

Una *introduzione* a Python

Simone Silvetti, DEAMS, Università di Trieste, aa. 2019-2020



Informazioni Utili

Email : simone.silvetti@gmail.com (<http://simonesilvetti.com/>)

Website del corso: <https://github.com/simonesilvetti/teaching>

Siti utili: <https://www.python.it/>



Cos'è Python?

«Python è un linguaggio di programmazione ad alto livello, orientato agli oggetti, adatto, tra gli altri usi, a sviluppare applicazioni distribuite, scripting, computazione numerica e system testing» (wikipedia)

«È un linguaggio multi-paradigma che ha tra i principali obiettivi: dinamicità, semplicità e flessibilità. Supporta il paradigma object oriented, la programmazione strutturata e molte caratteristiche di programmazione funzionale ...» (wikipedia)

«Python è un linguaggio di programmazione interpretato, interattivo, orientato agli oggetti. Include moduli, eccezioni, tipizzazione dinamica...» (python.it)

Cos'è Python?



Linguaggio di programmazione ad alto livello

Multi Paradigma

- Strutturata (imperativo)
- A oggetti
- Funzionale

Interpretato



Compilato

Tipizzazione
Dinamica



Tipizzazione
Statica

Linguaggio di programmazione

Linguaggio formale: un linguaggio che ha delle regole precise e non è ambiguo adatto a descrivere algoritmi (procedure che prendono dati in input e producono dati in output)

Questo permette al PC di interpretare un programma (insieme di frasi appartenenti al linguaggio) mediante l'analisi dei simboli e della struttura.

```

001001111011110111111111111100000
1010111110111111100000000000010100
101011111010010000000000000100000
101011111010010100000000000100100
10101111101000000000000000011000
10101111101000000000000000011100
10001111101011100000000000011100
10001111101110000000000000011000
00000001110011100000000000011001
00100101110010000000000000000001
00101001000000010000000001100101
10101111101010000000000000011100
00000000000000000111100000010010
00000011000011111100100000100001
0001010000100000111111111110111
10101111101110010000000000011000
00111100000001000001000000000000
10001111101001010000000000011000
000011000001000000000000011101100
00100100100001000000010000110000
10001111101111110000000000010100
00100111101111010000000000100000
0000001111100000000000000001000
0000000000000000000001000000100001

```

Codice Macchina a 32 bit

```

        .text
        .align 2
        .globl main
main:
    subu    $sp, $sp, 32
    sw      $ra, 20($sp)
    sd      $a0, 32($sp)
    sw      $0, 24($sp)
    sw      $0, 28($sp)
loop:
    lw      $t6, 28($sp)
    mul     $t7, $t6, $t6
    lw      $t8, 24($sp)
    addu    $t9, $t8, $t7
    sw      $t9, 24($sp)
    addu    $t0, $t6, 1
    sw      $t0, 28($sp)
    ble     $t0, 100, loop
    la      $a0, str
    lw      $a1, 24($sp)
    jal     printf
    move    $v0, $0
    lw      $ra, 20($sp)
    addu    $sp, $sp, 32
    jr      $ra

        .data
        .align 0
str:
    .ascii  "The sum from 0 .. 100 is %d\n"

```

Codice Assembly

```

a = 5
b = 10
c= a+b
print("la somma è "+str(c))

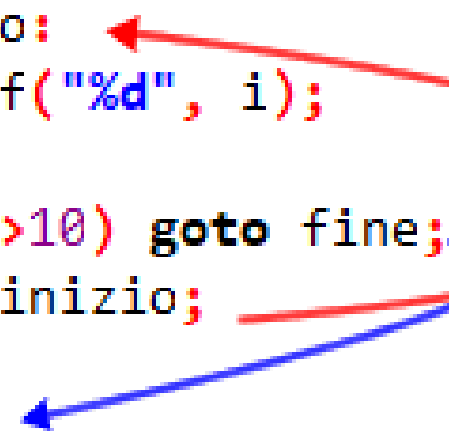
```

Python

Paradigmi di programmazione

PROGRAMMAZIONE NON STRUTTURATA

```
1  #include <stdio.h>
2  int main() {
3      int i=0;
4
5      inizio:
6      printf("%d", i);
7      i++;
8      if (i>10) goto fine;
9      goto inizio;
10
11     fine:
12     printf("fine");
13 }
```



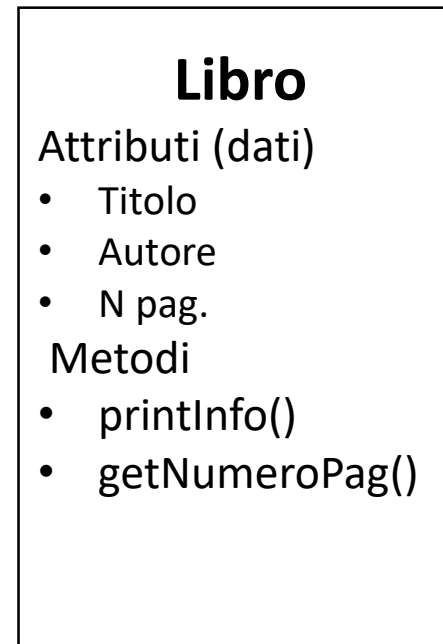
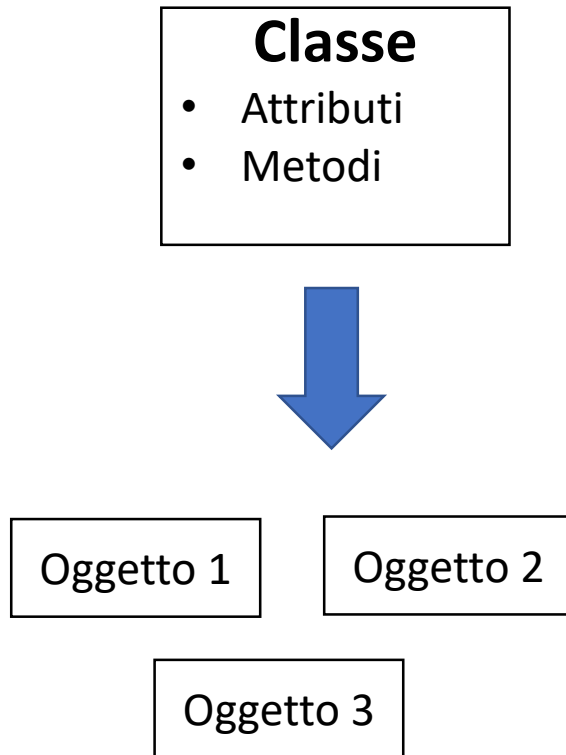
PROGRAMMAZIONE STRUTTURATA

```
1  #include <stdio.h>
2  int main() {
3      int i=0;
4
5      while(i<=10) {
6          printf("%d", i);
7          i++;
8      }
9
10     printf("fine");
11
12 }
13
```

WWW.OKPEDIA.IT

«Spaghetti Programming» o «goto programming»

Linguaggio orientato ad oggetti



```
(23234).printInfo()
```

```
>>> Alessandro Manzoni, «I Promessi Sposi», numero di pagine 500.
```

```
(23234).getNumeroPag()
```

```
>>> 500
```

Linguaggio funzionale

Paradigma di programmazione, dove il flusso di esecuzione assomiglia a una serie di valutazioni di funzioni matematiche



- Output (Y) è determinato esclusivamente dall' input (X)

Interpretato



Compilato

Linguaggio compilato

```
#include <stdio.h>

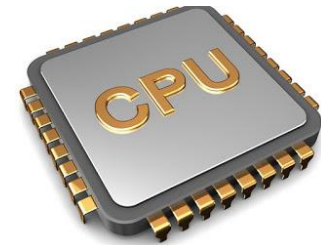
int main(void){
    printf("hello, world\n");
}
```



Compilatore

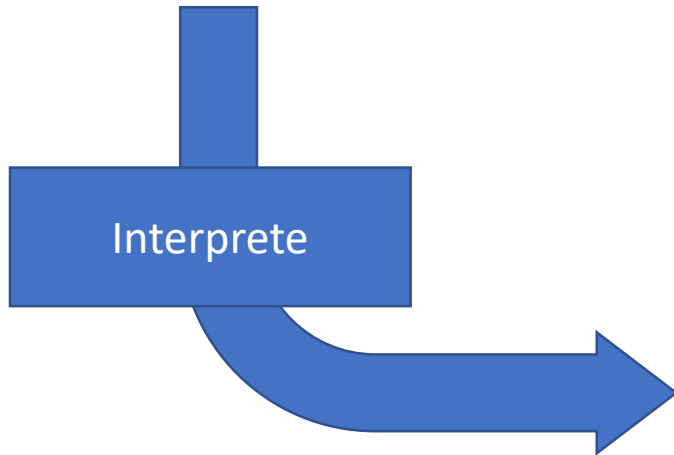


```
00100111101111011111111111110000
101011111011111110000000000010100
1010111110100100000000000100000
10101111101001010000000000100100
1010111110100000000000000011000
1010111110100000000000000011100
1000111110101110000000000011100
10001111101110000000000011000
0000000111001110000000000011001
001001011100100000000000000001
0010100100000010000000001100101
1010111110101000000000000011100
0000000000000000111100000100010
00000011000011111100100000100001
0001010000100000111111111110111
1010111110111001000000000011000
0011110000000100000100000000000
1000111110100101000000000011000
0000110000010000000000011101100
00100100100001000000010000110000
100011111011111000000000010100
0010011110111101000000000100000
00000011111000000000000001000
00000000000000000001000000100001
```



Linguaggio Interpretato

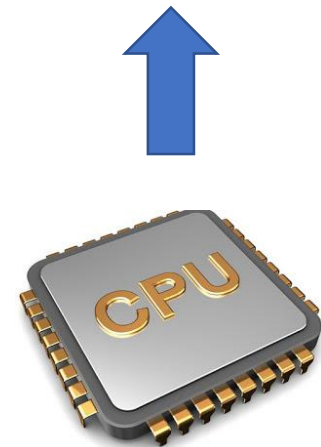
```
>>> print("Hello world!")
```



```
001001111011110111111111111100000
101011111011111110000000000010100
10101111101001000000000000100000
10101111101001010000000000100100
1010111110100000000000000011000
1010111110100000000000000011100
1000111110101110000000000011100
10001111101110000000000011000
0000000111001110000000000011001
001001011100100000000000000001
0010100100000010000000001100101
1010111110101000000000000011100
0000000000000000111100000100010
00000011000011111100100000100001
0001010000100000111111111110111
1010111110111001000000000011000
0011110000000100000100000000000
1000111110100101000000000011000
00001100000100000000000011101100
00100100100001000000010000110000
1000111110111111000000000010100
0010011110111101000000000100000
0000001111100000000000000011000
000000000000000000001000000100001
```



```
>>> print("Hello world!")
Hello world!
```



Benevolent dictator for «life»

Guido Van Rossum

- Olandese
- Crea Python nel 1989



Perché creare Python?



Per avere un linguaggio

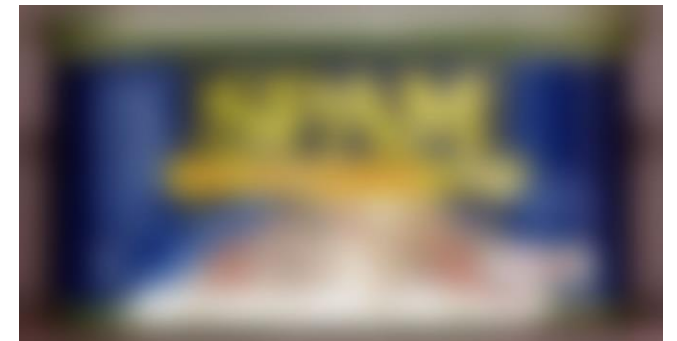
- semplice, intuitivo e potente
- open source, aperto allo sviluppo condiviso
- facilmente comprensibile, come l'inglese parlato
- ottimo per risolvere problemi quotidiani (degli sviluppatori)

Voleva essere un linguaggio di rottura con il passato,
lievemente provocatorio

Perché chiamarlo Python?

Guido V.R prende ispirazione da un gruppo comico inglese i Monty Python

«Il programma televisivo fu una rivoluzione — i Monty Python hanno rappresentato quello che i Beatles sono stati per la musica: un punto di partenza e di non ritorno. Il loro umorismo — anarchico, a tratti demenziale, sempre intellettuale, mai volgare, corrosivo, capace di toccare le vette rarefatte dell'assurdo — sfidò tutte le convenzioni comiche dell'epoca, sia in termini di stile che di contenuti» Corriere della Sera, 22/01/20



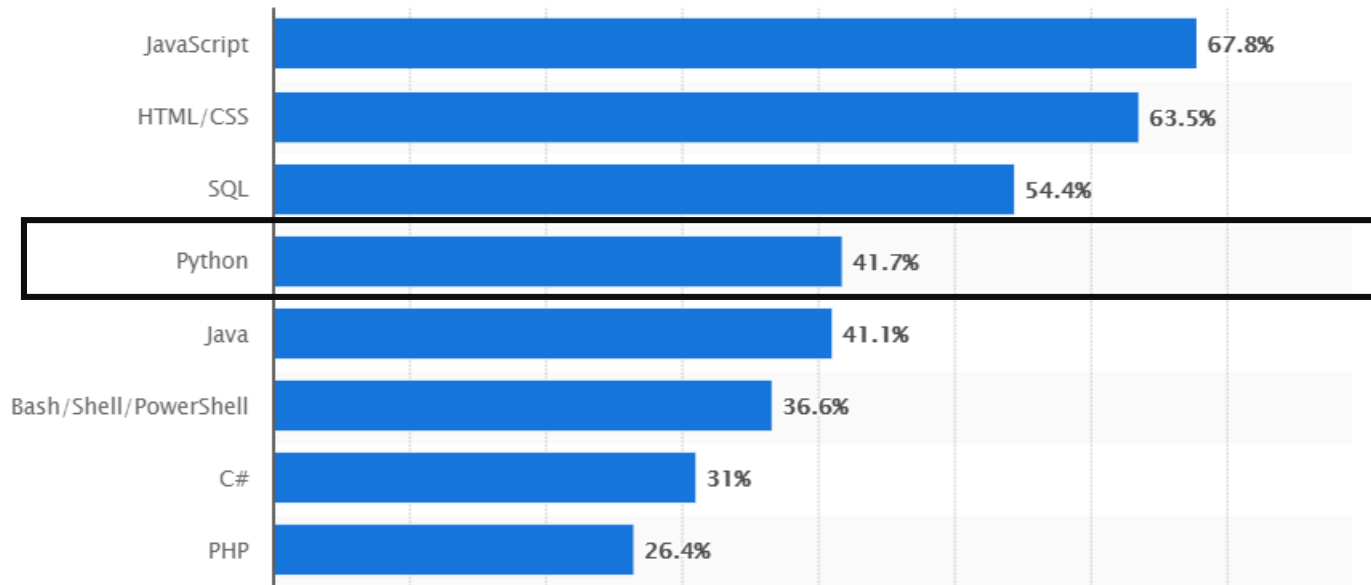
Perché chiamarlo Python?

Guido V.R prende ispirazione da un gruppo comico inglese i Monty Python

«Il programma televisivo fu una rivoluzione — i Monty Python hanno rappresentato quello che i Beatles sono stati per la musica: un punto di partenza e di non ritorno. Il loro umorismo — anarchico, a tratti demenziale, sempre intellettuale, mai volgare, corrosivo, capace di toccare le vette rarefatte dell'assurdo — sfidò tutte le convenzioni comiche dell'epoca, sia in termini di stile che di contenuti» Corriere della Sera, 20/01/22



Perché Python in università?



Grande comunità di sviluppatori
Buona curva di apprendimento
Richiesto dalle aziende
Usato in diversi campi

**Most used programming languages among developers worldwide,
as of early 2019**

Ecosistema Python



Python

Ecosistema Python



Python

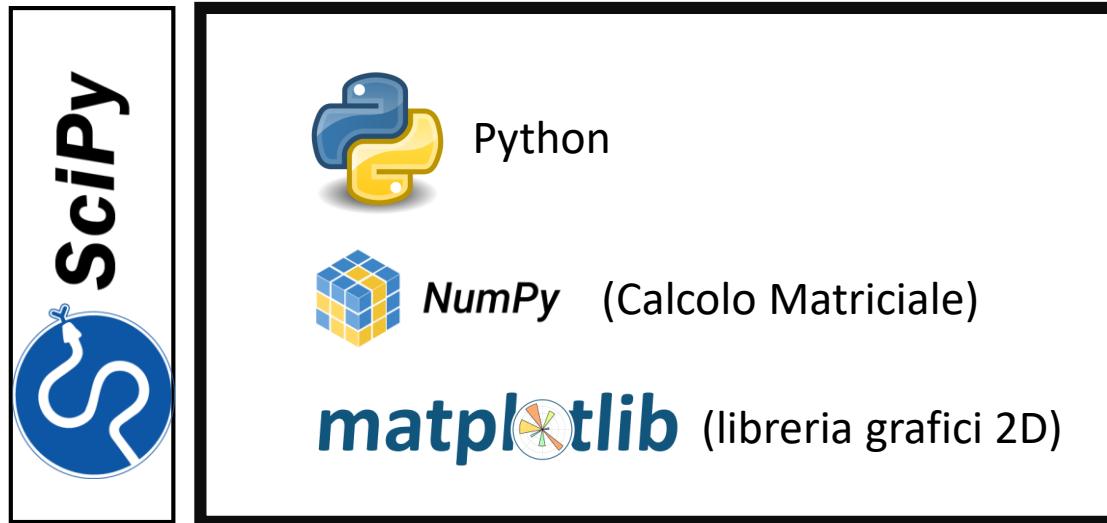


NumPy (Calcolo Matriciale)

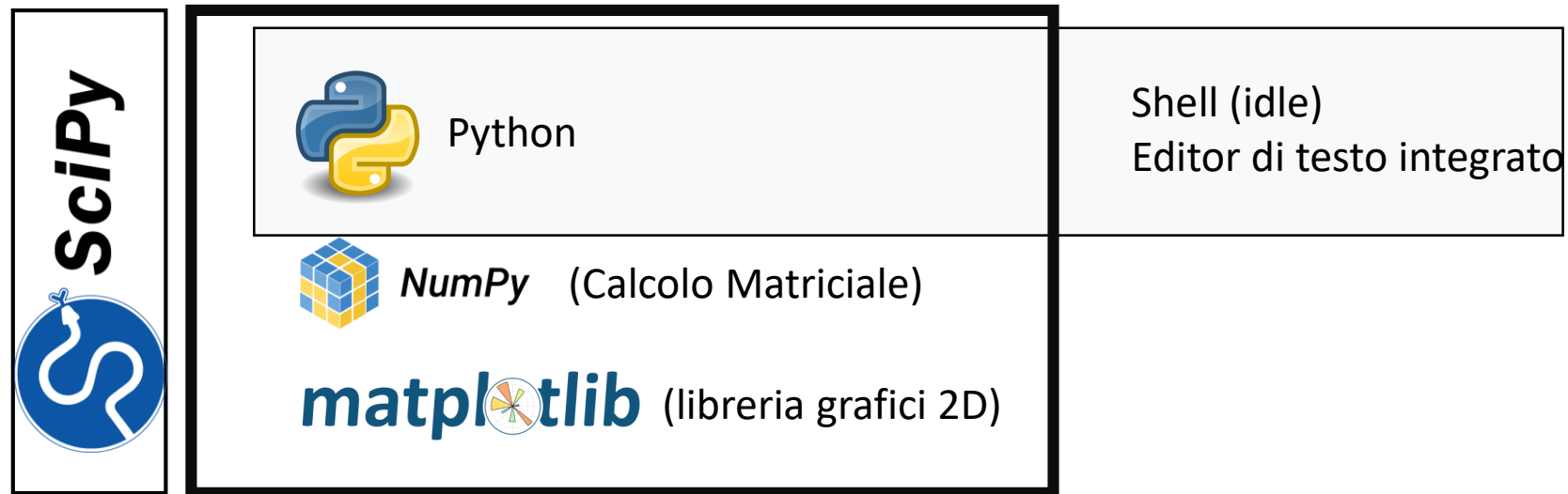


(libreria grafici 2D)

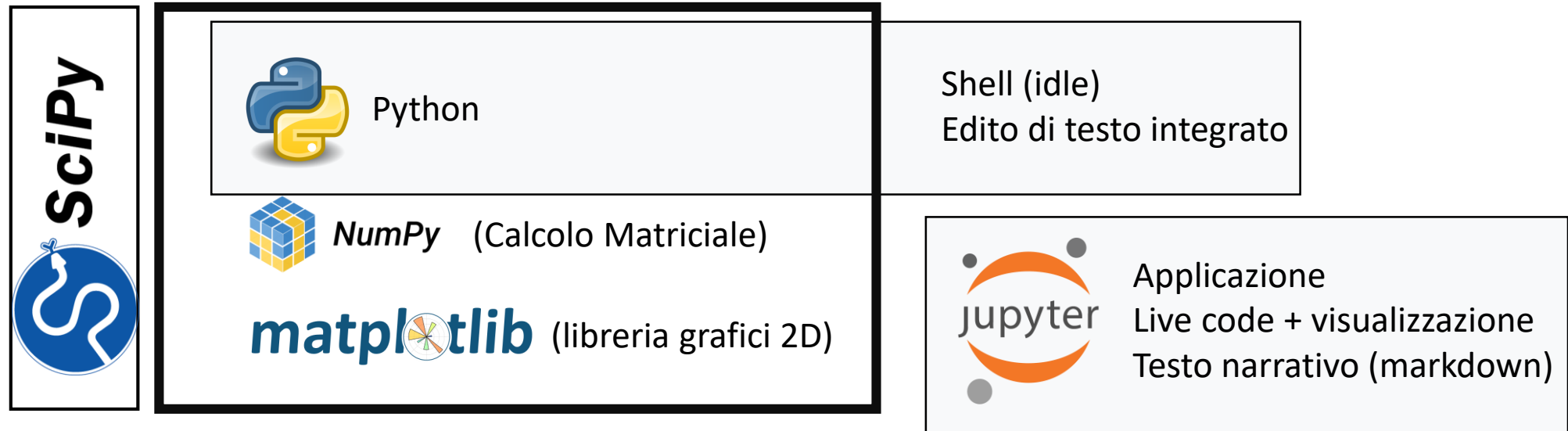
Ecosistema Python



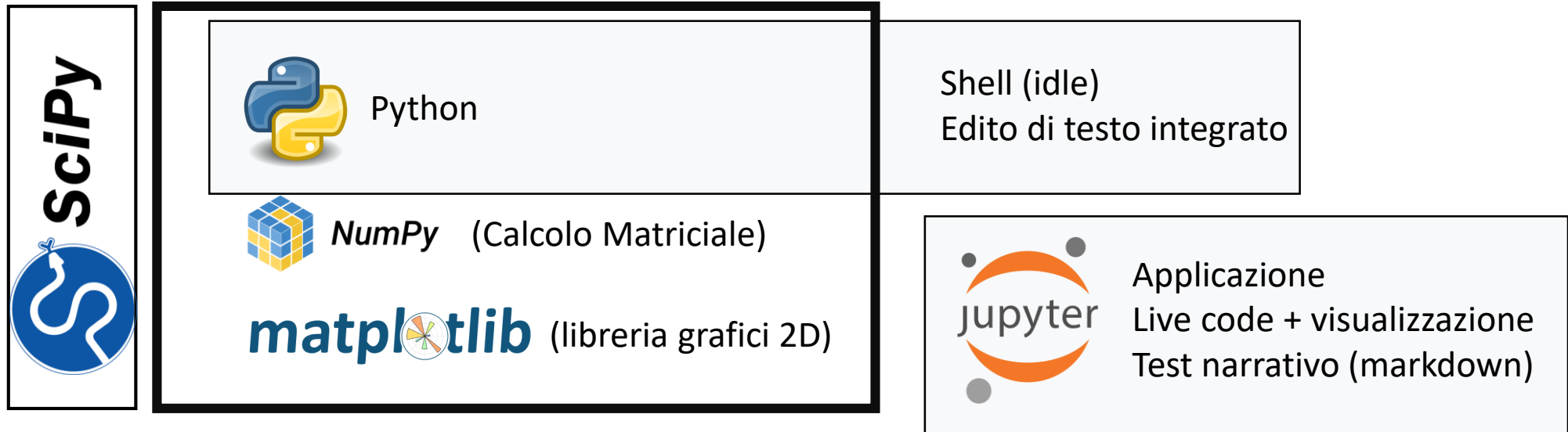
Ecosistema Python



Ecosistema Python



Ecosistema Python





<https://www.anaconda.com/distribution/>



Windows



macOS



Linux

Anaconda 2019.10 for Windows Installer

Python 3.7 version

Download

64-Bit Graphical Installer (462 MB)

32-Bit Graphical Installer (410 MB)

Python 2.7 version

Download

64-Bit Graphical Installer (413 MB)

32-Bit Graphical Installer (356 MB)



<https://www.anaconda.com/distribution/>



Windows



macOS



Linux

Anaconda 2019.10 for Windows Installer

Python 3.7 version

Download

64-Bit Graphical Installer (462 MB)

32-Bit Graphical Installer (410 MB)

Python 2.7 version

Download

64-Bit Graphical Installer (413 MB)

32-Bit Graphical Installer (356 MB)

Programma del Corso

Prima settimana +



Comandi tipi di base
Istruzioni condizionali
Iterazioni
Definiz. Funzione
Lambda expression
Scrittura/lettura file



Pillole Markdown

Seconda settimana



NumPy

matplotlib

