

Exercise Sheet #1

What are the exercise sheets? The exercise sheets are a (almost) weekly practice sheet meant to prepare you for the semester projects. Working through them is highly recommended but entirely voluntary. You can hand them in if you want feedback on your work, but you won't get a grade for it. You can also work on them during the tutorial sessions and ask questions.

Problem 1 (*Python – Getting started*)

If you are unfamiliar with Python and do not have Python installed on your system, here is a step-by-step guide to get you started:

- (1) To run Python programs you need to have the Python interpreter installed. Go to <https://python.org/> to download Python for your OS. There are plenty of online resources to help you with the installation.
- (2) To write Python programs you should use a text editor. There are plenty of choices, but the most popular is Visual Studio Code. You can download it from <https://code.visualstudio.com/>.
- (3) Test your installation in the next problem.

Problem 2 (*Hello World*)

The first step when dealing with a new programming language is usually to test the installation by running a *Hello World!*-program. Create a file in your text editor called `hello_world.py` and insert the code below. A *Hello World!*-program in Python is very simple

```
1 # hello_world.py
2
3 print("Hello World!")
```

Open a terminal (MacOS and Linux) or a Power Shell (Windows) and execute the program by running `$ python <file_name>.py`.¹

¹If that does not work, replace `python` with `python3`. If there are still problems, check the name of your Python executable. Hint for googeling: The location of the executable must be in the `PATH` variable.

Problem 3 (*Functions and Control Flow: Prime Tester*)

In this problem, we write a basic function using `if`-statements and loops. Consider the following code snippet and familiarize yourself with the syntax.

```
1 # function_example.py
2
3 # Define a function
4 def my_function(my_parameter):
5     if my_parameter < 0:
6         print("I hate your negativity !")
7     else:
8         for i in range(10):
9             print(my_parameter, "for the", i, "th time")
10    return my_parameter + 1
11 # Run the function
12 my_function(-1)
13 print(my_function(2))
```

A couple notes on that snippet:

- Python is, unlike C++, dynamically typed, i.e. you do not have to specify the type of a variable (or parameter) or the return type of a function.
- Again notice that Python does not have a `main()` function. By running the script using `$ python function_example.py`, all the code written above is executed.
- The `range()` function generates an iterable object with all integers strictly smaller than the passed number, starting from zero.

Try running the snippet and play around with it. Then write a function `is_prime(num)` that takes in an integer `num` and tests if `num` is prime or not. The result of the test should be printed to the console. Test your function with a couple meaningful examples.

Hint: You might need the modulo operator `%`.