Programming – Getting started guide

Zoltán Belső, Péter Rucz

August 2023

1 Introduction

This document is intended to provide a short guide for installing the tools needed for Python development. We will rely on two components:

1. A Python interpreter, which lets us run programs written in the Python language. The interpreter also comes with a Python environment that contains standard packages for various features and a package manager which allows for downloading packages from the internet. These are the essential tools for software development using Python.

2. An integrated development environment (IDE), which makes writing Python programs more comfortable. The development environment comes with a graphical user interface which facilitates running and debugging codes. Furthermore, the IDE offers a good amount of helpful tools in writing our programs, such as code auto-completion.

Important: during the course we will use Python 3. The older version, Python 2 still exists, but generally Python 3 programs cannot be run on a Python 2 interpreter. When downloading Python, or looking at Python codes online, always make sure that Python 3 is considered.

2 Installing the Python interpreter

2.1 Windows

2.1.1 Installing from download

For Windows operating systems, you can download the installer of the Python interpreter from `https://www.python.org/downloads/windows/`. At the time of writing, the newest version is 3.11.4. Download the installer for the latest version that is supported on your operating system. **Notice:** if you have an older version of Windows (such as Windows 7) you may have to choose an older Python version, for Windows 7, version 3.8.10 is recommended.

Figure 1 displays the steps of the installation process after starting the downloaded installer file. Note that the window decorations may be different on your system depending on your Windows version and the decoration settings of your operating system. In the first step, you can choose from a default installation and a customized one. In this screen it is highly recommended to check the "Add Python 3.x to PATH" checkbox. This will allow for starting the Python interpreter from the command line without entering its full path. After this, you can select either the "Install Now" or the "Customize installation" option. The subsequent parts of the figure show the steps of the customized installation. Note that if you want to install the interpreter into a directory other than what is shown under "Install now", you have to select the customized installation.

In Figure 1(b) you see the optional features that can be installed together with the Python interpreter. It is recommended to install all the features. During the semester we will use `tcl/tk` for developing programs with graphical user interfaces. Therefore, you will need this feature.
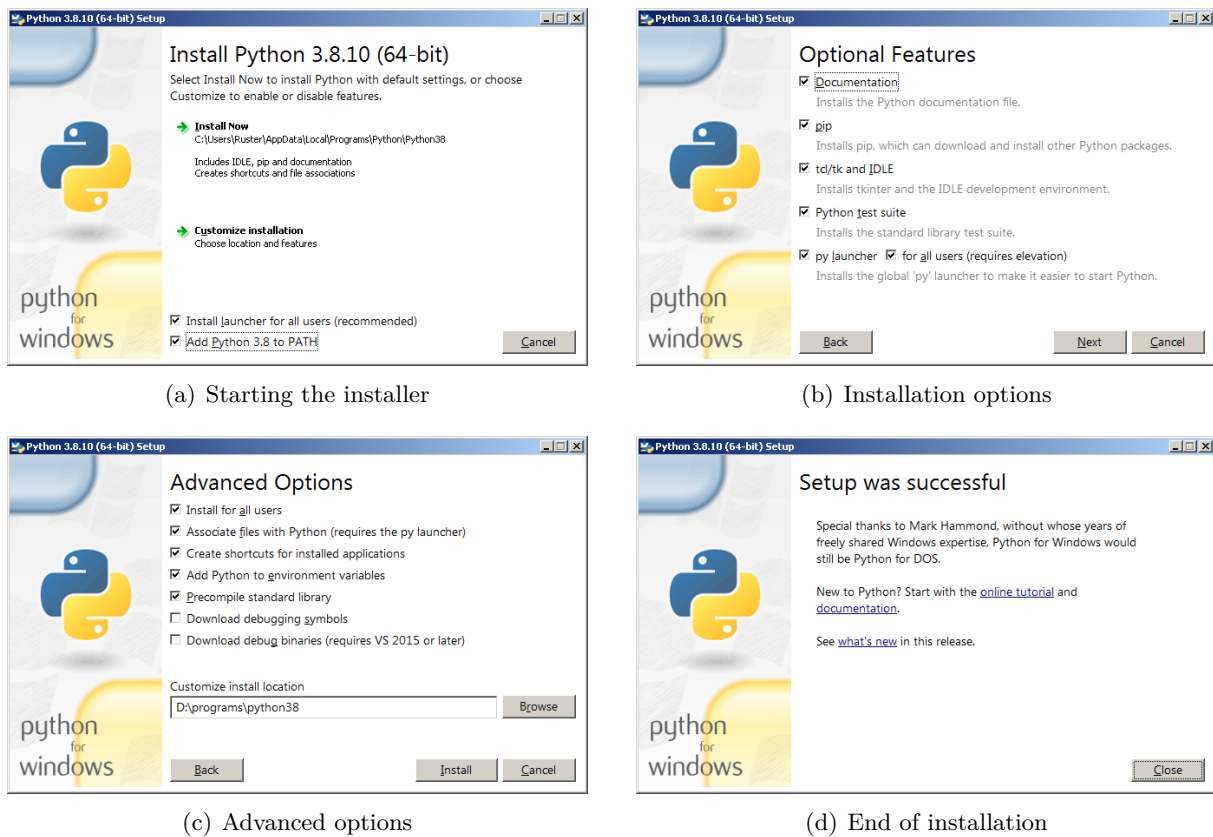
(a) Starting the installer

(b) Installation options

(c) Advanced options

(d) End of installation

Figure 1: Steps of installing the Python interpreter

Notice that if you check the "for all users" checkbox, you will need to have administrator rights on the computer. If you don't have admin privileges, uncheck that checkbox.

In the next step, shown in Figure 1(c) you can set up advanced options. Most importantly, you can select the installation folder in the "Customize install location" edit box, using the "Browse" button. You can leave the other options in their default state. Finally, the installer will finish, and you can start using Python.

2.1.2 Installing from Microsoft Store

You can also install the Python interpreter from the Microsoft Store, shown in Figure 2. Make sure to use the app provided by the Python Software Foundation.

2.1.3 Verifying the installation

After installation, it is very useful to check if the Python interpreter is detected by the system. Open a command prompt: you can do it by clicking on the "Start" button and executing the program called "cmd" by typing its letters after clicking on "Start". Then, running the command `where python` should give the path of the executable file of the Python interpreter. If the directory of your Python interpreter is on the system path, you should see the full path of the executable `python.exe` appearing in the next line of the command window, as shown in the top of Figure 3. If the interpreter is not found on the path you will see a message appearing in the command window that the file could not be found. In this case, you should adjust your path variable in order to include the Python install directory. (You can do this in the system settings, environmental variables section on your Windows. If you change the path variable, you will need to restart the command prompt application to see the effects of the changes.) If
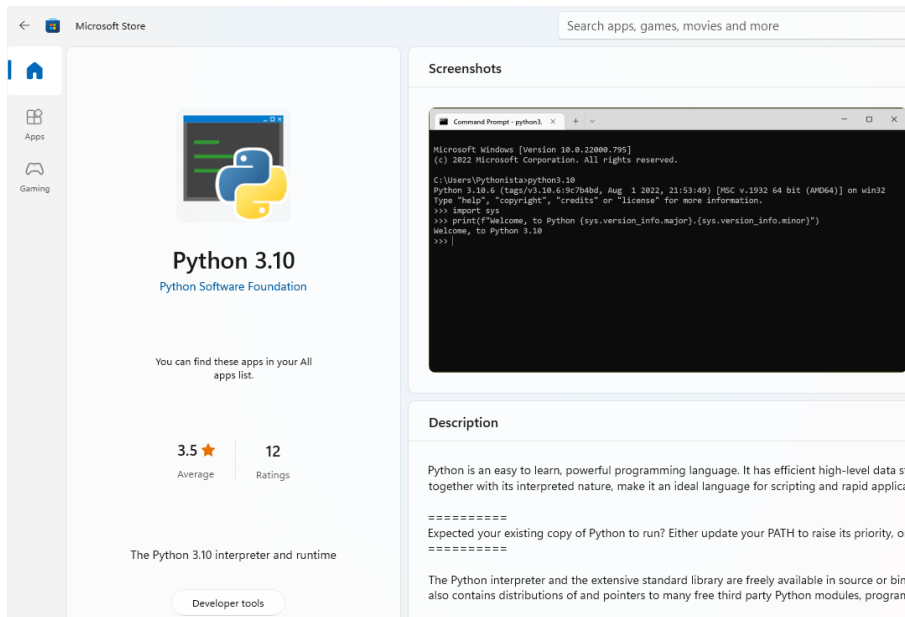
Figure 2: The Python interpreter in the Microsoft Store
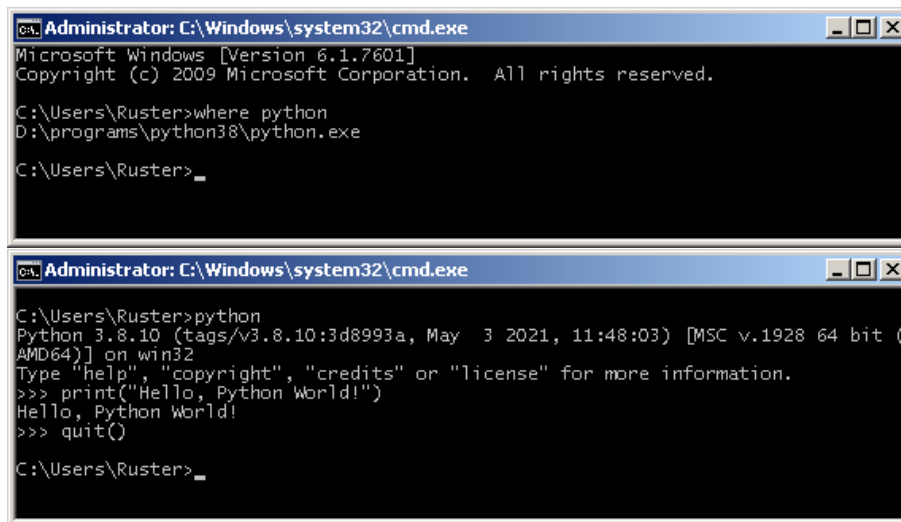

Figure 3: Checking Python installation and running the interpreter

multiple lines are shown, that means there are multiple interpreters installed on your computer. In this case, executing the command `python` will start the program listed in the topmost row. Once you see the correct path to the interpreter, you can proceed to the next step.

Finally, you can start the interpreter by typing `python` in the command window. This will print the version of the interpreter program and you will get a Python prompt indicated by `>>>` in the command window. Here, you can start typing Python commands. By typing `print("Hello, world!")` you should see the text "Hello world!" appearing in the next line. If this works, your Python interpreter is ready to be used. You can exit the interpreter by typing the command `quit()`. After that you can close the command window by using the `exit` command or clicking on the window closing "X" button.

2.2 Linux

In Linux, you can use your package manager to install Python 3. For example, on Ubuntu, you can install it by executing the following commands:

```
sudo apt-get update
sudo apt-get install python3
```

After the installation was successful, start the Python interpreter by running the command:

```
python3
```

This will start the Python 3 interpreter. In the interpreter, you can type Python commands. You can exit the interpreter using the command `quit()`.

In other Linux distribution, you should look for the package `python` or `python3` and install it using the package manager software coming with your Linux distro.

2.3 Mac OS

You can find the Python 3 installer for Mac OS here: `https://www.python.org/downloads/macos/`.

You can also find an instructional video on the step by step installation of Python 3 on Mac OS here: `https://www.youtube.com/watch?v=0hGzGdRQeak`

3 Getting started with the IDE

3.1 Installation

We recommend using the PyCharm development environment. It's a cross-platform environment, with a free community edition. If you prefer using an other IDE, feel free to do so, but make sure that you choose a program that you can comfortably work with. Support from us regarding the usage and configuration of the IDE will only be provided for PyCharm. Here we will show the steps of installation for Windows only, but you can also install it on Linux or MacOS.

You can download PyCharm from here: `https://www.jetbrains.com/pycharm/download/#section=windows`. Make sure you download the community (free) edition of PyCharm, it is under the professional edition, in the dark gray section of the download site. At the time of writing, the current version of PyCharm is 2023.2. Clicking on "Download" will download the installer. Once the downloading is complete, launch the PyCharm installer program.

When the installer is started, you have to go through the steps shown in Figure 4. If you want to change the installation path from the default one, you can do it in the second step, as shown in Figure 4(b). When the installation is completed, you can start the development environment from the Start menu, the Desktop shortcut just created, or by checking the "Run PyCharm Community Edition" checkbox in the last step of the installation process.

## 3.2 Setting up the first project

When starting PyCharm the first time, you will need to configure the interpreter. The steps of creating the first project are shown in Figure 5. At the first launch of PyCharm, you will see a welcome window, as shown in Figure 5(a). You can proceed by clicking on the "New Project" button. This will take you to the screen shown in Figure 5(b). First, configure the interpreter for the project, which can be done in two ways. It is recommended to use the option "New environment using" and selecting the "VirtualEnv" option. This will create a so-called virtual environment project that is beneficial as it allows installing packages locally and helps avoiding conflicts of package versions used by different projects. When choosing this option, make sure to select the correct Python 3 interpreter program installed in the previous section as the "Base interpreter". Alternatively, you can set up the system interpreter, by selecting the option "Previously configured interpreter" and then clicking the "Add interpreter" button and the "Add Local Interpreter" in the dropdown menu. Clicking on this menu will take you to the
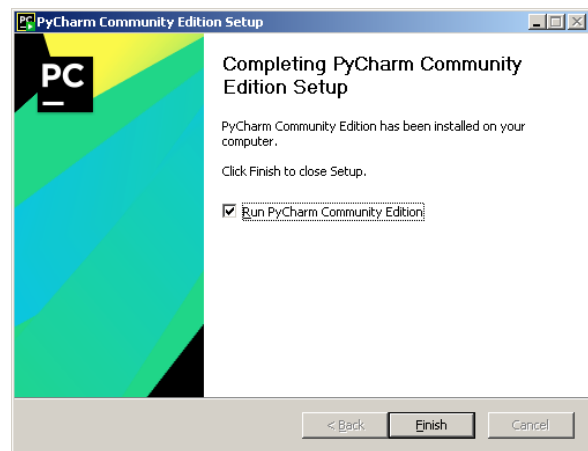
(a) Starting the PyCharm installer
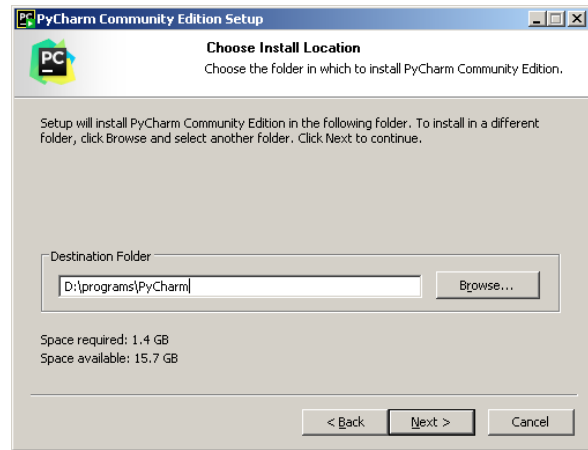
(b) Choosing installation location
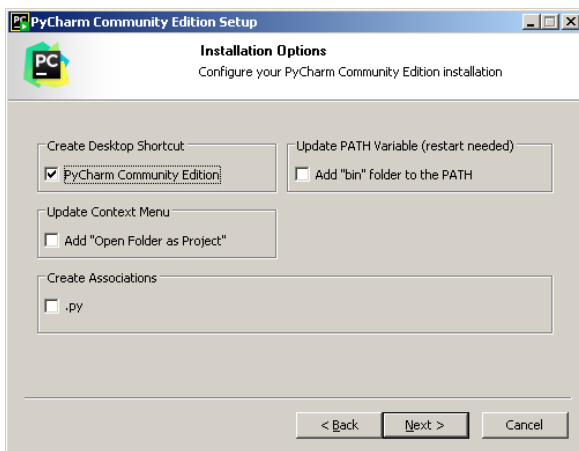
(c) Options

(d) Installation complete

**Figure 4:** Steps of installing the PyCharm development environment

next screen, shown in Figure 5(c). On this screen select the "System Interpreter" option, and you should see the full path of your Python 3 interpreter program (`python.exe` on Windows) appearing in the edit box on the top right. If the program does not appear, or the path is not correct, you can select the path where you have installed the interpreter in the previous section manually. Once the path of the interpreter program is set up correctly, you can close the window by clicking on the "OK" button. This will take you back to the previous screen, shown in Figure 5(d). Here, you may uncheck the checkbox "Create a main.py welcome script" on the bottom. You will also need to set up the path of the project in the top of the window. You can choose any directory of your preference.

We recommend that you create a main directory for the Python projects created during the semester. Then, you can add subfolders to this main folder for the exercises each week. In the first part of the semester we will solve more smaller tasks during the laboratory sessions. For these labs, you can create one file for each exercise. In the second half of the semester, one exercise will be longer and may contain multiple files. When you created the directory for your project and set it up in the window, you can create the project by clicking the "Create" button.

When the project is created, the main editor window is opened, shown in the screenshots of Figure 6. To start working with the project, you first need to add a Python file to the project. To add a file, right-click with the mouse on the name of the project and select the "New" → "Python file" options from the appearing menus, as shown in Figure 6. A small window will appear in the middle of the screen and you are asked to enter the name of the file, as depicted
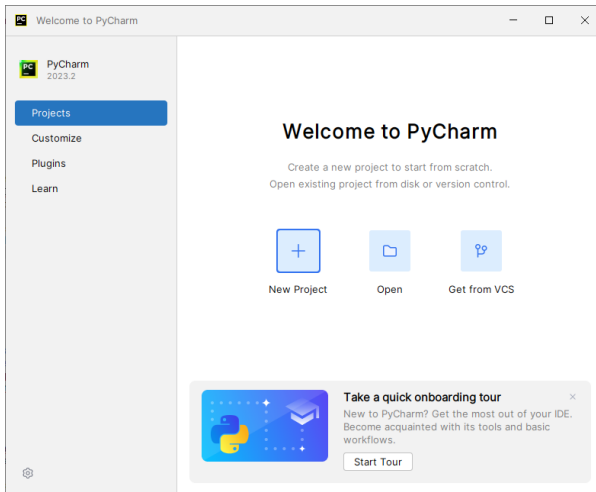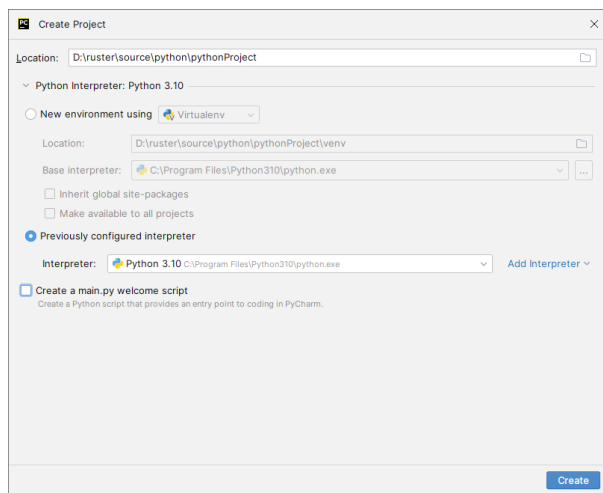
(a) Welcome window of PyCharm
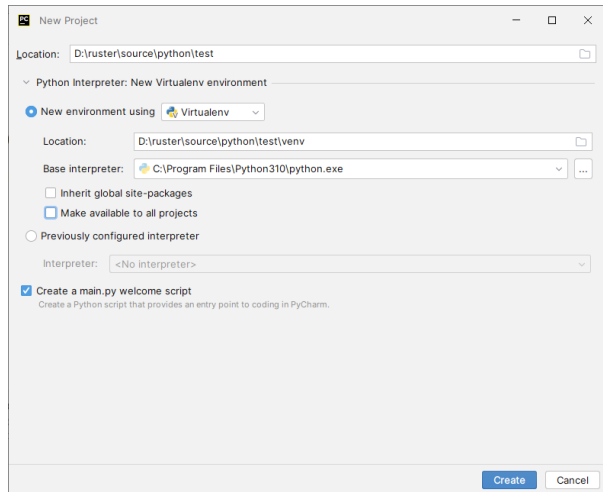
(b) Setting venv interpreter for the project
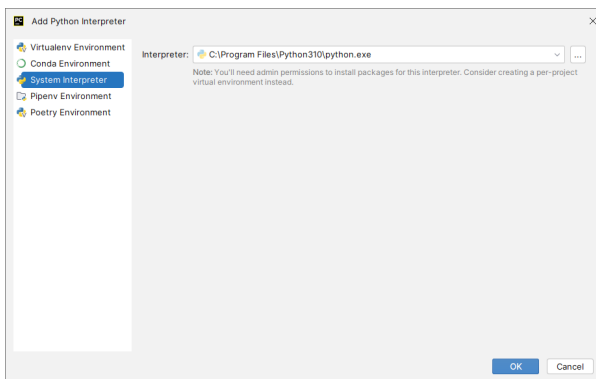
(c) Alternative: setting up system interpreter

(d) Finalizing settings for the project

**Figure 5:** Steps of setting up our first project in the PyCharm development environment

in Figure 6(b). When you have entered the name of the file (use the file extension `.py`, in our example `main.py`), the new file will be opened and you can start editing the file.

You can try out the "Hello, world!" program, as shown in Figure 6(c). When you have typed in the source code of the program, you can execute the program in the following ways:

1. Selecting the menu option "Run" from the menu accessed through top left menu button of the editor window.
2. Clicking the green triangle button in the top right region of the window.
3. Right-clicking in the editor window and choosing "Run" from the context menu.

You will see the output of the program in the bottom of the window, where the "Run" tab will be activated automatically.

Notice: PyCharm screenshots in this document were taken using the "IntelliJ Light" theme. You can change the appearance and the theme in the "File" → "Settings" menu, by selecting "Appearance & Behavior" → "Appearance" in the option tree on the left hand side. Then, you can change the theme in the settings panel appearing on the right.
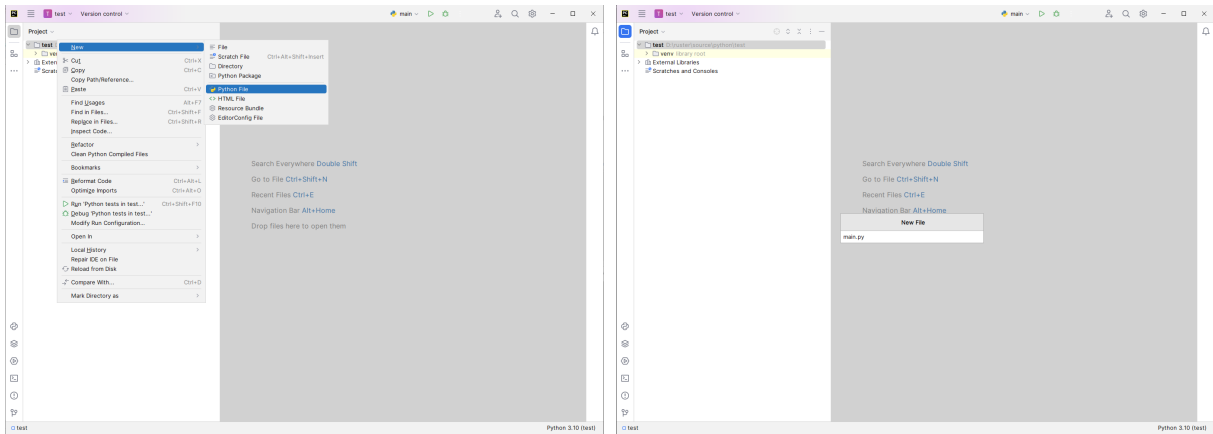
(a) Adding a file to the project

(b) Entering the name of the new Python file
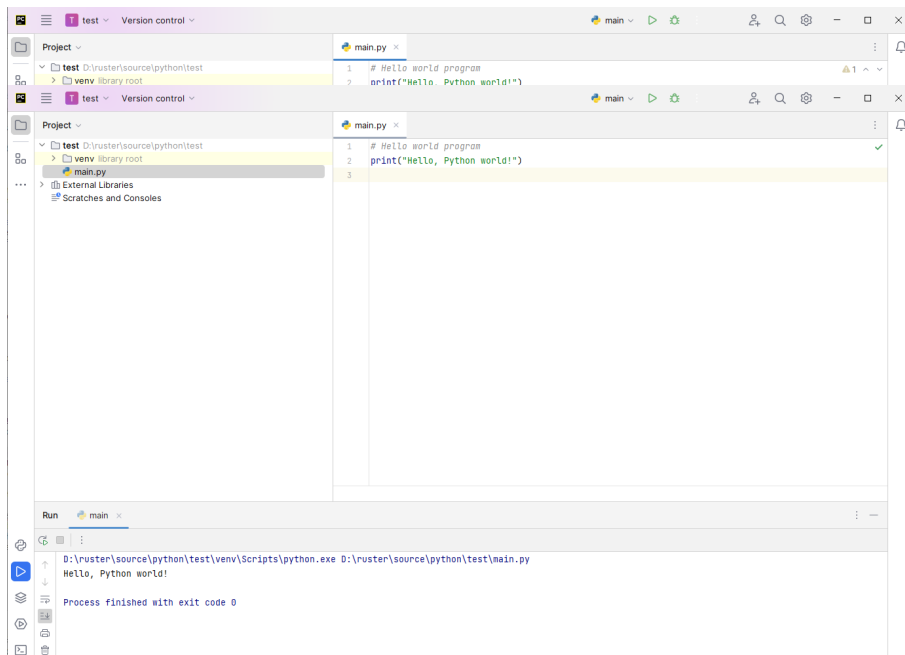
(c) Editiing and running the program

Figure 6: Creating and running the "Hello, world!" program in PyCharm

4   Installing packages

Later in the semester we will use some packages that are not installed as default. One such package is `numpy`, a package supporting matrix operations similar to Matlab. To install such packages you can use the program `pip` from the command line. For example, to install `numpy`, you can run:

```
pip install numpy
```

If you also have Python2 interpreters installed on your system, you need to call `pip3` instead of `pip`.

You will also be able to install packages using PyCharm, by left-clicking the name of the missing package after the `import` keyword and selecting the "install package" option in the context menu.

We will learn more about installing and using packages during the semester.

5 Closing words

This document was intended to help you getting started in using Python 3 on your computer. During the semester, we will learn how to write Python programs and how to use our development environment for writing Python programs, debugging, and testing in an efficient way. In the meantime, feel free to experiment with Python and the development environment.

If you have questions regarding the installation steps described in this document, or you encounter errors, please contact us by email: belso@hit.bme.hu (Zoltán Belső) and rucz@hit.bme.hu (Péter Rucz).