INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Introduzione a Python

by Martina Bellio - 2020

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Introduzione a Python

Comunicazioni di servizio:

1. Modalità WEBINAR (solo chat attiva),
ore 10.00-12.00 e 14.00-16.00
intervalli: 10:50-11:10 ; 14:50:15:10, per la chat
2. COME INTERAGIRE : negli intervalli e alla fine dell'incontro
3. E dopo questa giornata ? E-mail : **canella@lnl.infn.it**
4. Come è nato questo corso ? (Raspberry PI, laboratori e stage estivi, corso LNL, corso INFN)
5. A chi devo riconoscimenti ? (ACKNOWLEDGMENTS)

Ai miei colleghi LNL, alla RASPBERRY PI Foundation,

a : Dr. Charles R. Severance (MSU)

INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Generalità

Il corso ...

Questi incontri hanno lo scopo di fornire una prima introduzione all'uso del linguaggio di programmazione PYTHON.

Durata corso: 2 lezioni di 50' ciascuna, con la descrizione di esempi pratici di brevi programmi.

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Calendario

9 giugno 2020	: Per cominciare …
(mattina)	Nuts and Bolts
9 giugno 2020	: Programmi object-oriented
(pomeriggio)	Esempi

INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Contenuti 1

1 - Per cominciare …

- Introduzione a Python.
- Installazione ed uso dell'ambiente IDLE.
- Primi esempi ("Ciao Mondo!").
- Uso interattivo e tramite script.
- Uso diretto dell'interprete Python.

INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Contenuti 2

Nuts and Bolts

- I costrutti base per i programmi procedurali.
- Le variabili e i valori.
- Input/Output.
- Le strutture di controllo.
- Esempi ed esercizi.

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Contenuti 3

2 - I programmi object-oriented (OO)

- Introduzione alla programmazione OO.
- Predisposizione di IDLE per usare GUIZERO.
- Elementi base per programmi OO con Guizero.
- Primi esempi di programmi Guizero.
- Idee per proseguire con Guizero.

INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Contenuti 4

Esempi (di programmi object-oriented - OO)

- GUI : una finestra con un testo.
- Una finestra con un pulsante che modifica la finestra stessa.
- Un text editor.
- Esempi per proseguire.

Contenuti 5

Per Finire ...

- Cos'altro si può fare con Python ?
- Gestione di GUI e grafica in generale.
- Tools di comunicazione in rete.
- Tools di supporto alla gestione di siti web.
- Elaborazioni di immagini.

Suggerimenti

Suggerimenti ...

Dove seguire corsi dedicati:

Su https://www.futurelearn.com (corsi brevi)

- Raspberry Pi Foundation : 8 corsi (101, 102, 103)

3 o 4 settimane;

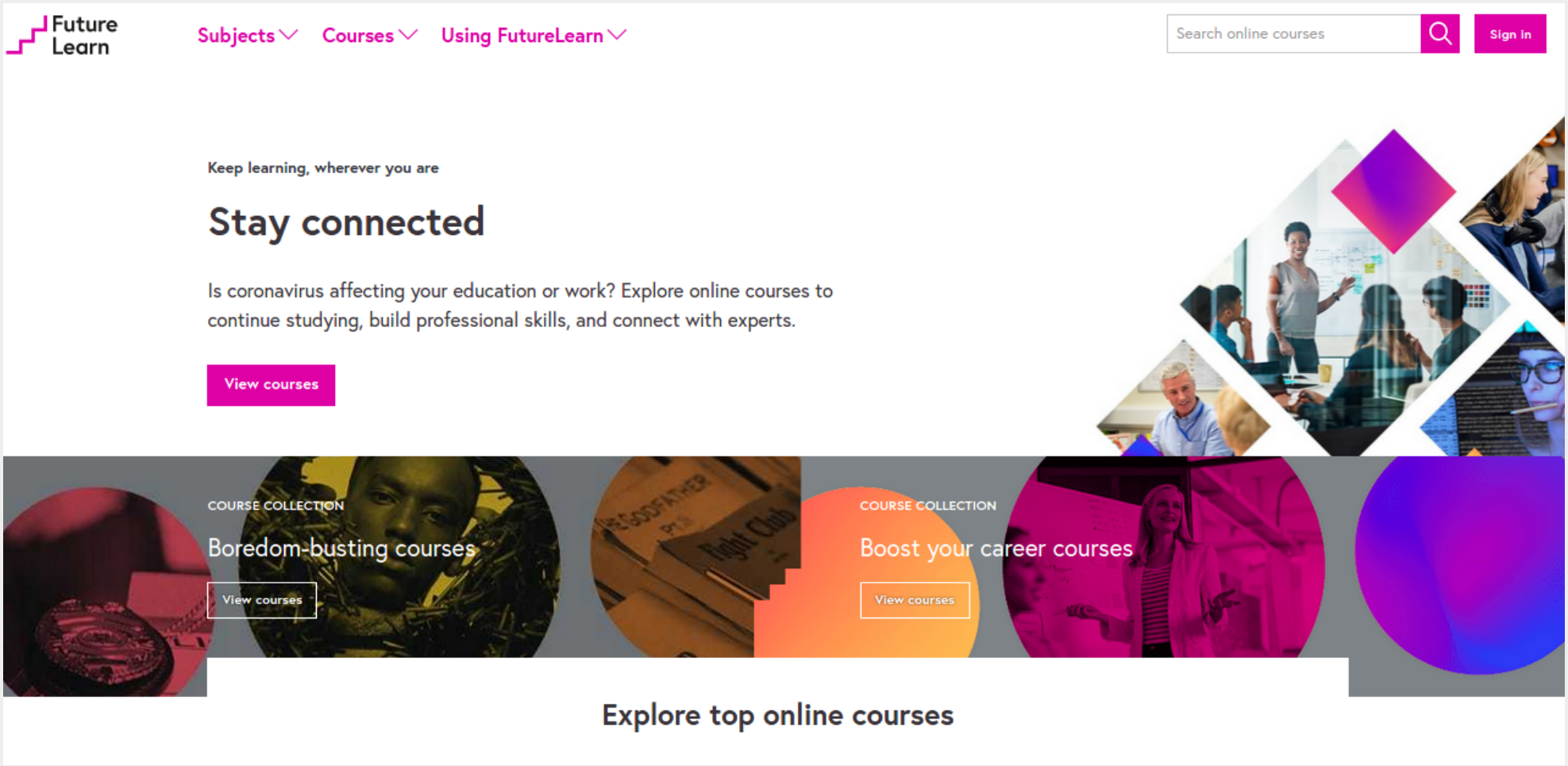
- **Programming for Everybody** (Univ. Michignan)

6 settimane.

Exploring Data Using Python 3, by Dr. Charles R. Severance (MSU)

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Altre risorse 1



https://www.futurelearn.com

Altre risorse 2

Raspberry Pi Foundation

About

26 corsi prodotti da RASPBERRY PI FOUNDATION

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Altre risorse 3

Tra I 26 corsi prodotti da RASPBERRY PI FOUNDATION:

Object-oriented Programming in Python: Create Your Own Adventure Game

Networking with Python: Socket Programming for Communication

Teaching Physical Computing with Raspberry Pi and Python

Programming with GUIs (GUIZERO)

Understanding Maths and Logic in Computer Science

Representing Data with Images and Sound: Bringing Data to Life (IMAGE)

Programming 101 ; Programming 102 ; Programming 103

Altre risorse 4

Dr. Charles R. Severance (MSU)

Future Learn

Subjects Courses Using FutureLearn

Search online courses

Sign in

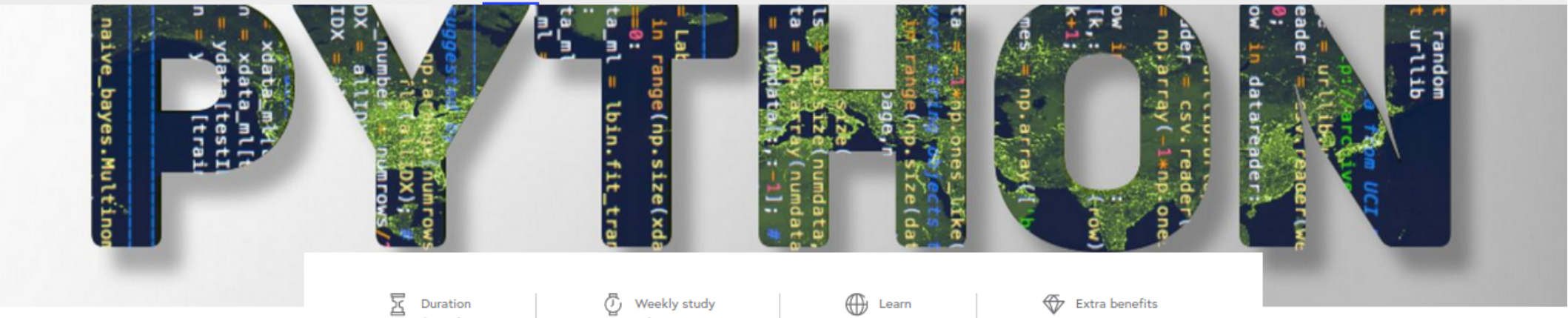
Courses / IT & Computer Science

Programming for Everybody (Getting Started with Python)

Master the basics of Python programming, and learn how to use programming tools and variables with the University of Michigan.

Join course for free

Overview Topics Start dates Requirements Educators Try it More courses



Duration
6 weeks

Weekly study
4 hours

Learn
Free

Extra benefits
From €59
Find out more

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Python

Python è un linguaggio di programmazione interpretato, “high-level”, “general-purpose”, e “open-source”.
E’ stato ideato da Guido van Rossum (un informatico olandese) all'inizio degli anni ‘90.
Il nome è legato alla passione di Van Rossum per i Monty Python (un gruppo comico britannico).

Python

Python enfatizza la leggibilità del codice con l'uso della spaziatura.
I suoi costrutti linguistici e l'approccio orientato agli oggetti mirano ad aiutare i programmatori a scrivere codice chiaro e logico per progetti su piccola e grande scala.
Python può essere considerato una versione moderna di Visual Basic: è adatto a sviluppare applicazioni distribuite, per la produzione di scripting, per la computazione numerica e la prototipazione.

INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Python

Python supporta molteplici paradigmi di programmazione: la programmazione procedurale classica e quella orientata agli oggetti. Python è spesso descritto come un linguaggio "batterie incluse" per la libreria standard completa e le molte librerie aggiuntive disponibili.
E' il linguaggio di programmazione principale su Raspberry PI con il sistema operativo Raspian.

INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

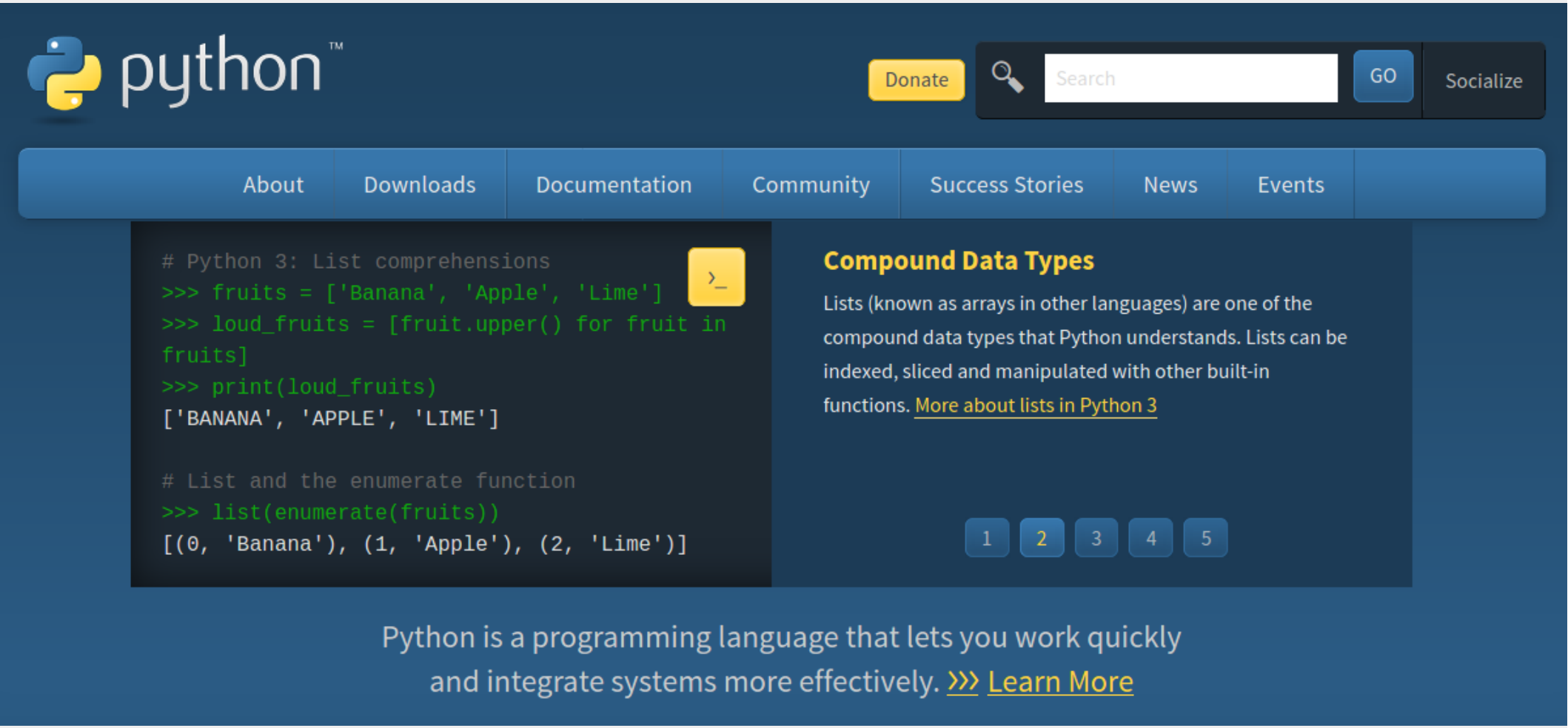
Python e IDLE

Per cominciare …

- Installazione ed uso di Python e IDLE.
- Primi esempi (“Ciao Mondo!”).
- Uso interattivo e tramite script.
- Uso diretto dell’interprete Python.

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

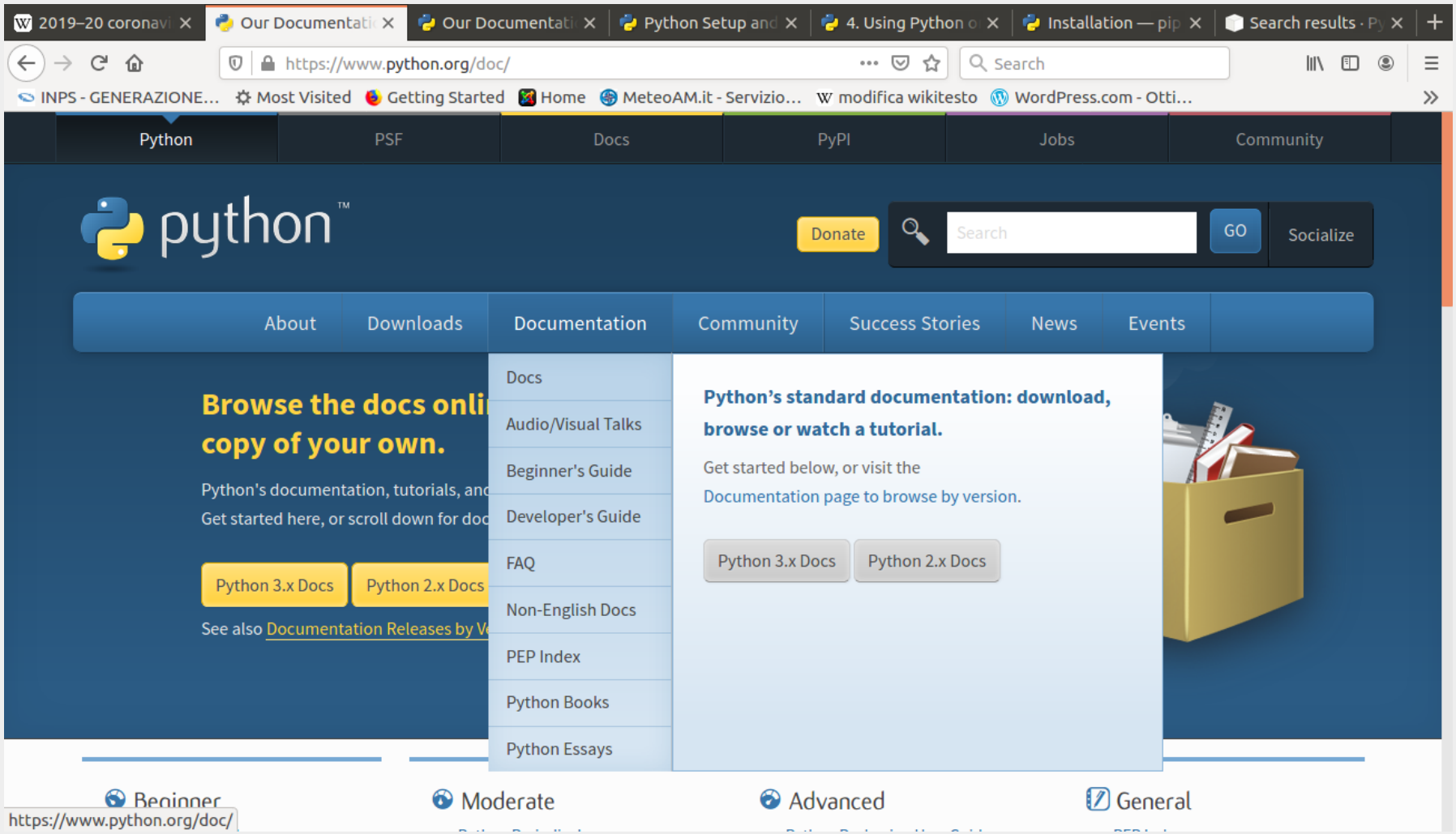
Per cominciare



https://www.python.org/

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

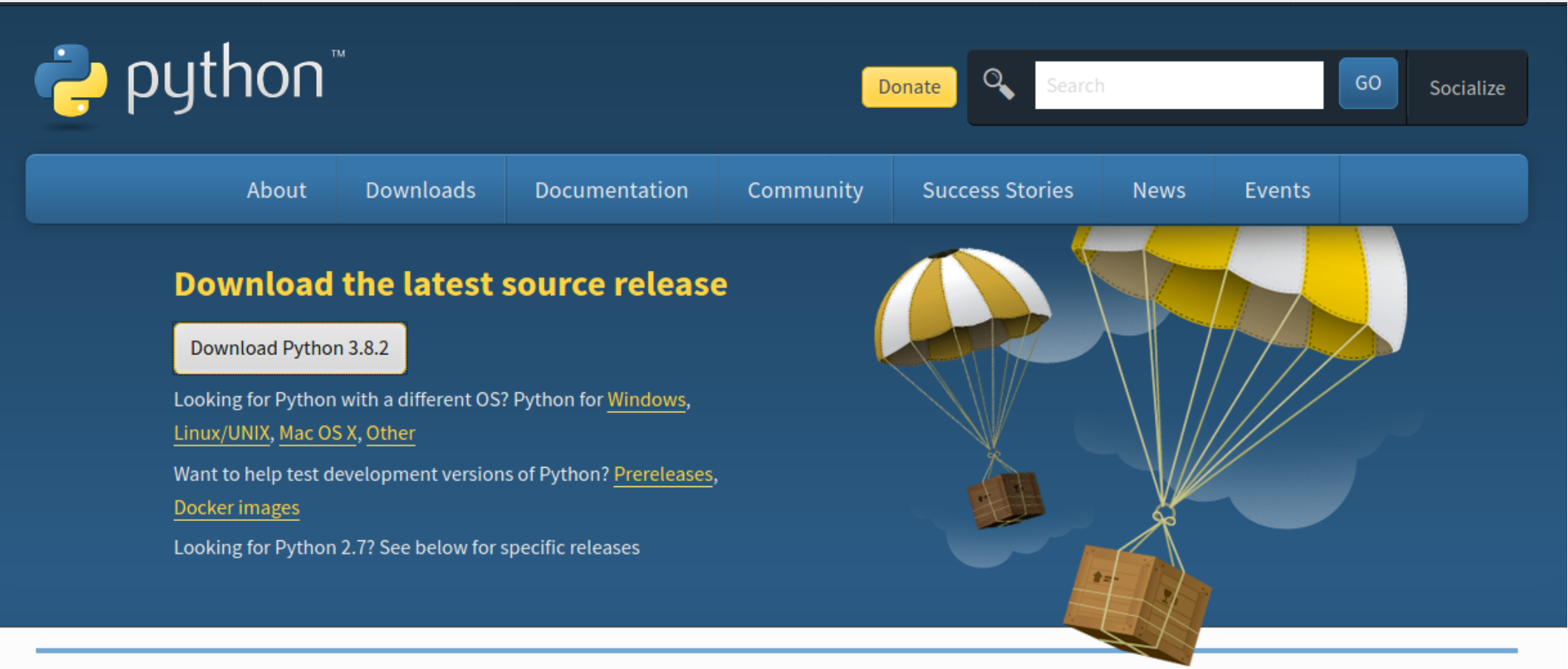
La documentazione



https://www.python.org/doc

INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

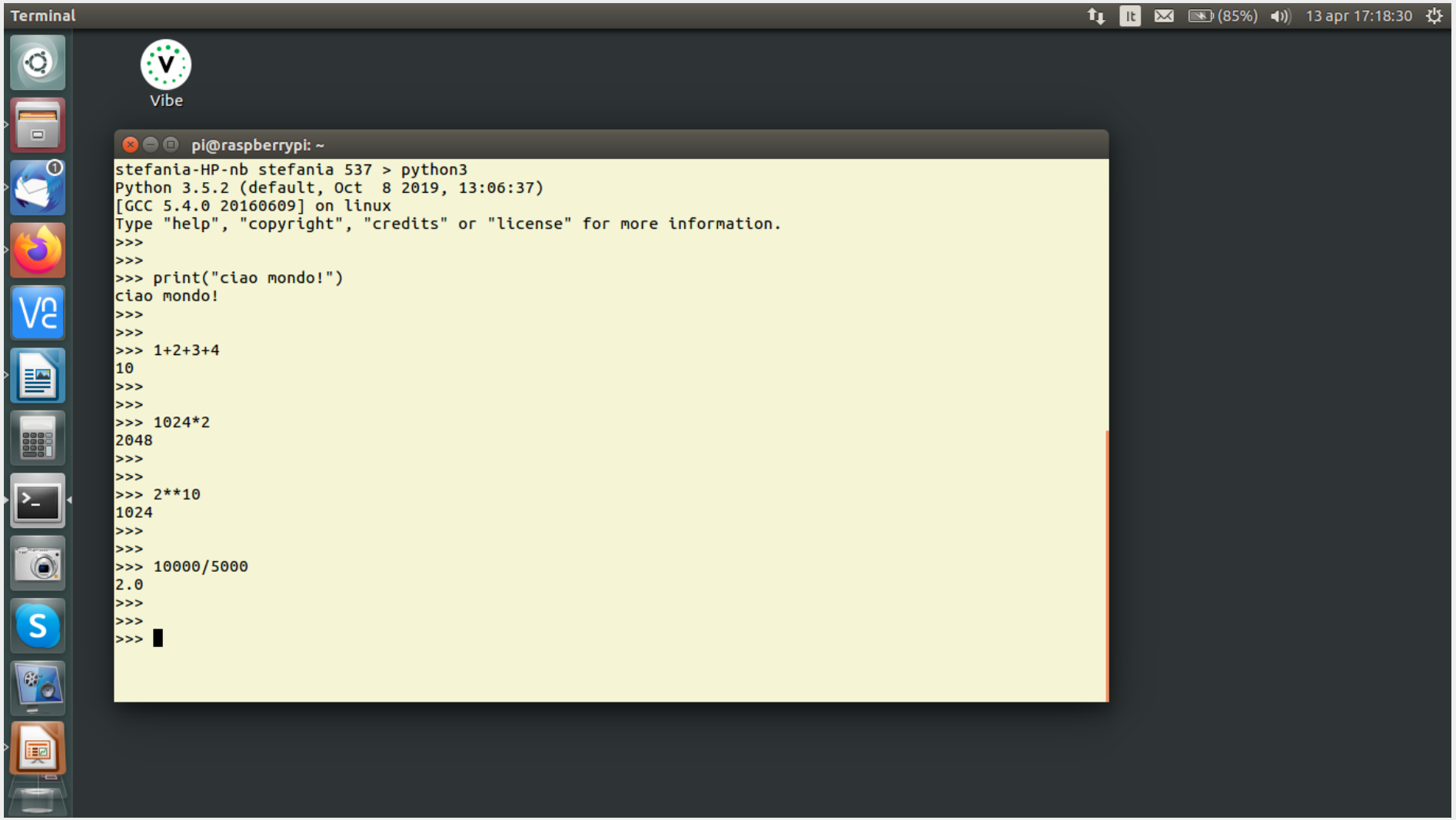
Installazione



https://www.python.org/downloads/

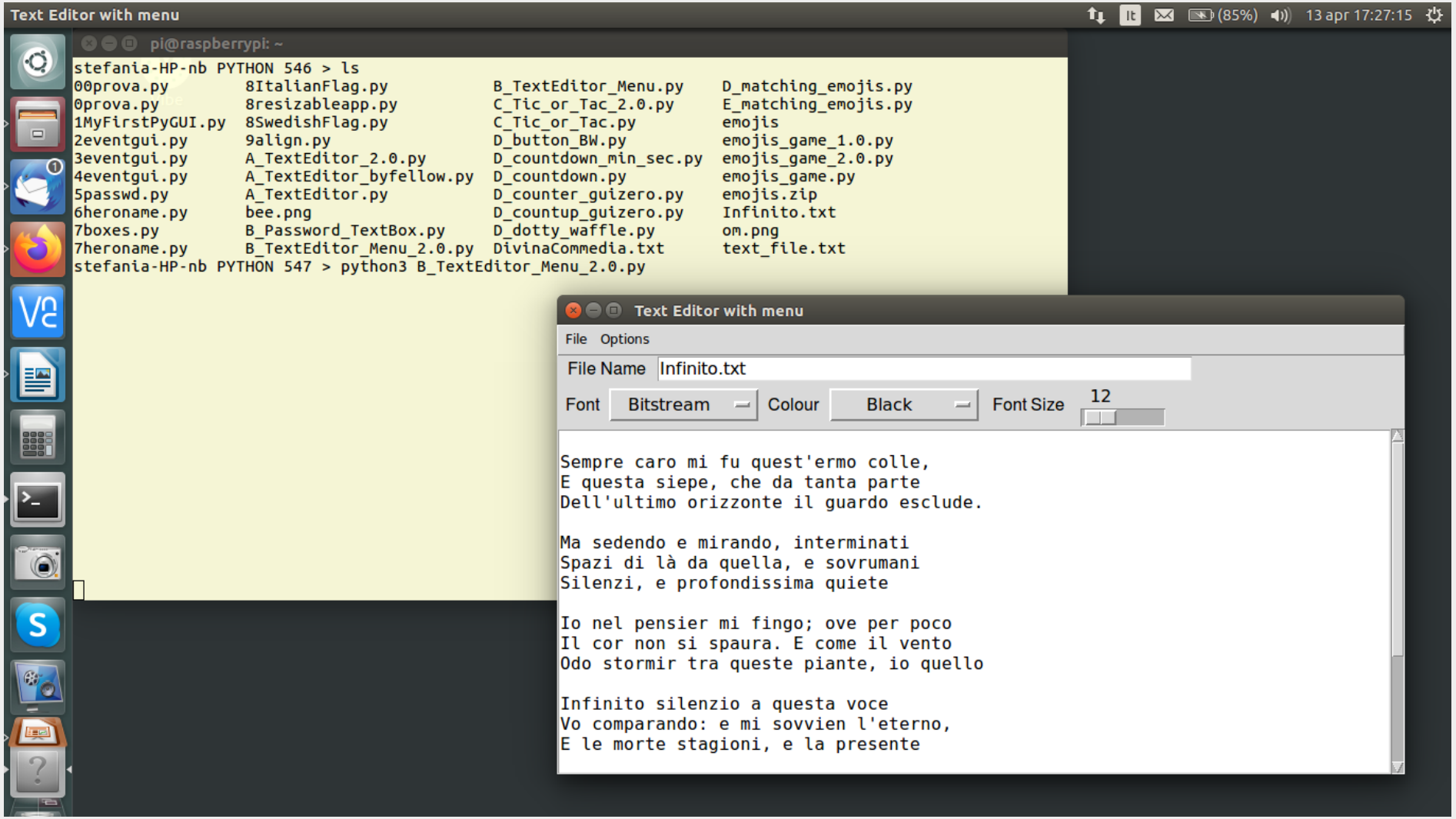
INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Interprete python



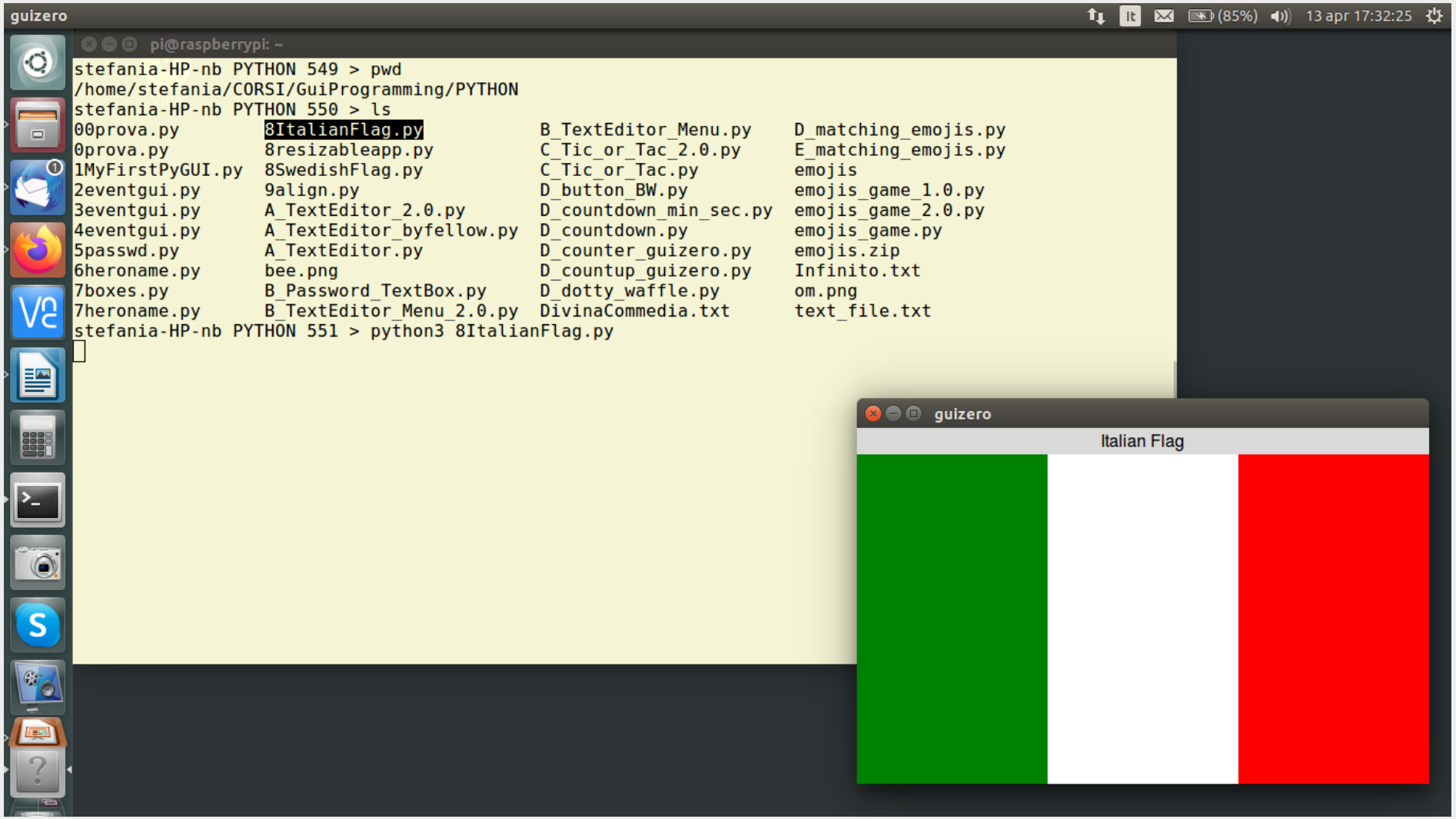
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Interprete python



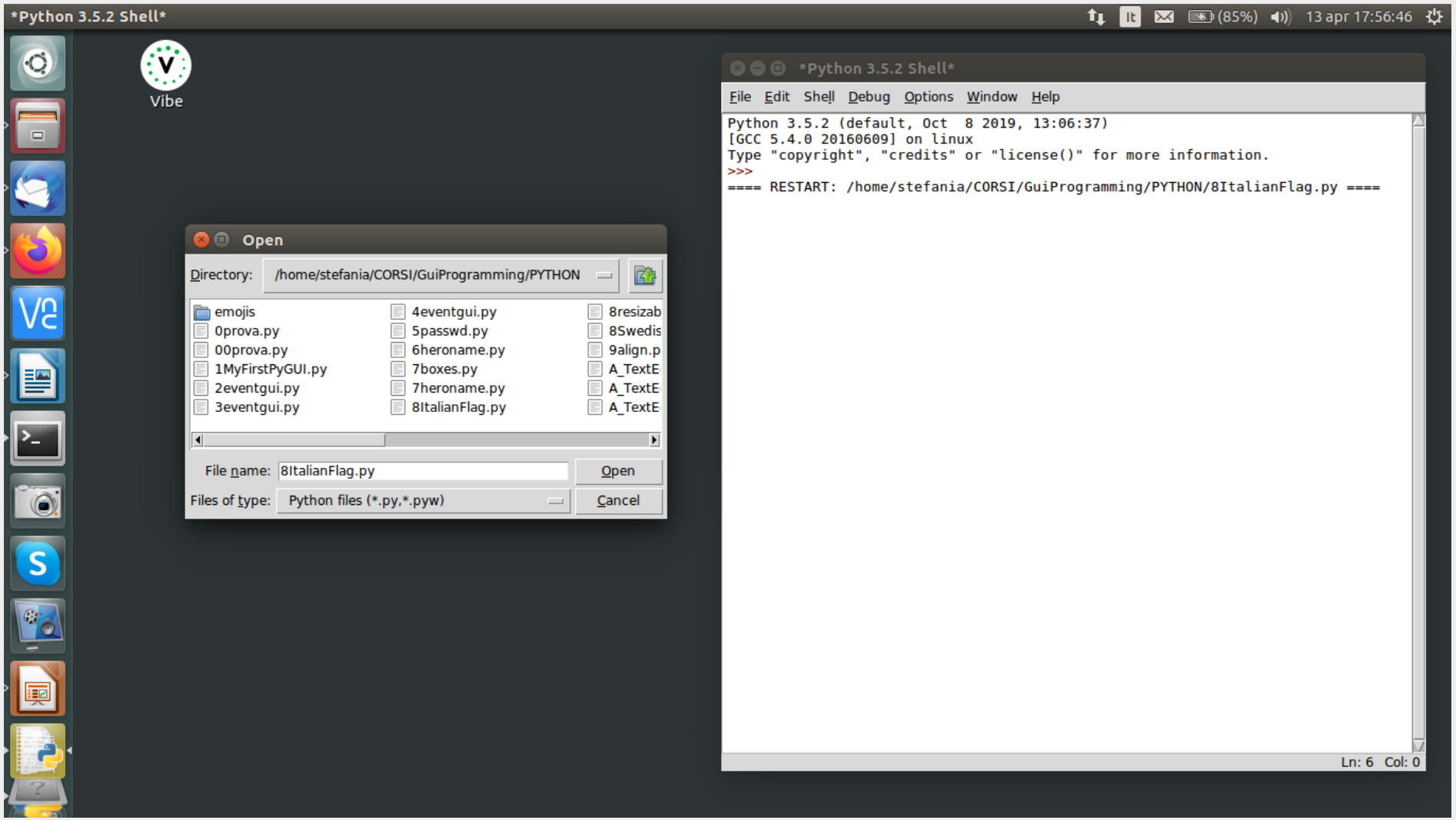
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Esecuzione di script python



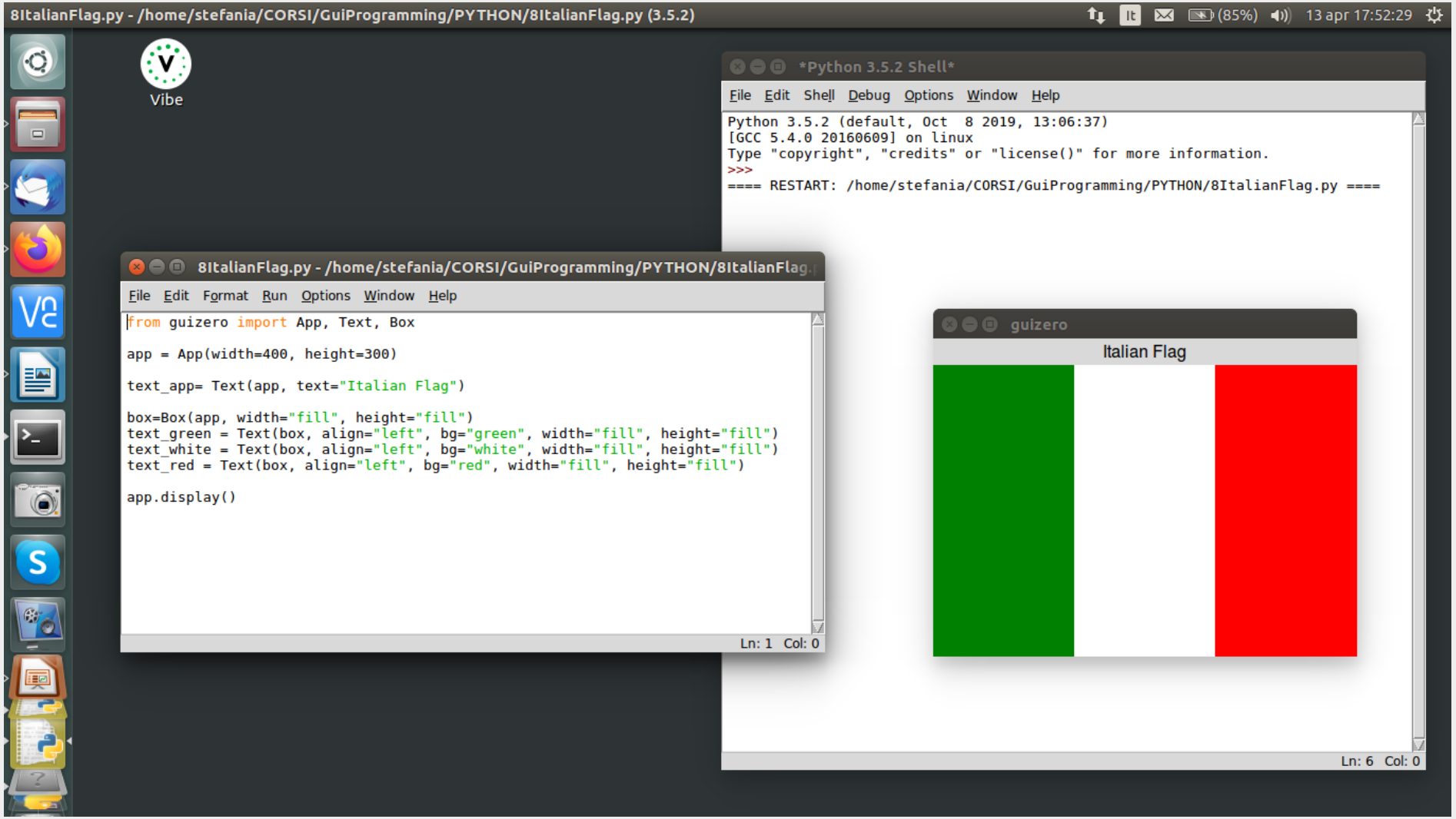
INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Usare IDLE 1



INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Usare IDLE 2



INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Perché ?

Perché imparare a programmare (in python) ?

- è creativo e remunerativo
- può aiutare a risolvere problemi pratici
- ci aiuta a capire come funzionano i dispositivi digitali che usiamo
- è il metodo con cui "addomesticare" computer, tablets e smart-phone

(vedi scheda 1)

INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Nuts and Bolts

Vocabolario e grammatica (minima)

- I costrutti base per i programmi procedurali.
- Le variabili e i valori.
- Input/Output.
- Le strutture di controllo.
- Esempi ed esercizi,

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Le Variabili

Con Python le variabili vengono “semplicemente usate”

```
>>> a = 10
>>> print(a)
10
>>> b = (a * 5)/2
>>> print(b)
25
>>>
```

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Le Variabili

Bisogna evitare di usare le parole chiave del linguaggio (33)

and	as	assert	break	class	continue	def
del	elif	else	except	false	finally	for
from	global	if	import	in	is	lambda
none	nonlocal	not	or	pass	raise	return
true	try	while	with	yield		

Operatori

Operatori aritmetici (e due operatori fra numeri interi)

Operator Name	Symbol	Try this	Evaluates to
Addition	+	5+2	7
Subtraction	-	5-2	3
Multiplication	*	5*2	10
Real division	/	5/2	2.5
Integer division	//	5//2	2
Modulus	%	5%2	1
Power	**	5**2	25

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Espressioni

Python rispetta il consueto ordine delle operazioni

Operation	Symbols	
Brackets	()	**B**
Indices	**	**I**
Division	/ or //	**D**
Multiplication	*	**M**
Addition	+	**A**
Subtraction	-	**S**

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Operatori di relazione

Operatori di relazione (o confronto), per le espressioni logiche

Usando gli operatori di relazione >, <, ==, !=, >=, <=
Si costruiscono espressioni logiche il cui risultato è un valore Booleano, ovvero determinano una condizione True o False

Operator name	Symbol	Try this	Evaluates to
Equality	==	5==2	? F
Less than	<	5<2	? F
Greater than	>	5>2	? T
Less than or equal to	<=	5<=2	? F
Greater than or equal to	>=	2>=2	? T
Not equal to	!=	5!=2	? T

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Le stringhe

Le stringhe di caratteri possono essere assegnate a variabili

Inoltre l'operatore + opera con le stringhe una concatenazione: ovvero unisce insieme le stringhe:

```
>>> a = 4
>>> b = 6
>>> print(a+b)
10
>>> c = '4'
>>> d = '6'
>>> print(c+d)
46
```

INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Stringhe e numeri

Si possono eseguire operazioni mescolando stringhe e numeri

Ad esempio l'operatore * può essere usato anche con le stringhe, moltiplicando il contenuto di una stringa per un numero intero. Per esempio:

```
>>> a = 'Prova '
>>> b = 3
>>> print(a * b)
Prova Prova Prova
```

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Input / Output

Per le operazioni di output : *print* ; per input : *input*

Abbiamo già visto in diversi casi che l'istruzione print produce l'output su terminale o console del suo argomento in modo sensato.
Per l'operazione di input si usa proprio l'istruzione input, che. di default legge stringhe:

```
>>> nome = input('Come ti chiami?\n')
Come ti chiami?
Gioia
>>> print(nome)
Gioia
```

INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Commenti

I commenti sono preceduti dal #

I commenti alle istruzioni possone essere posizionati al posto delle istruzioni (a inizio riga) oppure dopo le istruzioni.
Si possono inserire righe vuote, a piacere.

```
# percentuale sull'ora
percentage = (minute * 100) / 60
```

Oppure:

```
percentage = (minute * 100) / 60 # percentuale sull'ora
```

Input numerico

Per le operazioni input numerico occorre la “conversione”

L’istruzione input legge stringhe, da convertire in numeri, al bisogno:

```
>>> prompt = 'Quanti giorni ha il mese di giugno?\n'
>>> giorni = input(prompt)
Quanti giorni ha il mese di giugno??
30
>>> int(giorni)
30
>>> int(giorni) + 1
31
```

Controllo semplice: if

Istruzione condizionale IF-THEN

La forma sintattica della istruzione di controllo IF-THEN è illustrata dal seguente esempio:

```
if (x > 0) :
    print('x è un numero positivo')
```

Si noti che “then” è tradotto con : e si noti la tabulazione che identifica il blocco.

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Controllo if-then-else

Istruzione condizionale IF-THEN-ELSE

L'istruzione di controllo IF-THEN-ELSE è illustrata da:

```
if (x > 0) :
    print('x è un numero positivo')
else:
    print('x è un numero negativo o nullo')
```

Si noti che i : sono presenti anche dopo else

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

CICLI (while e for)

Le istruzioni per i cicli (loop) con while e for

```
# WHILE
n=10
while (n > 0):
    print(n)
    n = n - 1

print('ZERO!')
```

```
# FOR
n=10
for i in range(n):
    print((n-i))

print('ZERO!')
```

INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

ESEMPI

- Input/Output + If-then-else
- Reaction Game + Loop + Statistics
- Reaction Game + Until
- Calcolo dei fogli da ORIGAMI
- Indovina il numero, con suggerimenti

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

I mattoni

I mattoni con cui costruire i programmi (procedurali) sono:

INPUT : Get data from the "outside world"

OUTPUT : Display the results of the program

SEQUENTIAL exec : Perform statements one after another.

CONDITIONAL exec : check for conditions and then execute or not. (IF THEN ELSE)

REPEAT : Perform a set of statements repeatedly (LOOPS)

REUSE: Write a set of instructions once and give them a name and then reuse (FUNCTION)

Gestione degli errori

In Python la gestione degli errori usa il costrutto **try-except**

You can think of the `try` and `except` feature in Python as an "insurance policy" on a sequence of statements.
Example:

```
Fahrenheit_Temperature = input('Enter Fahrenheit Temperature:')
try:
    fahrenheit = float(Fahrenheit_Temperature)
    celsius = (fahrenheit - 32.0) * 5.0 / 9.0
    print(celsius)
except:
    print('Please enter a number for Fahrenheit Temperature')
```

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Funzioni predefinite

In Python esistono diverse funzioni predefinite (built-in)

type : funzione che ritorna il “tipo” di argomento passato
input : funzione per leggere una stringa (riga) da tastiera
print : funzione che scrive un dato (un risultato) sul video
max e min : ritornano rispettivamente massimo e minimo tra i valori passati
len : ritorna la lunghezza di una stringa (numero di caratteri)
Int: converte l’argomento in numero intero
float: converte l’argomento in numero reale
str: converte l’argomento in stringa

Anche i nomi delle funzioni built-in sono riservati ed è necessario evitare di usarli come nomi di variabili.

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Funzioni di libreria

In Python esistono diverse librerie di funzioni (math, time ...)

Per usare le funzioni di una libreria occorre "importare" la libreria
oppure le singole funzioni di libreria:
Esempio per usare sleep da time

```
import time # poi si usa time.sleep(60)
from time import sleep # poi si usa sleep(60)
```

Esempio per usare sqrt da math

```
import math # poi si usa math.sqrt(2.0)
from math import sqrt # poi si usa sqrt(2.0)
```

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Definizione di Funzioni

In Python le funzioni sono definite con **def** nome_funzione :

Esempio di funzione che somma due argomenti

```
def somma_di_2val( val1, val2) :
    print(val1+val2)
    return (val1+val2)
```

(attenzione all'indentazione che definisce il corpo della funzione)

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Parametri e argomenti

Parametri e argomenti sono termini usati per le funzioni

Gli argomenti sono le variabili che vengono passate alle funzioni

math.sqrt(x) è una funzione con 1 argomento e ritorna la radice quadrata di x

math.pow(x,y) è una funzione con 2 argomenti e ritorna x^y

Dal punto di vista delle funzioni, gli argomenti sono anche indicati come parametri delle funzioni. Il passaggio dei parametri avviene "per valore" - le funzioni NON modificano il valore degli argomenti, ma solo il risultato che eventualmente ritornano

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Funzioni void e non

Si possono definire funzioni che non ritornano valore (void)

Le funzioni che non ritornano alcun valore (ad esempio funzioni di puro output) sono le funzioni VOID.

Invece le funzioni che eseguono delle operazioni e ritornano un risultato (NON void o fruttuose – fruitful) hanno un tipo: intero, oppure float, oppure bool, oppure stringa.

Normalmente il valore di ritorno di una funzione "fruttuosa" viene assegnato ad una variabile, altrimenti va perso e l'esecuzione della funzione è inutile.

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

break e continue

break e/o continue sono direttive per modificare i cicli (loop)

Esempio di modifica di un ciclo infinito con un break (per terminare il ciclo)

```
while True:
    line = input('> ')
    if line == 'done': break
    print(line)
print('DONE!')
```

Esempio di uso del continue (per terminare il solo ciclo corrente), il ciclo NON termina mai

```
while True:
    line = input('> ')
    if line[0] == '#': continue # non stampa le righe di solo commento
    print(line)
```