PHOT 110: Introduction to programming

Midterm exam questions (version A)

Michaël Barbier, Spring semester (2024-2025)

Before you start

The midterm exam counts for 10% of your total grade of the course. The exam is performed on the computer, and you do not need to provide any answers on paper. There is one question (the third question) which asks to correct a script which contains various errors, other questions require you to write Python scripts.

Take the following points into account before you start the exam.

- Your exam is located in a folder on the computer:
  - The questions: `midterm_questions_x.pdf`
  - The script with errors for question 3: `script_with_errors.py`
  - Three cheat-sheets: for Python, Numpy, and Matplotlib.
- Please make sure that you save all the scripts and output files of the exam, so you can upload them at the end of the exam.
- Let us know if during the exam there is any issue with the computer, PyCharm, or libraries. We will try to verify this up front, but please inform us in case of any issues.

You will be asked to save plots to your folder, this can be done using the `savefig` method, see the following example (the output of this example is a simple line plot):

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()
x = [1, 2, 3, 5]
y = [2, 5, 1, 3]
ax.plot(x, y)
fig.savefig("output_plot_example.png")
```

Questions

Question 1: Print numbers series

Write a script that prints the series $(n-1)/n$ for first positive numbers $n = 1, \ldots, N$, that is, for a given N (you can take N as a parameter of your script) prints the numbers:

$$0/1,\ 1/2,\ 2/3,\ 3/4,\ \ldots\ ,(N-1)/N$$

where each number is printed on a separate line. If you set `N = 6`, then the output should look similar to:

```
0.0
0.5
0.6666666666666666
0.75
0.8
0.8333333333333334
```

Save your solution as a script with file name: `solution_1.py`.

Question 2: Count number of PNG and JPEG files

Define a list with (image) file names as follows:

```
file_names = ["eye.eps", "garden.png", "horse.jpg", "ball.png", "arrow.pdf", "apple.svg"]
```

Implement a script that counts the number PNG-files (".png") and JPEG-files (".jpg") by verifying the extension. Use a loop structure to verify each file name of the list. You can use the `endswith()` method to verify whether the email address is valid:

```
# Example showing how to use the "endswith()" method
my_string= "hello.txt"
# The endswith() method returns True if the string ends in a
# specific substring (here "txt"), otherwise it returns False
print(my_string.endswith(".txt"))
print(my_string.endswith(".zip"))
```

```
True
False
```

Save your solution as a script with file name: `solution_2.py`. This is the output of a correct working script:

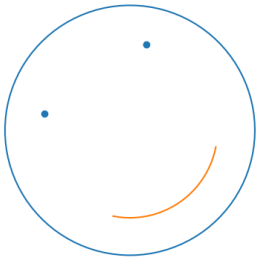
```
Count of PNG and JPEG files
    Number of PNG-files: 2
    Number of JPEG-files: 1
```

Question 3: Correct a Python script

Open the script with name: `script_with_errors.py` and correct the errors.

The correct script prompts the user to type an angle in degrees. It then plots a smiley under that angle and saves it as a PNG-file.

The output plot for 30 degrees is for example:



Question 4: Repeated input

Repeatingly prompt the user to input an integer number between 1 and 100 until the provided integer is divisible by 9. Verify whether the number is within the given range. If the number is divisible by 9, then print a message telling how many times 9 fits in the number (see example), otherwise let the user try again. Allow the user also to try again when: the user types a number which is too small, too large, or types invalid input.

Example:

```
Give an integer divisible by 9 number in interval [1, 100]: 125
The number 125 is invalid, please try again.

Give an integer divisible by 9 number in interval [1, 100]: eighteen
The number eighteen is invalid, please try again.
```

```
Give an integer divisible by 9 number in interval [1, 100]: 18
The number you provided is divisible by 9 and:
  18 = 2 times 9
```

Save your solution as a script with file name: `solution_4.py`.

Question 5: Plot functions

Plot a graph with the following functions (in interval $[0.1, 10]$):

$$f_1(x) = \frac{\sin(x)}{x}$$
$$f_2(x) = x^2 \exp(-x)$$
$$f_3(x) = \frac{1}{1+x^2}$$

You can make use of the functions in the `numpy` library: `exp()`, `sin()`, and the value of `pi`. Take a sufficiently high number of x values so the curve looks smooth, you can use the `numpy` function `linspace()`. **Save the plot** under the file name: `output_plot_math.png`. **Save your solution** as a script with file name: `solution_5.py`.