

# Laboratorio di Python

Introduzione al laboratorio, al pair programming, a Python

Lab01

23 Febbraio 2018



# Outline

## Laboratorio

Struttura e programma delle lezioni

## Pair programming

## Python

Introduzione

Programmazione



# Outline

## Laboratorio

Struttura e programma delle lezioni

Pair programming

Python

Introduzione

Programmazione



# Outline

## Laboratorio

Struttura e programma delle lezioni

Pair programming

Python

Introduzione

Programmazione



# Lezione tipo

- ▶ Correzione degli esercizi svolti autonomamente e consegnati all'indirizzo di posta elettronica: **labinfomate@gmail.com**
  - ▶ Veloce rassegna di alcuni concetti spiegati a lezione
  - ▶ Assegnazione di esercizi da svolgere in laboratorio/aula
    - ▶ Poiché è naturale che ognuno di voi svolga gli esercizi a velocità diversa, sulla pagina di laboratorio verranno resi disponibili i testi di tutti gli esercizi di quel giorno. Usare con cautela ;)
  - ▶ Assegnazione di esercizi da svolgere a casa e consegnare **prima dell'inizio del laboratorio successivo**
- 
- ▶ Tutto il materiale di laboratorio su <http://lodi.web.cs.unibo.it/labinfomate.html>
  - ▶ Il codice di volta in volta caricato (soluzioni esercizi in classe e a casa) è più completo e dettagliato (e a volte con più soluzioni alternative) di quello sulle slide, ridotto per questioni di spazio.

# Esercizi per casa

- ▶ Fare gli esercizi di volta in volta (tentando e ritentando) è l'unico modo per imparare (e superare l'esame)
- ▶ Consegnare gli esercizi ogni volta a: **labinfomate@gmail.com**
- ▶ La prossima volta daremo indicazioni precise sulla consegna: sbagliare a consegnare equivale a non consegnare
- ▶ Verranno effettuati controlli automatici sul codice consegnato (controlli di copiatura, controlli sul contenuto...)
- ▶ A campione, verranno corretti alcuni esercizi
- ▶ La consegna di almeno 10 (su circa 14) esercitazioni permette di ottenere un voto in più all'esame.



# Outline

## Laboratorio

Struttura e programma delle lezioni

## Pair programming

## Python

Introduzione

Programmazione



# Programmazione in coppia: come?

## Driver

- ▶ Colui che controlla mouse e tastiera (il “volante”)
- ▶ Ha l'obiettivo di scrivere un programma funzionante
- ▶ Scrive il codice e spiega al compagno (navigator), passo passo, ciò che sta facendo
- ▶ Prende le decisioni

## Navigator

- ▶ Osserva ciò che fa il driver e ascolta quanto egli dice
- ▶ Presta attenzione, individua errori e suggerisce correzioni
- ▶ Suggerisce strade alternative
- ▶ Si concentra sulla “visione generale” dell'esercizio

**Ogni tanto (all'inizio vi guideremo), ci si scambiano i ruoli!**





# Programmazione in coppia: perché?

- ▶ Usato nelle aziende: pagare due persone per lavorare meglio su un solo progetto... conviene!
- ▶ Si individuano e correggono più errori
- ▶ Il driver può concentrarsi sui dettagli, il navigator sulla soluzione generale
- ▶ Spiegare ad un'altra persona il codice è un ottimo modo per capire meglio cosa si sta facendo e se si sta andando nella direzione giusta

# Programmazione in coppia: un'occasione per voi

- ▶ Si tratta di uno strumento per imparare: se la coppia non funziona, bisogna essere abbastanza maturi da cambiarla
- ▶ Un driver che non considera il navigator, o un navigator prepotente non sono auspicabili
- ▶ Gli esercizi a casa vanno fatti **da soli**: usateli come esercizi in preparazione alla prova pratica in laboratorio e allo scritto (anch'essi andranno fatti da soli !!)



# Outline

## Laboratorio

Struttura e programma delle lezioni

## Pair programming

## Python

Introduzione

Programmazione



# Outline

## Laboratorio

Struttura e programma delle lezioni

## Pair programming

## Python

**Introduzione**

Programmazione



# Cos'è Python?



È un linguaggio di programmazione.

- ▶ Useremo Python 3.\*
- ▶ <https://www.python.org/downloads/>
- ▶ su Linux: `sudo apt-get install idle3`, che installa sia Python 3 sia IDLE, l'IDE che useremo

# Python

Si può eseguire un file Python (già composto usando un editor di testo) digitando da terminale `python nomefile.py` (o il comando `python3` per essere sicuri di usare Python 3).

Per programmare più comodamente useremo IDLE (Ambiente di sviluppo integrato di Python).

Per mandare in esecuzione IDLE:

- ▶ da terminale eseguiamo il comando `idle` (o il comando `idle3` per essere sicuri di usare Python 3)
- ▶ controlliamo la versione nella prima riga della shell di Python che si apre
- ▶ oppure apriamo IDLE 3 dall'interfaccia grafica



# Outline

## Laboratorio

Struttura e programma delle lezioni

## Pair programming

## Python

Introduzione

Programmazione



# Shell vs. File

Alcuni comandi:

- ▶ Nella shell, scrivere `print("Hello World!")` e premere invio.





# Shell vs. File

Alcuni comandi:

- ▶ Nella shell, scrivere `print("Hello World!")` e premere invio.
- ▶ File → New window, si apre un nuovo file, scrivere `print("Hello World!")` e premere invio.



# Shell vs. File

Alcuni comandi:

- ▶ Nella shell, scrivere `print("Hello World!")` e premere invio.
- ▶ File → New window, si apre un nuovo file, scrivere `print("Hello World!")` e premere invio.
- ▶ Non succede nulla! Bisogna salvare il file ed eseguirlo (con Run → Run module, oppure con F5)



# Comandi

Alcuni comandi:

- ▶ `print("Hello World!")` (scrive Hello World!)
- ▶ `x = 2` (assegna il valore intero 2 alla variabile x)
- ▶ `x = x+5` (somma 5 al valore di x; assegna tale valore a x)

Ogni operazione definisce un comando specifico a seconda del tipo di dato che si sta usando.



# Tipo di dato

Insieme di valori + operazioni che manipolano tali valori.

- ▶ interi (int) → operazioni tra interi
- ▶ numeri razionali (float) → operazioni tra float
- ▶ numeri complessi (complex) → operazioni tra complessi
- ▶ valori booleani (bool) → operazioni tra booleani
- ▶ valori stringa (str) → operazioni tra stringhe
- ▶ ...

# Operazioni su interi, virgola mobile, complessi

## Rappresentazioni:

- ▶ interi:  $M=3$
- ▶ razionali (float):  $M=3.0$
- ▶ complessi:  $M=(3+1j)$

## Operazioni:

- ▶  $M+M \rightarrow$  somma (interi, float, complessi)
- ▶  $M*M \rightarrow$  prodotto (interi, float, complessi)
- ▶  $M/M \rightarrow$  divisione con risultato float, o complesso
- ▶  $M//M \rightarrow$  quoziente intero
- ▶  $M\%M \rightarrow$  modulo (resto della divisione intera, solo tra interi)
- ▶  $M**M \rightarrow$  elevamento a potenza (interi, float, complessi)



# Operazioni stringhe

Rappresentazione:

- ▶  $M = \text{"Prova"}$
- ▶  $N = \text{"casa"}$

Operazioni:

- ▶  $M+N \rightarrow$  concatena la stringa  $M$  ed  $N$  (es. Provacasa)
- ▶  $M*3 \rightarrow$  concatena 3 volte la stringa  $M$  (es. ProvaProvaProva)
- ▶  $\text{len}(M) \rightarrow$  restituisce la lunghezza di  $M$  (es. 5)
- ▶  $M[0], \dots, M[\text{len}(M)-1] \rightarrow$  restituisce i singoli caratteri della stringa.  
es:  $(M[0] \rightarrow P)$



# Esercizio di lettura 1

Cosa si visualizza a video dopo la print?

---

```
1 a=2
2 b=3
3 area=a*b
4 perimetro=a*2+b*2
5 print(area , perimetro)
```

---

## Esercizio di lettura 2

Cosa si visualizza a video dopo la print?

---

```
1 a = " ciao"  
2 b = "  mondo"  
3 aeb=a+b  
4 print(aeb)
```

---



## Esercizio di lettura 3

Cosa si visualizza a video dopo la print?

---

```
1 a = "ciao"  
2 b = 5  
3 aperb = a*b  
4 print(aperb)
```

---

# Esercizi da svolgere in classe

- ▶ Stampare a video il perimetro di un quadrato avente lato  $l=4$ .
- ▶ Stampare a video l'area di un quadrato avente lato  $l=5$ .
- ▶ Stampare a video  $n$  volte, con  $n=10$ , la stringa  $s$  (es. con  $s="ciao"$  stamperà `ciaociaociaociaociaociaociaociaociao`)



E' ora di scambiarsi i ruoli

Pair programming: cambio!



# Esercizi da svolgere in classe

- ▶ Stampare a video una stringa lunga 4 caratteri al contrario (es. se `s="lodi"`, il programma stampa `idol`)
- ▶ Supponete di correre 10 km in 42 min e 42 sec. Stampate la vostra velocità media in km/minuto, km/h, miglia/minuto e miglia/h.



# Esercizi da svolgere in classe

- ▶ Stampare a video una stringa lunga 4 caratteri al contrario (es. se `s="lodi"`, il programma stampa `idol`)
- ▶ Supponete di correre 10 km in 42 min e 42 sec. Stampate la vostra velocità media in km/minuto, km/h, miglia/minuto e miglia/h.
  - ▶ Calcolate quanti secondi ci sono in 42 minuti e 42 secondi.
  - ▶ A quante miglia corrispondono 10 km? (Suggerimento: ci sono 1,61 km in un miglio)
  - ▶ La vostra velocità media è calcolata come distanza/tempo.

