

Assignment 1: Programming Basics (II)

Deadline: Sep 24, 2019; 12:00 CET

Introduction This exercise is about improving your basic Python programming skills from the initial tutorial. Please follow the instructions and complete the following tasks.

Rationale With this exercise you will apply the knowledge and skills you have learned from the initial tutorial. You will implement some simple functions and algorithms yourself and learn how to use already existing functions from external libraries. The goal is to make you feel more comfortable with implementing your own Python programs and solving some basic algorithmic problems.

Prerequisites Warm-Up Assignment, `numpy` package (Task 3), and the *Booksite* library (Task 4)

- ① **Task 1** Create a file `max_array.py` and inside this file a function `max_element(array)` that returns the largest element of the array contains `int` or `float`. Create a client function to call the `max_element()` function and test your code with different arrays. (1 Point)
- ② **Task 2** Write a simple coin flip simulator and betting system. Create a file `gambling.py`. The program should take two arguments the user enters from the command line: 1) the stake, and 2) the cash goal to be reached. In the beginning the current cash should be initialized with the stake. We will use the function call `random.randrange(0, 2)` to get a random integer of 0 or 1 to simulate the coin flip. If we get a 0 we win and increase the current amount of cash by 1. If we get a 1 we lose and decrease the current amount of cash by 1. The coin flip should be repeated until we reach the goal or are broke, i.e. we don't have any cash left. Count the number of flips and print them on the console. Note that in order to use the `random.randrange()` function we need to add the statement `import random` at the beginning of the program. (2 Points)
- ③ **Task 3** In this exercise we will use the Python `numpy` package to solve some simple mathematical problems. To use and manage external libraries in Python a program called *Pip* is usually used. By typing `pip --version` on the terminal of your computer you can see if it is already installed. Note that if you use an IDE (e.g., Thonny or PyCharm) you will probably have to install the libraries in a way specific to the IDE. To install `numpy` type `pip install numpy` on your terminal; for Thonny go to *Tools – Manage Packages*¹ in Thonny and search and install *numpy*. Once installed add the statement: `import numpy` to the beginning of your Python file. (3 Points)

- Task 3.1: Create a file `max_arrays.py` and inside this file a function `maximum_arrays(a,b)` that compares both arrays `a` and `b` element-wise and returns a new array containing the larger elements. Use the `insert`² function to add new elements to a list.

Example: from applying the function to the arrays `a=[12,5,8,19,6]` and `b=[3,6,2,12,4]` the result should be `c=[12,6,8,19,6]`.

¹<https://github.com/thonny/thonny/wiki/InstallingPackages>

²<https://www.programiz.com/python-programming/methods/list/insert>

- Task 3.2: Create a file `max_arrays_simple.py` and create a function `maximum_arrays(a,b)` that does the same as in Task 2.1. Use the function `maximum(a,b)`³ from the `numpy` library to solve the task. Be careful here: the `numpy.maximum` function returns a specific array⁴ that needs to be converted to the built-in list (`.tolist()`).
- Task 3.3: Create a file `loan_calculator.py`. Inside this file create a function `calculate_payment(i,m,p)` that calculates the monthly payment rate for a loan with interest `i`, for `m` months and the borrowed principal `p`. Can you find suitable functions from Python libraries for that? Your function should return the absolute value of the monthly payment rounded to two decimal points.

④ **Task 4** In the lectures and exercises we will sometimes refer to an external library as part of the *Introduction to Programming in Python* book by Sedgewick et al. Follow the instructions given under “To get started”⁵ to install the *Booksite* library (introc-1.0)⁶ for your operating system and/or IDE. To install Sedgewick’s booksite library, download and unzip the `introc-1.0.zip`⁷ file and follow the instructions according to your operating system. To verify that it has been installed successfully, open a terminal, enter `python` and then enter `import stdio`. If this does not bring up an error then everything works fine. Note that if you use an IDE to execute your Python code, you will have to install the library in a way specific for the IDE (e.g., in Thonny go to Tools – Manage Packages – Install from local file and select the downloaded zip-file). (4 Points)

- Task 4.1: Study the `stdio.py` class from the library. Create a file `read_integers.py` with a function called `read_file()`. This function should read in a text file via the standard input, convert the contents to integers and return the integers in an array. Can you find a suitable function in the library to help you with this task? Find out how to pass a text file to the standard input when executing your program. Try your program with a text file containing just regular integers separated by whitespaces. Print the result on the console. What happens if the text file also contains other strings? Can you explain what the function in the library does? Explain these questions in the `README.txt` file (see below).
- Task 4.2: Study the file `scale.py`⁸ from Sedgewick’s book. You should be able to explain what each line is doing. Create a copy of the file with the new name `ratioscale.py` and modify the file such that the user does not need to provide the new height of the picture via the terminal. The height of the target picture should be calculated based on the width-to-height ratio of the source image. Test your program with the `mandril.png`⁹ file and different new target sizes. Note that you may have to install the `pygame` package via Pip for the program to run.

⑤ **Hand-in Instructions** Make sure to put your student ID and name at the beginning of all the Python files as below.

```
#!/usr/bin/env python3
# XX-YY-ZZZ (Your ID)
# John Doe (Your name)
...
```

Create a file `README.txt` that states 1.) any reference that you used to complete this assignment, 2.) pitfalls you encountered, and 3.) short explanations of your solution if necessary.

³<https://docs.scipy.org/doc/numpy/reference/generated/numpy.maximum.html>

⁴<https://docs.scipy.org/doc/numpy/reference/generated/numpy.ndarray.html>

⁵<https://introc.cs.princeton.edu/python/home/>

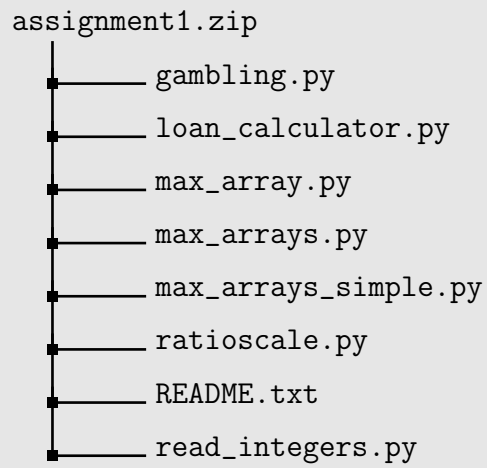
⁶<https://introc.cs.princeton.edu/python/code/>

⁷<https://introc.cs.princeton.edu/python/code/dist/introc-1.0.zip>

⁸<https://introc.cs.princeton.edu/python/31datatype/scale.py.html>

⁹<https://introc.cs.princeton.edu/python/31datatype/mandrill.png>

Compress all the Python files and the `README.txt` to a `zip` file. Upload the zip file to your exercise group's Course page on Canvas. (See Figure. 1 for the list of files should be included in the zip file.)



```
assignment1.zip
├── gambling.py
├── loan_calculator.py
├── max_array.py
├── max_arrays.py
├── max_arrays_simple.py
├── ratioscale.py
├── README.txt
└── read_integers.py
```

Figure 1: The required files for submission