GHENT
UNIVERSITY

DEPARTMENT OF GEOLOGY
PYTHON DIVISION

A GENTLE INTRODUCTION TO PYTHON WITH PYCHARM

Robin Thibaut 30/03/2022

GHENT
UNIVERSITY

OVERVIEW

- This presentation will give you a basic introduction to Python and PyCharm.
- We will go over the absolute basics of Python syntax, data types, and file input and output.
- An introduction to PyCharm will follow.
- Finally, you'll be invited to try out some of the things you've learned in hands-on exercises.

It's smooth and sleek, and always ready to strike

With powerful code that's easy to read and write

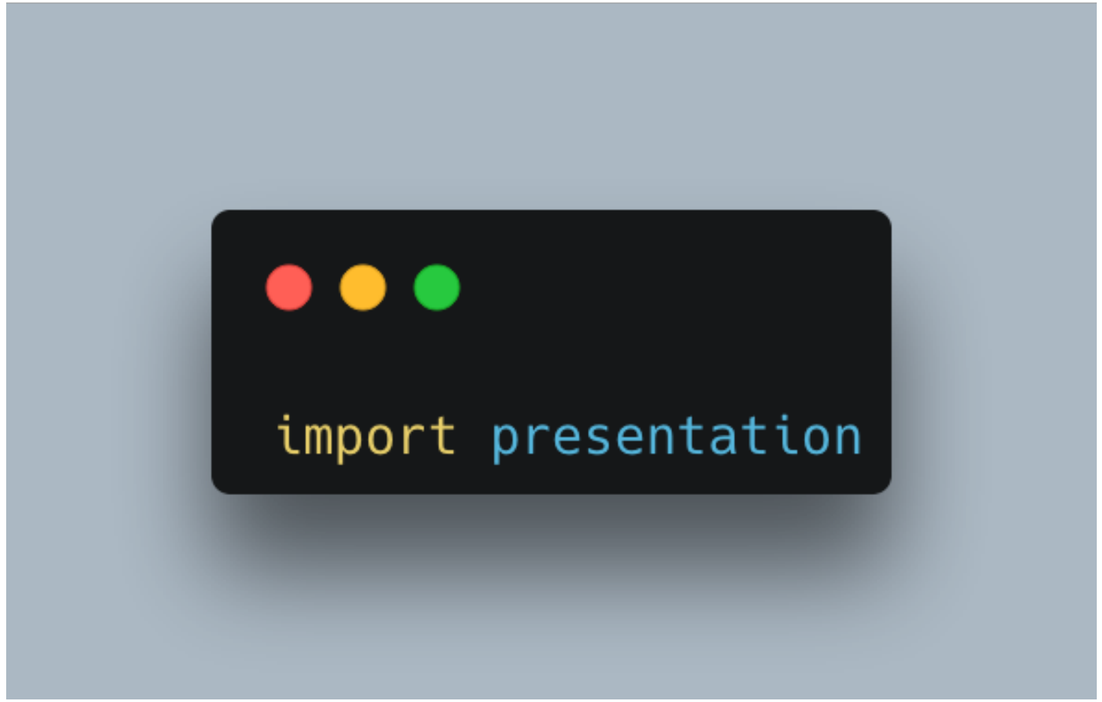
Python is the perfect tool for any task

Big or small, it can handle anything you ask

GHENT UNIVERSITY

INTRODUCTION TO PYTHON – WHAT IS PYTHON AND WHY IS IT A GOOD LANGUAGE TO LEARN?

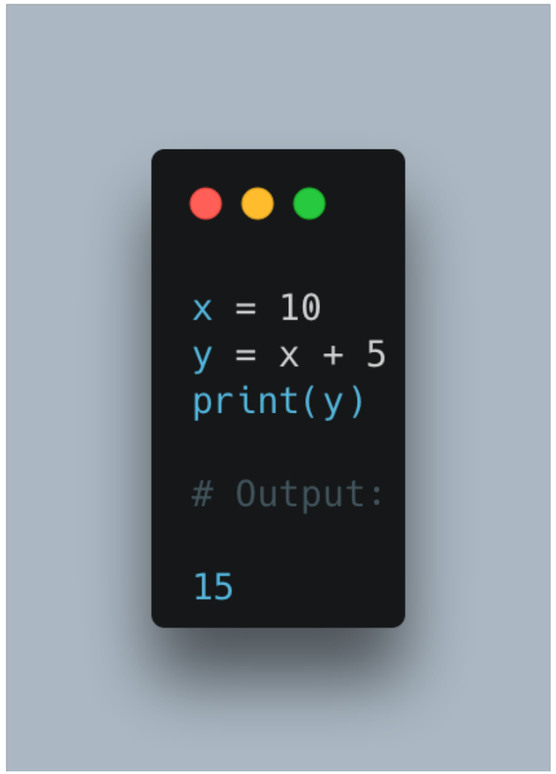
- Python is a programming language with a simple syntax and powerful data structures.
- It is easy to learn for beginners and has many modules and libraries that allow for robust programming.
- Python is a popular language for web development, scientific computing, artificial intelligence, and more.



GHENT UNIVERSITY

PYTHON SYNTAX – AN INTRODUCTION TO THE BASIC SYNTAX OF THE PYTHON LANGUAGE

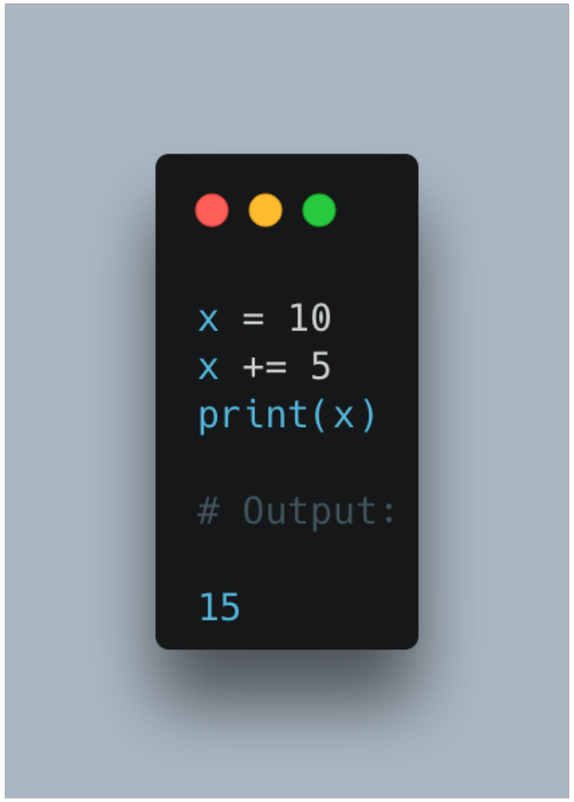
- Python syntax is very simple and easy to learn.
- The basic syntax consists of statements and expressions.
- A statement is a single line of code that performs an action.
- An expression is a combination of values, variables, and operators that evaluates to a single value.



GHENT UNIVERSITY

PYTHON VARIABLES – HOW TO DECLARE AND USE VARIABLES IN PYTHON

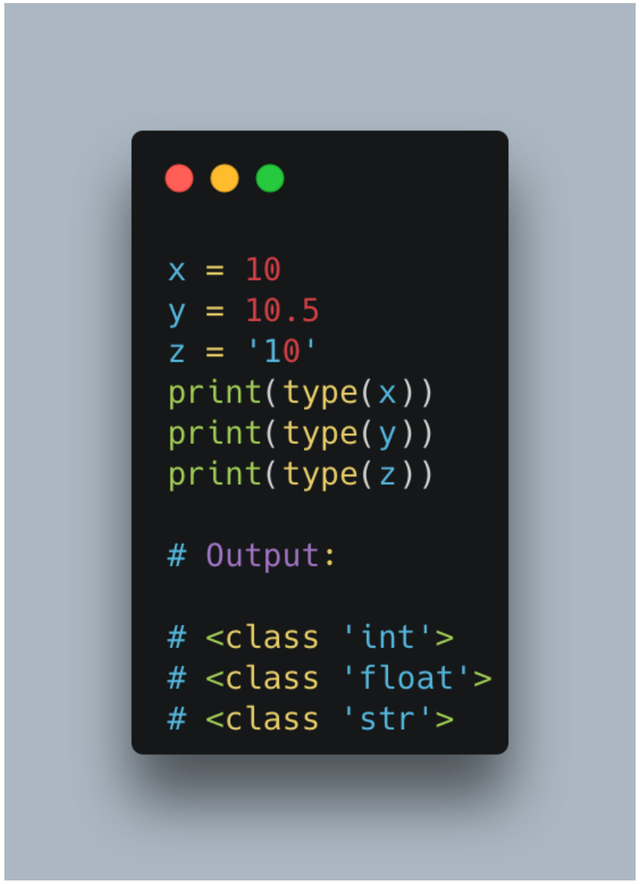
- A variable is a name that refers to a value.
- In Python, you can declare variables by assigning a value to a name.
- For example, to declare a variable named 'x' that refers to the value '10', you would write 'x = 10'. You can then use the variable 'x' in your code.



GHENT UNIVERSITY

PYTHON DATA TYPES – AN INTRODUCTION TO THE VARIOUS DATA TYPES AVAILABLE IN PYTHON

- Python has many built-in data types, such as integers, floats, strings, and lists.
- These data types are used to represent different kinds of information.
- Python also allows you to create your own data types, called user-defined types.



```
x = 10
y = 10.5
z = '10'
print(type(x))
print(type(y))
print(type(z))

# Output:

# <class 'int'>
# <class 'float'>
# <class 'str'>
```

GHENT UNIVERSITY

PYTHON LISTS – HOW TO CREATE AND USE LISTS IN PYTHON

- A list is a collection of values.
- In Python, you can create lists by enclosing values in square brackets.
- For example, to create a list of the numbers from 1 to 10, you would write '[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]'.
- You can access individual values in a list by using their index.

GHENT UNIVERSITY

PYTHON TUPLES – HOW TO CREATE AND USE TUPLES IN PYTHON

- A tuple is a immutable sequence of values.
- In Python, you can create tuples by enclosing values in parentheses.
- For example, to create a tuple of the numbers from 1 to 10, you would write '(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)'.
- You can access individual values in a tuple by using their index.

```
my_tuple = ("mouse", "python", "rabbit")
print(my_tuple[1])

# Output:

"python"
```

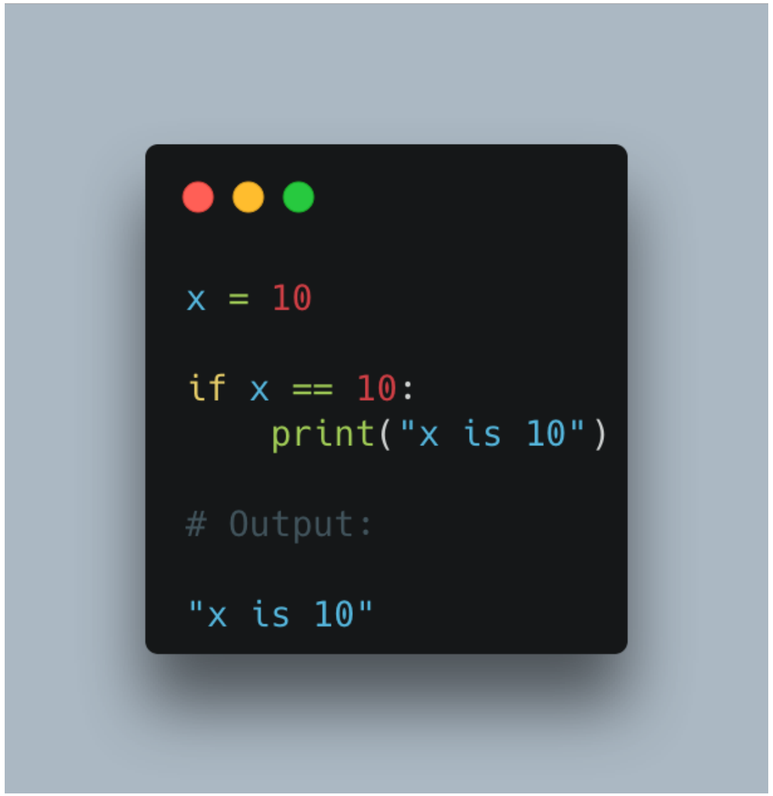
GHENT UNIVERSITY

PYTHON DICTIONARIES – HOW TO CREATE AND USE DICTIONARIES IN PYTHON

- A dictionary is a mutable collection of key-value pairs.
- In Python, you can create dictionaries by using the 'dict' keyword.
- For example, to create a dictionary of the numbers from 1 to 10, you would write 'dict(1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five', 6: 'six', 7: 'seven', 8: 'eight', 9: 'nine', 10: 'ten')'.
- You can access individual values in a dictionary by using their key.

PYTHON CONDITIONALS – HOW TO USE CONDITIONALS IN PYTHON

- Conditionals allow you to execute code only if a certain condition is met.
- In Python, you can use conditionals by using the 'if' keyword.
- For example, to print 'Hello, world!' if the variable 'x' is equal to '10', you would write 'if x == 10: print('Hello, world!')'.



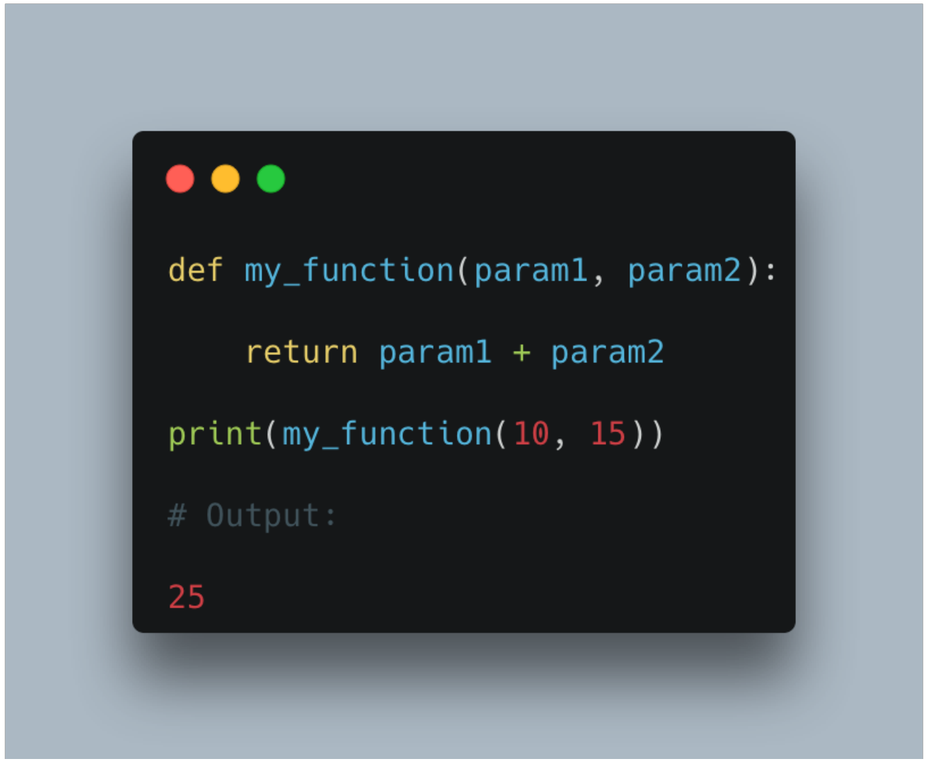
LOOPS - FOR AND WHILE LOOPS

- Loops allow you to execute a block of code multiple times.
- In Python, you can use loops by using the 'for' and 'while' keywords.
- For example, to print the numbers from 1 to 10, you would write 'for i in range(1, 11): print(i)'.

GHENT UNIVERSITY

PYTHON FUNCTIONS – HOW TO CREATE AND USE FUNCTIONS IN PYTHON

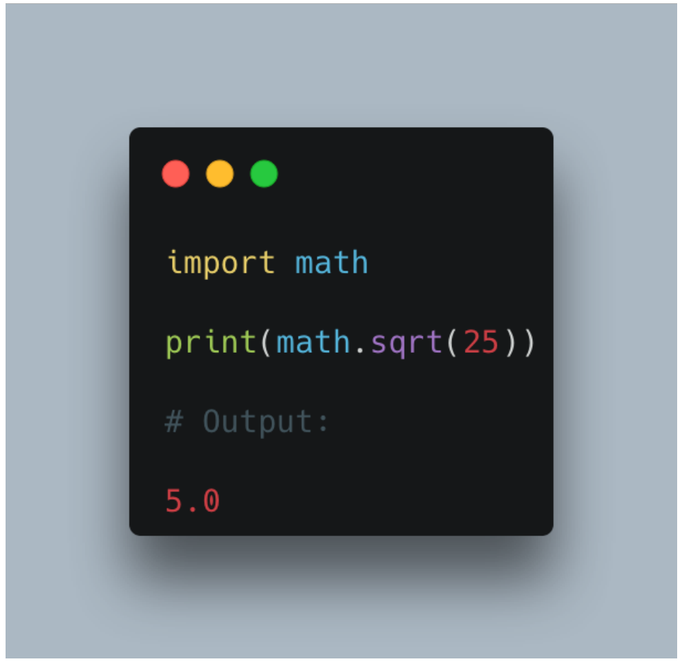
- A function is a block of code that performs a specific task.
- In Python, you can create functions by using the 'def' keyword.
- For example, to create a function that prints 'Hello, world!', you would write 'def print_hello(): print('Hello, world!')'.
- You can call a function by using its name.



GHENT UNIVERSITY

PYTHON MODULES – HOW TO IMPORT AND USE MODULES IN PYTHON

- A module is a Python file that contains code.
- You can import a module into your Python code by using the 'import' keyword.
- For example, to import the 'math' module, you would write 'import math'. You can then access the code in the module by using the module's name.



GHENT UNIVERSITY

PYTHON CLASSES AND OBJECTS – AN INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING IN PYTHON

- Object-oriented programming is a programming paradigm that uses objects and their interactions to design programs.

- In Python, you can create classes to define new types of objects. For example, you could create a 'Person' class to represent people.

- Objects can have attributes and methods. Attributes are values associated with an object, and methods are functions that can be called on an object.

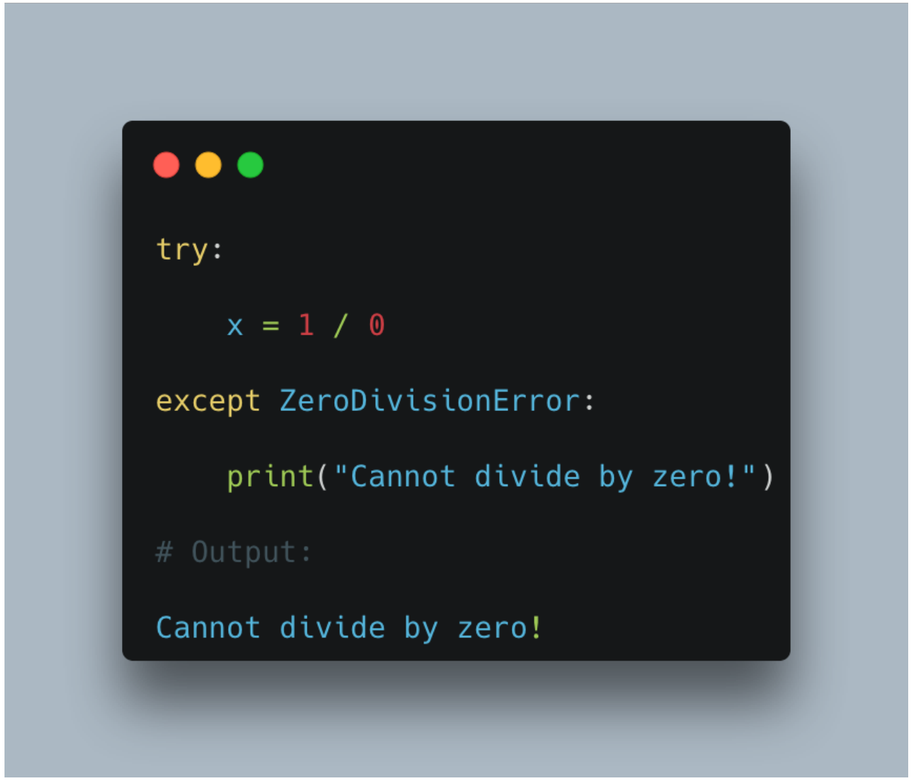
```python
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def print_info(self):
        print(f"Name: {self.name}")
        print(f"Age: {self.age}")

person1 = Person("Bob", 30)
person1.print_info()

# Output:
Name: Bob
Age: 30
```

PYTHON EXCEPTION HANDLING – HOW TO HANDLE ERRORS AND EXCEPTIONS IN PYTHON

- When an error occurs in Python, an exception is raised.
- You can handle exceptions by using the 'try' and 'except' keywords.
- For example, to handle an exception that is raised when you try to open a file that does not exist, you would write 'try: open('filename') except FileNotFoundError: print('File not found')'.



GHENT UNIVERSITY

PYTHON FILE INPUT AND OUTPUT – HOW TO READ AND WRITE FILES IN PYTHON

- File input and output allows you to read and write data from and to files.
- In Python, you can open files by using the 'open' function.
- For example, to open a file for reading, you would write 'f = open('filename', 'r')'. You can then read the file by using the 'read' method. To write to a file, you would use the 'write' method.

```
f = open('filename', 'w')
f.write('Hello, world!')
f.close()
f = open('filename', 'r')
data = f.read()
print(data)
f.close()
# Output:
Hello, world!
```

GHENT UNIVERSITY

SETTING UP PYCHARM – HOW TO DOWNLOAD AND INSTALL PYCHARM

- Pycharm is an Integrated Development Environment (IDE) for Python.
- It is a cross-platform IDE that is available for Windows, macOS, and Linux.
- Pycharm has many features for Python development, such as code completion, code formatting, and debugging.

CREATING YOUR FIRST PYTHON PROJECT – HOW TO CREATE A NEW PROJECT IN PYCHARM AND WRITE YOUR FIRST PYTHON PROGRAM

- To create a new project in Pycharm, go to File > New Project.
- Enter a project name and location, and select a Python interpreter.
- Pycharm will create a new project with a default Python file.
- To write your first Python program, open the Python file and enter some code.
- To run the program, go to Run > Run '<file name>'.

- A project is a directory that contains all of the files related to a single Python application.

GHENT UNIVERSITY

VIRTUAL ENVIRONMENTS

- A virtual environment is a tool to keep the dependencies required by different projects in separate places, by creating virtual Python environments for them.
- It solves the “Project X depends on version 1.x but, Project Y needs 4.x” dilemma, and keeps your global site-packages directory clean and manageable.

Virtual environments are like forests where trees are the **packages** and each forest is a **project**.

Python is the sun that shines down on the forest and gives the trees life.

GHENT UNIVERSITY

HOW TO CREATE A VIRTUAL ENVIRONMENT?

- To create a virtual environment, go to your project's directory and run venv.

python3 -m venv my_env

- This will create a my_env directory in your project directory.
- To activate a virtual environment, go to your project's directory and run source.

source my_env/bin/activate

- The virtual environment-specific python and pip executables will be added to your shell's PATH after you activate it.

- **This step is not necessary if you are using the IDE PyCharm, it will automatically create the virtualenv and activate it for you when the project is created.**
- **Alternatively, use Anaconda.**

HOW TO INSTALL PACKAGES IN A VIRTUAL ENVIRONMENT?

– To install packages in a virtual environment, run pip install in the terminal.

pip install <name of the package you need>

- Neat little trick:

pip install -r requirements.txt

- This command installs all of the packages listed in requirements.txt.

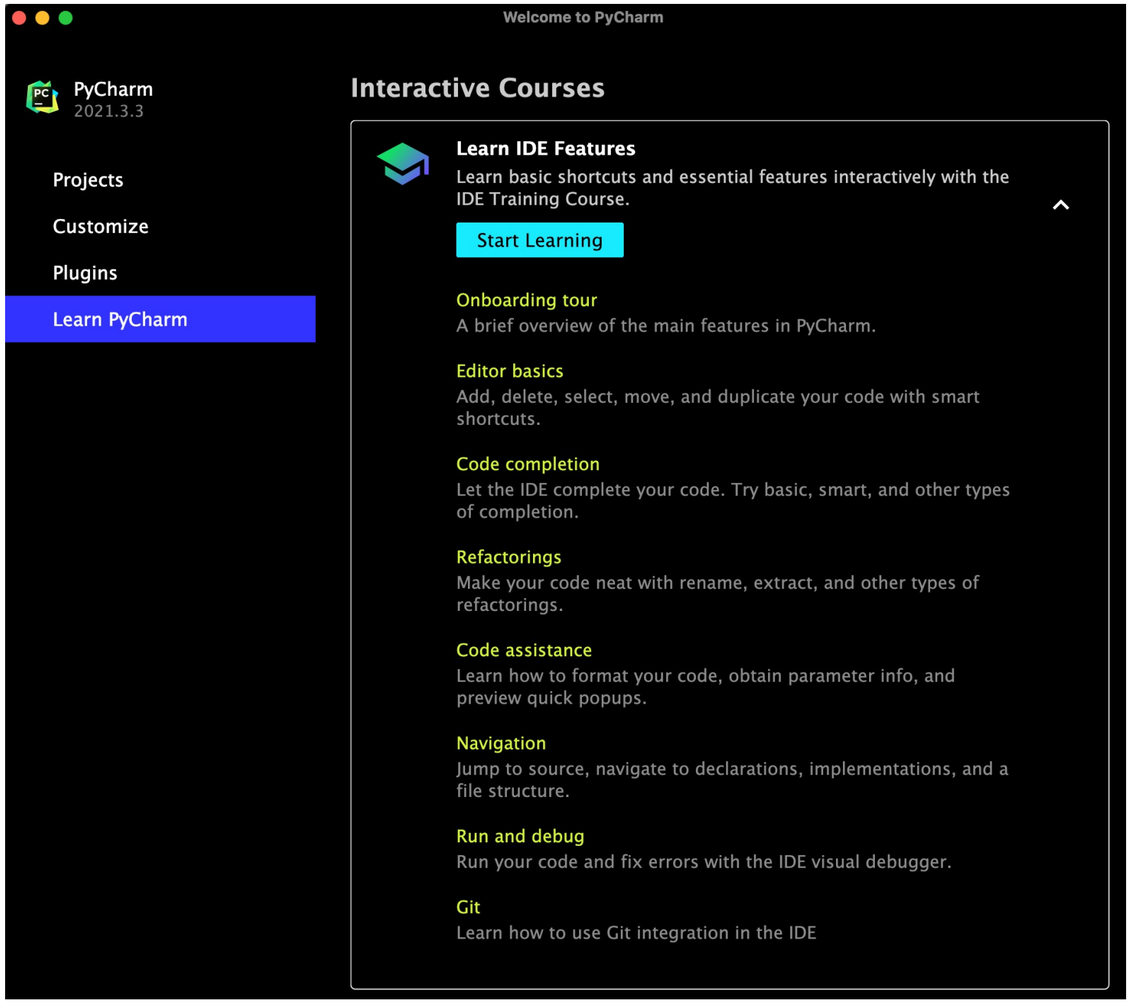
- You can create the file manually or by running pip freeze > requirements.txt in the terminal, which outputs all currently installed packages in requirements.txt

PIP is the lumberjack who cuts down trees and brings them to the forest

GHENT UNIVERSITY

EXERCISES

- Complete the PyCharm interactive course until “Git” (not included – afternoon session)



GHENT UNIVERSITY

EXERCISES

- Solve the basic exercises in a PyCharm project
- Make sure you understand the procedure of creating the virtual environment
- Install the packages you need with PIP through the terminal

- Feel free to focus on the parts that matter the most to you
- Work alone or in group

CONCLUSION – A SUMMARY OF WHAT YOU LEARNED IN THE WORKSHOP

- In this workshop, you learned about Python and how to use Pycharm for Python development.
- You learned about the basic syntax of the Python language and how to declare and use variables.
- You also learned about Python data types and how to create and use lists and tuples.
- Additionally, you learned about dictionaries, functions, and modules.
- Finally, you learned about object-oriented programming and exception handling.

Robin Thibaut
PhD Student

DEPARTMENT OF GEOLOGY

E robin.thibaut@ugent.be

www.ugent.be

Universiteit Gent
@ugent
@ugent
Ghent University

GHENT
UNIVERSITY