Advanced School in Artificial Intelligence

Introduction to Python

Ing. Zese Riccardo
riccardo.zese@unife.it

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Regione Emilia-Romagna

Università degli Studi di Ferrara

Outline

- Introduction to Python
- Introduction to Neural Networks
- Convolutional NN
- Recurrent NN
- Autoencoders and self supervised learning

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Outline

- Introduction to Python
- Introduction to Neural Networks
- Convolutional NN
- Recurrent NN
- Autoencoders and self supervised learning

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

The history of the language

- The first version of Python was called ABC, defined in the first 80s at the National Research Institute for Mathematics and Computer Science of Amsterdam.
- A couple of years later (end 80s) Guido van Rossum, one of the developers of ABC, improved it by defining a new language: **Python**.
  - The name derives by Monty Python's Flying Circus, Van Rossum was a big fan of it.

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

The history of the language

- In 2000 the version 2.0 was developed.
- In 2008 the version 3.0, also called Python 3000 or Python 3k, introduced many changes and improvements, the use of UNICODE, and many functions were completely redefined.
  - Python 3 is not compatible with Python 2!
  - In this course we **consider Python 3**.

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Install Python

- Home page: https://www.python.org/
- You can download the installer for Windows and Mac and the tarball for Linux.
- As regards Linux systems, usually it is already installed or can be easily installed using the package manager of the distro:
  - `sudo apt install python3`
  - `sudo yum install python3`
  - `sudo pacman -S python`
  - …

IDE

- There are lots of IDEs that you can use:
  - Pydev, a plug-in for Eclipse IDE
  - PyCharm (https://www.jetbrains.com/pycharm/), in two versions one free and one commercial. One of the most used and complete.
  - Python plug-in for Visual Studio Code
  - …
- Online Jupiter notebooks:
  - https://colab.research.google.com/notebooks/welcome.ipynb#recent=true, connected with your google or UniFE account

Tutorial and references

- Tutorials:
  - https://docs.python.org/3/tutorial/index.html (English)
  - https://www.html.it/guide/guida-python/ (Italian)
  - …

- Books:
  - Think Python by Allen B. Downey
    - https://github.com/AllenDowney/ThinkPython (source, English)
    - https://github.com/AllenDowney/ThinkPythonItalian (source and PDF, Italian)
    - Many of these slides are taken from this book and from the Python tutorial.

Run Python

- Two ways of running the **interpreter**:
  - **Interactive mode**: executable by running the **`python`** command without arguments in a console. A new prompt will be shown: **>>>**

```
Python 3.7.3 (default, Mar 26 2019, 21:43:19)
[GCC 8.2.1 20181127] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

  - After the prompt you can execute your code

```
Python 3.7.3 (default, Mar 26 2019, 21:43:19)
[GCC 8.2.1 20181127] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 1 + 1
2
>>>
```

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Run Python

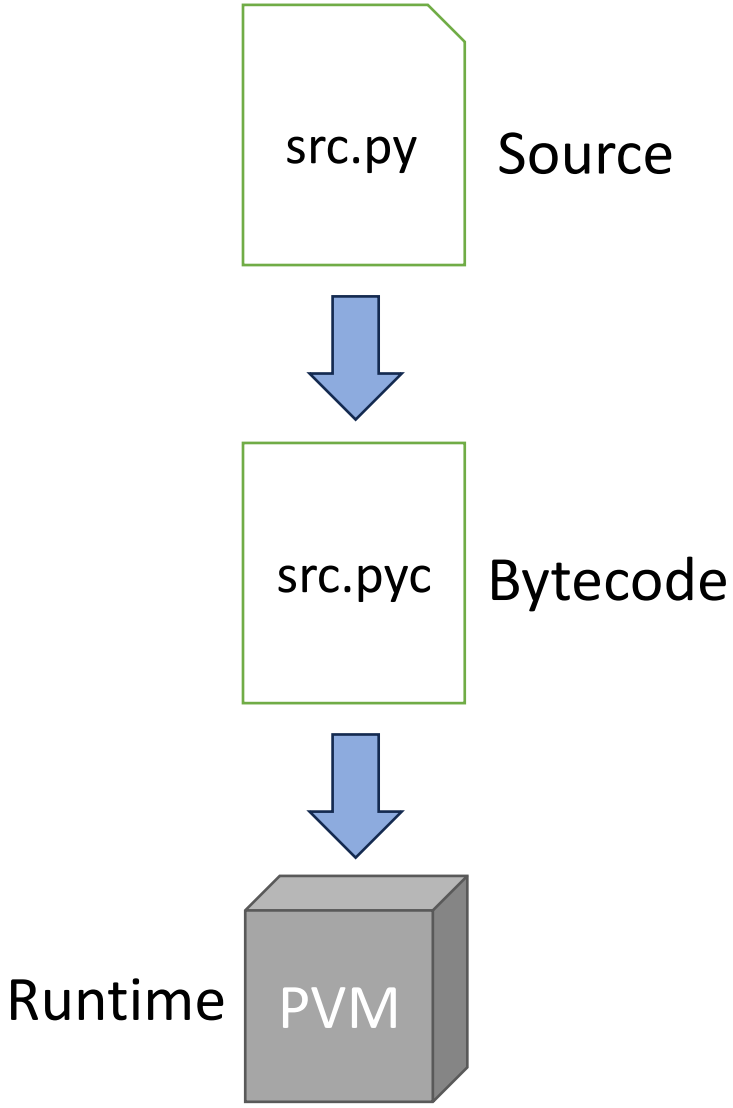
- Two ways of running the **interpreter**:
  - **Script mode**: executable by running the **`python`** command with the name of the file as argument.
  - Python files uses **`.py`** extension.

```
# python myscript.py
Hello world!
#
```

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Executing a Python script

- Each time the **`python`** command is invoked, the written code is scanned for tokens, each of which is parsed into a logical tree structure that represents the program.
- This structure is then transformed into a bytecode (file with the extension **`.pyc`** or **`.pyo`**).
- To be able to execute these bytecodes, a special interpreter known as the Python Virtual Machine (PVM) is used.



Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Hello World!

- Using interactive interpreter

```
>>> print('Hello World!')
Hello World!
```

- Using a script file

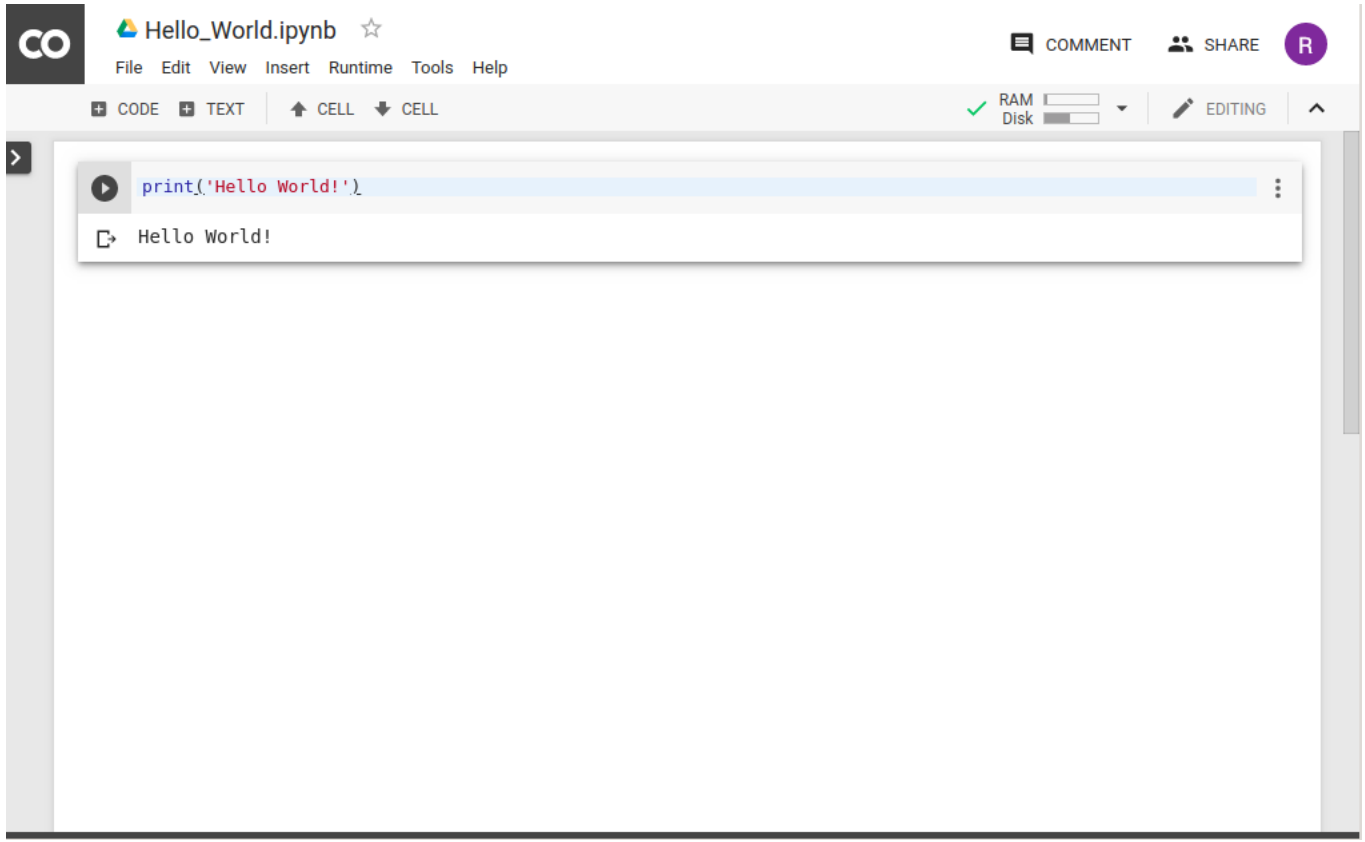
script.py

```
print('Hello World!')
```

```
# python script.py
Hello World!
```

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Hello World!



Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Variables

- Variable names can contain both letters and numbers.
  - They have to begin with a letter.
- There is not a limit in the number of characters and more words can be used
  - To divide words in the name one can use the underscore → using uppercase letters is not an error but goes against the naming conventions and should be avoided.

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Variables

- There are 31 keywords that cannot be used as name of variables:

```
and      del      from     not    while
as       elif     global   or     with
assert   else     if       pass   yield
break    except   import   print
class    in       raise    continue
finally  is       return   nonlocal
def      for      lambda   try
```

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Basic Input/Output

- We have already seen the instruction to write on the standard output → function `print`

```
>>> print(1)
1
>>> print('hello')
hello
>>> s = 'hi'
>>> a = 3
>>> b = 5
>>> print(a,b,a+b,s)
3 5 8 hi
```

- If we work in the interactive interpreter we can write the name of a variable to print it

```
>>> a = 3
>>> a
3
```

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Basic Input/Output

- To allow users to write data using the standard input we can use the function `input`

```
>>> in = input()
This is my first input
>>> in
'This is my first input'
```

- We can also pass an argument to the function to prompt a message before waiting for the user's input

```
>>> mess = input('Write your message\n')
Write your message
Hi!
>>> mess
'Hi!'
```

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Python types

- In Python everything is an object.
- In languages like C, variables refer to specific memory locations that have a fixed size that depends on their type.
  - You need to specify the type when declaring a variable.
- In Python objects have a specific type (number, string, list, etc.), while variables are just labels, references that refer to a given object.
  - Variables do not have types.

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Python types

- Let's consider the following code where we assign different values to variable **x**

```
x = 10
x = 7
x = "Python"
x = [1,2,3]
```

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Python types

- Let's consider the following code where we assign different values to variable **x**

```
x = 10
x = 7
x = "Python"
x = [1,2,3]
```

The object 10 is created and assigned to the variable (reference) called **x**

Python types

- Let's consider the following code where we assign different values to variable **x**

```
x = 10
x = 7
x = "Python"
x = [1,2,3]
```

Then, the object 7 is created and assigned to the variable (reference) **x**. Now, the object 10 is no more referenced as **x**, the unique reference to 10, now refers to 7. The Garbage Collector will remove this object from the memory.

Python types

- Let's consider the following code where we assign different values to variable **x**

```
x = 10
x = 7
x = "Python"
x = [1,2,3]
```

Since variables do not have types, they are just references to objects, they can be assigned to objects of different types.
The type of the variable will be inferred thanks to the referenced object, which specifies what one can do with the variable.

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Python types

- Python uses duck typing, which follows the rule «if it walks like a duck and quacks like a duck, then it must be a duck».
- The suitability is determined by checking the presence of methods and properties.

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Python types

Type	Mutable	Decription
`int`	Immutable	Integer number: `1`
`bool`	Immutable	Boolean: `True`, `False`
`float`	Immutable	Floating point number: `2.3`
`complex`	Immutable	Complex number with real and immaginary part: `1+2.3j`
`str`	Immutable	String: `'hi'`
`tuple`	Immutable	Can contains different types: `(1,'hi',False)`
`bytes, bytearray`	Immutable	Sequence of bytes
`list`	Mutable	List, can contain different types: `[1,'hi',False]`
`set, frozenset`	Mutable	Unordered set, can contain different types: `{1,'hi',False}`
`dict`	Mutable	Associative array, map: `{'key':1,3:'string'}`

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Python types

- One can identify the type of a value or variable by using the function **`type`**

```
>>> type('Hello')
<type 'str'>
>>> type(7)
<type 'int'>
>>> type('7')
<type 'str'>
>>> x = 10
>>> type(x)
<type 'int'>
```

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Python types

- One can check if a value or a variable is of a certain type by using the function **`isinstance`**

```
>>> isinstance('Hello',str)
True
>>> isinstance(7,str)
False
>>> isinstance(7,int)
True
>>> x = 10
>>> isinstance(x,int)
True
```

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021

Python types

- Python has also a special type called `None`.
- This is the type returned by void functions (functions that have no return statement) or a return statement without an argument.
- Similar to Null of other languages, can be used also to initialize empty variables.
- Its type is **NoneType**

Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021