Programming techniques for NLP

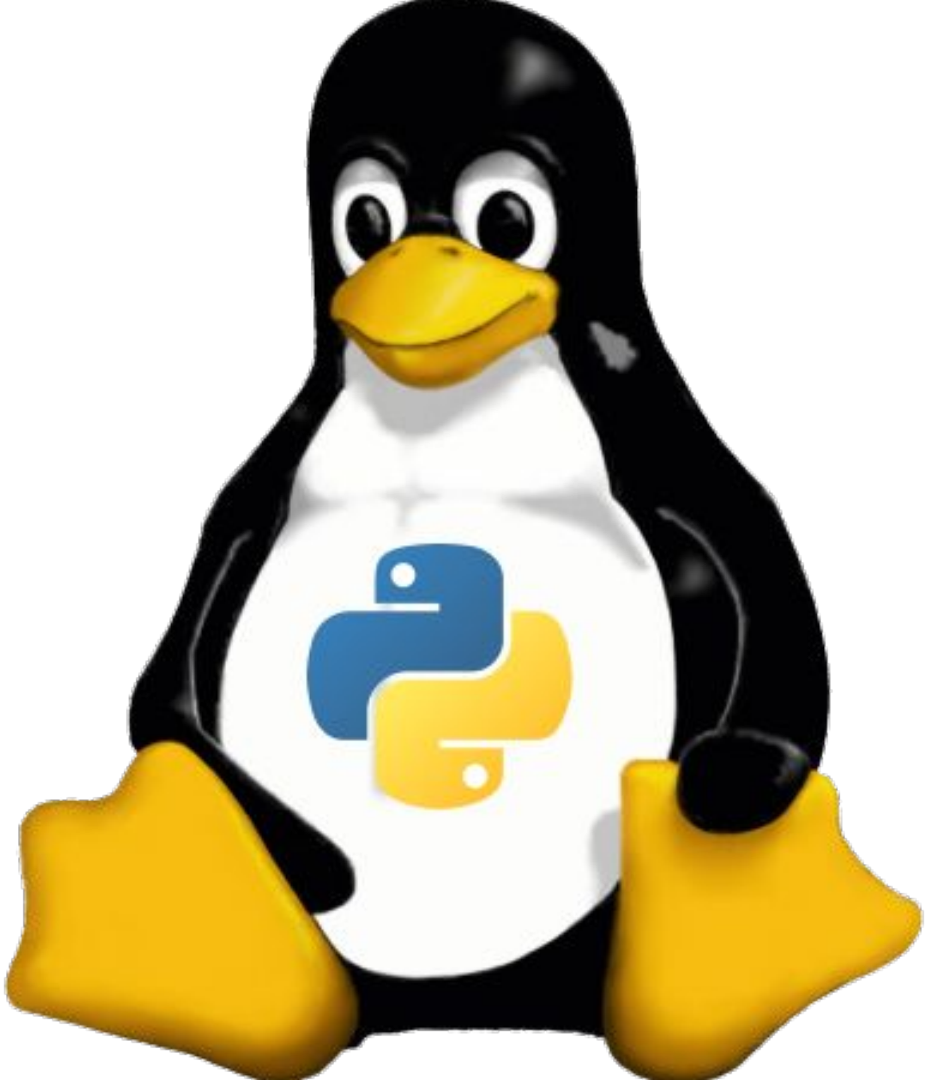
Olatz Perez de Viñaspre Garralda
Ander Soraluze Irureta

Evaluation

- Part I exercises: 40 %
- Part II assessments: 60 %

Python programming language

Open source



–

Language **simple**, **understandable** and **powerful**

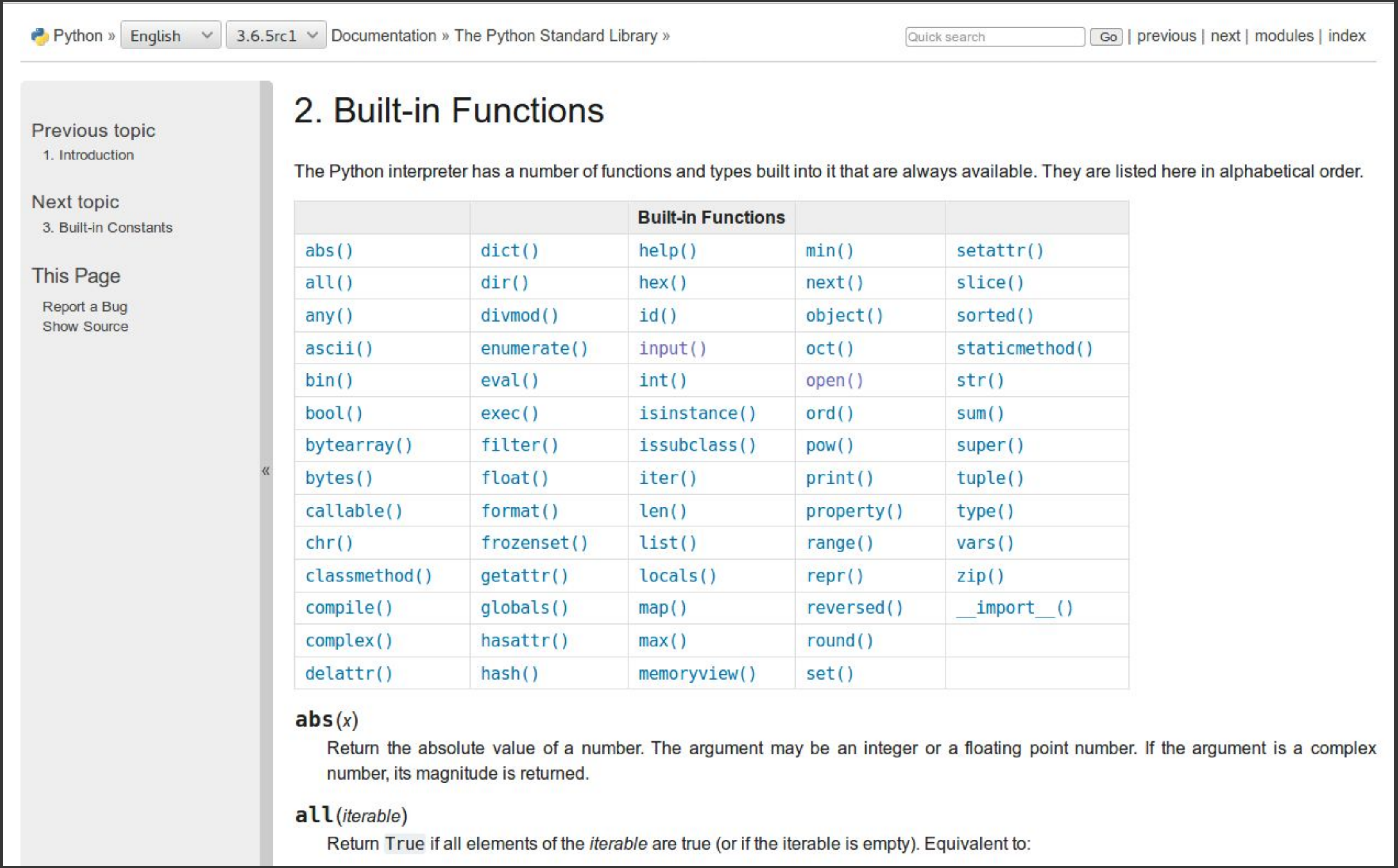
Beautiful is better than ugly. Explicit is better than implicit. Simple is better than complex. Complex is better than complicated. Flat is better than nested. Sparse is better than dense. Readability counts. Special cases aren't special enough to break the rules. Although practicality beats purity. Errors should never pass silently. Unless explicitly silenced. In the face of ambiguity, refuse the temptation to guess. There should be one — and preferably only one — obvious way to do it. Although that way may not be obvious at first unless you're Dutch. Now is better than never. Although never is often better than right now. If the implementation is hard to explain, it's a bad idea. If the implementation is easy to explain, it may be a good idea. Namespaces are one honking great idea — let's do more of those!

There should be one-- and preferably only one --obvious way to do it.

python

TM

Wide and clear documentation



This Page

Report a Bug
Show Source

2. Built-in Functions

The Python interpreter has a number of functions and types built into it that are always available. They are listed here in alphabetical order.

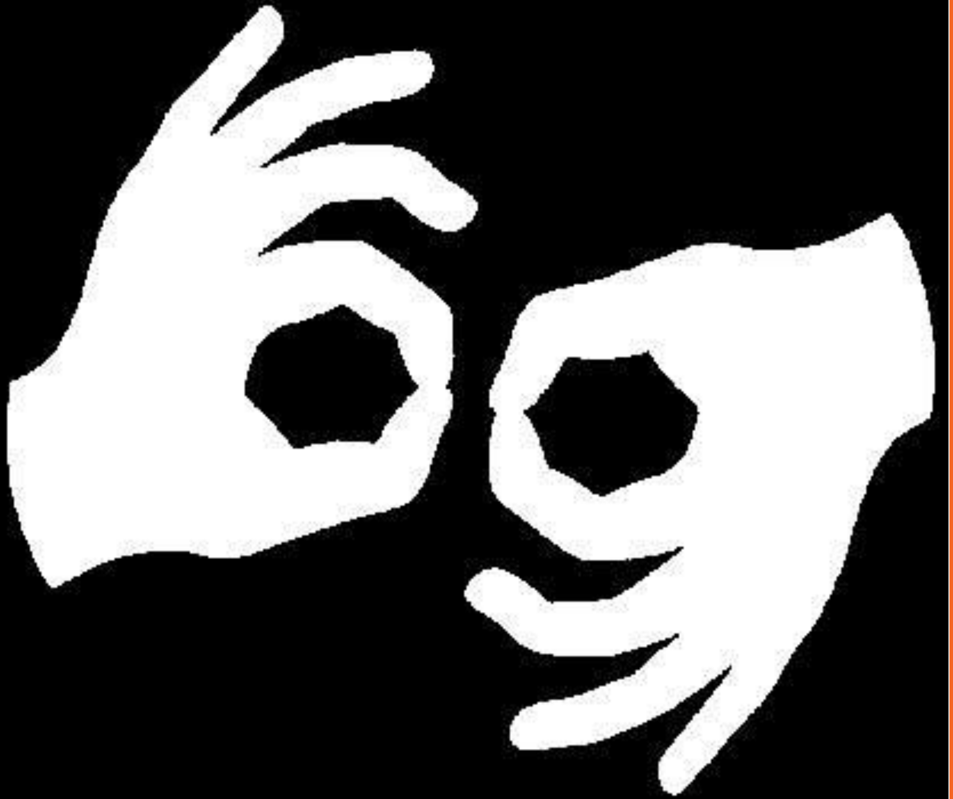
		Built-in Functions		
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

abs(*x*)

Return the absolute value of a number. The argument may be an integer or a floating point number. If the argument is a complex number, its magnitude is returned.

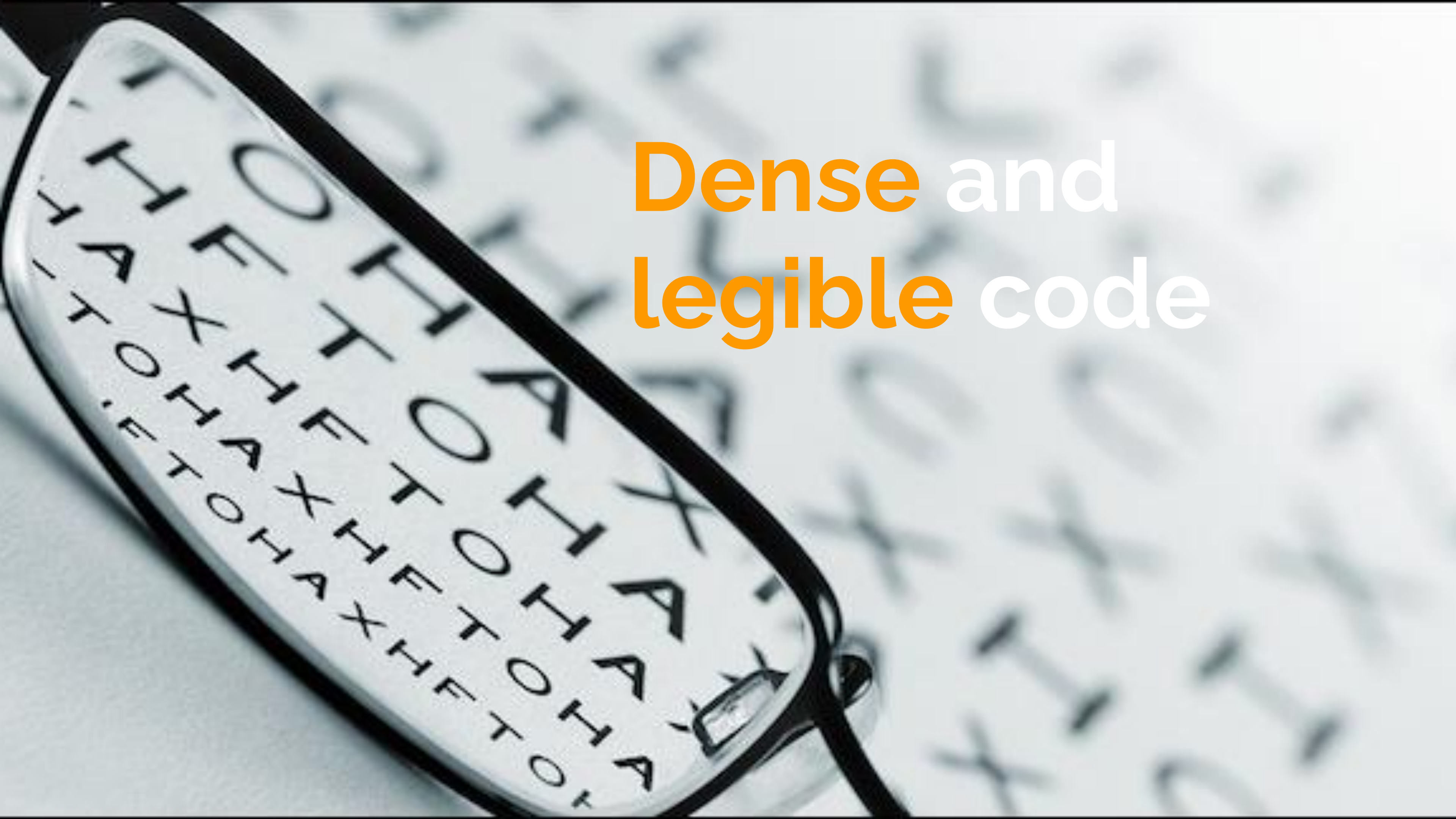
all(*iterable*)

Return `True` if all elements of the *iterable* are true (or if the iterable is empty). Equivalent to:



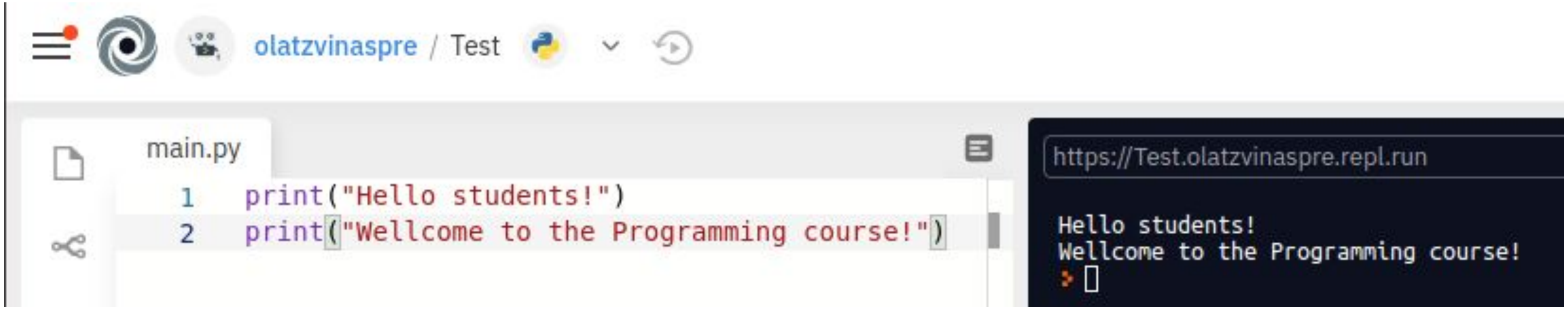
Scripting language (Interpreter)

Dense and legible code



Syntax like any languages

Grammar & Composition III
THEMES in Literature
Vocabulary, Spelling, Poetry III
SCIENCE Matter and Energy Laboratory Manual
SCIENCE Matter and Ener
WORLD HISTORY AND CULTURE



```python
print("Hello students!")
print("Wellcome to the Programming course!")
```

```
Hello students!
Wellcome to the Programming course!
>
```

Differences 2 vs 3

Main **differences** between Python 2 and Python 3

- ➔ **Encoding**
  Much simpler (and less problems) in Python 3
- ➔ **Display**
  print function with parenthesis print()
- ➔ **Internal optimizations**
  Memory, efficiency,...

We will work with...

- ➔ Basic data and operations
- ➔ Variables
- ➔ Conditionals
- ➔ Iterations
- ➔ Strings
- ➔ Lists and dictionaries
- ➔ Files
- ➔ Functions
- ➔ Libraries

Some basic definitions

- **Algorithm:** A finite sequence of well-defined and ordered operations to solve a problem (or perform a computation)
  It is the definition of a process
  - When you define **WHAT** to do
  - It defines **HOW** to do it

- **Program/app(lication):**
  The sequence of operations the computer is going to execute

Some basic definitions

- ➔ **Programming language:** Artificial language that can be used to control the computer behaviour.
  A grammar rule set is needed, as well as some symbols and reserved words.
- ➔ We will use Python.

- ➔ **Elements of a program:**
  - ◆ Data: to represent information and results
  - ◆ Operations: to change data and create procedures

LINUX
PCLinuxOS

Operating System

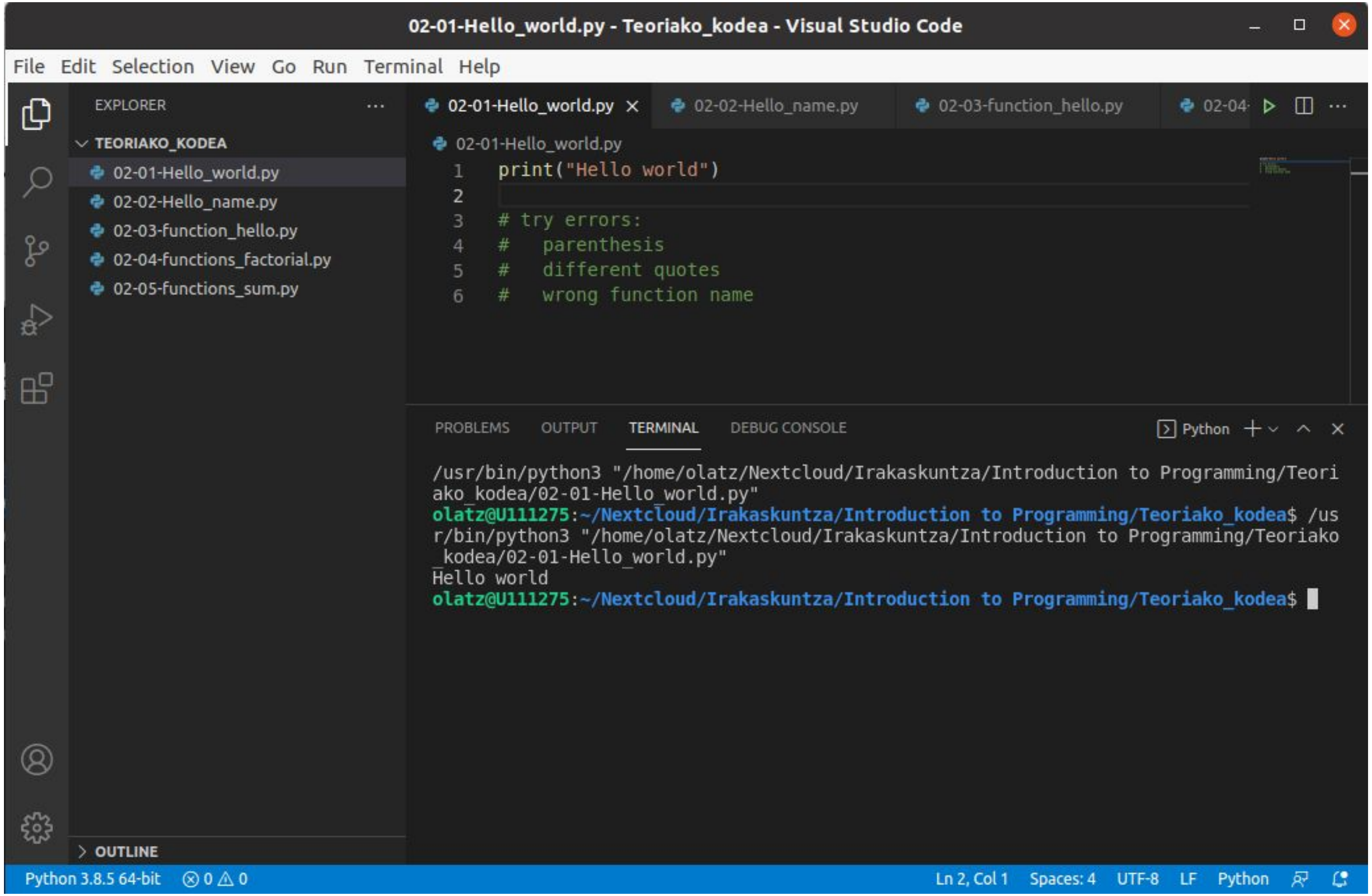
We recommend you to install Ubuntu (or other Linux distribution) in your computer:

- In a new partition
- In a virtual machine (VMware, VirtualBox,...)

Plenty tutorials on the Internet

Environment (IDE)

You can use the IDE of your choice. The lecturers will be using **Visual Studio Code** (installed in the classroom’s machines)



02-01-Hello_world.py - Teoriako_kodea - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TEORIAKO_KODEA
02-01-Hello_world.py
02-02-Hello_name.py
02-03-function_hello.py
02-04-functions_factorial.py
02-05-functions_sum.py
OUTLINE
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Python
Python 3.8.5 64-bit ⊗ 0 ⚠ 0
Ln 2, Col 1 Spaces: 4 UTF-8 LF Python

```python
print("Hello world")

# try errors:
#   parenthesis
#   different quotes
#   wrong function name
```

```
/usr/bin/python3 "/home/olatz/Nextcloud/Irakaskuntza/Introduction to Programming/Teori
ako_kodea/02-01-Hello_world.py"
olatz@U111275:~/Nextcloud/Irakaskuntza/Introduction to Programming/Teoriako_kodea$ /us
r/bin/python3 "/home/olatz/Nextcloud/Irakaskuntza/Introduction to Programming/Teoriako
_kodea/02-01-Hello_world.py"
Hello world
olatz@U111275:~/Nextcloud/Irakaskuntza/Introduction to Programming/Teoriako_kodea$
```

Ready to start working!

http://www.parentesis.com