

# Laboratorio di Python

## Alcuni moduli utili, Esercizi vari

Università di Bologna

9 aprile 2015

# Sommario

- 1 Correzione esercizi
- 2 Alcuni Moduli Utili
- 3 Scegliere una struttura dati
- 4 Esercizi per casa

# Esercizi per casa

Scrivere e documentare le funzioni che risolvano i seguenti problemi:

- 1 Definire una funzione che presa una sequenza, come parametro restituisca il numero dei valori valle; un elemento appartenente a una determinata sequenza si definisce valle se  $s[i] < s[i-1]$  e  $s[i] < s[i+1]$
- 2 Definire una funzione che presa una sequenza, come parametro restituisca il numero dei valori apice; un elemento appartenente a una determinata sequenza si definisce apice se  $s[i] > s[i-1]$  e  $s[i] > s[i+1]$



# Correzione

```
def numero_valle(s):  
    n=0  
    for i in range(1,len(s)-1):  
        if s[i]<s[i-1] and s[i]<s[i+1]:  
            n=n+1  
    return n
```



# Correzione

```
def numero_apice(s):  
    n=0  
    for i in range(1,len(s)-1):  
        if s[i]>s[i-1] and s[i]>s[i+1]:  
            n=n+1  
    return n
```



# Modulo random

- 1 `import random`
- 2 `random.random()` (numero casuale tra [0.0, 1.0))
- 3 `random.uniform(a, b)` (numero float casuale tra a e b)
- 4 `random.randint(n, m)` (intero casuale tra n e m, estremi inclusi)
- 5 `random.randrange(n, m, k)` (intero casuale in range(n, m, k))
- 6 `random.choice(s)` (sceglie a caso un elemento da una sequenza s)
- 7 `random.sample(s, n)` (estrazione casuale senza reinserimento di n elementi da una sequenza s)

Maggiori informazioni:

<https://docs.python.org/3/library/random.html>.



# Manipolazione stringhe

- Il tipo `str` ha dei metodi interessanti come `split` e `splitlines`.
  - 1 `line='Il mio cane si chiama: Fido.'`
  - 2 `line.split()` → `['Il', 'mio', 'cane', 'si', 'chiama:', 'Fido.']`
  - 3 `line.split(':')` → `['Il mio cane si chiama', 'Fido.']`
  - 4 `text='Il testo che segue,  
vi mostra il funzionamento  
del metodo splitlines.  
Ecco cosa accade'`
  - 5 `text.splitlines()` → `['Il testo che segue,', 'vi  
mostra il funzionamento', 'del metodo  
splitlines.', ' Ecco cosa accade.']`
  - 6 `line[0:2].isalpha()` → `True`
- Vedi anche <https://docs.python.org/3/library/stdtypes.html#string-methods>.



# Leggere file

Python ha un tipo di dato File, completo di una serie di metodi.

- 1 `f = open('./esempio.py', 'r')`  
# apre il file `'./esempio.py'` in lettura (`r=read`);
  - 2 `lines = f.readlines()`  
# contiene la lista di tutte le righe del file.
- `f.close()` richiamo il metodo `close` sull'oggetto `f`.
  - Per i più curiosi, altre informazioni su <https://docs.python.org/3.3/tutorial/inputoutput.html#methods-of-file-objects>.





# Scrivere su file

```
fo=open('prova.txt', 'w')  
fo.write('Primo file di testo:\n\n')  
fo.write('terza riga,\n')  
fo.write('quarta riga riga,\n')  
fo.write('fine del file.')  
fo.close()
```

- 1 `fo = open('prova.txt', 'w')` apro il file prova in scrittura, se non esiste lo crea. `fo` ora è un'istanza della classe file. (`fo` oggetto)
- 2 `fo.write('Primo file di testo:  
n  
n')` richiamo il metodo `write` col testo che desidero scrivere (Primo file di testo:  
n  
n) della classe file al mio oggetto `fo`.
- 3 `fo.close()` richiamo il metodo `close` sull'oggetto `fo`.



# Esercizio

- 1 Scrivere una funzione che dato il nome di un file scelga casualmente una parola presente in tale file.



# Scompongo l'esercizio in sotto-problemi

- 1 Scrivere una funzione che apra un file e restituisca tale file come una lista.
- 2 Scrivere una funzione che presa una stringa elimini alcuni caratteri speciali e restituisca la stringa pulita.
- 3 Scrivere una funzione che data una stringa definisca se questa è una parola (si definisce parola una concatenazione di sole lettere.)
- 4 Scrivere una funzione che data una lista di stringhe restituisca la lista delle sole parole distinte presenti tra le stringhe.
- 5 Scrivere una funzione che dato un file restituisca casualmente una parola appartenente al file



## Apri il file; Elimina punteggiatura

---

```
import random
import string
def openfile(nome):
    f=open(nome, 'r')
    r=f.readlines()
    f.close()
    return r
```

---

```
def eliminapunteggiatura(s):
    #elimino la punteggiatura nella parola
    s=s.lstrip("`'.,;:!?"+'"}'-'').rstrip("`'.,;:!?"+'"}'')
    return s
```

---



## Dalla lista di stringhe alla lista di parole

---

```
def suddivido_parole(r):  
    #data una lista di stringhe prende solo le parole  
    w=[] #variabile per la lista delle parole  
    for line in r: #per ogni riga del file  
        for s in line.split(): #per ogni stringa  
            s=eliminapunteggiatura(s)  
            parola=isalpha(s) #verifico se e parola  
            if (s not in w) and parola:  
                w.append(s)  
  
return w
```

---



# Estrazione casuale di una parola da un file

---

```
def random_word_from_file(nome):  
    return random.choice(suddivido_parole(openfile(nome)))
```

---

# Esercizio

Scrivere una funzione che stampi la lista delle linee contenute nel file dove è salvata la funzione stessa. Cos'è un programma? Sapreste spiegare come funziona questa funzione?

# Matrici diagonali

Quale può essere una buona struttura dati per la memorizzazione e l'implementazione delle operazioni con le matrici diagonali? Una matrice diagonale è una matrice quadrata in cui solamente i valori della diagonale principale possono essere diversi da 0.



# Soluzione

- 1 Matrice = [ [1,0,0], [0,2,0], [0,0,4] ]
- 2 Matrice = [1,2,4]

Perchè scegliere la seconda rappresentazione?

# Esercizio

Definiamo le funzioni che implementino le operazioni di:

- 1 somma
- 2 moltiplicazione

tra matrici diagonali. Definire le funzioni che data una matrice diagonale ne calcoli il:

- 1 determinante
- 2 la matrice inversa



# Esercizi

Scrivere e documentare le funzioni che risolvano i seguenti problemi:

1 Definiamo le funzioni che implementino le operazioni di:

- 1 somma
- 2 moltiplicazione

indipendentemente dalla rappresentazione della matrice che è stata utilizzata.

2 Scrivere una funzione che presi due file copi il primo nel secondo file.

3 Scrivere una funzione che simuli l'estrazione del lotto, tale funzione deve scrivere su un file il risultato dell'estrazione.

Inviare gli esercizi svolti a: **labinfo.mat.unibo@gmail.com**



# Cosa abbiamo fatto?

- 1 Correzione esercizi
- 2 Alcuni Moduli Utili
- 3 Scegliere una struttura dati
- 4 Esercizi per casa