Assignment 1

Status: Completed
Your identity is visible during marking.

Deadline: 10 Sep 2018, 23:55

Weekly assignment 1

This weeks assignments are meant to make you become a little bit familiar with running Python, typing code in an editor, running into some bugs and maybe even fixing a couple. Since there are no programs to be handed in, please submit your answers in a text file (extension .txt) through the GUL entry for this assignment.

Part A

Terminal and REPL

Open up the terminal and start the Python REPL by typing python. You should see `>>>` when you are in the REPL. Try some expressions (for instance `1+1`) to see their evaluation and some statements to be executed (for instance `print('hello brave new world!')`). You can exit the REPL by typing CTRL-D (hold the control key down and press the d key).

Editor, Terminal and Script

Open up your favorite programming editor (Textmate, Emacs, anything else?) and create a script with the statement

```
print('hello brave new world!')
```

Save the script to a file with the extension `.py`, for instance `a1_parta_print.py` and run it by typing `python a1_parta_print.py`. What happens?

Also create a file with only the expression `1+1`, save it as, for instance, `a1_parta_plus.py` and run it. What happens now?

Comparison

What differences did you note between running just an expression and a statement with `print` in the REPL on the one hand and running them as a script on the other? What do you think is going on? Hand in your answers as prose.

Part B

Solve the exercises from Chapter 1 in Think Python (Exercise 1.1 and 1.2) and Chapter 2 (Exercise 2.1 and 2.2) and hand in the answers as prose and/or code snippets in the textfile.

Part C

For the following part of A1, it is good to know that there is an "emergency brake" to force a program to quit: If a program doesn't respond or terminate, press CTRL+C (hold the control key and press the c key) to force it to stop. This also works inside the Python REPL - any computation that is ongoing will then stop with a so called KeyBoardInterrupt-error.

Now, with this precaution, make a script `binary_search.py` that contains the card searching example from the lecture, with some extra print statements to try the program. Make sure you copy it with the exact indentation, otherwise Python may give syntax errors.

```
def binary_search(my_card,whole_deck):
    top = 0
    bottom = len(whole_deck)
    while True:
        middle = (top+bottom)//2
        if whole_deck[middle] == my_card:
            return middle
        elif my_card < whole_deck[middle]:
            bottom = middle
        else:
            top = middle+1

print(binary_search(11,[10,11,13,14])) # should print 1
print(binary_search(14,[10,11,13,14])) # should print 3
```

Recall from the lecture that you do not have to understand the whole program at this point. However, you should at this point be able to enter the code in file and get python to execute it. Run the program as a script to see if it does as intended: the expected result of the two print statements at the end are given as comments. When you are happy it does, and you feel confident, add the following line to the script:

```
print(binary_search(12,[10,11,13,14]))
```

Run the program to see what happens... and if you get tired of waiting for the program to stop running, remember you can interrupt it with CTRL-C!

Describe what happens. Why do you think this is? What kind of error are we dealing with here (syntax/runtime/semantic)? Look again at page 7 of the lecture slides. Is this a problem with the implementation or with the algorithm? Hand in your answers as prose.

Hemanth Kumar Battula , 6 Sep 2018 18:06

File name: Assignment_1.txt (8,8 KB)

Status set to: To be marked

Comment: PartA, PartB, PartC are all present in the same file. Copied the snippets of code, related to partB from the REPL I used and pasted in text file.

Yuri Bizzoni , 7 Sep 2018 09:59

Status set to: Completed

Comment: Well done