



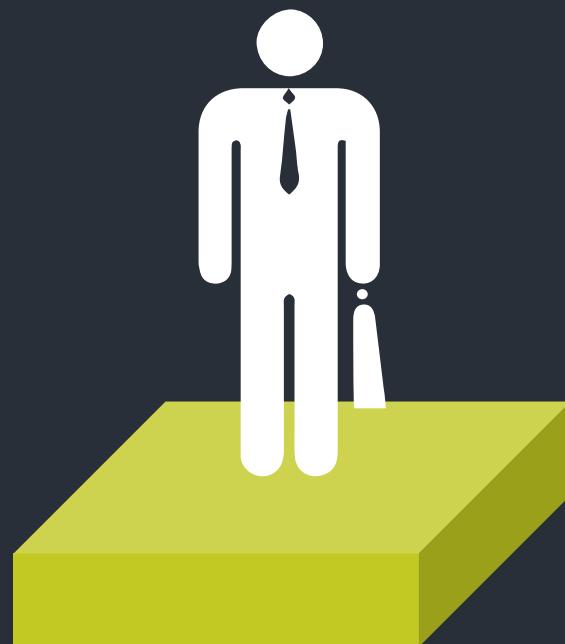
LABORATORIO DI: METODI E MODELLI MATEMATICI IN PYTHON

A CURA DI: ANTONIO MIRARCHI & GIUSEPPE TROTTA

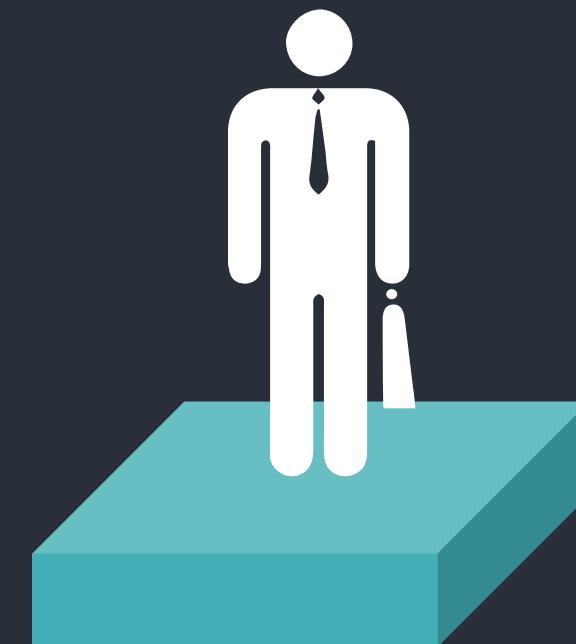
<https://www.labmetodiemodelli.it/>

Chi Siamo?

Antonio Mirarchi
Sr Software Architect
Thinkopen s.p.a.



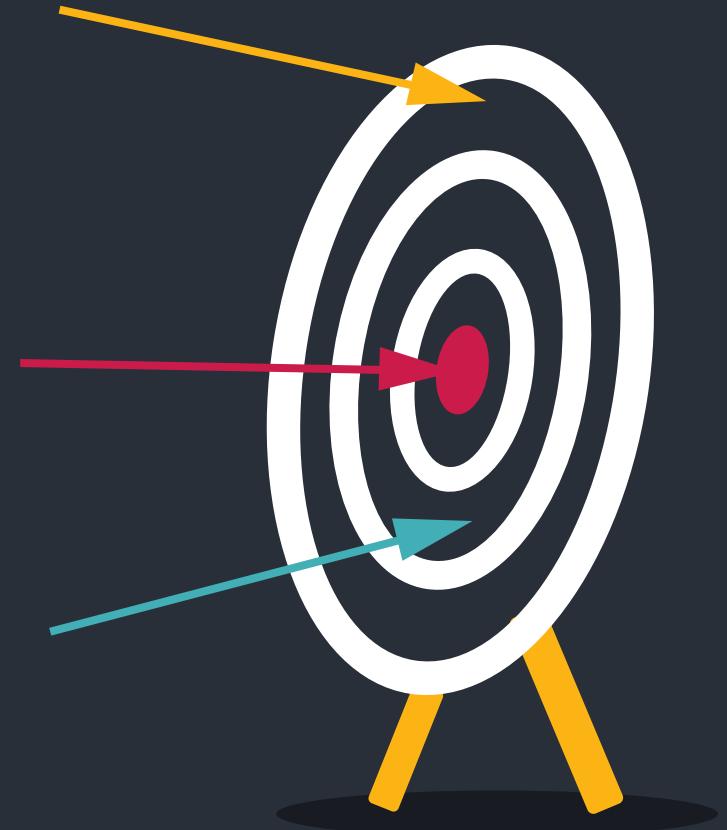
Giuseppe Trotta
COO & AI Architect
HR-Executive s.r.l.



Scopo del Corso



- 01** Fornire una solida conoscenza di Python from “Zero to Hero”, permettendovi di utilizzare in primis Python per gli scopi matematici
- 02** Una panoramica concreta degli algoritmi di machine learning in Python attraverso lo studio di librerie apposite di Data Science
- 03** Alcuni Cenni di Deep Learning attraverso i principali Framework



Ottenimento Crediti

Manuale di sopravvivenza al laboratorio



Almeno 7
laboratori su 10



3 Esercitazioni
Intermedie



Nessun Voto
Finale

Eventi Da Non Perdere



**SCIENZE E TECNOLOGIE
DIPARTIMENTO DI MATEMATICA
«FEDERICO ENRIQUES»**

Seminari di orientamento al lavoro dedicati agli studenti dei Corsi di Laurea in Matematica

13 OTTOBRE 2021 dalle 17:00 alle 18:45

Sala di Rappresentanza, via Saldini n. 50

**KEEP CALM AND LISTEN TO DATA:
un'introduzione al mondo della data science, per comprendere le sfide e le opportunità del mondo data driven**

Intervengono:

LUCA MARIA GIORDANO
Data Scientist | Xnext
Laurea Magistrale in Matematica, Università degli studi di Milano

CLAUDIA LUNINI
Data Scientist, AI, IoT & VR | NTT DATA
Laurea Magistrale in Matematica, Università degli studi di Milano

Modera l'incontro:

TIZIANO PENATI
Docente | Università degli Studi di Milano

PER INFORMAZIONI:
placement.aziende@unimi.it

Evento in collaborazione con il

COSP
Centro di Servizio di Ateneo per l'Orientamento allo Studio e alle Professioni
UFFICIO ORIENTAMENTO POST-LAUREA E PLACEMENT

PER ISCRIVERSI:
lastatalenews.unimi.it > Calendario eventi
Tipologia «Career Service»

La partecipazione è gratuita su iscrizione, con obbligo di **Green Pass**

<https://www.labmetodiemodelli.it/>

01

Il sito di riferimento per il materiale del corso

1

1 Informazioni e News

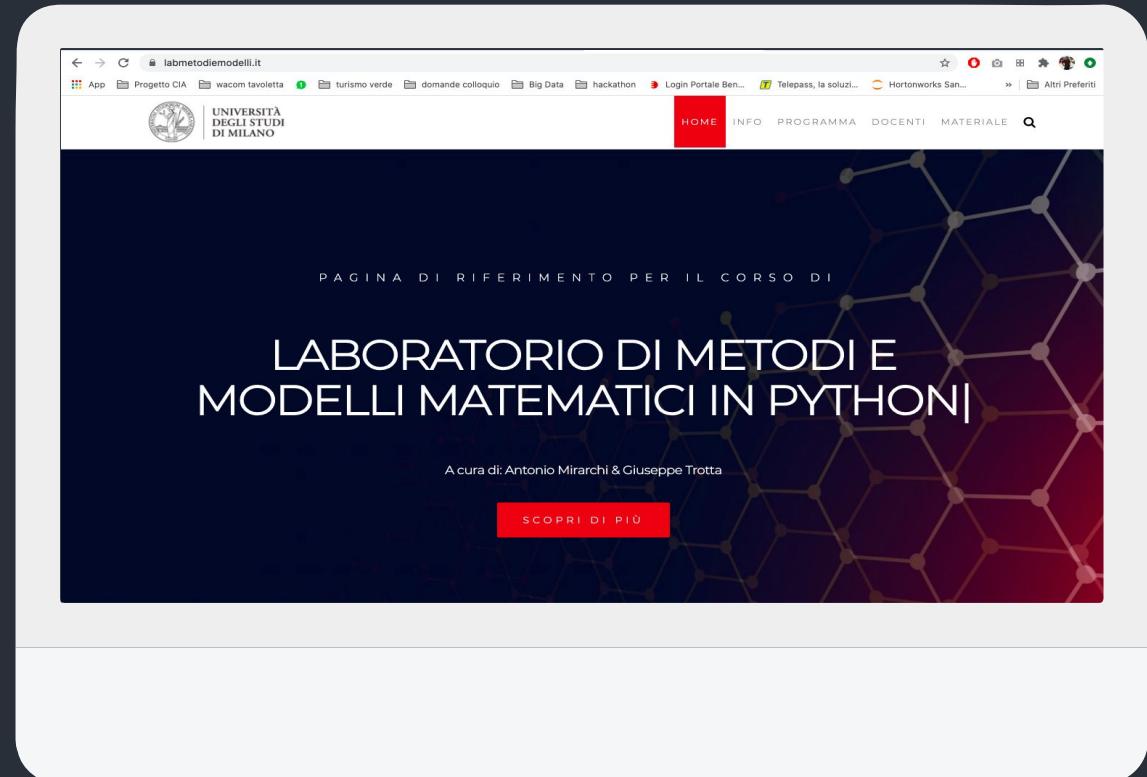
2 Comunicazioni ufficiali

3 Slide delle lezioni

4 Materiale vario del corso

5 Programma Laboratorio

6 Esercizi



<https://www.labmetodiemodelli.it/>

IL PROGRAMMA

INTRODUZIONE A
PYTHON

01

LE STRUTTURE DATI
IN PYTHON

02

LE LIBRERIE PER LA
DATA SCIENCE
(PARTE 2)
+ Test Intermedio

04

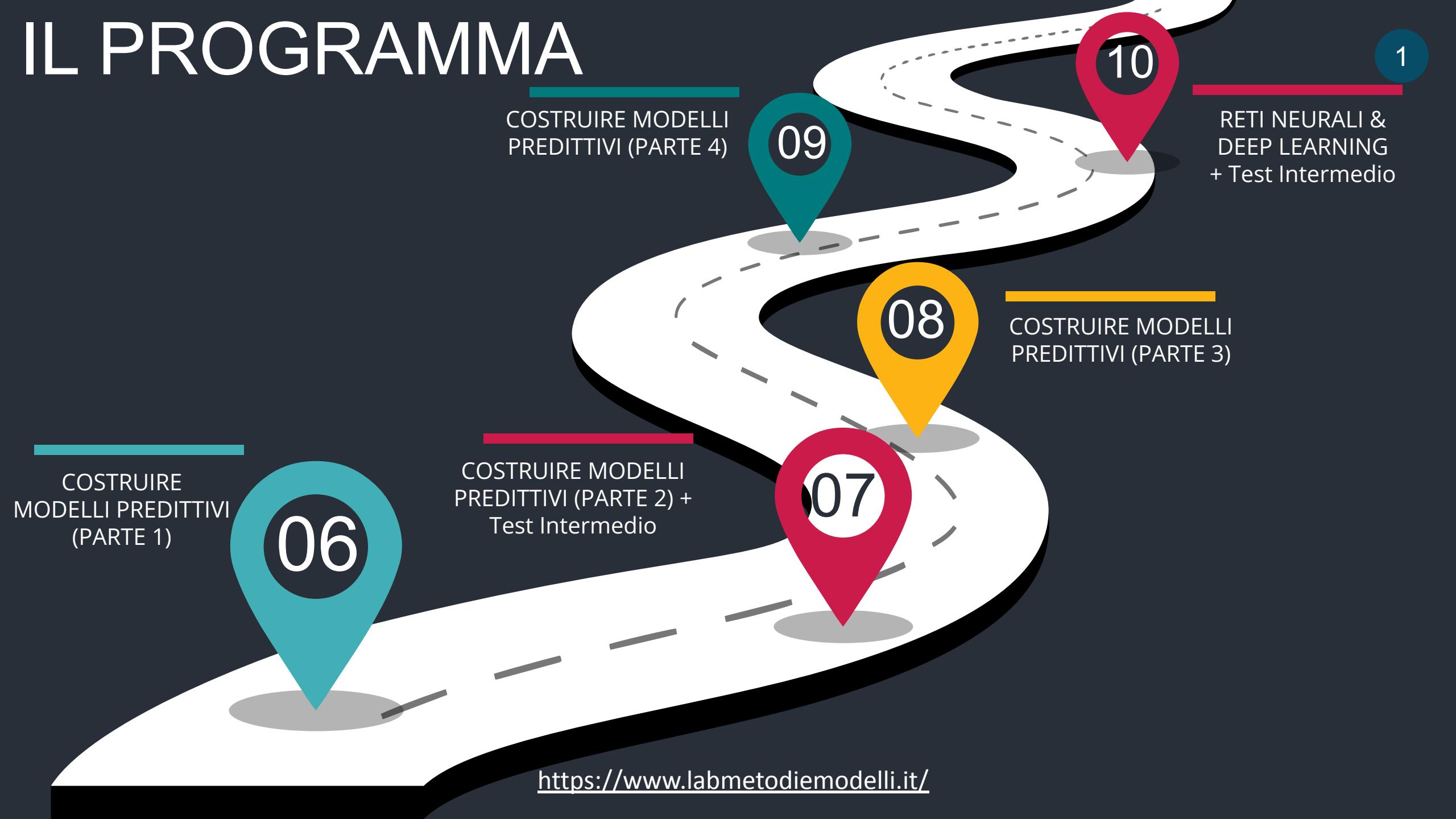
LE LIBRERIE PER LA
DATA SCIENCE
(PARTE 1)

03

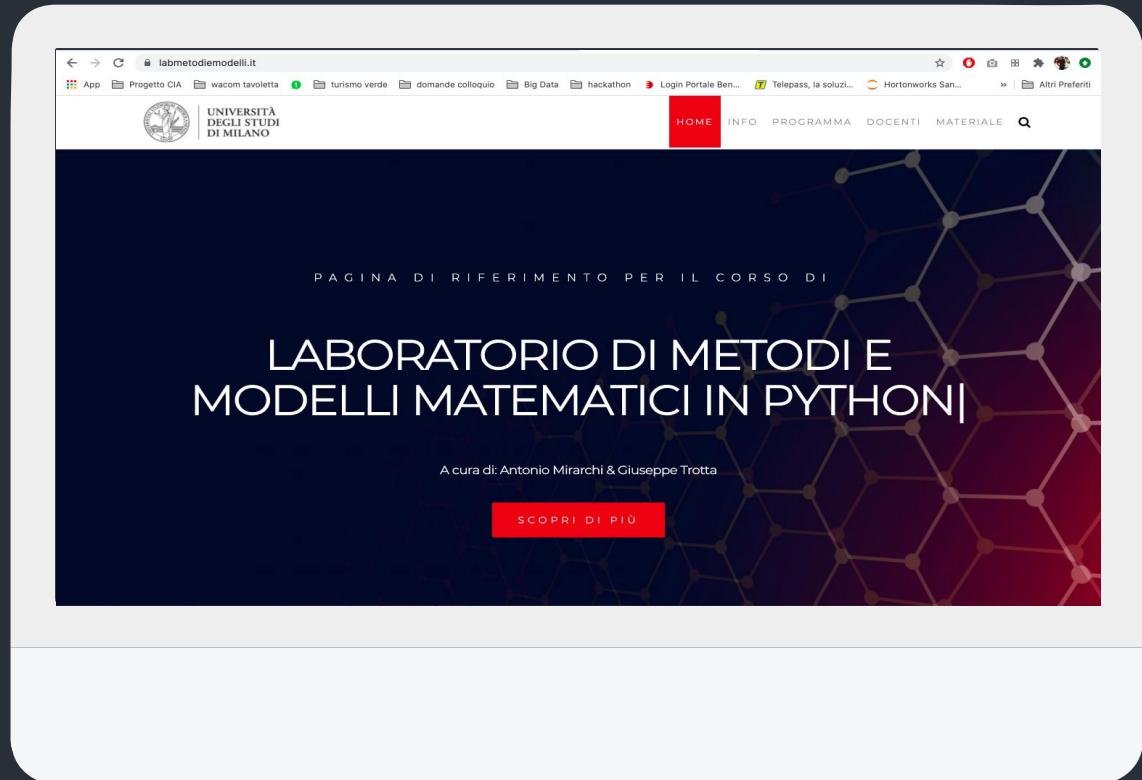


IL PROGRAMMA

1



- 1 Installazioni e Configurazioni
- 2 Cos'è Python?
- 3 Introduzione alla Programmazione
- 4 I Commenti
- 5 Gli Identifier
- 6 L'indentazione

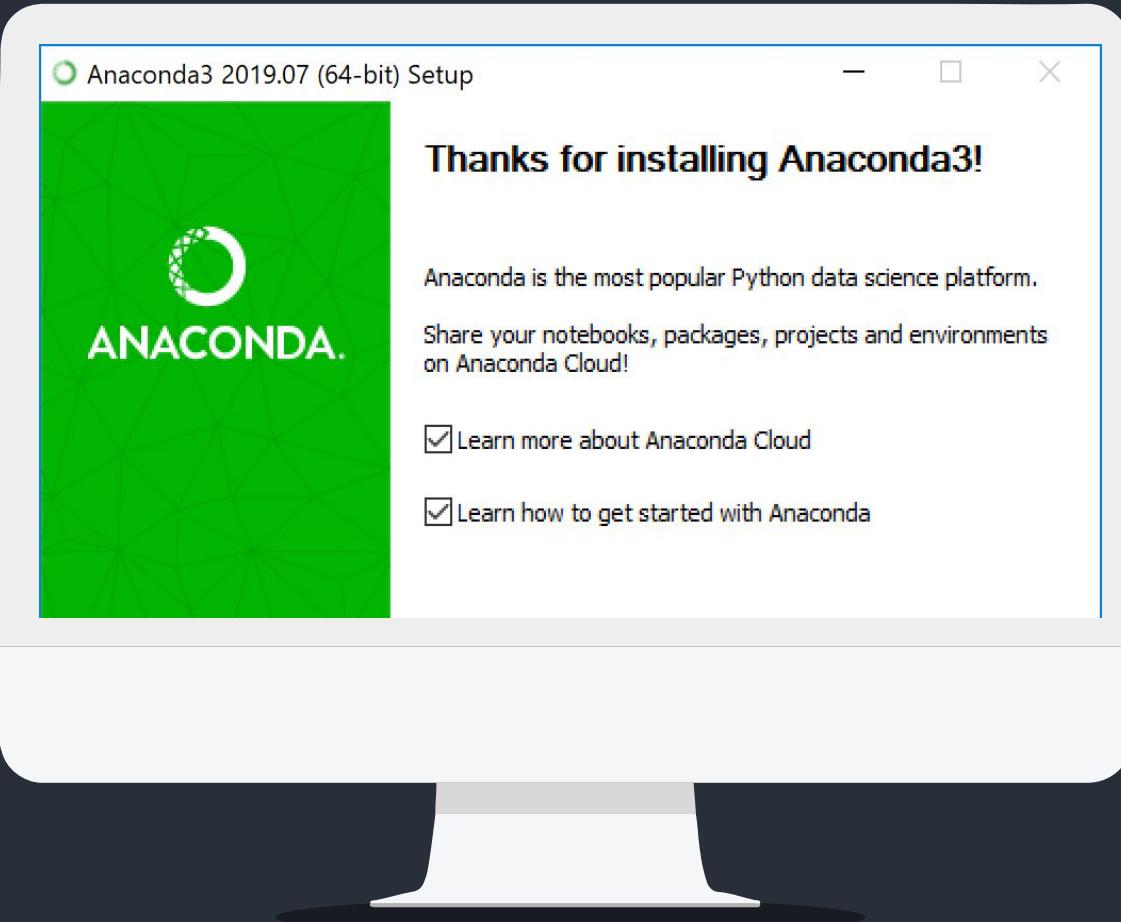


1. Installazioni e Configurazioni

1



Installing on windows



1

Download the Anaconda installer.

2

Double click the installer to launch and follow the steps

3

Choose whether to add Anaconda to your PATH environment variable. Install Anaconda to a directory path that does not contain spaces or unicode characters.

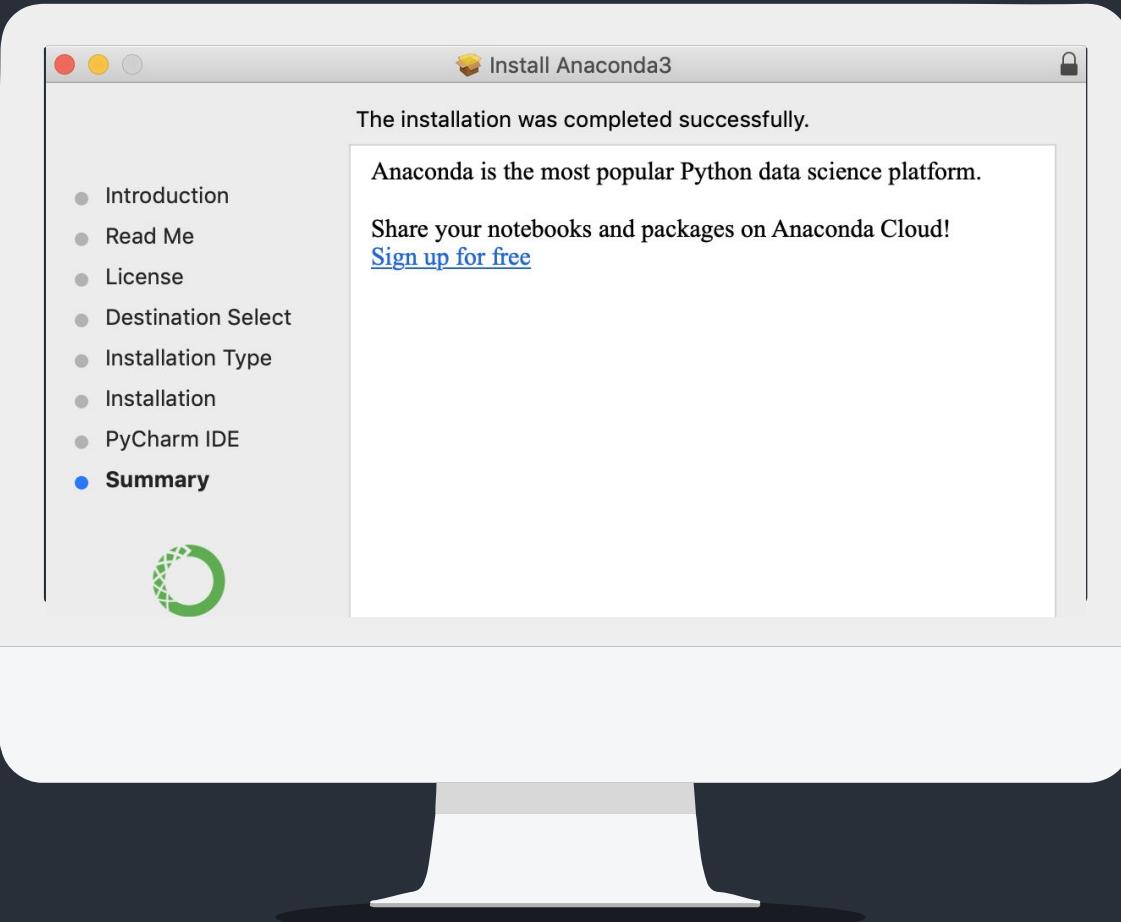
4

Choose whether to register Anaconda as your default Python.

5

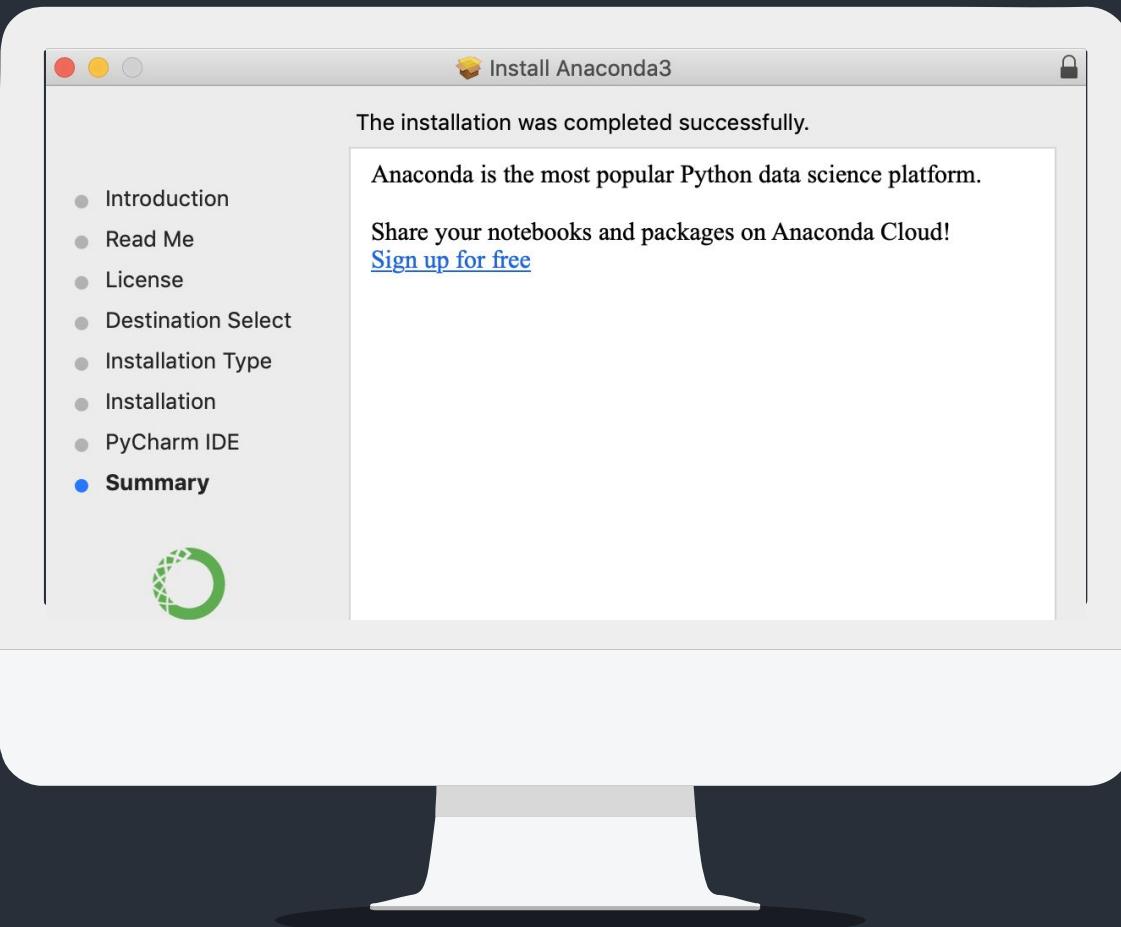
Click the Install button

Installing on MacOS



- 1 Download the graphical [macOS installer](#) for your version of Python.
- 2 Double-click the downloaded file and click continue to start the installation.
- 3 Click the Install button to install Anaconda in your home user directory (recommended)
- 4 OR, click the Change Install Location button to install in another location (not recommended).
On the Destination Select screen, select Install for me only.
Click the Install button

Installing on Linux



1

Check the Prerequisites at :

<https://docs.anaconda.com/anaconda/install/linux/>

2

In your browser, download the Anaconda installer for Linux.

3

Open a terminal and run the following:
sha256sum /path/filename

4

Enter the following to install Anaconda for Python 3.7:
bash
~/Downloads/Anaconda3-2019.07-Linux-x86_64.sh
Follow the steps

5



Let's
Code!

<https://www.labmetodiemodelli.it/>

2. Cos'è Python

01

Python is a popular, multi-paradigm, high level programming language. It was developed by Guido van Rossum in the late 1980's and officially released in 1991.

02

Python supports several programming paradigms such as object-oriented, structured, aspect-oriented, and functional programming.

03

Python is a high-level language that allows codes to be written in nearly regular English statements. This makes Python programs highly readable even to beginners and those without prior programming experience.

04

Python is an interpreted language. Its interpreter runs and converts codes into machine-readable byte codes.

05

Python is a powerful and flexible programming language. It can be used to develop web applications and productivity programs, create games, and write GUIs.

Syntax



- 1 Keywords
- 2 Python Identifiers
- 3 Indentation
- 4 Comments

3. Introduzione alla Programmazione - Identifiers

01

An identifier is a name for a variable, function, class, module, and similar objects. Any object or entity that you intend to use in your program should be appropriately identified or named.

02

An identifier can be a combination of uppercase and lower case letters, underscores, and digits (0-9).

03

An identifier should never start but may end in a number
You may not use special characters such as \$, %, and @ within identifiers.

04

Python is case-sensitive.

05

Almost all identifiers start in lowercase except Class identifiers which, by convention, start in an uppercase letter.

3. Introduzione alla Programmazione - Identifiers

1

06

Quotation marks are used to indicate string literals in Python.
You can use single ('), double (""), or triple ("""") quotes

07

Statements are expressions within a program that can be read and executed by the Python interpreter. Python supports statements such as assignment statement, if statement, for statement, and while statement

08

An identifier should never start but may end in a number
You may not use special characters such as \$, %, and @ within identifiers.

09

Statements may sometimes spill over several lines. To tell Python implicitly that the lengthy expression is a single statement, you can wrap it inside braces {}, brackets [], or parentheses () .

10

Statements may sometimes spill over several lines. To tell Python implicitly that the lengthy expression is a single statement, you can wrap it inside braces {}, brackets [], or parentheses () .



Let's
Code!

<https://www.labmetodiemodelli.it/>

3. Introduzione alla Programmazione - Indentation

Unlike programming languages such as C, C++, and Java which signify blocks of code with braces {}, Python programs are structured through indentation. You can easily identify blocks of code such as loops and functions because they start on the same distance going to the right. If you need to write a more deeply nested code, you'll simply indent another block to the right. The ending is marked by the first unindented line. This rigid language requirement on indentation further enhances the readability of Python codes.

The amount of indentation is not fixed and you may use your preferred indent level. By convention, Python programmers use four white spaces instead of tabs and you might like to consider that option in your own program. Python, however, does require consistency – you need to maintain the same level of indentation within the block.

3. Introduzione alla Programmazione - Comments

Comments are notes that you put into a program to describe a process, step, or other important details. Comments are useful for providing more documentation to your work which can be invaluable in the future when you or other programmers decide to review or revisit the program. A comment is appropriately marked with a hash (#) symbol to instruct the interpreter to ignore the line when you run the code.

For long comments spanning over several lines, you can use a hash (#) symbol at the start of each connected line to wrap them together:

Alternatively, you can wrap multi-line comments with triple quotes.

Paradigmi di programmazione

Programmazione
imperativa



Programmazione
strutturata

Programmazione
procedurale

Programmazione
modulare

Programmazione
dichiarativa



Programmazione
logica

Programmazione
funzionale