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.

Lesson 1 - Introduction

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**.

Lesson 1 - Introduction

Requirements

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

Lesson 1 - Introduction

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

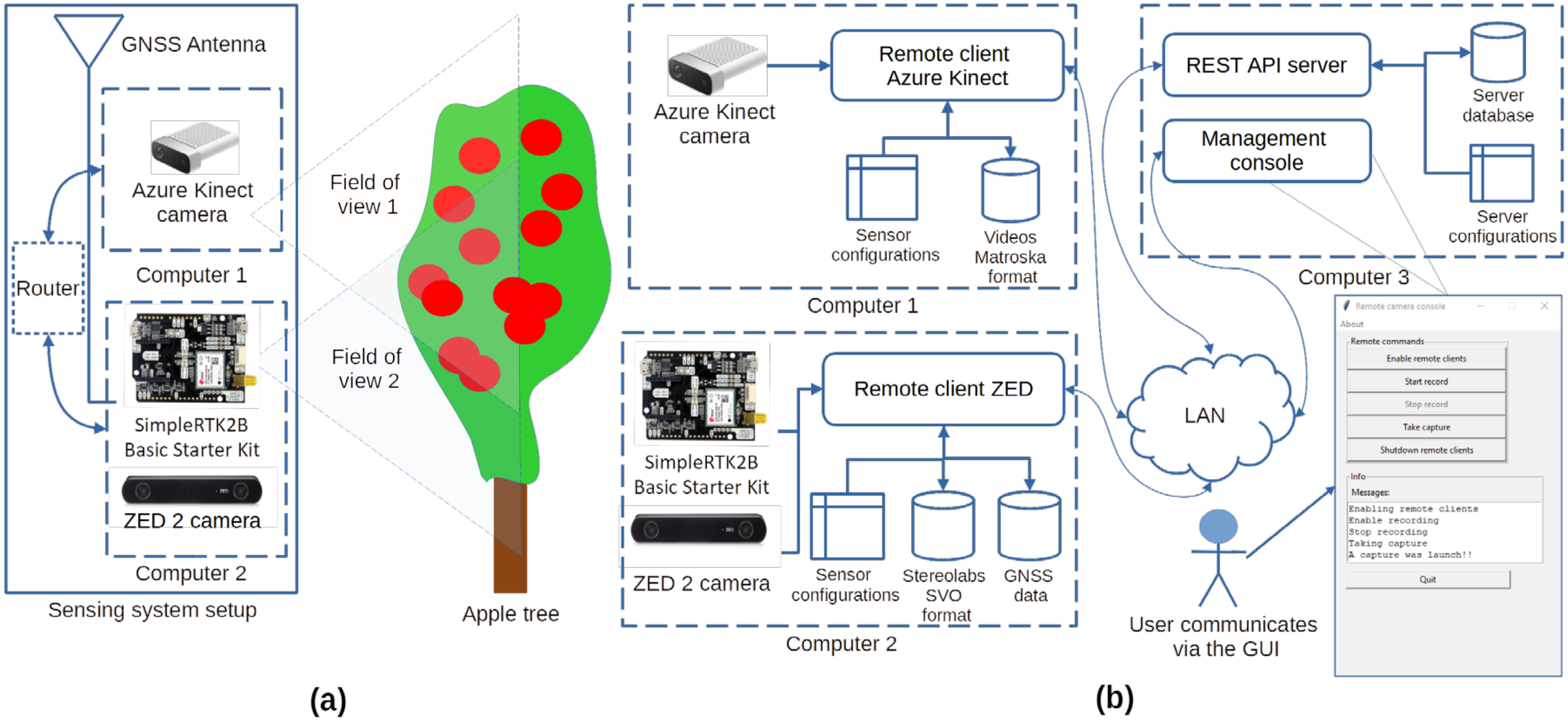
It is a general-purpose, high-level programming language. The first version was released in 1991 and was created by Guido van Rossum. The official site is [https://www.python.org/]

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.

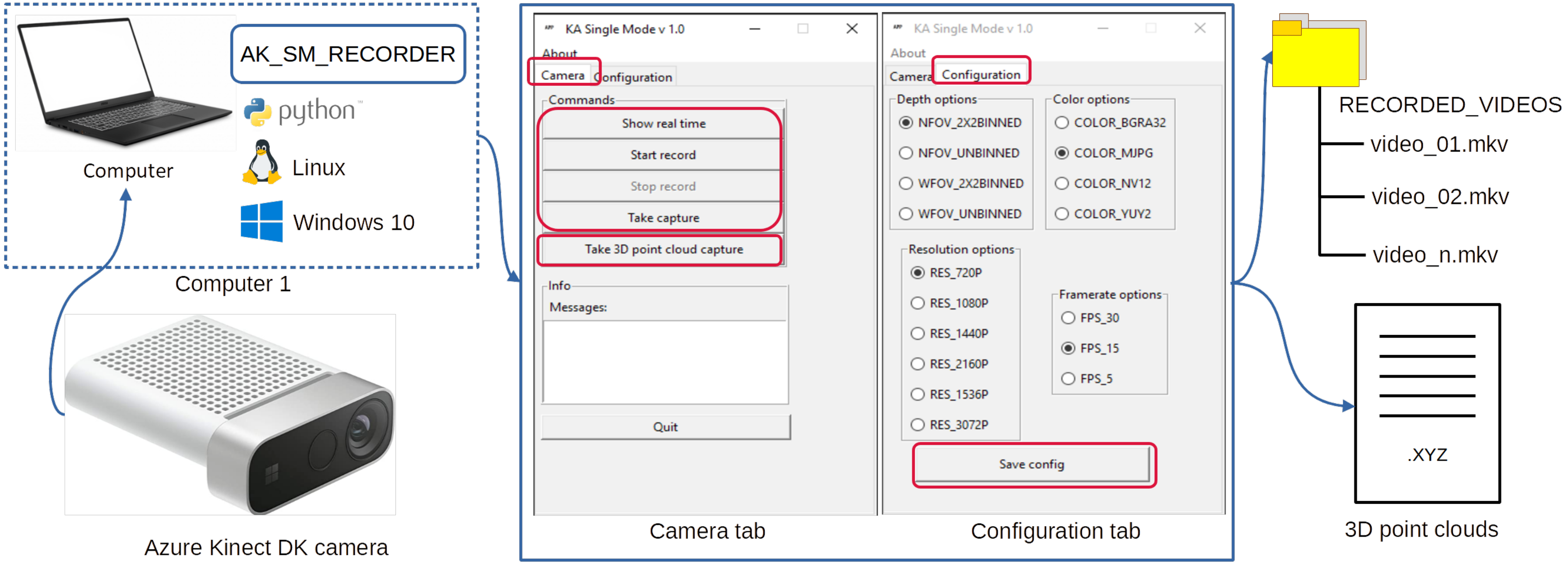
AK_ACQS – data acquisition with multiple sensors



```
Enabling remote clients
Enable recording
Stop recording
Taking capture
A capture was launch!!
```

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

AK_SM_RECORDER – data acquisition with Azure Kinect sensor



KA Single Mode v 1.0

About

Camera | Configuration

Commands
- Show real time
- Start record
- Stop record
- Take capture
- Take 3D point cloud capture

Info

Messages:

Quit

Camera tab

KA Single Mode v 1.0

About

Camera | Configuration

Depth options: [✓ NFOV_2X2BINNED] NFOV_UNBINNED WFOV_2X2BINNED WFOV_UNBINNED

Color options: COLOR_BGRA32 [✓ COLOR_MJPG] COLOR_NV12 COLOR_YUY2

Resolution options: [✓ RES_720P] RES_1080P RES_1440P RES_2160P RES_1536P RES_3072P

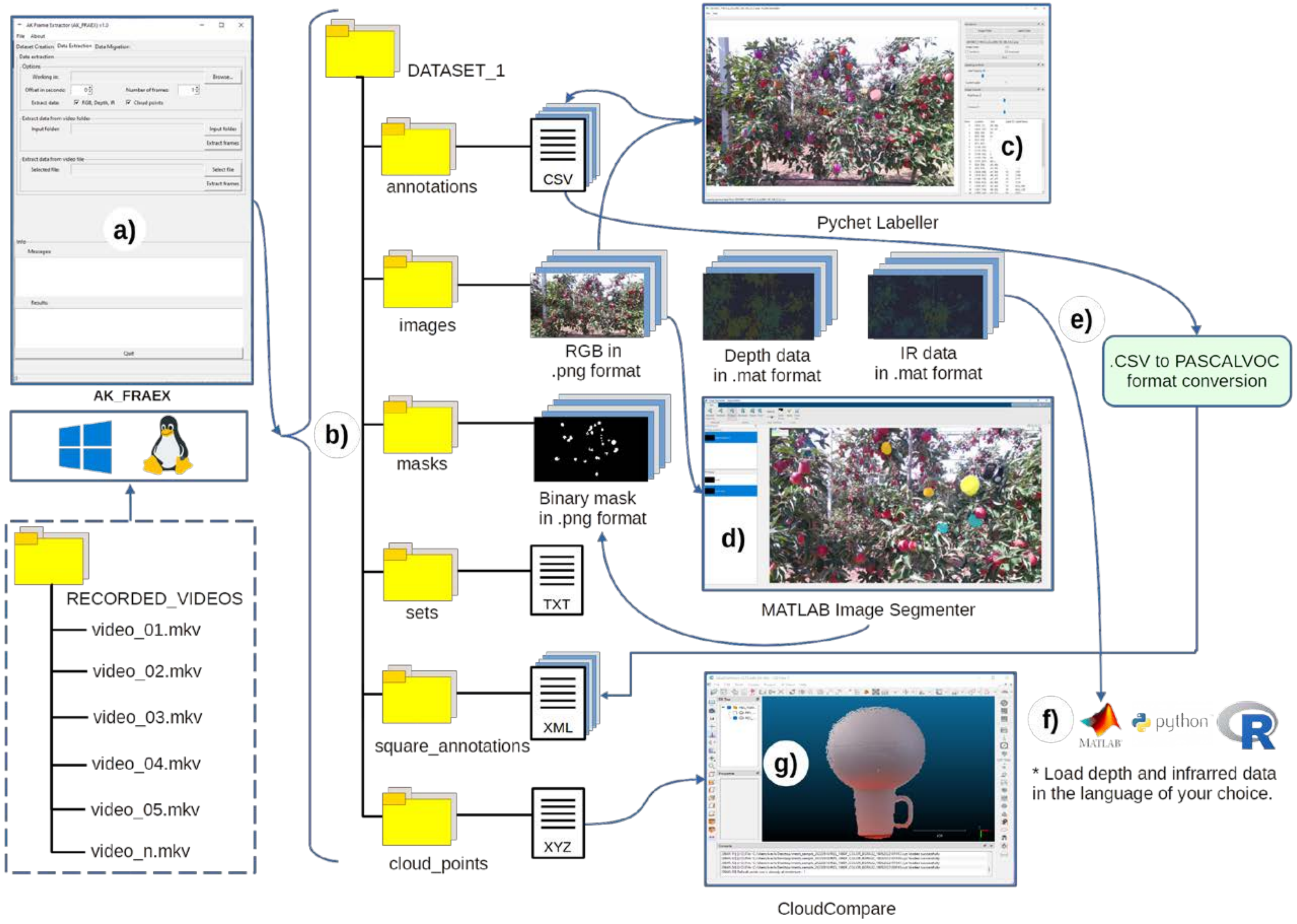
Framerate options: FPS_30 [✓ FPS_15] FPS_5

Save config

Configuration tab

```
https://pypi.org/project/ak-sm-recorder/
pip install ak-sm-recorder
```

AK_FRAEX - Extraction of data recorded with the Azure Kinect sensor

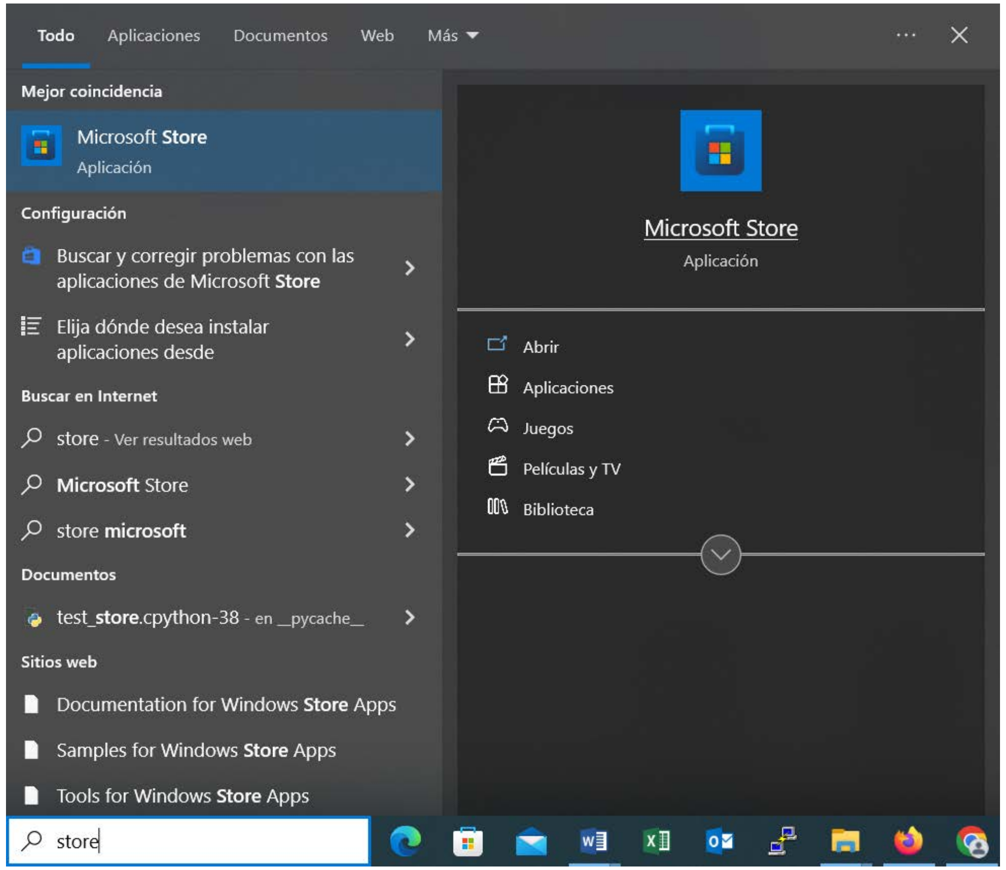


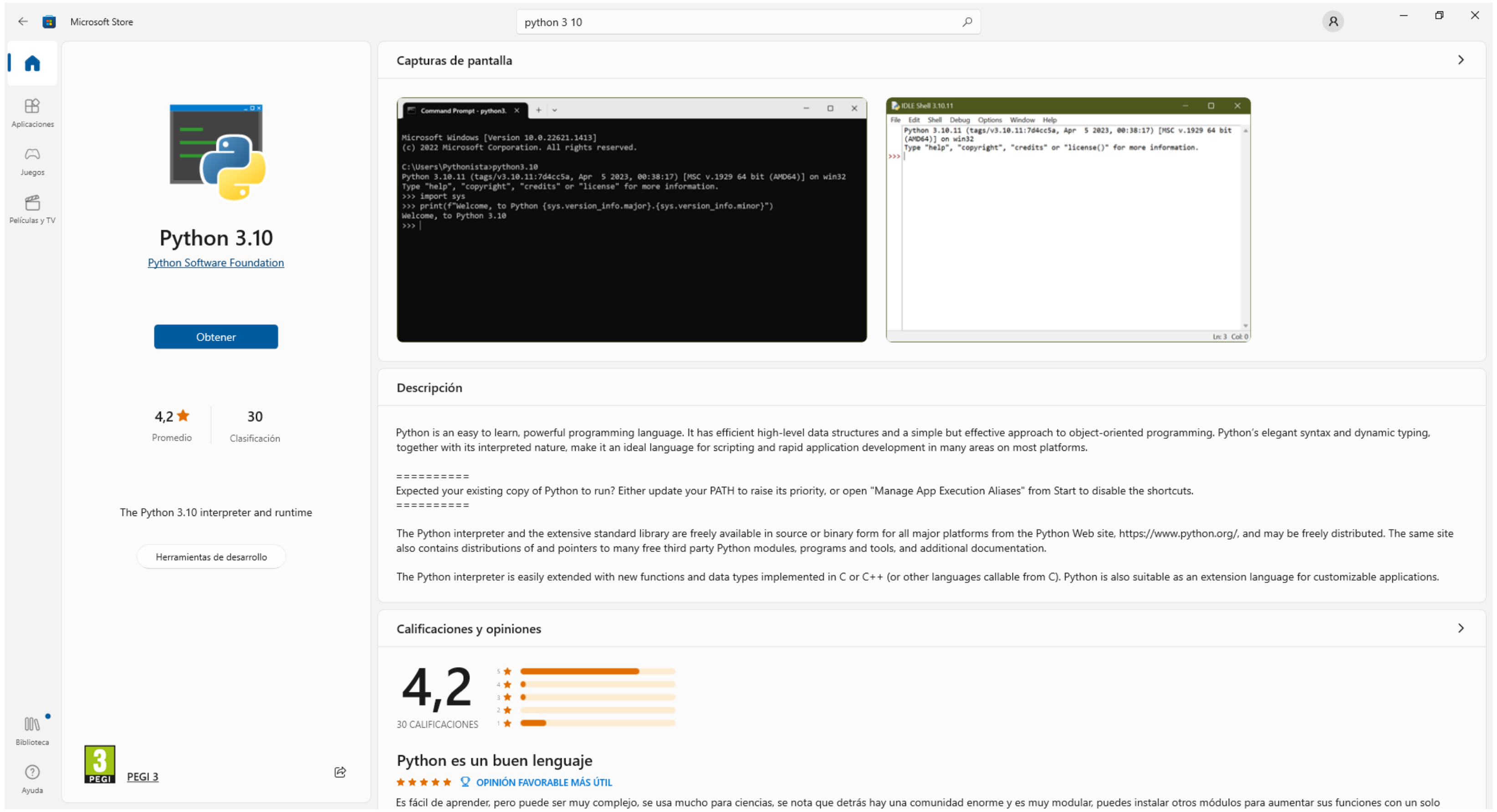
```
https://pypi.org/project/ak-frame-extractor/

pip install ak-frame-extractor
```

Lesson 2 – First steps with Python

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.


Todo
Aplicaciones
Documentos
Web
Más
Mejor coincidencia
Microsoft Store
Aplicación
Configuración
Buscar y corregir problemas con las aplicaciones de Microsoft Store
Elija dónde desea instalar aplicaciones desde
Buscar en Internet
store - Ver resultados web
Microsoft Store
store microsoft
Documentos
test_store.cpython-38 - en __pycache__
Sitios web
Documentation for Windows Store Apps
Samples for Windows Store Apps
Tools for Windows Store Apps
store
Microsoft Store
Aplicación
Abrir
Aplicaciones
Juegos
Películas y TV
Biblioteca



The Python 3.10 interpreter and runtime

```
Microsoft Windows [Version 10.0.22621.1413]
(c) 2022 Microsoft Corporation. All rights reserved.

C:\Users\Pythonista>python3.10
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> print(f"Welcome, to Python {sys.version_info.major}.{sys.version_info.minor}")
Welcome, to Python 3.10
>>> 
```

```
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
```

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

==========
Expected your existing copy of Python to run? Either update your PATH to raise its priority, or open "Manage App Execution Aliases" from Start to disable the shortcuts.
==========

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

Python es un buen lenguaje

Es fácil de aprender, pero puede ser muy complejo, se usa mucho para ciencias, se nota que detrás hay una comunidad enorme y es muy modular, puedes instalar otros módulos para aumentar sus funciones con un solo

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:

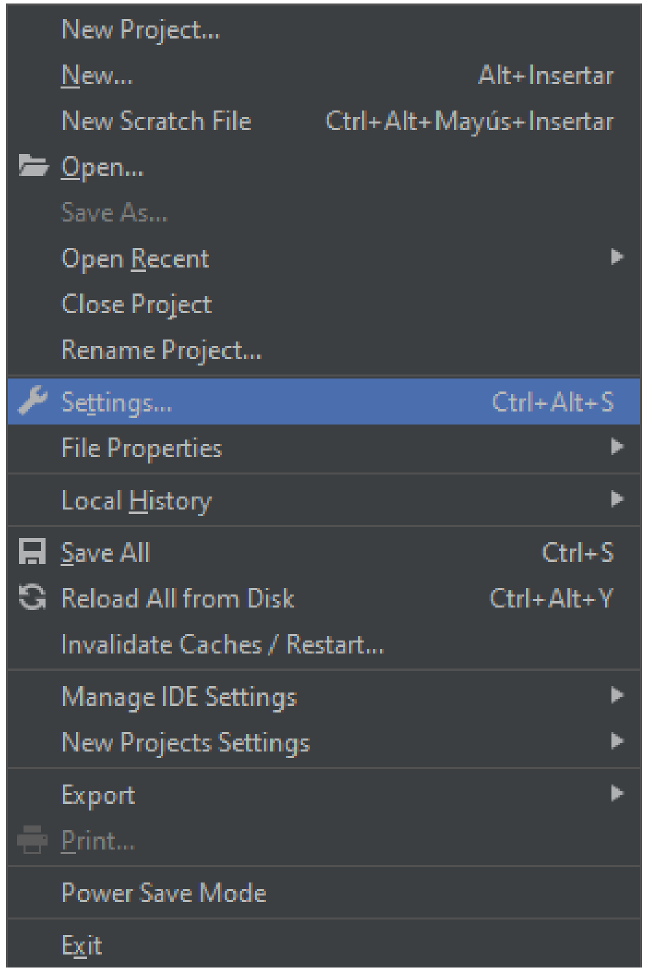
- Using the tools that Pycharm offers.
- Manual creation of the virtual environment from the command line.

The main components of the Pycharm tool are explained bellow.

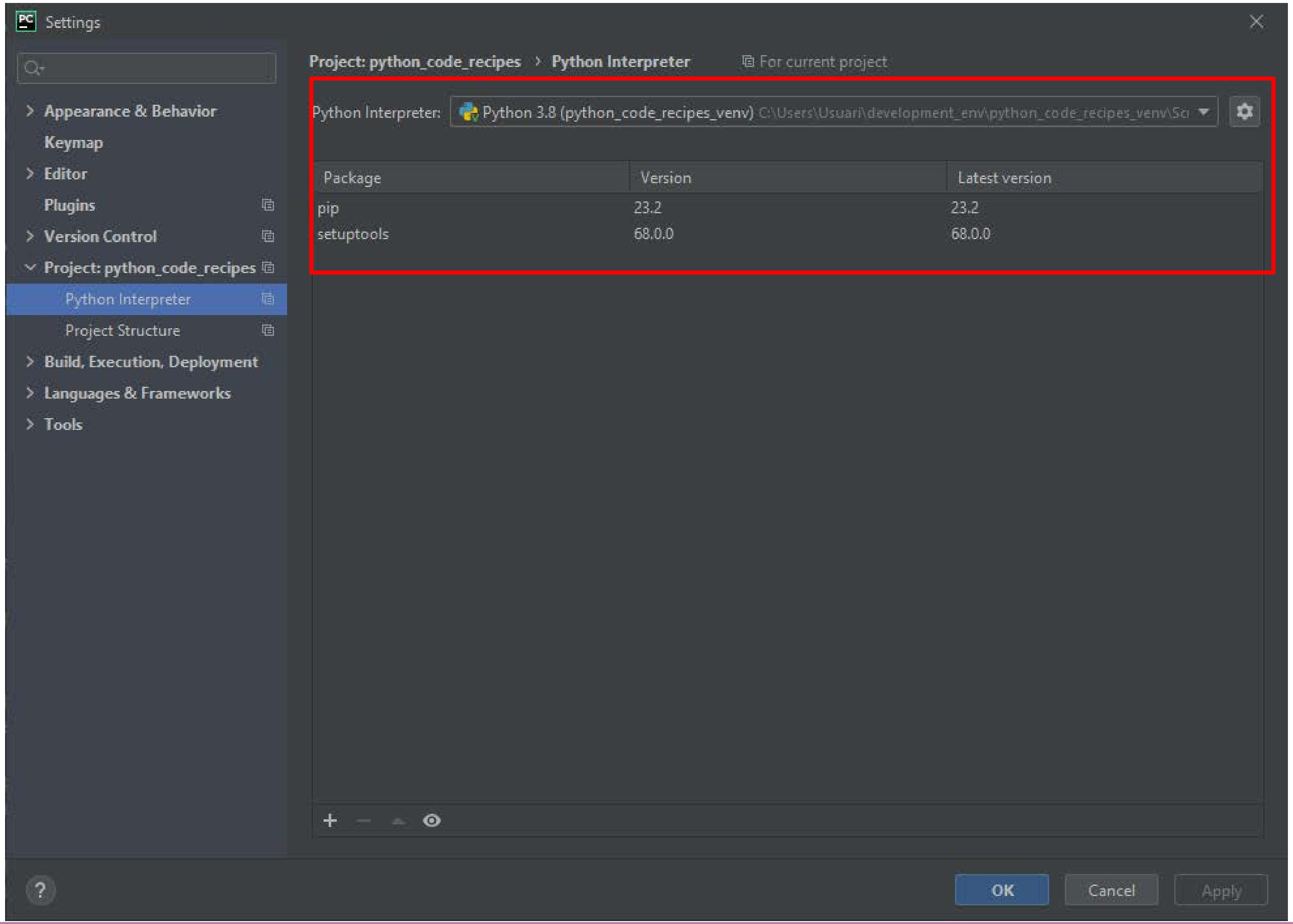
Lesson 2 – First steps with Python

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

Create a new project.



Configure the virtual environment

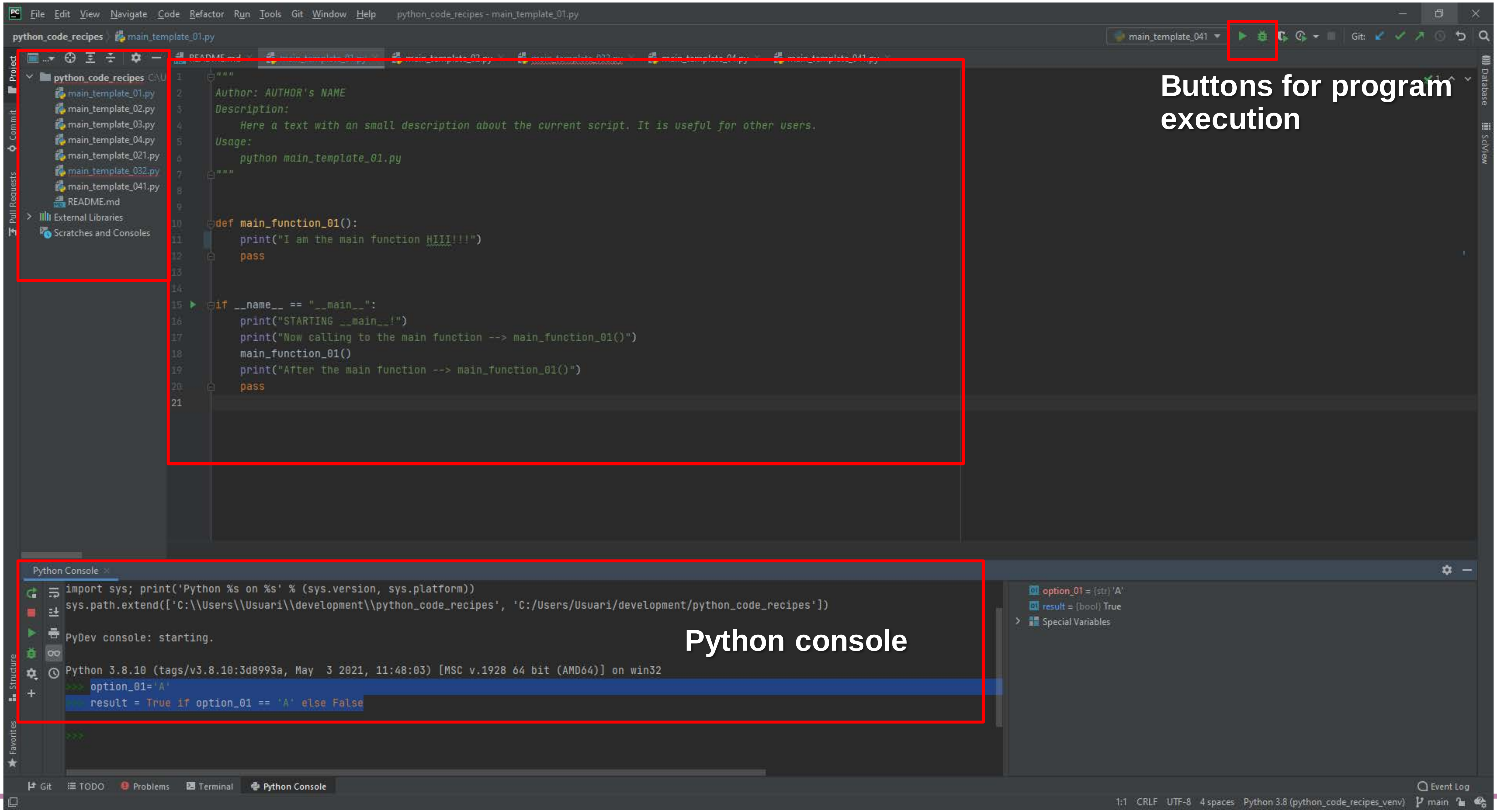


Project: python_code_recipes › Python Interpreter · For current project

Python Interpreter: Python 3.8 (python_code_recipes_venv) C:\Users\Usuari\development_env\python_code_recipes_venv\Sc

Package	Version	Latest version
pip	23.2	23.2
setuptools	68.0.0	68.0.0

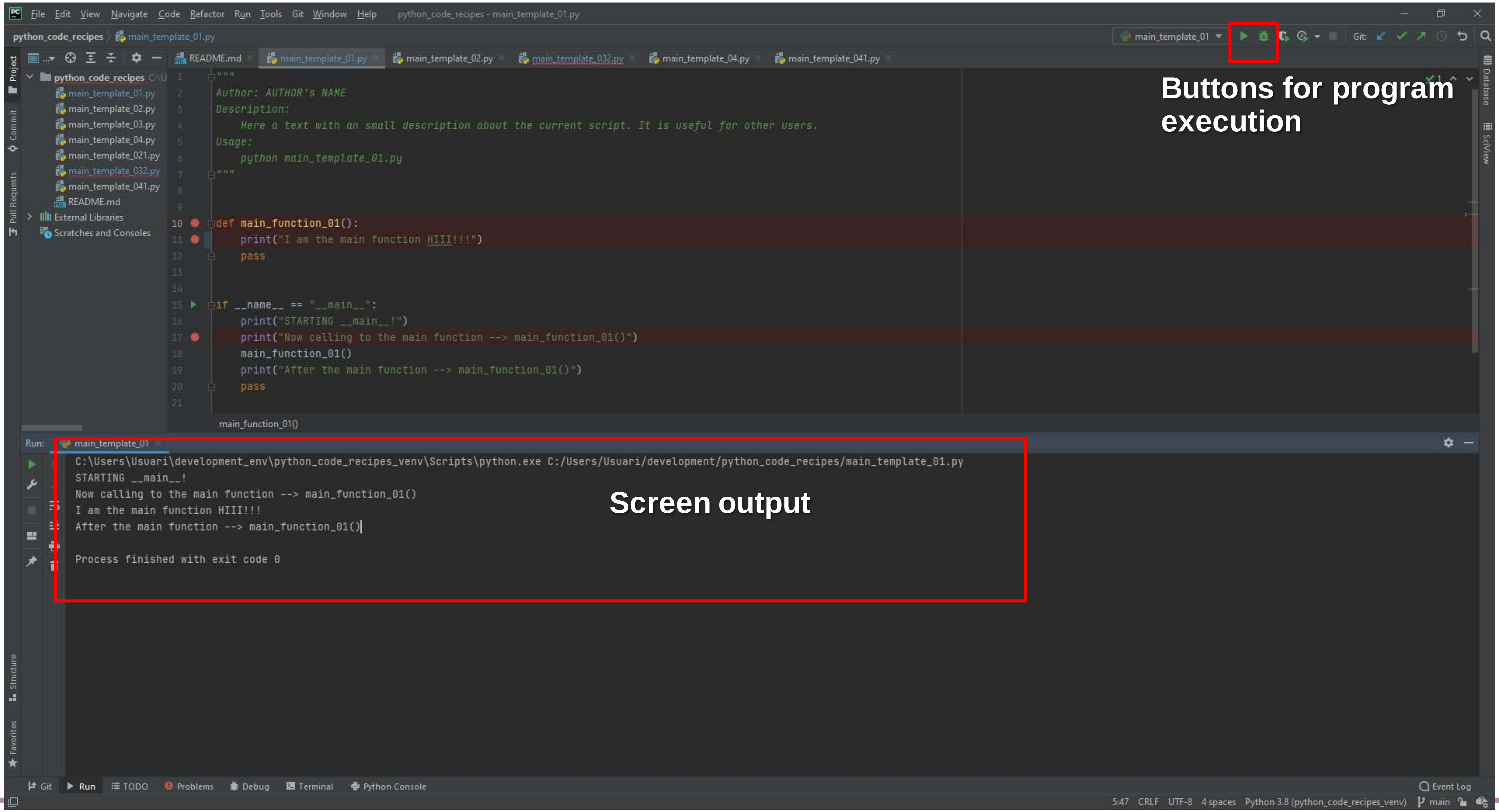
Lesson 2 – First steps with Python



```python
"""
Author: AUTHOR's NAME
Description:
    Here a text with an small description about the current script. It is useful for other users.
Usage:
    python main_template_01.py
"""


def main_function_01():
    print("I am the main function HIII!!!")
    pass


if __name__ == "__main__":
    print("STARTING __main__!")
    print("Now calling to the main function --> main_function_01()")
    main_function_01()
    print("After the main function --> main_function_01()")
    pass

```

```
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['C:\\Users\\Usuari\\development\\python_code_recipes', 'C:/Users/Usuari/development/python_code_recipes'])

PyDev console: starting.

Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
>>> option_01='A'
>>> result = True if option_01 == 'A' else False

>>>
```

Lesson 2 – First steps with Python



```python
"""
Author: AUTHOR's NAME
Description:
    Here a text with an small description about the current script. It is useful for other users.
Usage:
    python main_template_01.py
"""


def main_function_01():
    print("I am the main function HIII!!!")
    pass


if __name__ == "__main__":
    print("STARTING __main__!")
    print("Now calling to the main function --> main_function_01()")
    main_function_01()
    print("After the main function --> main_function_01()")
    pass
```

```
C:\Users\Usuari\development_env\python_code_recipes_venv\Scripts\python.exe C:/Users/Usuari/development/python_code_recipes/main_template_01.py
STARTING __main__!
Now calling to the main function --> main_function_01()
I am the main function HIII!!!
After the main function --> main_function_01()

Process finished with exit code 0
```

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.

Lesson 2 – First steps with Python

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**".

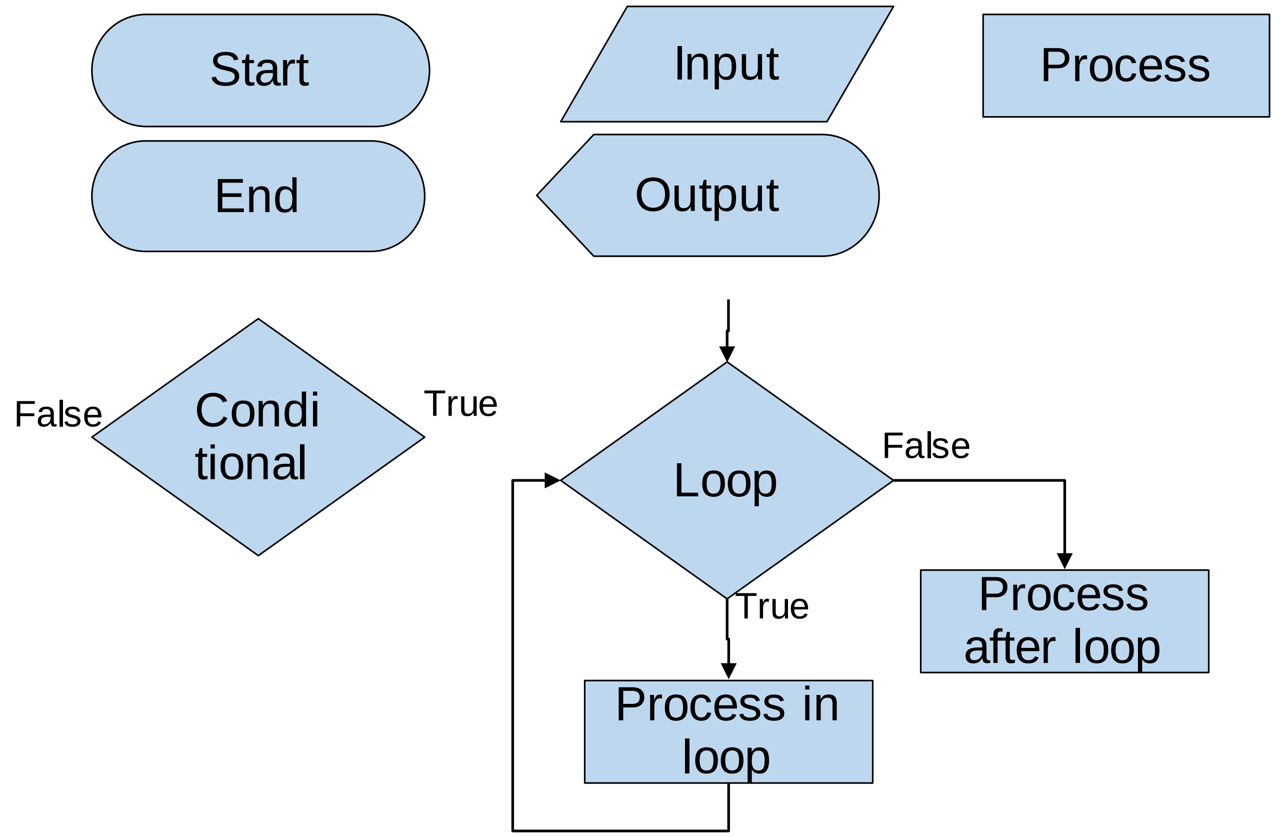
Lesson 3 – Fundamentals of Python

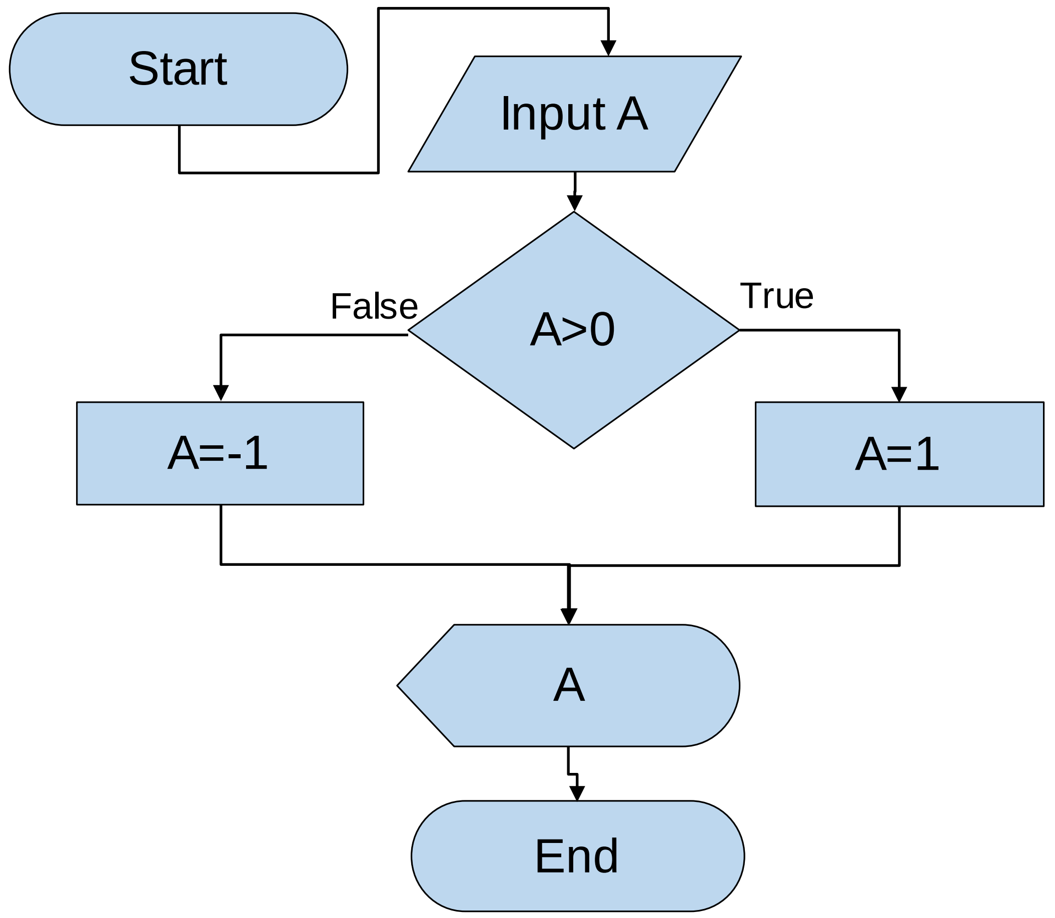
- Introduction to algorithmic thinking: flow diagrams, flow control, conditionals, loops, inputs and outputs.
- Variables: assignment, names, scope. Types: numbers, strings, lists, dictionaries.
- Control flow sentences, conditionals and loops, functions.
- Outputs: text files, CSV files.

Lesson 3 – Fundamentals of Python

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**

Lesson 3 – Fundamentals of Python

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.

Lesson 3 – Fundamentals of Python

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 3 – Fundamentals of Python

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

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