

Tutorial del laboratorio

Informatica@DSS 2019/2020 — Il canale

Massimo Lauria <massimo.lauria@uniroma1.it>
<http://massimolauria.net/courses/informatica2019/>

I software che useremo

- ▶ Editor di testi (per scrivere programmi)
- ▶ Terminale (per eseguire i programmi)
- ▶ Browser web (questionari e documentazione)

I software che useremo

- ▶ Editor di testi (per scrivere programmi)
- ▶ 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`

Editor di testo

**Programma per scrivere file contenenti testo
"semplice"**

- ▶ testi privi di formattazione
- ▶ adatti a testi con struttura logica (i.e., programmi)
- ▶ strumenti di supporto alla scrittura di software

**Sono molto diversi dai "word processor" come Office,
che permettono di produrre documenti formattati ma
non di manipolare efficientemente testi strutturati.**

Browser web

Questo lo conoscete sicuramente

- Firefox
- Internet explorer (Edge)
- Google Chrome

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`

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  
/Users/massimo/lavori/didattica/2018.IntroInformatica/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
/Users/massimo/lavori/didattica/2018.IntroInformatica/lectures

$ cd /Users/massimo/Documents

$ pwd
/Users/massimo/Documents
```

Esempi (3)

Salire alla cartella genitore

```
$ pwd
/Users/massimo/lavori/didattica/2018.IntroInformatica/lectures

$ cd ..

$ pwd
/Users/massimo/lavori/didattica/2018.IntroInformatica

$ cd ..

$ pwd
/Users/massimo/lavori/didattica
```


Esempi (4)

Andare in una sottocartella

```
$ cd /Users/massimo/
```

```
$ pwd  
/Users/massimo
```

```
$ cd lavori
```

```
$ pwd  
/Users/massimo/lavori
```

```
$ cd didattica
```

```
$ pwd  
/Users/massimo/lavori/didattica
```

Esempi (5)

Concatenare i movimenti relativi

```
$ pwd
/Users/massimo/lavori/didattica/2018.IntroInformatica/lectures

$ cd ../../2017.IntroInformatica/lectures/../../sito/

$ pwd
/Users/massimo/lavori/didattica/2017.IntroInformatica/sito
```

Esempi (6)

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

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

Il simbolo ~ è un alias per la cartella utente

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

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. Scegliere e avviare il sistema operativo Debian-9
2. entrare con utente studente (senza password)
3. trovare *terminale* ed *editor di testo*
4. scrivere il programma nell'editor di testo
5. salvarlo su un file in qualche cartella
6. trovare la cartella da terminale
7. eseguire

```
$ python3 nomeprogramma.py
```

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

Questionario

- ▶ info preliminari
- ▶ feedback sull'aula
- ▶ confidenza con gli strumenti del laboratorio

<http://bit.ly/INF02019-02>