python™

Practical Workshop
Introduction to Python Fundamentals

Juan Carlos Miranda – UdL GRAP

Universitat de Lleida
Grup de Recerca
AgròTICa i Agricultura de Precisió (GRAP)

ΓΕΩΠΟΝΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
AGRICULTURAL UNIVERSITY OF ATHENS

SFT GROUP
Smart Farming Technology Group

EXCELLENCE IN FOOD AND AGRICULTURE RESEARCH

CERCA
Centres de Recerca de Catalunya

Lesson 1 - Introduction

- Workshop objectives.
- Target audience.
- Methodology and evaluation.
- Requirements.
- Schedule proposed.

Workshop objectives

The objective of this workshop is to introduce to the fundamentals of programming in Python. **The contents are focused on acquiring experience through practice**. After this workshop the participant is expected to be able to:

- Organize source code logically.
- Recognize the use of variables and data types used in Python.
- Make use of flow statements: conditionals, loops, functions.

The participant is encouraged to continue to increase their knowledge through **practice and self-study**.

Target audience

- This workshop is for participants who want to improve their Python programming skills. **The fundamentals needed to start coding small programs are explained**.
- Basic concepts of algorithmic thinking are explained, but the participant must improve this ability on their own through practice.

Methodology and evaluation

- The lessons are designed **to motivate the participants** to apply the Python language in real life problems after the course. **Links to Internet sites are provided** as supplementary material. All the lessons include exercises, in order to fix the knowledge acquired.

- **Two sessions will be presented with explanation in class**, the rest will be done as consultation and accompaniment classes. **This workshop does not have evaluation exams**.

Requirements

- Participants **must have their computers**; they **must be enabled to install the tools proposed in the workshop**. Operating systems supported by Python interpreter.

Schedule proposed

Lesson	Objective	Duration	Date	Hour	Place
Lesson 1 - Introduction	Introduction to the workshop	30 minutes.	28/07/2023	15:00 – 15:30	Room 1.1
Lesson 2 – First steps with Python	Setting up the development environment	30 minutes	28/07/2023	15:30 – 16:00	Room 1.1
Lesson 3 – Fundamentals of Python	Introduction to programming in Python	1 hour	02/08/2023	15:00 – 16:00	Room 1.1
	Practical exercises	1 hour	04/08/2023	15:00 -16:00	Virtual meeting
Accompaniment class 1	Optional	1 hour	08/08/2023	15:00 – 16:00	Virtual meeting
Accompaniment class 2	Optional	1 hour	16/08/2023	15:00 – 16:00	Virtual meeting

Lesson 2 – First steps with Python

- About Python.
- Some examples developed using Python.
- Setting the development environment.
- First lesson to debug and find errors.

About Python

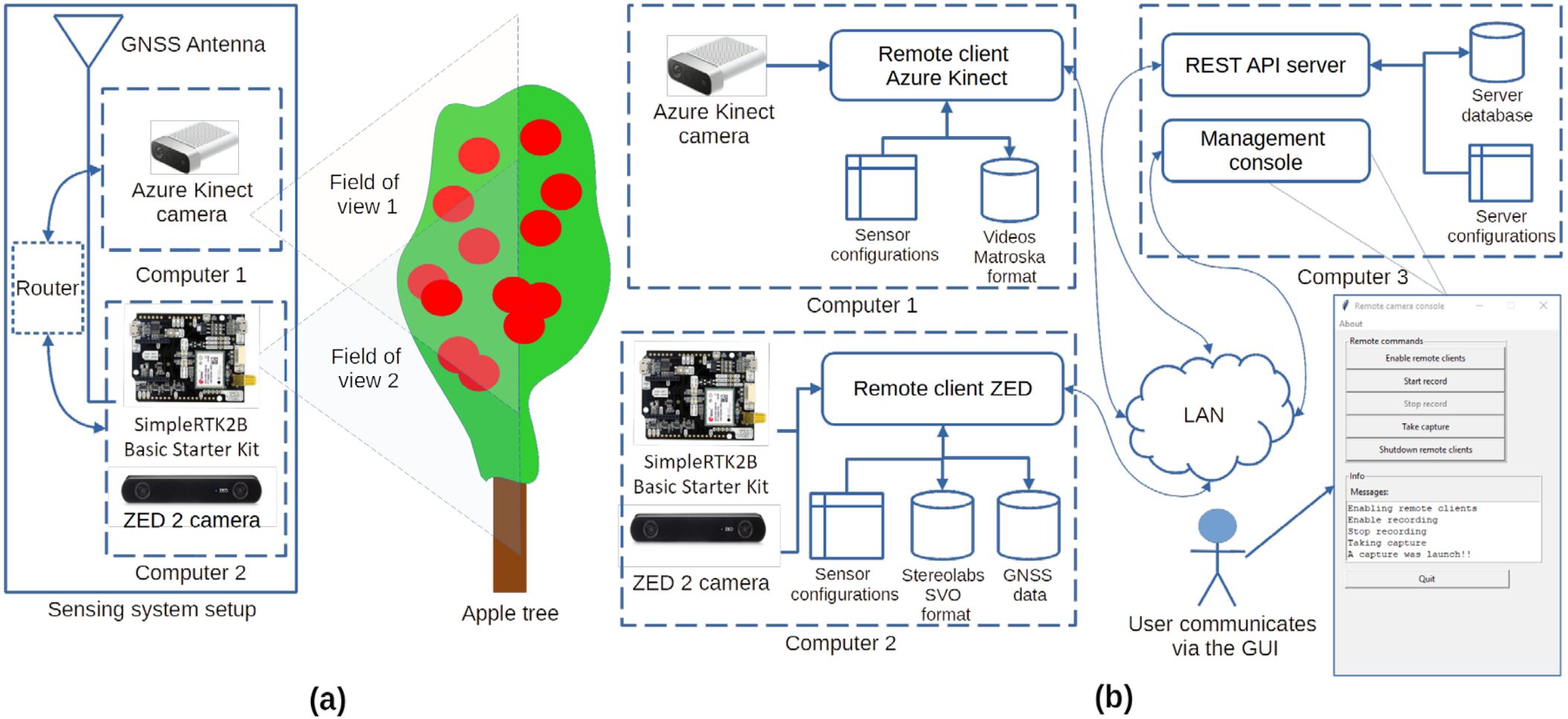
It is a general-purpose, high-level programming language. The first version was released in 1991 and was created by Guido van Rossum. The official site is [https://www.python.org/]

A large community supports the language for different uses. **There is a centralized repository for download ready-to-use packages**. [https://pypi.org/]

Some examples developed using Python

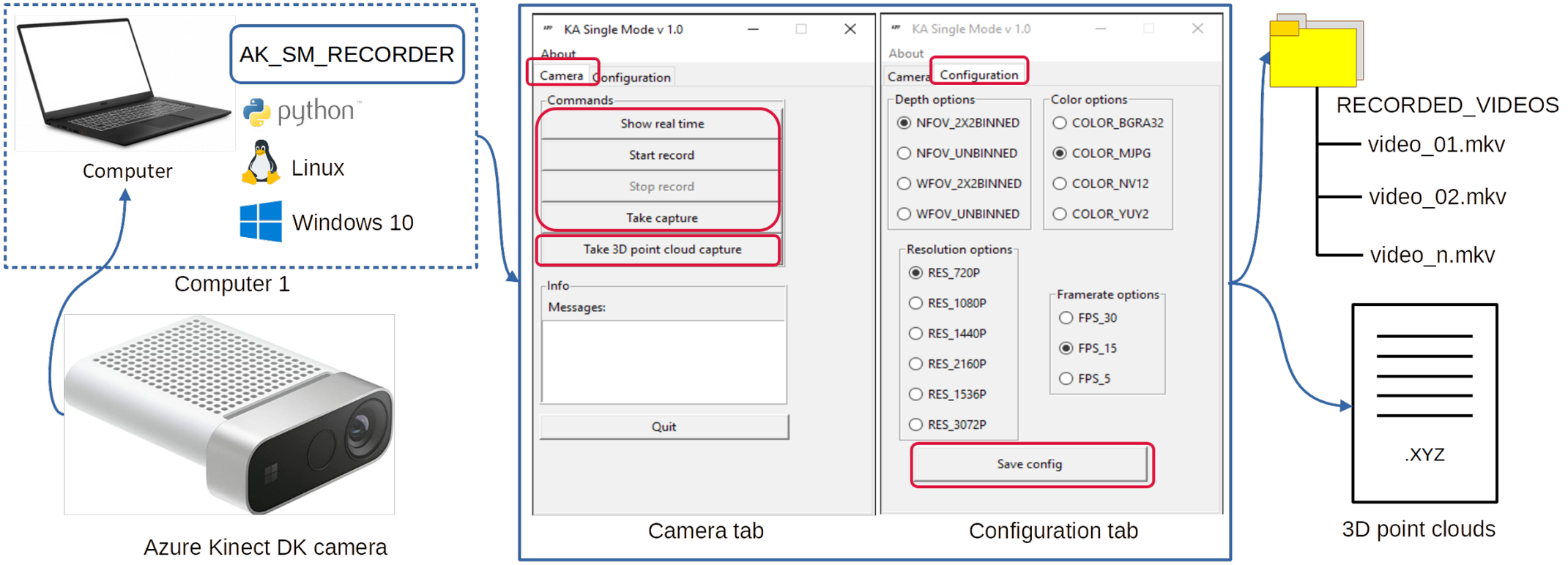
Below, in order to motivate the participants, some examples are shared, where I have applied Python to solve problems in agriculture.

AK_ACQS – data acquisition with multiple sensors



https://doi.org/10.1016/j.softx.2022.101231

AK_SM_RECORDER – data acquisition with Azure Kinect sensor

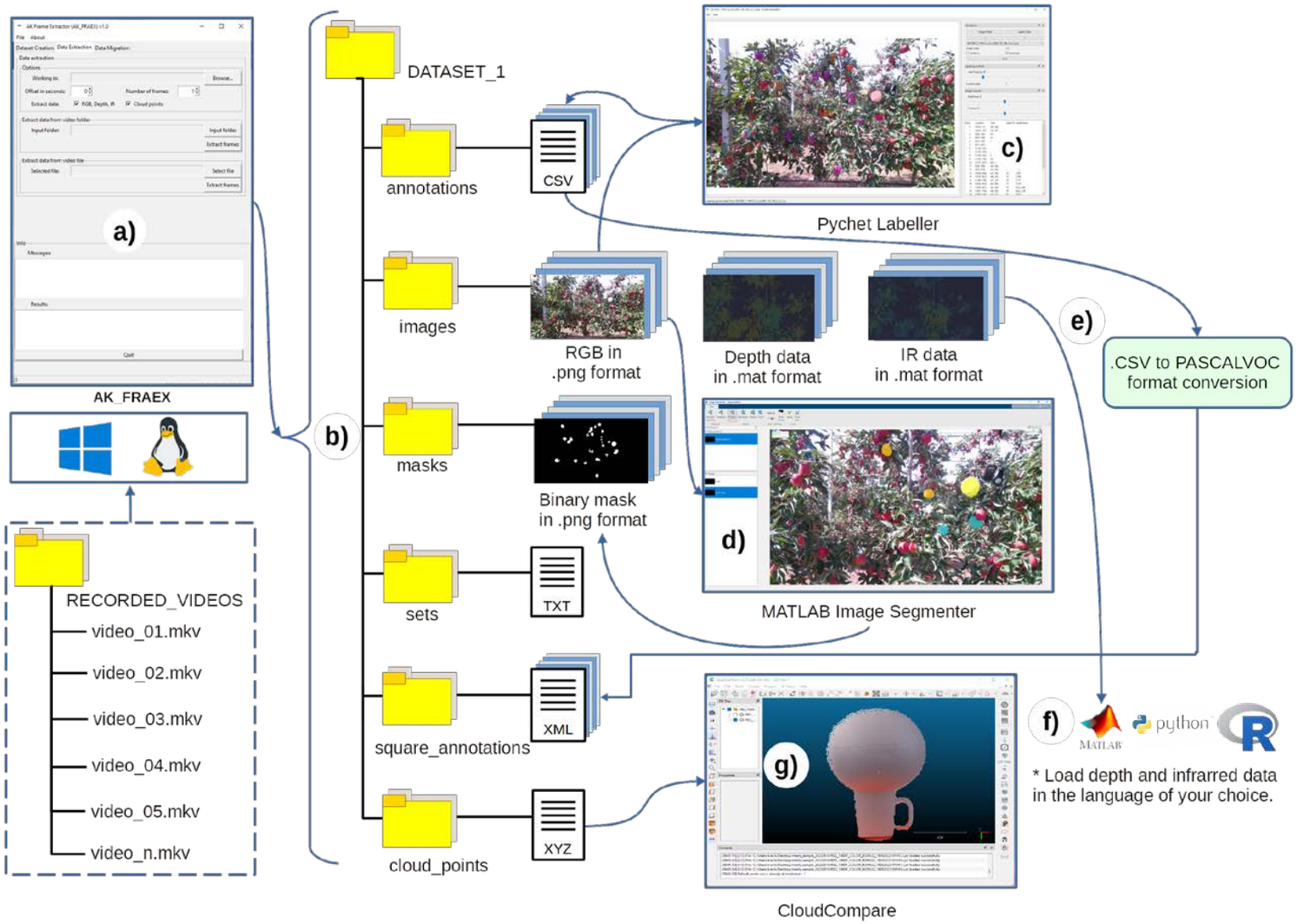


python Package Index

https://pypi.org/project/ak-sm-recorder/

pip install ak-sm-recorder

AK_FRAEX - Extraction of data recorded with the Azure Kinect sensor



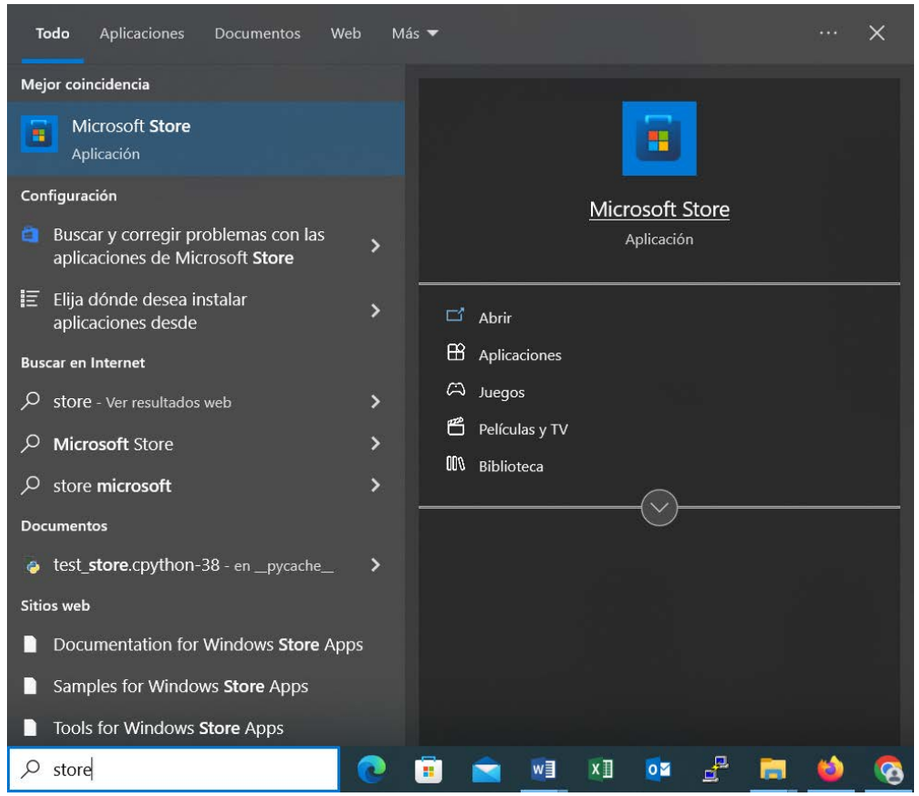
python Package Index

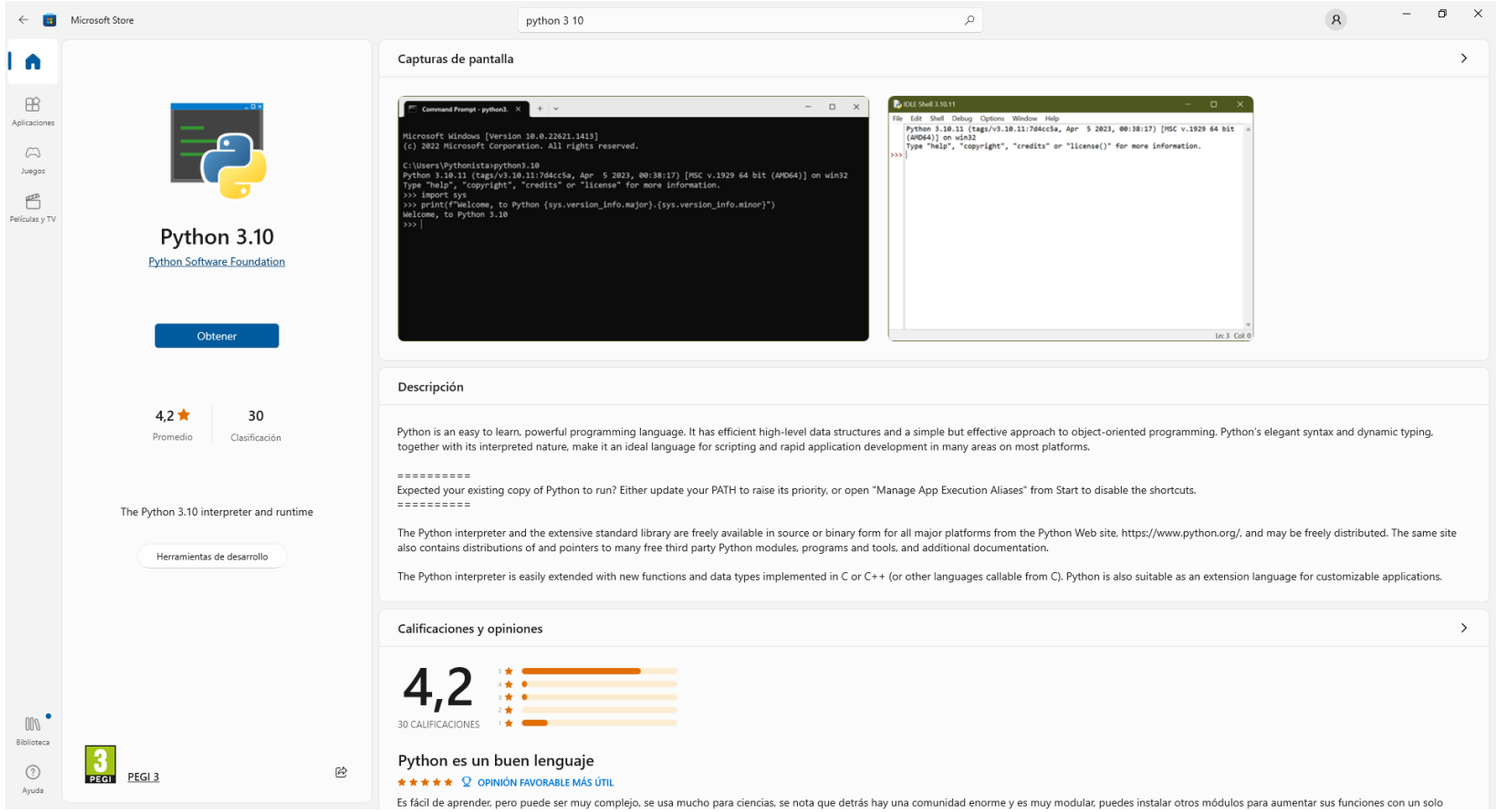
https://pypi.org/project/ak-frame-extractor/

pip install ak-frame-extractor

Setting the development environment

Download the interpreter from the official site [https://www.python.org/] or follow the instructions to install it on your favorite operating system.





Setting the development environment

There are many tools to start programming in Python, for this course we will use Pycharm [https://www.jetbrains.com/pycharm/]. Other tools suggested by the official site are listed at [https://wiki.python.org/moin/IntegratedDevelopmentEnvironments].

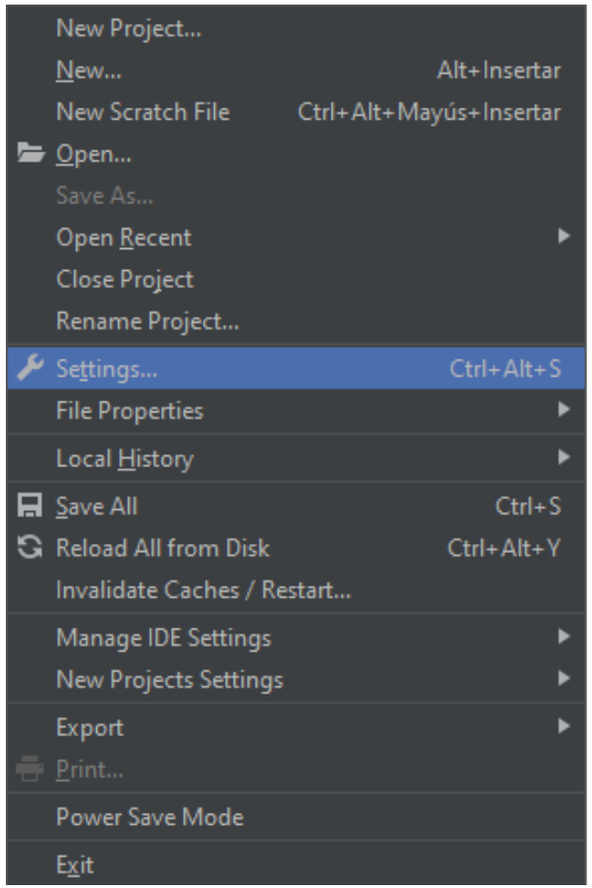
There are two ways to créate virtual environments:

- Using the tools that Pycharm offers.
- Manual creation of the virtual environment from the command line.

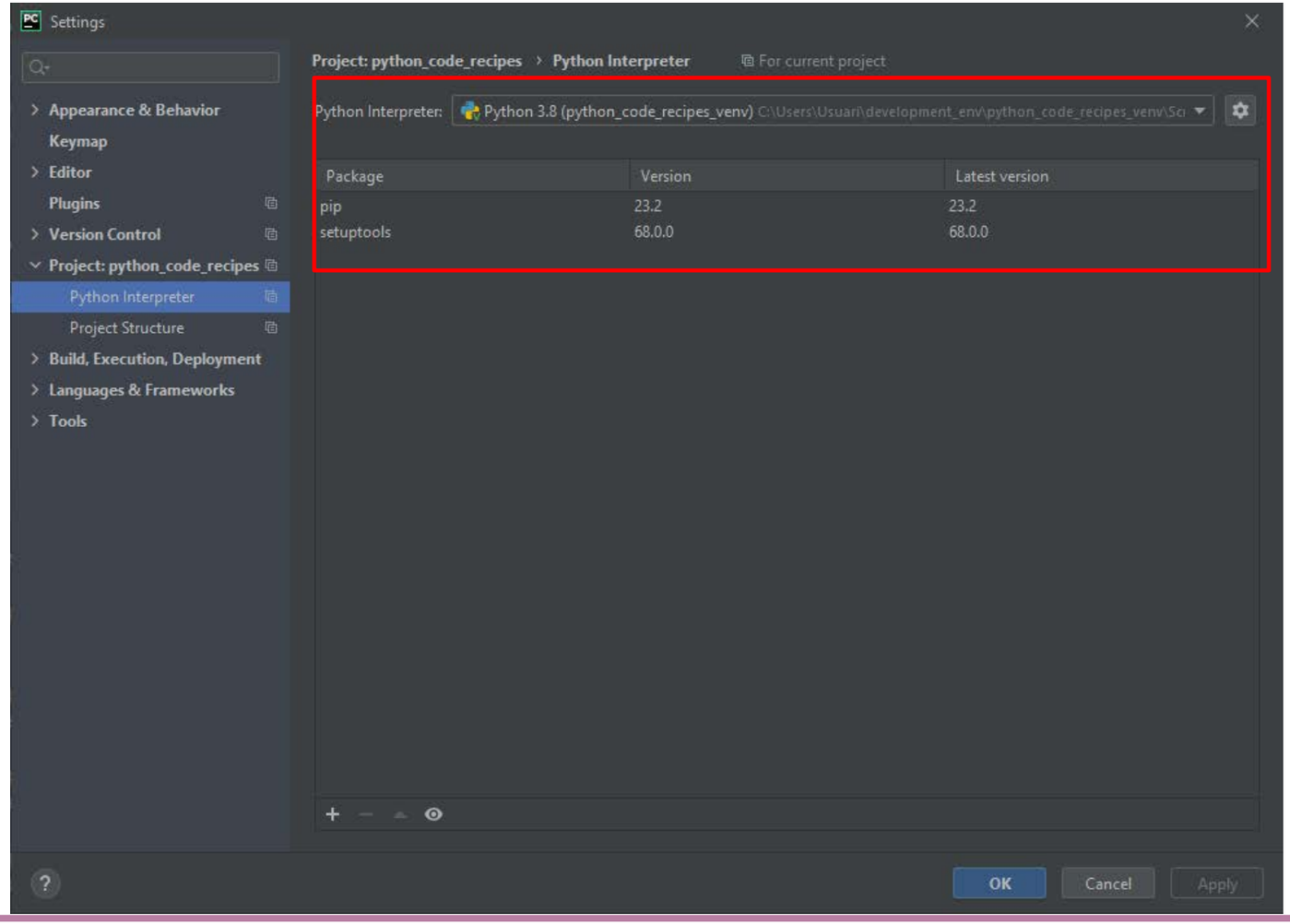
The main components of the Pycharm tool are explained bellow.

Setting the development environment -> Using the tools that Pycharm offers.

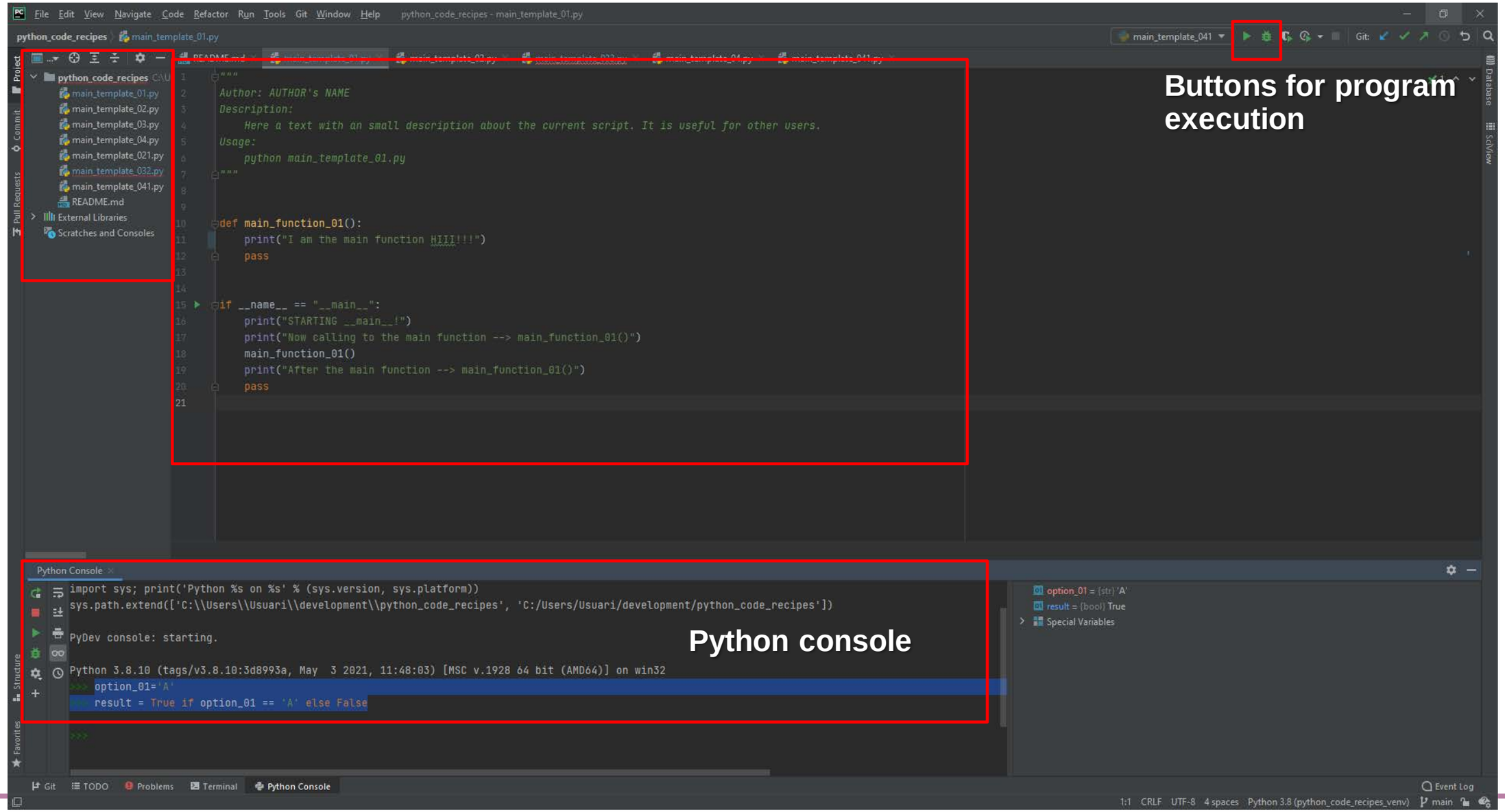
Create a new project.



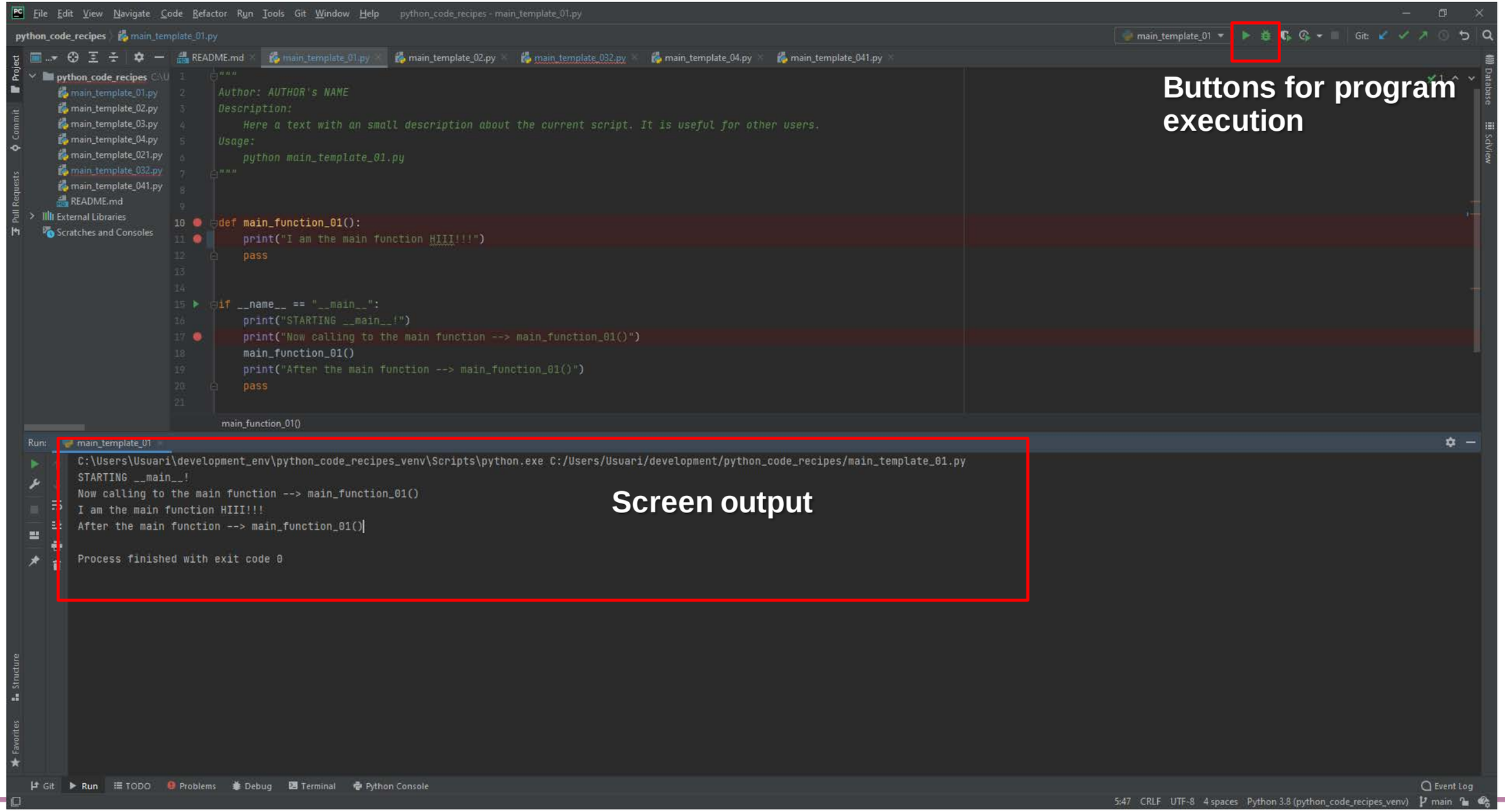
Configure the virtual environment



Lesson 2 – First steps with Python



Lesson 2 – First steps with Python



Lesson 2 – First steps with Python

Setting the development environment -> Manual creation of the virtual environment from the command line.

Creating the virtual environment

```
python3 -m venv ./python_code_recipes_venv
source ./python_code_recipes_venv/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
```

Activating the virtual environment

```
source ./python_code_recipes_venv/bin/activate
```

Windows

```
cd ./python_code_recipes_venv/Scripts
activate
```

Checking the current version

```
python --version
```

Saving installed libraries and their version number

```
pip freeze
pip freeze > requirements.txt
```

Command line

Commands used in the console to manage virtual environments for Python development.

First lesson to debug and find errors

This is a practical exercise to demonstrate the use of the debugging option.

Participants must download the repository to follow the activities. All the examples used in this workshop are published at [https://github.com/juancarlosmiranda/python_code_recipes].

Practical activity

1. Open with Pycharm "**activity_01_01.py**" that contains a simple template for Python programs and execute it.
2. Continue practicing with debugging techniques with files: "**activity_02_01.py**" and "**activity_02_02.py**". Finally solve the error in "**activity_02_03.py**".

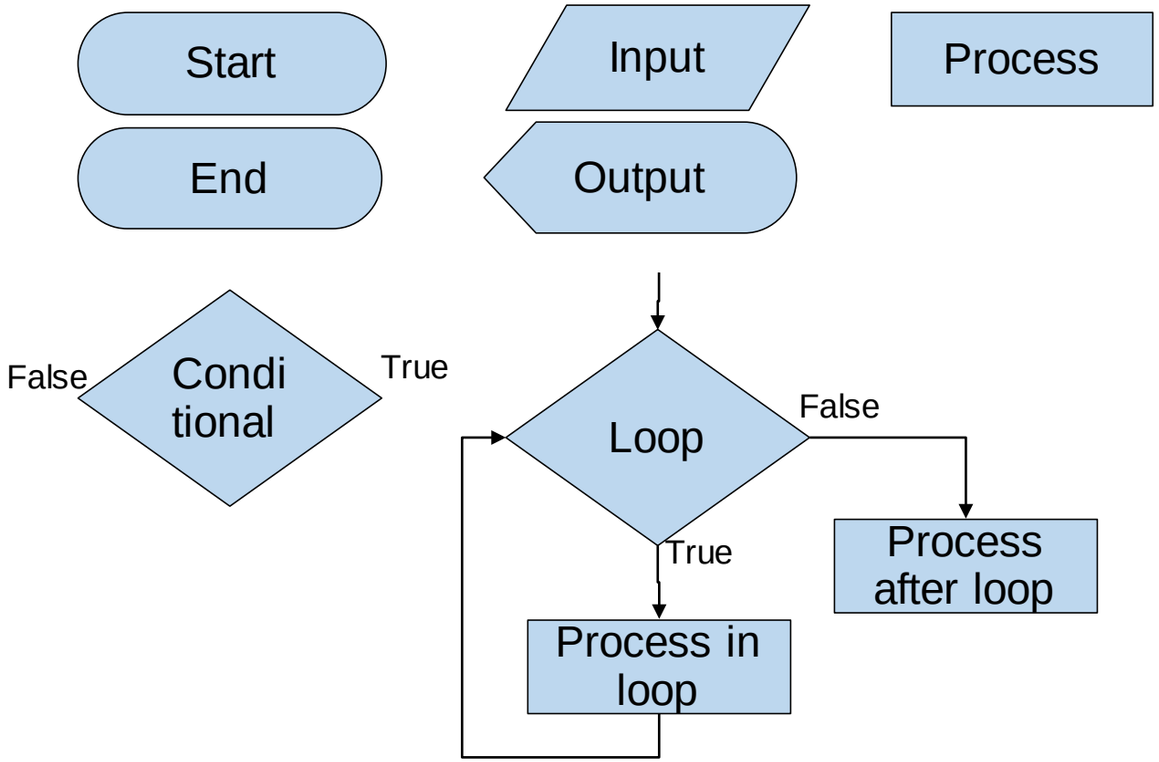
Lesson 3 – Fundamentals of Python

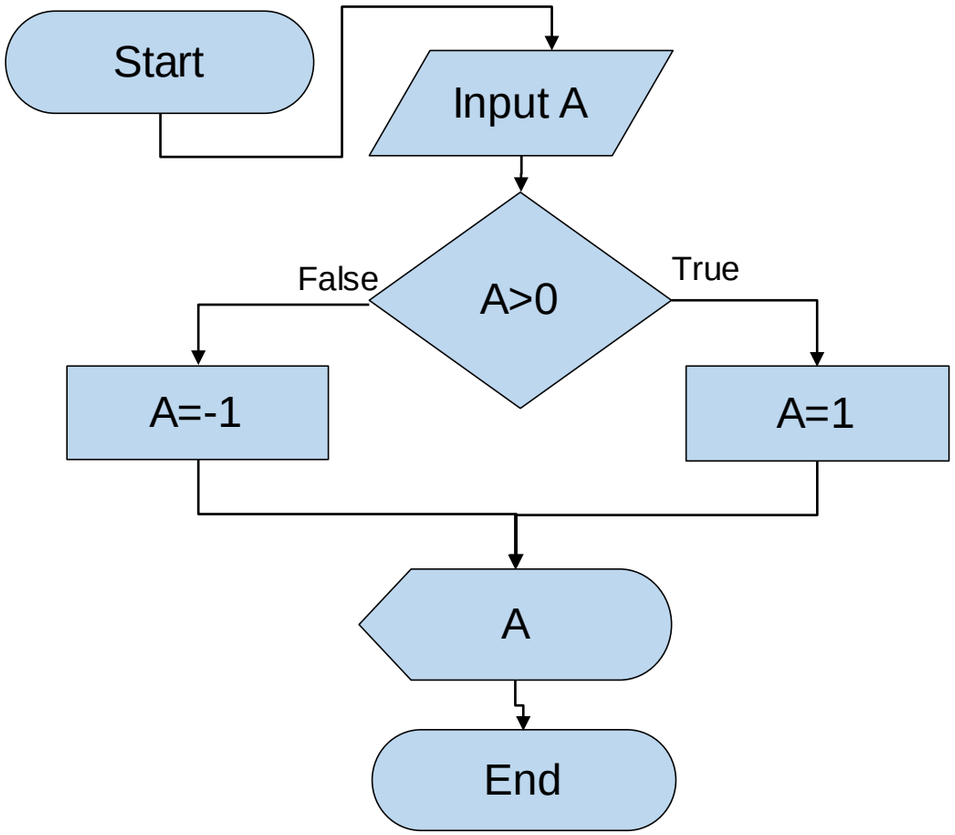
- Introduction to algorithmic thinking: flow diagrams, flow control, conditionals, loops, inputs and outputs.
- Variables: assignment, names, scope. Types: numbers, strings, lists, dictionaries.
- Control flow sentences, conditionals and loops, functions.
- Outputs: text files, CSV files.

Introduction to the algorithmic thinking

Flowcharts are an abstraction to show the operation of a program graphically. The main structures are: conditionals, loops, inputs, outputs. More information about flowcharts in [

https://en.wikipedia.org/wiki/Flowchart]





Introduction to the algorithmic thinking

To remember some concepts, we review the most used operators in programming, which will be used later.

Relational operators

- > greather than.
- < less than.
- == equal to
- >= greather than or equal
- <= less than or equal
- != not equal

Logic operators

- AND, OR, NOT.

A	B	A AND·B
False	False	**False**
False	True	**False**
True	False	**False**
True	True	**True**

A	B	A OR·B
False	False	**False**
False	True	**True**
True	False	**True**
True	True	**True**

B	NOT A
False	**True**
True	**False**

Variables: assignment, names, scope. Types: numbers, strings, lists, dictionaries

In the assignment of variables will use to encode the most used data types: integer numbers **(int),** float numbers **(float),** strings, lists, dictionaries. A full list supported by **"The Python Standard Library"** is described at [https://docs.python.org/3/library/stdtypes.html].

Practical activity

1. Open "**activity_02_01.py**", that show examples of variable management. Check the sentence structure for variable assignment.
2. Open "**activity_02_02.py**", that show examples of variable scope, local and global variables.

Control flow sentences: conditionals

This is a practical exercise to follow with Integrated Developmen Environment (IDE).

1. Open "**activity_03_01.py**", that show examples of variable management. Optional code "**activity_03_02.py**", it requires Python > 3.10.

Control flow sentences: loops

1. Open "**activity_04_01.py**" and "**activity_04_02.py**", that show examples of loops.

Control flow sentences: functions

1. Open "**activity_04_03.py**", that show examples of functions.

Managing files: text files, CSV files

1. Open "**activity_05_01.py**", that show examples of file management.
2. Open "**activity_06_01.py**", that show examples of comma-separated (CSV) file management.

Managing files: text files, CSV files

1. Open "**activity_05_01.py**", that show examples of file management.
2. Open "**activity_06_01.py**", that show examples of comma-separated (CSV) file management.

- To remember the syntax of the language, cheat sheets are helpful, here you can download **"Python Cheat Sheet for Beginners" from "Datacamp"** [https://images.datacamp.com/image/upload/v1673614153/Marketing/Blog/Python_Cheat_Sheet_for_Beginners.pdf]

Managing files: text files, CSV files

1. Open "**activity_05_01.py**", that show examples of file management.
2. Open "**activity_06_01.py**", that show examples of comma-separated (CSV) file management.

Lesson 4 – Useful information

- Official site for Python language. https://www.python.org/
- The Python tutorial. https://docs.python.org/3/tutorial/index.html
- PEP 8 – Style Guide for Python Code. https://peps.python.org/pep-0008/
- Beginner's Guide to Python. https://wiki.python.org/moin/BeginnersGuide
- Python – Introductory Books. https://wiki.python.org/moin/IntroductoryBooks
- Rosetta Code. Is a site with general information about programing languages, it is included Python. https://rosettacode.org/wiki/Category:Programming_Tasks
- Data Science Rosetta Stone. http://www.datasciencerosettastone.com/index.html
- W3 Schools. Good site for reference https://www.w3schools.com/python/
- Python Cheat Sheet for Beginners. https://www.datacamp.com/cheat-sheet/getting-started-with-python-cheat-sheet

Lesson 4 – Useful information

- The Hitchniker's Guide to Python. https://docs.python-guide.org/

Lesson 4 – Useful information

Keep practicing, improvement will come with practice.

Acknowledgements

- This work was partly funded by the Department of Research and Universities of the Generalitat de Catalunya (grants 2017 SGR 646 and 2021 LLAV 00088) and by the Spanish Ministry of Science and Innovation / AEI/10.13039/501100011033 / ERDF (grants RTI2018-094222-B-I00 [PAgFRUIT project], PID2021-126648OB-I00 [PAgPROTECT project] and TED2021-131871B-I00 [DIGIFRUIT project]).
- The Secretariat of Universities and Research of the Department of Business and Knowledge of the Generalitat de Catalunya and European Social Fund (ESF) are also thanked for financing Juan Carlos Miranda's pre-doctoral fellowship (2020 FI_B 00586).
- Thanks to the Agricultural University of Athens - Smart Farming Technology Group (SFT GROUP) for support my stay research and providing me the space and the necessary tools to carry out this workshop.

Email: juancarlos.miranda@udl.cat
Twitter: @mirandajuancar

www.grap.udl.cat

Universitat de Lleida
Grup de Recerca
AgròTICa i Agricultura
de Precisió (GRAP)

@grapudl

EXCELLENCE IN FOOD AND AGRICULTURE RESEARCH

CERCA
Centres de Recerca
de Catalunya