanding of docstrings
- Walkthrough 3 - Tuples were proved immutable (TupleExercise.py) and methods were tested in MethodsExercise.py and DictionaryExercise.py.
- Walkthrough 4 – Created own dictionary item (my_for_dictionary.py). List comprehension exercise was challenging but the conversion of temperatures was completed and checked against a calculator online (list_comprehension_exercise.py).
- Walkthrough 5 – The toughest exercise was on writing a function to search for an even number and return true if found from a list with odd and even numbers.
- Walkthrough 6 – The cube.py and square.py were amended to best practices.

Conclusion

The purpose of this report was to document and review the experience of learning the Python programming language through six technical walkthroughs looking at the learning environment and focusing on the content and learning outcomes [1].

In terms of the learning environment, the VDI proved unsuitable as the lack of admin rights meant it was not configurable and the Python plug-in could not be installed. Notepad++ was recommended as an editor and this was found to be a very friendly easy-to-run tool, particularly after it was configured with the NPPExec plugin to run scripts from within Notepad++ itself [2]. For the later walkthroughs, Visual Studio Code was recommended as an editor but while it is more complicated to use, there is a wealth of supporting documentation available [3].

The first walkthrough [1] covered the fundamental concepts of Python (assignments, data types, operators, strings, etc.) and as any experienced programmer will tell you, it is very important to get these foundations right [4]. They were a good mixture of conceptual and practical learning, with print screens of code only available in the earlier walkthroughs, forcing the student to type the code and pick up the nuances of the language. The problem of floating numbers was highlighted very succinctly by a small piece of code but the recommended reading, while it was informative some of the material was difficult due to its math content [5]. Dynamic typing, and its merits and associated debates, to the novice, seemed a very friendly approach to programming but became more of a foe than a friend as the `print(type(variable))` command became the most typed in troubleshooting [6]. The budget exercise was a test of creativity and highlighted the problem of great ideas but short on skills and the need to learn the lesson from experienced programmers to keep it simple, get it running and then improve upon it [4].

The second walkthrough [1] highlighted the importance of documenting and commenting on your code as if not done at the time of writing, it will rarely get done as programmers get busy with other projects, which leads to problems for them and their colleagues in the future [7]. There was some ambiguity around this module, as students were instructed to create an exercise folder but there were no obvious exercises to complete.

The third walkthrough focused on Python data structures and there were no problems with either content or exercises [1]. The complications came later when these data structures were used or referenced in loops and functions and the niceties and limitations of each type became more apparent. The main issue was remembering the characteristics and context for each, but this should hopefully come with practice aided by the recommended reading document [8].

In walkthrough four and subsequent ones, life became a lot more challenging as concepts were combined [1]. A good understanding of the basics was required to understand the logic of flow control, loops and list comprehension and flow control diagrams may have been useful. Flow control and `if/elif/else` was perfectly demonstrated in the exercise by changing the variables to witness the results. The 'for' loop and 'list comprehension' were a little trickier to understand as confused by variable names created to hold data while looping (e.g., for each item), not sure if they were text or part of the command but this was resolved by research, making changes and observing the results [9]. The pros and cons of a for loop vs list comprehension were investigated and it was determined that while list comprehension may be faster in calculations, a for loop is the best choice when the logic is more complicated [10]. Also, looked at the range function to determine that it operated in a similar way to "start:stop: step". The temperature conversion exercise proved a challenge but was completed and checked using an online calculator [11].

The function walkthrough proved the most exacting as functions can include logic, loops calculations etc [1]. It brought home the importance of correct notation as a misplaced indent or colon could trip up even a seasoned programmer. There was confusion over parameters, arguments and returns but an online website proved very useful in helping to understand the syntax and workings of functions [12]. The key to understanding functions was looking at other sources, typing up the scripts and making individual changes to elements within functions to clarify exactly their purpose within the functions [13]. The topic on scope cleared up some of the issues experienced with changing variable values and types but it did take a while and a bit of research to understand local and enclosing [14]. The exercises also proved challenging as some problem-solving skills were required particularly in finding an even number in a list of numbers.

The final walkthrough covered the subject of modules and standard libraries [1]. There were no real problems creating the custom modules or importing the standard libraries or just the specific function. The Python standard libraries document was reviewed and found that it gave very valuable information on the various functions and their use [15]. There were also no problems in creating `_init_.py` but it did require some further research as was unsure of the context of its use from the one example and subsequently concluded that it will be a very useful file in the future [16].

Overall, the walkthroughs were excellent tools for learning the fundamental skills in programming as a great deal of knowledge was acquired in a short space of time [1]. Their ultimate success will be judged by how much is retained and applied in the future when students are faced with more demanding concepts and tasks. Hopefully, solid foundations have been laid for the future.

- [1] J. O'Raw, "Content." https://lyitbb.blackboard.com/ultra/courses/_58891_1/cl/outline (accessed Sep. 24, 2022).
- [2] *How to run Python Program in Notepad++ ?*, (Jun. 17, 2021). Accessed: Sep 24, 2022. [Online Video]. Available: https://www.youtube.com/watch?v=6J8gXEu_lpE
- [3] "Editing Python Code in Visual Studio Code." <https://code.visualstudio.com/docs/python/editing> (accessed Sep. 24, 2022).
- [4] "How to Learn Programming?," *GeeksforGeeks*, Jan. 14, 2020. <https://www.geeksforgeeks.org/how-to-learn-programming/> (accessed Sep 24, 2022).
- [5] "15. Floating Point Arithmetic: Issues and Limitations — Python 3.10.7 documentation." <https://docs.python.org/3/tutorial/float.html> (accessed Sep. 24, 2022).
- [6] G. Myrianthous, "Dynamic Typing in Python," *Medium*, May 01, 2021. <https://towardsdatascience.com/dynamic-typing-in-python-307f7c22b24e> (accessed Sep. 24, 2022).
- [7] R. Python, "Writing Comments in Python (Guide) – Real Python." <https://realpython.com/python-comments-guide/> (accessed Sep. 26, 2022).
- [8] "5. Data Structures — Python 3.10.7 documentation." <https://docs.python.org/3/tutorial/datastructures.html> (accessed Sep. 30, 2022).
- [9] P. Geeks, "Loops in Python with Examples," *Python Geeks*, Jul. 30, 2021. <https://pythongeeks.org/loops-in-python/> (accessed Sep. 30, 2022).
- [10] S. Witowski, "For Loop vs. List Comprehension," *Sebastian Witowski*, Sep. 17, 2020. <https://switowski.com/blog/for-loop-vs-list-comprehension/> (accessed Sep. 30, 2022).
- [11] "Celsius to Fahrenheit conversion (°C to °F)." <https://www.rapidtables.com/convert/temperature/celsius-to-fahrenheit.html> (accessed Sep. 31, 2022).
- [12] "Python Functions." https://www.w3schools.com/python/python_functions.asp (accessed Oct. 01, 2022).
- [13] "Python Functions," *GeeksforGeeks*, Sep. 08, 2018. <https://www.geeksforgeeks.org/python-functions/> (accessed Oct. 01, 2022).
- [14] P. Geeks, "Python Variable Scope," *Python Geeks*, Jun. 29, 2021. <https://pythongeeks.org/python-variable-scope/> (accessed Oct. 06, 2022).
- [15] "The Python Standard Library — Python 3.10.7 documentation." <https://docs.python.org/3/library/> (accessed Oct. 05, 2022).
- [16] "Understand Python __init__.py for Beginners - Python Tutorial," *Tutorial Example*, Oct. 21, 2019. <https://www.tutorialexample.com/understand-python-init-py-for-beginners-python-tutorial/> (accessed Oct. 07, 2022).