

Informatica

Appunti dal laboratorio 2

Concetti fondamentali

- Esistono programmi, come Microsoft Word oppure Acrobat Reader, che vengono usati dagli utenti per far eseguire al computer determinate operazioni (come ad esempio quelle legate alla scrittura e lettura di documenti).
- Esistono anche programmi più particolari, come l'interprete Python, che vengono sempre usati dagli utenti per far eseguire operazioni al computer, solo che, a differenza degli altri programmi, la libertà sulle operazioni è molto più ampia, perché si permette all'utente di specificarle completamente da zero.
- Gli interpreti, tra cui quello Python, permettono agli utenti di scrivere programmi: nel corso del laboratorio gli studenti imparano (seppur in maniera rudimentale) ad essere programmatori e non solo semplici utenti di computer.

Come installare l'interprete Python sul proprio computer

- L'interprete Python si scarica sul sito python.org
- È disponibile in due versioni, la 2.7 e la 3.5
- La 3.5 è più recente, ma le uniche differenze non sono particolarmente significative per il contesto di questo corso. Il libro di testo fa comunque riferimento ad essa.
- Qualunque tipo di sistema operativo governi il vostro computer (Windows, Mac OS, Linux), c'è una versione dell'interprete Python
- Al momento dell'installazione ricordatevi dove viene installato (questa informazione servirà dopo)

Uso dell'interprete Python

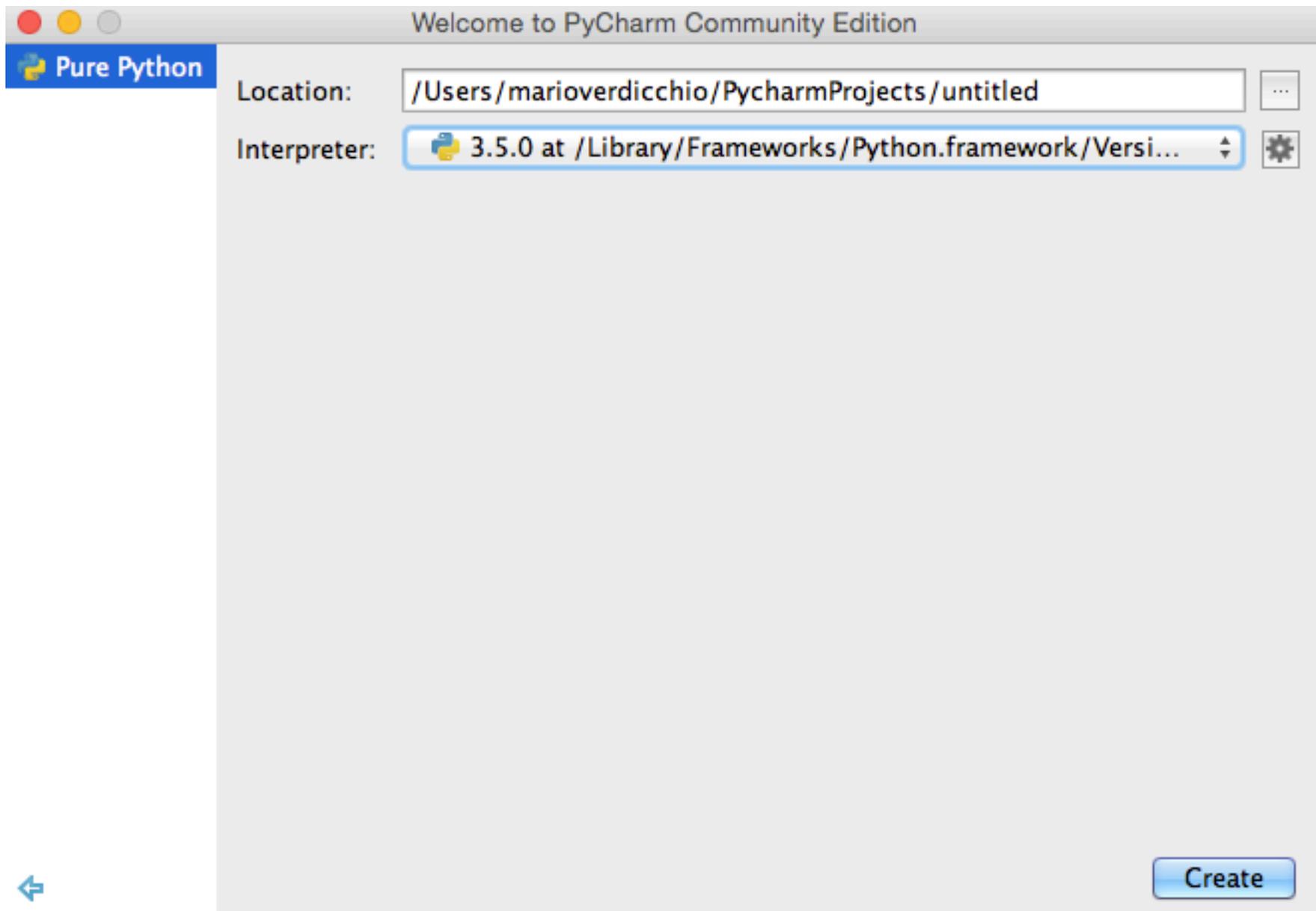
- Nella cartella in cui l'interprete viene installato, troverete un file chiamato "IDLE"
- Fateci doppio click sopra per attivarlo
- Si aprirà una finestra in cui potete inserire comandi che l'interprete Python eseguirà
- Python è di fatto il nome di un linguaggio di programmazione
- Questo interprete riceve comandi scritti in questo linguaggio e li esegue
- L'esecuzione è immediata: appena premete "invio" alla fine del comando l'interprete lo esegue (se esso è scritto secondo le regole sintattiche di Python)

Esempio di uso

```
Python 3.5.0 Shell
Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015, 11:00:19)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.
4 + 3
7
>>> print ("ciao")
ciao
>>> x = 7
>>> x
7
>>> x = x + 1
>>> x
8
>>> print ("x vale ", x)
x vale 8
>>>
```

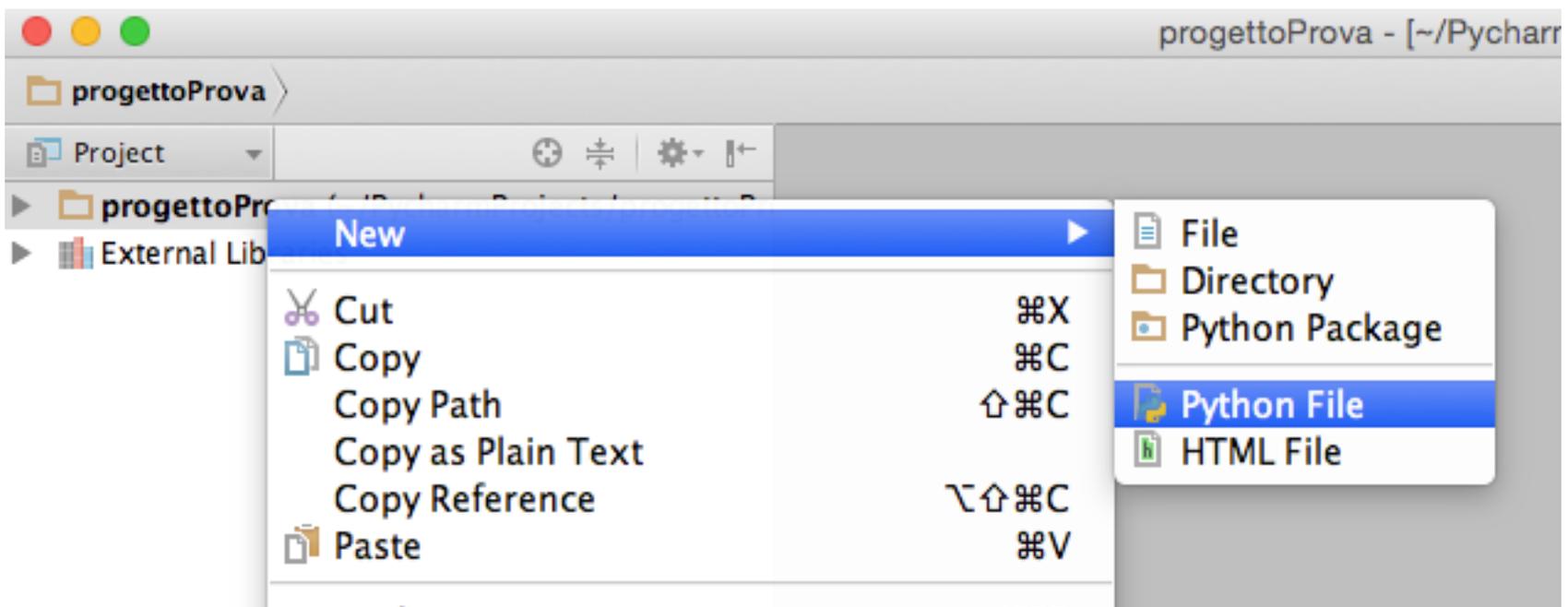
Ambiente di programmazione PyCharm

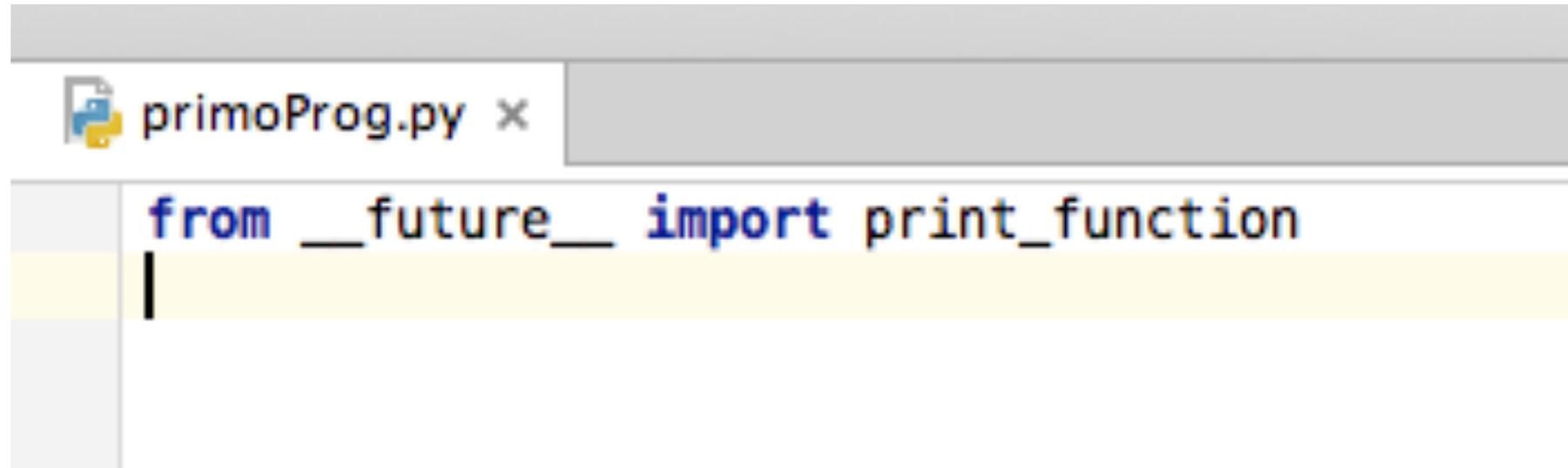
- Per avere un'interazione più sofisticata con l'interprete Python, che ci permetta di avere suggerimenti sulla sintassi del linguaggio, di scrivere programmi, salvarli, riaprirli e modificarli, fate il download dell'ambiente di programmazione (IDE: Integrated Development Environment) PyCharm sul sito www.jetbrains.com/pycharm
- PyCharm costruisce un ambiente intorno all'interprete Python con il suddetto scopo
- Quando create un nuovo progetto con PyCharm vi chiederà quale interprete Python usare: indicate il percorso per raggiungere quello che avete installato in precedenza
- PyCharm nel menù a tendina "Interpreter" dovrebbe individuare le varie possibilità per voi



Scrittura di un programma in Python

- Una volta stabilito il nome del progetto (è la parte finale del contenuto della riga “location” della schermata precedente), cliccate su “create” per creare la cartella dedicata al progetto
- Tale cartella è destinata a contenere i vari programmi che fanno capo al progetto
- Per creare il vostro primo programma cliccate destro sul nome del progetto nella colonna a sinistra, poi, nei menù che compaiono, selezionate “new” e poi “Python file” (vedi figura successiva)





```
primoProg.py x  
from __future__ import print_function  
|
```

- Se state usando l'interprete Python 2.7 e volete usare il comando di output (print) come si fa nella versione 3.5 e sul libro, nella prima riga del programma inserite la dicitura di cui sopra
- Essa serve a forzare l'uso della nuova versione della print anche nel vecchio interprete

Operazioni di output

- Per stampare su schermo, il comando è “print”
`print(“ciao a tutti”)`
`print(x)`
- Il comando è seguito dagli elementi da stampare (se ve n’è più d’uno, separati da una virgola) racchiusi tra parentesi tonde
- Gli elementi da stampare così come sono (stringhe di caratteri) vanno messi tra virgolette, altrimenti vengono interpretati come nomi di variabili e il loro valore viene stampato su schermo
- Ogni volta che viene eseguita una print, il computer va a capo automaticamente; se volete farlo rimanere sulla stessa riga dopo una print, dovete specificare che la print viene conclusa con uno spazio
`print(“ciao a tutti”, end=“ ”)`

Operazioni di input

- Il comando per eseguire un input si chiama, appunto, “input”
- In Python si può eseguire in concomitanza con un output che specifica all’utente che tipo di dato viene richiesto per il computer

```
x = input(“inserisci un numero”)
```

- Con questo comando si visualizza su schermo la stringa, e ciò che l’utente inserisce prima di premere invio viene salvato nella variabile x
- Python 2.7 dà per scontato che in arrivo dalla tastiera ci sia un numero, mentre Python 3.5 lo considera una stringa (sequenza di caratteri), quindi se state usando 3.5 e volete che in x venga salvato un numero, dovete scrivere

```
x = int(input(“inserisci un numero”))
```

dove la funzione “int” attorno alla funzione “input” specifica che il dato in arrivo va considerato come un numero intero.

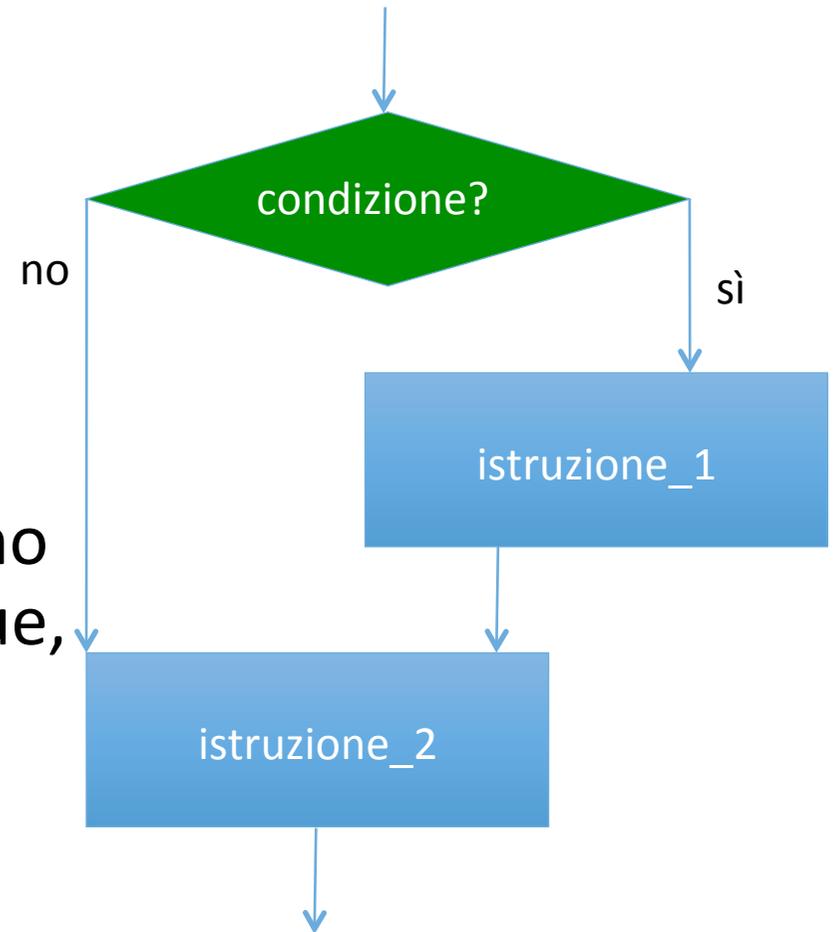
Operazioni condizionali: if

if <condizione>:

 istruzione_1

istruzione_2

- Questo codice funziona come in figura
- Al posto di <condizione> possiamo inserire una condizione qualunque, anche composta, come $x \geq 0$ and $y < z$



Operazioni condizionali: if-else

```
if <condizione>:  
    istruzione_1  
else:  
    istruzione_2
```

- Il codice funziona come in figura
- Attenzione: l'allineamento verticale delle istruzioni (indentazione) ne determina la dipendenza (o meno) dalla condizione

