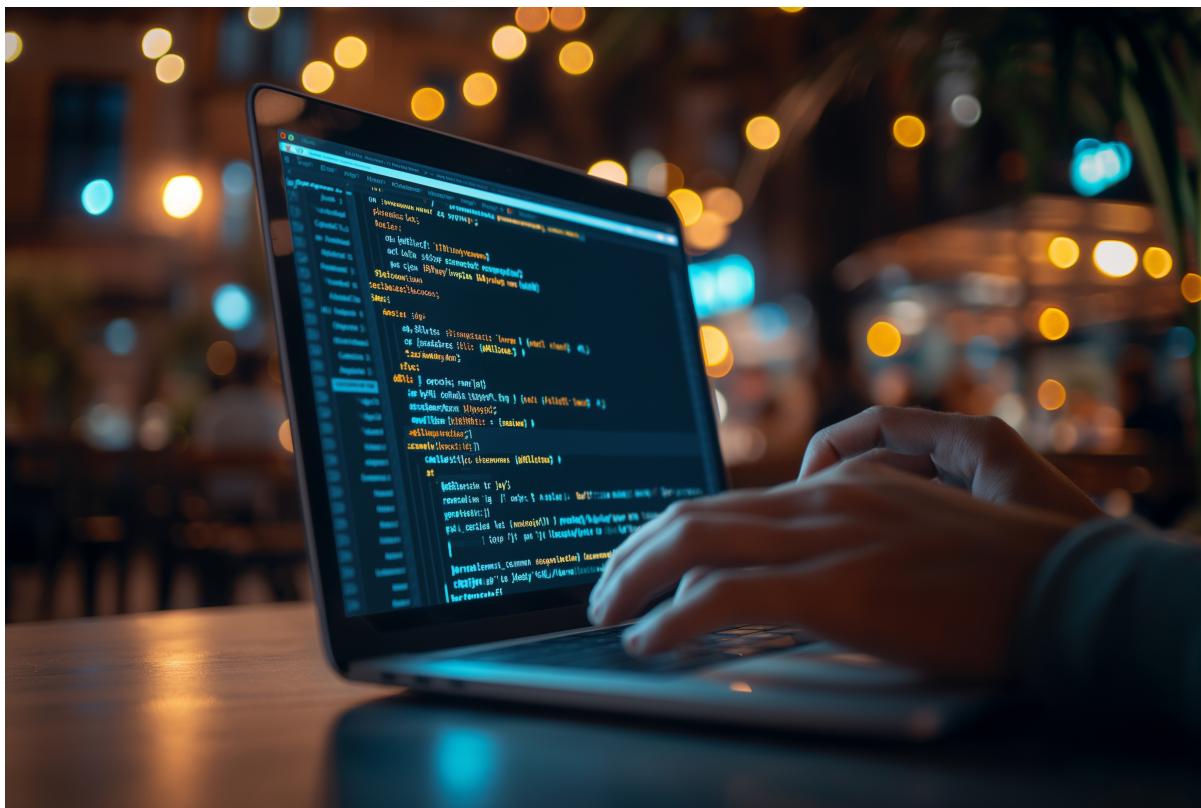


Guida all'installazione

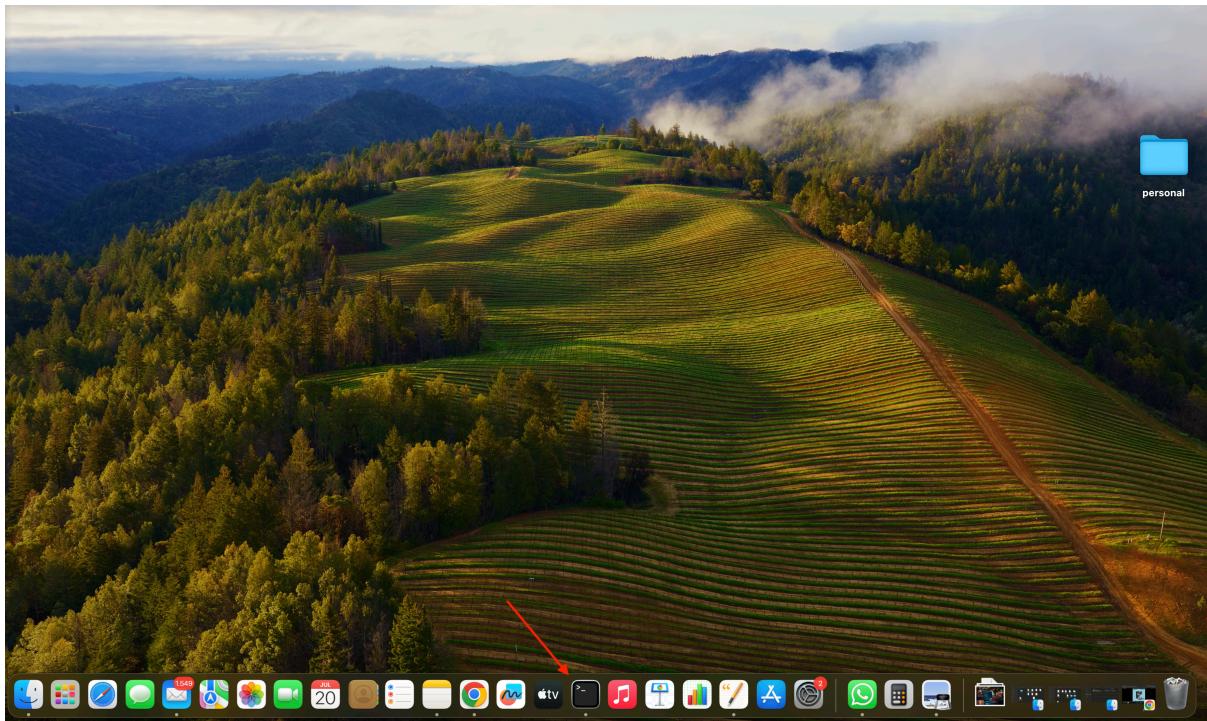
Introduzione a

Python

Questo documento vuole fungere da guida per l'installazione della versione di python 3.8.9 o superiore. Vedremo passo dopo passo come ottenere la versione più recente di python, ottenere i "pacchetti" utili per effettuare le principali operazioni e la piattaforma per programmare. Vedremo come effettuare questa procedura sia per Macbook sia per Windows.



Per prima cosa, e' necessario verificare la versione di python presente sul vostro computer. Per fare ciò, apri il terminale del tuo PC: su **Macbook**, il terminale è presente nella barra delle applicazioni (Dock) come indicato nell'immagine.



Su **Windows** si può digitare “prompt dei comandi” nella barra di ricerca in modo da trovare un’icona simile a quella in figura. A questo punto basta digitare nel prompt “python –version” per verificare quale versione è installata di default sul proprio PC

```
simopi~/Users/simonepiscitelli ~ python --version
Python 3.8.9
simopi~/Users/simonepiscitelli ~
```

Nel caso non avessi python 3 è possibile scaricarlo dal sito ufficiale. Ecco gli steps da seguire per i due sistemi operativi.

Windows:

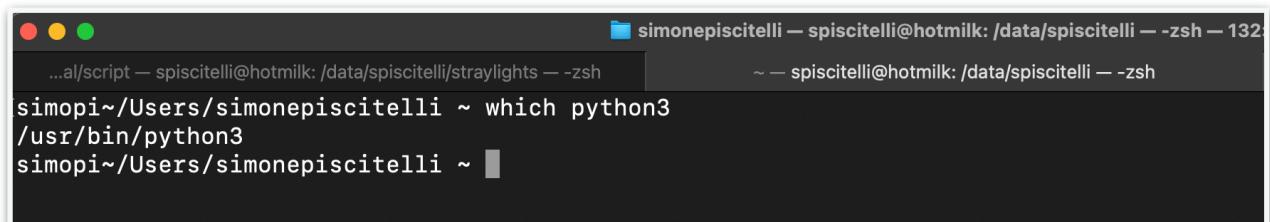
- Scarica l’installatore di Python 3.8:** Vai sul sito ufficiale di Python <https://www.python.org/downloads/> e scarica l’installatore per la versione a 32 o 64 bit del tuo sistema operativo. La versione a 32 bit dovrebbe bastare per i nostri scopi.
- Esegui l’installatore:** Esegui il file .exe scaricato e segui le istruzioni del wizard di installazione. Assicurati di selezionare l’opzione per aggiungere Python 3.8 al tuo percorso di sistema durante l’installazione.

3. **Verifica l'installazione:** Apri il prompt dei comandi e digita `python3 --version`. Se l'installazione è stata eseguita correttamente, dovresti vedere la versione di Python 3.8 installata.

macOS:

1. **Scarica il file macOS Installer:** Vai sul sito ufficiale di Python <https://www.python.org/downloads/> e scarica il file macOS Installer per la tua versione di macOS.
2. **Esegui l'installer:** Apri il file `.pkg` scaricato e segui le istruzioni del programma di installazione.
3. **Verifica l'installazione:** Apri il Terminale e digita `python3 --version`. Se l'installazione è stata eseguita correttamente, dovresti vedere la versione di Python 3.8 installata.

Perché python funzioni correttamente è necessario piazzarlo in una determinata posizione nel tuo PC. Per verificare dov'è può essere utile eseguire il comando “which python3” come in figura.



```
simonepiscitelli — spiscitelli@hotmilk: /data/spiscitelli — zsh — 132
...al/script — spiscitelli@hotmilk: /data/spiscitelli/straylight — zsh
simopi~/Users/simonepiscitelli ~ which python3
/usr/bin/python3
simopi~/Users/simonepiscitelli ~
```

Questo comando andrà a verificare dove è stato installato python. Se è nella cartella `/usr/bin/` allora il procedimento è andato a buon fine.

NOTA: può essere importante verificare sia la versione del proprio sistema operativo sia la tipologia di terminale! Diversi sistemi operativi hanno terminali differenti (bash, zsh..) con linguaggi differenti. Pertanto, comandi come “which” possono essere diversi a seconda del terminale. Verifica quale terminale possiedi digitando “echo \$SHELL”.

Installazione dei pacchetti

Spesso nell'informatica se hai avuto un'idea qualcuno l'ha già realizzata; magari pure meglio. Python infatti si serve dei cosiddetti "pacchetti" o più correttamente "moduli" per svolgere particolari operazioni. Questi moduli contengono in generale comandi che possono effettuare operazioni complicate. Lo scopo di questo corso è infatti imparare ad usare i pacchetti principali.

Il comando per l'installazione dei pacchetti è "pip", un comando a sua volta preinstallato nelle versioni più recenti di python. In figura è presente un esempio di installazione di un pacchetto noto come "numpy", uno dei principali pacchetti per svolgere operazioni matematiche

```
pip install numpy
```

Assicurati di installare i pacchetti con questo comando e verificarne la presenza con il comando *which*. Puoi installare anche più pacchetti allo stesso tempo separando i nomi con la virgola.

Ecco una lista dei moduli principali che utilizzeremo in questo corso:

- 1) numpy
- 2) matplotlib
- 3) Pandas

Ambiente Virtuale

In python, l'ambiente virtuale può essere visto come una stanza di una casa dedicata solamente a certi utilizzi. Talvolta, per questioni organizzative, è molto conveniente "creare la propria stanza" per ogni progetto. In teoria, le versioni dalla 3.3 in poi di python dovrebbero avere il comando "venv" incluso. Se ne può verificare la presenza con il solito comando "which" nel terminale.

Nel caso il comando “venv” non venga trovato, è possibile installare “virtualenv” digitando

```
pip install virtualenv
```

e verificarne la presenza sempre nel solito modo (which). Il comando pip è a sua volta preinstallato nelle versioni recenti di python.

A questo punto è possibile creare il proprio ambiente virtuale utilizzando il comando virtualenv come nella figura seguente (ho scelto “mio_ambiente” come nome)

```
personal ~ spiscitelli@hotmilk: /data/spiscitelli ~ zsh ~ 132x39
...al/script ~ spiscitelli@hotmilk: /data/spiscitelli/straylight ~ zsh
...esktop/personal ~ spiscitelli@hotmilk: /data/spiscitelli ~ zsh
...itelli@hotmilk: /data/spiscitelli/straylight/straylight ~ zsh +[

simopi~/Users/simonepiscitelli ~/Desktop/personal virtualenv mio_ambiente
created virtual environment CPython3.8.9.final.0-64 in 686ms
  creator CPython3macOsFramework(dest=/Users/simonepiscitelli/Desktop/personal/mio_ambiente, clear=False, no_vcs_ignore=False, globall=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/Users/simonepiscitelli/Library/Application Support/virtualenv)
    added seed packages: pip==24.1, setuptools==70.1.0, wheel==0.43.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
simopi~/Users/simonepiscitelli ~/Desktop/personal ls
Bayesian Codes      Python Course      books      mio_ambiente      typeMstars
Deep Learning       Simone Piscitelli.pdf  documenti   script
PhD                bashexercise     merger_ringdown_test  spagnolo
simopi~/Users/simonepiscitelli ~/Desktop/personal
```

Il comando utilizzando ha quindi creato l’ambiente “mio_ambiente” nella cartella in cui ero (Desktop/personal).

Una volta creato, l’ambiente va “attivato” in modo tale da poter interagirci. Ecco di seguito i comandi che attivano l’ambiente:

- 1) Macbook: source mio_ambiente/bin/activate
- 2) Windows: source mio_ambiente/Scripts/activate

A questo punto, l’attivazione dell’ambiente sarà completata nel momento in cui comparirà il nome dello stesso nel terminale

```
mio_ambiente ~ spiscitelli@hotmilk: /data/spiscitelli ~ zsh ~ 132x39
...al/script ~ spiscitelli@hotmilk: /data/spiscitelli/straylight ~ zsh
...al/mio_ambiente ~ spiscitelli@hotmilk: /data/spiscitelli ~ zsh
...itelli@hotmilk: /data/spiscitelli/straylight/straylight ~ zsh +[

simopi~/Users/simonepiscitelli ~/Desktop/personal/mio_ambiente ls
bin      lib      pyvenv.cfg
simopi~/Users/simonepiscitelli ~/Desktop/personal/mio_ambiente source bin/activate
(mio_ambiente) simopi~/Users/simonepiscitelli ~/Desktop/personal/mio_ambiente
```

Una volta fatto l’accesso nel tuo ambiente è possibile installare i pacchetti desiderati come indicato nel paragrafo precedente. Il comando pip, una volta

dentro l'ambiente, installerà pacchetti solo nell'ambiente stesso. Questo può portare dei vantaggi in quanto alcuni pacchetti potrebbero essere in conflitto tra di loro. La separazione degli ambienti risulta, quindi, essere conveniente in queste circostanze.

Nel caso si volesse creare un nuovo ambiente è necessario disattivare l'ambiente in cui si è. Per farlo si può semplicemente digitare il comando *deactivate*.

Iniziare a programmare: Jupyter

Tra le tante piattaforme che ci sono per programmare in python trovo particolarmente comodo *Jupyter Notebook*, ovvero un interfaccia molto ben organizzata che consente di compilare codice in maniera molto intuitiva.

Per installare questa piattaforma utilizziamo sempre il comando pip

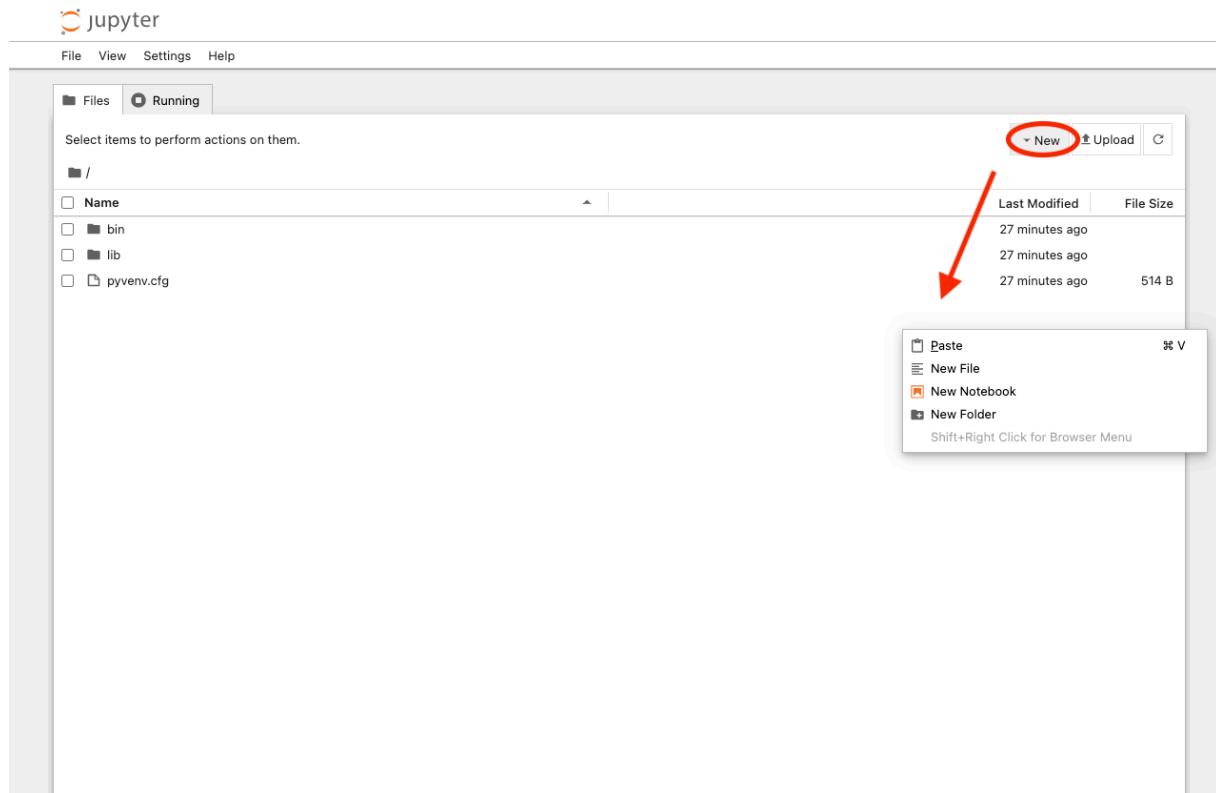
```
pip install jupyter notebook
```

Verifichiamone la presenza e la versione con i soliti comandi *which* e *—version*, per poi avviare il programma con il seguente comando

```
jupyter notebook
```

Questo comando avvia automaticamente la piattaforma o restituisce un link “*http...*” da inserire nel browser.

In figura un esempio di come appare la piattaforma *Jupyter Notebook*. È possibile creare cartelle per organizzare i propri progetti ma soprattutto notebook in cui programmare.



Con questo è tutto! Ora hai compiuto il primo passo verso la programmazione in Python. Buon divertimento!

SP