Department of Computer Science
University of Pretoria

Programming Languages
COS 333

Practical 6: PHP and Python

October 7, 2022

1 Objectives

This practical aims to achieve the following general learning objective:

- To gain and consolidate some experience writing programs in several different imperative scripting languages, including: PHP and Python;
- To consolidate a variety of basic concepts related to imperative programming languages and scripting languages, as presented in the prescribed textbook for this course.

2 Plagiarism Policy

The Department of Computer Science considers plagiarism as a serious offence. Disciplinary action will be taken against students who commit plagiarism. Plagiarism includes copying someone else's work without consent, copying a friend's work (even with consent) and copying material (such as text or program code) from the Internet. Copying will not be tolerated in this course. For a formal definition of plagiarism, the student is referred to `http://www.ais.up.ac.za/plagiarism/index.htm` (from the main page of the University of Pretoria site, follow the `Library` quick link, and then click the `Plagiarism` link). If you have any form of question regarding this, please consult the lecturer, to avoid any misunderstanding. Also note that the principle of code re-use does not mean that you should copy and adapt code to suit your solution. Note that all assignments submitted for this module implicitly agree to this plagiarism policy, and declare that the submitted work is the student's own work. Assignments will be submitted to a variety of plagiarism checks. Any typed assignment may be checked using the Turnitin system. After plagiarism checking, assignments will not be permanently stored on the Turnitin database.

3 Submission Instructions

Upload your practical-related source code files to the appropriate assignment upload slot on the ClickUP course page. For your PHP submission you must implement and submit a single file named `s99999999.php`, where `99999999` is your student number. For your Python submission you must implement and submit a single file named `s99999999.py`, where `99999999` is your student number. Include comments at the top of your program source file, explaining how your program should be executed, as well as any special requirements that your program has. Multiple uploads are allowed, but only the last one will be marked. The submission deadline is **Tuesday 18 October, at 23:00**.

4 Background Information

For this practical, you will be writing programs in PHP 7.2 and Python 3.6. You will have to compare these languages in terms of their support for different programming language concepts. To do this, you will have

to write short programs to demonstrate how each language handles the concept under consideration. These programs demonstrate the concepts under consideration adequately, and allow you to describe the language's support (or lack of support) for the appropriate features. Manuals (which include tutorials) for PHP [1] and Python [2] are available on the course ClickUP page and online, respectively.

5 Practical Tasks

For this practical, you will have to write programs in PHP and Python to accept a user-provided string and perform some processing on it. First the general characteristics of the program are described, and then the specific details related to the two languages.

The programs first accept a string of arbitrary length. All uppercase characters are converted to lowercase, while all non-alphabetic characters are stripped from the string. For example, if the user specifies the string "Madam in Eden, I'm Adam.", the converted string would be:

```
madaminedenimadam
```

The program then prints out this converted string, and outputs "Palindrome" if the string is a palindrome, and "Not a palindrome" if the string is not a palindrome. A palindrome is a word or sentence that reads the same both forwards and backwards (if punctuation and spaces are ignored). For example, given the above input string, the output would be:

```
madaminedenimadam
Palindrome
```

There are, of course, several ways in which this test can be performed, and you are free to use whichever approach you prefer. You may use any built-in functionality provided by either language.

PHP

While PHP programs are typically written for execution on a web server, we will be using the command line interface of PHP 7.2 to interpret your script. Your script must accept the user input as a command line parameter. The command line parameter is the un-converted input string surrounded by double quotes.

The conversion of the input string should be handled by a function that is called `convertInput`, while the testing of the converted string should be handled by a second function called `isPalindrome`. The user input is handled as a string that is passed to the `convertInput` function. The `convertInput` function should receive the input string, and modify it according to the conversion described above. The `isPalindrome` function should receive the converted string, and performs a palindrome test as described above. The function then returns a Boolean value indicating whether the converted string is a palindrome or not.

If you use the PHP command line interface to interpret your PHP script, it should generate HTML output which can be opened in a web browser. Note that this output must include HTML tags (it can't just be the text output, even though a Web browser will open a file containing just the output). Also note that this output is generated on the command line, and not output to a file. Include a comment in your program explaining how to run it using the command line interface, as well as any ways in which your program deviates from the above program specification. Check the `php` man page under Linux for information on how to use PHP from the command line.

Python

Write a script in Python 3.6 that generates the converted string and performs the search as described above. Use two functions called `convertInput` and `isPalindrome` that are similar to the ones described for the PHP implementation.

Your Python script should simply print the required output to the screen (i.e. it does not have to generate HTML output). Include a comment in your program explaining how to run it using the command line interface, as well as any ways in which your program deviates from the above program specification.

6 Marking

Each of the implementations will count 5 marks for a total of 10 marks. Both the implementation and the correct execution of the programs will be taken into account. Your program code will be marked offline by the teaching assistants. Make sure that you upload your complete program code to the appropriate assignment upload slot, and include comments at the top of your program source file, explaining how your program should be executed, as well as any special requirements that your program has. Do not upload any additional files other than your source code.

References

[1] Mehdi Achour, Friedhelm Betz, Antony Dovgal, Nuno Lopes, Hannes Magnusson, Georg Richter, Damien Seguy, and Jakub Vrana. PHP manual, 2020. Access at `http://php.net/download-docs.php`.

[2] Python Software Foundation. Python 3.6.0 documentation, 2021. Access at `https://docs.python.org/release/3.6.0/`.