

# Tutorial del laboratorio

## Informatica@DSS 2023/2024

Massimo Lauria <massimo.lauria@uniroma1.it>  
<https://massimolauria.net/informatica2023/>

# I software che useremo

- ▶ Ambiente di programmazione Python (Thonny)
- ▶ Terminale (per eseguire i programmi)
- ▶ Browser web (questionari e documentazione)

# I software che useremo

- ▶ Ambiente di programmazione Python (Thonny)
- ▶ Terminale (per eseguire i programmi)
- ▶ Browser web (questionari e documentazione)

Troviamoli sui PC del laboratorio!

# Terminale

Interfaccia a caratteri, che permette di eseguire comandi. I comandi sono **scritti** e la risposta a questi comandi viene fornita in maniera **scritta**

- ▶ Più "antica" rispetto alle interfacce moderne
- ▶ Molto più espressiva

Permette di

- ▶ esplorare il **filesystem** (i file nel computer)
- ▶ eseguire e combinare comandi
- ▶ eseguire `python3`

# Ambiente di programmazione

Programma per scrivere file contenenti "programmi"

- ▶ testi **senza** formattazione **con** struttura logica
- ▶ strumenti di supporto alla **scrittura di software**
- ▶ strumenti per testare la correttezza

Thonny (<https://thonny.org>) è molto adatto per iniziare a programmare in Python.

# Browser web

Questo lo conoscete sicuramente

- ▶ Firefox
- ▶ Internet explorer (Edge)
- ▶ Google Chrome
- ▶ Safari

# Prompt e Linea di comando

# Prompt e linea di comando

Quando lanciate il terminale vi trovate davanti all'interprete dei comandi per lavorare su file.

Su Mac e Linux

```
blabla@bla: ~/$
```

Su Windows

```
C:\Users>
```



# File e cartelle

I comandi `ls` (Mac/Linux) e `dir` (Windows) vi dicono che file ci sono nella **cartella corrente**.

Ci si può muovere tra le cartelle con il comando `cd` (Change Directory).

```
massimo@lauria:~$  
  
massimo@lauria:~$ ls  
Applications      Dropbox            Pictures           personal  
Desktop            Library            Public  
Documents          Movies             config  
Downloads          Music              lavori  
  
massimo@lauria:~$ cd Documents  
massimo@lauria:~/Documents$
```

# Piccolo tutorial sulla linea di comando

Un piccolo tutorial su come fare le operazioni di base e come muoversi tra cartelle.

## **Django Girls Tutorial (italiano)**

- ▶ muoversi tra le cartelle
- ▶ copiare, muovere, rinominare, cancellare file

# File e cartelle (Linux e Mac)

Il file, documenti e programmi nel computer sono organizzati in maniera gerarchica detta **filesystem**.

- ▶ Esiste una cartella “root” indicata con uno slash /
- ▶ La cartella root contiene sotto cartelle e file.
- ▶ Ogni sottocartella può contenere sottocartelle e file.

**Percorso assoluto**: posizione nel filesystem

Es. `python3` è un file in `bin` che è in `usr` che è in /

```
/usr/bin/python3
```

# Posizione corrente "dell'utente"

Quando si lavora usando il terminale si è sempre **posizionati** in una cartella del filesystem. All'inizio si è posizionati nella propria **cartella utente** o **home directory**.

L'utente può **muoversi** nella gerarchia con il comando `cd`

# Due cartelle speciali

Cartella HOME: è la cartella predefinita dell'utente, nel nostro caso

```
/home/studente/
```

La cartella che contiene i file posizionati sul desktop

```
/home/studente/Scrivania
```

# Alcuni comandi del terminale

- ▶ `pwd` indica in quale cartella si è posizionati
- ▶ `ls` elenca il contenuto della cartella corrente
- ▶ `cd <nome>` va alla cartella `<nome>`

Il terminale ha migliaia di altri comandi

- ▶ `mkdir <nome>` crea la cartella `<dir>`
- ▶ `rmdir <nome>` elimina la cartella `<dir>`
- ▶ `mv <old_pos> <new_pos>` muove o rinomina file
- ▶ `man <cmd>` istruzioni su come usare il comando
- ▶ ...

# Esempi (1)

```
$ pwd  
/home/massimo/lavori/informatica2023/lectures
```

```
$ ls  
Makefile  
lecture01-print.pdf  
lecture01-slides.pdf  
lecture01.org  
lecture01b.org  
lecture01b.pdf  
...
```

## Esempi (2)

Cambiare cartella di lavoro usando il **percorso assoluto**

```
$ pwd
/home/massimo/lavori/informatica2023/lectures

$ cd /home/massimo/Documents

$ pwd
/home/massimo/Documents
```



# Esempi (3)

## Salire alla cartella genitore

```
$ pwd  
/home/massimo/lavori/informatica2023/lectures
```

```
$ cd ..
```

```
$ pwd  
/home/massimo/lavori/informatica2023
```

```
$ cd ..
```

```
$ pwd  
/home/massimo/lavori/didattica
```

# Esempi (4)

## Andare in una sottocartella

```
$ cd /home/massimo/
```

```
$ pwd  
/home/massimo
```

```
$ cd lavori
```

```
$ pwd  
/home/massimo/lavori
```

```
$ cd didattica
```

```
$ pwd  
/home/massimo/lavori/didattica
```

# Esempi (5)

## Concatenare i movimenti **relativi**

```
$ pwd
/home/massimo/lavori/informatica2023/lectures

$ cd ../../2022.IntroInformatica/lectures/../../sito/

$ pwd
/home/massimo/lavori/informatica2022/sito
```

## Esempi (6)

cd senza argomenti riporta alla cartella home. Sui PC del laboratorio è /home/studente/.

```
$ pwd
/home/massimo/lavori/informatica2023/lectures
$ cd
$ pwd
/home/massimo
```

Il simbolo ~ è un alias per la cartella utente

```
$ pwd
/home/massimo/lavori/didattica/2018.IntroInformatica/lectures
$ cd ~
$ pwd
/home/massimo
$ cd ~/lavori
$ pwd
/home/massimo/lavori
```

# Nomi di file e cartelle: maiuscole e minuscole

Su Linux e Mac un file o una cartella di nome "Pippo" è considerata different da una di nome "pippo".

Provate a posizionarvi nella cartella `/home/studente/` con il comando

```
cd
```

Se il comando successivo è

```
cd scrivania
```

oppure

```
cd Scrivania
```

c'è differenza?

# Differenze tra Linux/Mac e Windows

Il terminale di Windows funziona diversamente.

- ▶ il comando `ls` non esiste ma esiste il comando `dir`
- ▶ le cartelle sono separate da `\` invece che `/`
- ▶ ci sta una cartella `root` per ogni disco
- ▶ la cartella `root` del disco principale è `C:\`
- ▶ la cartella `home` è ad esempio `C:\Users\massimo\`

# Prompt, terminale e Python

## Interfaccia testuale di Mac/Linux

```
blabla@bla: ~/$
```

## Interfaccia testuale di Windows

```
C:\Users>
```

## Python3

```
>>>
```

# Eseguire python

Per eseguire un programma

```
blabla@bla:~/ $ python3 nomeprogramma.py
```

Per lanciare l'interprete interattivo

```
blabla@bla:~/ $ python3
```



# Interfaccia testuale S.O. vs Python

```
lauria@macbook15: ~/$ cd Documenti
lauria@macbook15: ~/Documenti$ cd ..
lauria@macbook15: ~/$
```

# Interfaccia testuale S.O. vs Python

```
lauria@macbook15: ~/$ cd Documenti
```

```
lauria@macbook15: ~/Documenti$ cd ..
```

```
lauria@macbook15: ~/$ python3
```

```
>>>
```

# Interfaccia testuale S.O. vs Python

```
lauria@macbook15: ~/$ cd Documenti  
lauria@macbook15: ~/Documenti$ cd ..  
lauria@macbook15: ~/$ python3
```

```
>>> print(5+ 0.2)  
5.2  
  
>>> exit() # oppure premo Ctrl-D
```

# Interfaccia testuale S.O. vs Python

```
lauria@macbook15: ~/$ cd Documenti
```

```
lauria@macbook15: ~/Documenti$ cd ..
```

```
lauria@macbook15: ~/$ python3
```

```
>>> print(5+ 0.2)  
5.2
```

```
>>> exit() # oppure premo Ctrl-D
```

```
lauria@macbook15: ~/$
```

# Interfaccia testuale S.O. vs Python

```
lauria@macbook15: ~/$ cd Documenti
```

```
lauria@macbook15: ~/Documenti$ cd ..
```

```
lauria@macbook15: ~/$ python3
```

```
>>> print(5+ 0.2)  
5.2
```

```
>>> exit() # oppure premo Ctrl-D
```

```
lauria@macbook15: ~/$ python3 nomeprogramma.py
```

# Interfaccia testuale S.O. vs Python

```
lauria@macbook15: ~/$ cd Documenti
lauria@macbook15: ~/Documenti$ cd ..
lauria@macbook15: ~/$ python3
```

```
>>> print(5+ 0.2)
5.2

>>> exit() # oppure premo Ctrl-D
```

```
lauria@macbook15: ~/$ python3 nomeprogramma.py
  blah blah blah output del programma blah blah
  blah blah blah output del programma blah blah
  blah blah blah output del programma blah blah

lauria@macbook15: ~/$
```

# Python come ambiente interattivo

```
lauria@macbook$ python3

Python 3.7.3 (default, Jun 17 2019, 12:55:33)
[GCC 7.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 4 + 5
9
>>> 'Ciao mondo'
'Ciao mondo'
>>> x = 7
>>> x*2 - 5
9
>>> print('Ciao mondo')
Ciao mondo
>>> print(x)
7
>>> [ qui potete premere Control-D per uscire ]

lauria@macbook$
```

# Il computer non vi giudica. Fate errori!

- ▶ per esplorare il linguaggio Python
- ▶ per capire la diagnostica

```
>>> x
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'x' is not defined
>>> x = 5
>>> x
5
>>> 3 / 0
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: division by zero
>>>
```



# Programma Python su file

Scriviamo il file `primoprogramma.py`

```
4 + 5 1
'Ciao mondo' 2
x = 7 3
x*2 - 5 4
print('Ciao mondo') 5
print(x) 6
```

Andiamo nella cartella che contiene `primoprogramma.py`

```
lauria@macbook$ python3 primoprogramma.py
Ciao mondo
7
```

# Sequenza operativa

1. Avviare il sistema operativo Statistica-Laboratorio
2. entrare con utente **studente** (senza password)
3. trovare *terminale ed ambiente di programmazione*
4. scrivere il programma
5. salvarlo su un file in qualche cartella
6. (per Thonny) eseguirlo dentro Thonny
7. trovare la cartella da terminale
8. eseguire

```
$ python3 nomeprogramma.py
```

Dove `nomeprogramma.py` va cambiato con il nome del file su cui avete salvato il vostro programma.