

## Esercitazioni di Fondamenti di Informatica - Lez. 10 03/12/2020

### Programmazione Python

Il codice degli esercizi e' contenuto nella cartella **codice10**

LINK UTILI
<a href="https://www.python.it">https://www.python.it</a> —> download
<a href="https://www.jetbrains.com/pycharm/">https://www.jetbrains.com/pycharm/</a>
<a href="https://gist.github.com/ricpol/2ca0ae46f02bfddf08036fa85519aa97">https://gist.github.com/ricpol/2ca0ae46f02bfddf08036fa85519aa97</a> guida per windows
<a href="https://docs.python.it/html/lib/contents.html">https://docs.python.it/html/lib/contents.html</a>
<a href="https://www.w3schools.com/python/python_reference.asp">https://www.w3schools.com/python/python_reference.asp</a>

### Dal metalinguaggio al linguaggio in pratica

in metalinguaggio	in Python
include	import
{ } blocchi di codice	INDENTAZIONE SENZA FINE RIGA
dichiarazione variabili ( tipo nome valore )	CREO NOME e USO
leggo da input	INPUT()
scrivo su output	PRINT()
if..then else	IF cond : ... ELIF cond :...ELSE :
while ..do	WHILE cond :
for ...	FOR i in RANGE(1,max) FOR _ in RANGE(max)
funzioni	def nomefunzione( parametro ) :
array/lista	ls = []

## 1. VERIFICA SE ESISTE SUL NOSTRO SISTEMA ambiente Linux:

```
$ which python  
/usr/bin/python
```

oppure:

```
$ which python3  
/usr/bin/python3
```

se riceverete un output simile a quelli descritti, Python è già presente nel vostro computer.

### Linux e Mac OSX :

da console, richiamate le informazioni chiedendole **direttamente** all'interprete Python:

```
$ python -V oppure % python3 -V
```

```
>> Python 3.9.0 (la versione installata)
```

### Windows

```
c:> py
```

se presente lancia la shell

## 2. INSTALLAZIONE

installare da <https://www.python.it.....download>

in Windows installa diversamente se fatto per singolo o per tutti gli utenti

**c:/Programm file/python3** ( come admin visibile da ogni utente)

- altrimenti si troverà in una cartella locale

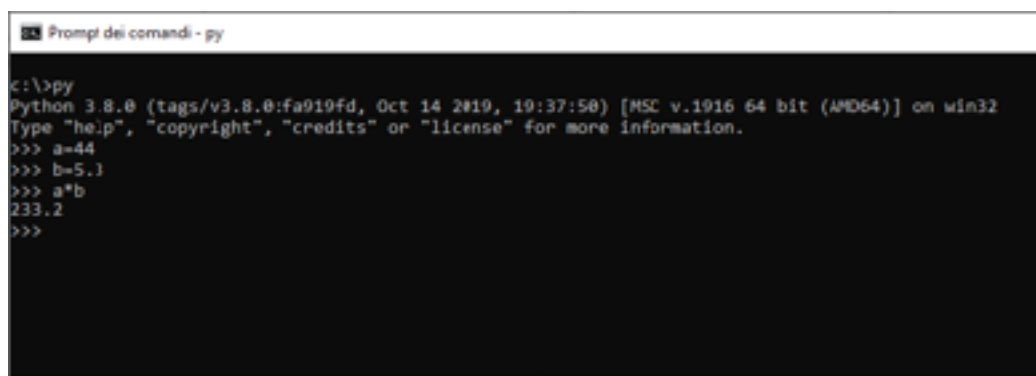
### 3. USO

- lanciare l'interprete in due modi

**c:> py**

**c:> python3**

**>>>** interpreta i comandi uno per volta

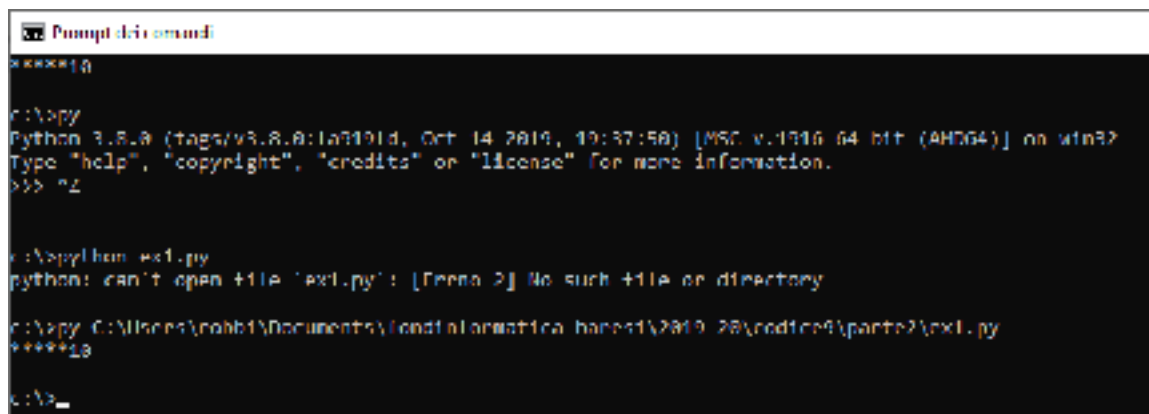


```
Prompt dei comandi - py
c:\>py
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a=44
>>> b=5.1
>>> a*b
233.2
>>>
```

**c:> python3 script.py** oppure

**c:> python3 c:/users/ciccio/doc/ex1.py**

esegue i comandi in uno script



```
Prompt dei comandi
c:\>py
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> ^Z
^Z

c:\>python ex1.py
python: can't open file 'ex1.py': [Errno 2] No such file or directory

c:\>py C:\Users\ciccio\Documents\fondamenti di informatica - esercizi\2019-20\credito4\parte2\ex1.py
****18

c:\>_
```

- usare IDLE che è l'IDE di Python integrato  
si trova come app installata con PythonX
- PyCharm gratuito per utenti politecnico

#### 4. ESERCIZI DI RISCALDAMENTO

variabili, istruzioni strutturate, funzione, main, lettura e scrittura utente

– LEGGO E SALUTO

```
name = input('come ti chiami ? ')
print('benvenuto/a',name)
```

– STAMPA 5 STAR E IL VALORE DI UNA VARIABILE

```
for _ in range(5):
    l=10
    print('*', end='')
print(l)
```

– CREO UNA LISTA LEGGENDO I VALORI DA STANDARD INPUT

– LA STAMPO

```
lista2 = []
for i in range(4):
    el = input('leggo ')
    lista2.append(el)

print('lista2 ')
for i in lista2:
    print(' ', i, end = ' ')
print(' ')
```

– svolto a lezione

```
for i in range(1,6):
    el = int(input(' valore '))
    list.append(el)
# for i in list:
print(list)
print('sostituisci un elemento i')
i = int(input('> '))
# el = int(input(' valore '))
list[i] = int(input(' valore '))
print(list)
```

# inizializzo e modifico

```
lista1 = [1]*5
for i in lista1:
    print(i, end = ' ')
lista1[0] = input('cambia il primo ')

for i in lista1:
    print(i, end = ' ')
```

## 5. Indovina un numero

usiamo random per generare il numero da indovinare

```
# myNumber.py
# indovina un numero
```

```
import random
computer_num = random.randint(1, 10)
```

```
# Creare la funzione is_same()
```

```
# Start the game
print("Ciao .\n Ho pensato un numero da 1 and 10.")
```

```
# Collect the user's guess as an integer
guess = int(input("Vuoi indovinarlo ? "))
```

```
# Use our function

if computer_num == guess:
    result = "Vinto"
elif computer_num > guess:
    result = "Basso"
else:
    result = "Alto"

# gioca finchè vince

while result != "Vinto":
    if result == "Basso":
        guess = int(input("troppo basso ..prova ancora "))
    else:
        guess = int(input("troppo alto ..prova ancora "))

    if computer_num == guess:
        result = "Vinto"
    elif computer_num > guess:
        result = "Basso"
    else:
        result = "Alto"
# quando indent cambia ripete la condizione e il blocco

# End the game
input("Giusto!\nbene \n\n\nPremi INVIO per finire .")
```

## 6. Leggere un prezzo e un valore di sconto in input e calcolare il prezzo finale

```
p= float(input('prezzo '))  
s= float(input('sconto '))  
p-=20/100*p  
print(p," prezzo finale")
```

### Scambio valori

```
a=int(input('Inserisci a: '))  
b=int(input('Inserisci b:'))  
print('I valori inseriti sono a:', a, ' e b: ', b)  
print('Adesso scambio i valori',a,b)  
a,b=b,a  
print('I valori scambiati sono a:', a, ' e b: ', b)
```

lettura carattere

```
vocali = 'aeiou'  
g = input("scrivi car ")  
print(g)  
if g in vocali:  
    print(g,"c'è")  
else:  
    print(g,"no")
```

## 7. Partendo dal seguente dizionario:

```
prezzi = {'mela':0.50, 'banana':0.60, 'ananas':1, 'fragola':0.2}
```

Scrivere un programma Python 3 che definisce una appposito dizionario spesa= { ....} prodotto: qtà, per una spesa specifica, calcola il costo e lo stampa a schermo (usando due cifre decimali per i centesimi di euro).

```
# Dizionario prezzi
prezzi = {'mela':0.50, 'banana':0.60, 'ananas':1, 'fragola':0.2}
# Struttura spesa
spesa = {'mela':3, 'banana':1, 'ananas':2, 'fragola':10}

totale = 0

for prodotto, quantita in spesa.items():
    totale +=prezzi[prodotto]*quantita

print("Totale spesa %.2f" % totale)
```

8. ESAME 31/08/2017 es. 6. Scrivere un programma Python3 che definisce un appposito dizionario che contiene, attraverso le coppie chiave-valore, i costi di un insieme di prodotti orto-frutticoli. Il programma poi deve chiedere all'utente le quantità richieste dei prodotti disponibili (contenuti nel dizionario) e calcolare il costo totale dei prodotti, e quantità, selezionati.

ALGORITMO possibile

Definisco il dizionario {prodotto, prezzo}

per ogni elemento in dizionario -> visualizza prodotto e prezzo

chiedere in input prodotto e quantità

se il prodotto non è nel dizionario ripetere la richiesta



per ogni prodotto inserito

spesa= spesa+dic.prezzo \*quantità

quantità\_tot = quantità\_tot+quantità

## SOLUZIONE

```
dic = {"mela":0.2, "pera":0.3,"kiwi":0.5}

fattura = {}
costoTot = 0

print("frutta disponibile")

for x,y in dic.items():
    print(x,"prezzo: %.2f" %y)

while True:
    key=input("Inserire il frutto che si vuole comprare: ")

    if not key:
        break

    if key in dic.keys():

        q=int(input("inserire la quantita': "))
        costoTot += float(dic[key])*int(q)
        if key not in fattura.keys():
            fattura[key] = 0
            fattura[key] += float(dic[key])*int(q)
        else:
            fattura[key] += float(dic[key])*int(q)
    else:
        print("frutto non disponibile")

print("Il costo totale e' %.2f" %costoTot)

for x,y in fattura.items():
    print(x,"costo: %.2f" %y)
```

## 9. Scrivere il codice Python che:

- (a) legge un intero n
- (b) legge n stringhe di testo
- (c) salva le stringhe piu' lunghe di 5 caratteri in un file "long.txt", mentre quelle piu corte in un file "short.txt"

```
# Leggo un numero intero
n = int(input("Inserisci un numero intero: "))

# apro entrambi i file in scrittura
fshort = open("short.txt", "w")
flong = open("long.txt", "w")

# ripeto il ciclo per n volte
for _ in range(n):
    # leggo una stringa di testo
    s = input("Inserisci una stringa di testo: ")
    # e confronto la sua lunghezza
    if len(s)>5:
        flong.write(s+"\n") # la scrittura nel file avviene
                           # aggiungendo un \n per andare
                           # a capo
    else:
        fshort.write(s+"\n")

fshort.close()
flong.close()
```

**domanda** : cosa accade se non chiudo il file con close?

**risposta** : al termine dello script non è possibile leggere il contenuto in chiaro, sembra vuoto

**10. Scrivere il codice Python che legga un file "preventivo.csv" nel seguente formato:**

```
Prodotto,Importo unitario, Quantita'  
Bicchiere,3.00,6  
Piatto,2.50,6  
Set posate,21.00,1
```

**Legga tutte le righe e calcoli il totale per ogni prodotto (installazione esclusa) e dell'intero preventivo. I risultati devono essere poi salvati in un secondo file "fattura.txt" in questo formato**

```
Prodotto,Importo unitario,Quantita',Totale  
Bicchiere,3.00,6,18.00  
Piatto,2.50,6,15.00  
Set posate,21.00,1,21.00  
Totale,,,54.00
```

**SOLUZIONE**

```
# Apro il file e leggo tutte le righe  
  
preventivo = open("preventivo.csv", "r")  
linee = preventivo.readlines()  
preventivo.close()  
  
totale_fattura = 0  
for indice in range(len(linee)): # Per ogni riga del file  
    lista = linee[indice].strip().split(",") # Separo i dati  
    # strip() elimina gli spazi  
    # split(",") separa i dati secondo le virgole  
    if indice == 0: # Se e' la prima riga devo solo aggiungere la  
        # dicitura Totale  
        lista.append("Totale")  
    else: # Altrimenti recupero prezzo e quantita' e  
        # aggiorni i totali  
        totale = float(lista[1])*int(lista[2])  
        totale_fattura += totale
```

```

        lista.append("%.2f" % totale)
        linee[indice] = ",".join(lista) + "\n" # Trasformo la lista in stringa

        linee.append("Totale,,,%0.2f" % totale_fattura)

# Apro il file in scrittura e stampo tutte le righe
fattura = open("fattura.csv", "w")
fattura.writelines(linee)
fattura.close()

```

## 11. Scrivere il codice Python che legge un file di testo "testo.txt" e lo stampa al contrario(dall'ultima riga alla prima)

```

1.
2.  # Metodo 1: leggo tutte le righe del file e le scorro in ordine contrario
3.
4.  def contrario_uno():
5.      file = open("testo.txt")
6.      for l in file.readlines()[::-1]:
7.          print(l.rstrip()) # rstrip ritorna una copia della stringa dove
8.                             # tutti i caratteri sono stati tolti dalla fine della
9.                             # stringa (caratteri di spazio di default)
10.         file.close()
11.  contrario_uno()
12.
13. # Metodo 2: leggo tutte le righe del file e le scorro creando una
14. # nuova lista dove l'ultima riga letta e' messa sempre in testa
15.
16. def contrario_due():
17.     file = open("testo.txt")
18.     contrario = []
19.     for l in file.readlines():
20.         contrario = [l.rstrip()] + contrario
21.     for l in contrario:
22.         print(l)
23.     file.close()
24.  contrario_due()
25.
26. # Metodo 3: dichiaro una funzione ricorsiva che uso per
27. # scorrere tutte le righe del file
28.
29. def contrario_tre():
30.     file = open("testo.txt")
31.     def f(lines):
32.         if lines:
33.             f(lines[1:]) # Dalla riga successiva fino alla fine
34.             print(lines[0].rstrip()) # stampa la prima riga
35.     f(file.readlines())
36.     file.close()
37.  contrario_tre()

```

**12. Creare del codice Python3 che legga tutto ciò che viene scritto sullo standard input e riporti ogni carattere letto su un file passato come argomento. La lettura continua fino alla lettura di un carattere terminatore '\$'**

```
1. import sys # Parametri e funzioni specifiche per il sistema
2.
3. file = open("log.txt", "w") # apro il file in scrittura
4.
5. c=""
6. while c!='$': # fino a che non trovo il carattere '$'
7.     c = sys.stdin.read(1) # leggo un carattere da stdin
8.     # c = input()
9.     if c!='$':
10.         file.write(c)
11. file.close()
```

**13. Scrivere un programma in Python3 che legge da un file di testo dato (testo.txt) e stampa il numero di parole lette, il numero di caratteri non spazi bianchi letti, e la lunghezza media delle parole lette.**

```
1.
2. # Apro il file in lettura
3. file = open("testo.txt", "r")
4.
5. # Leggo tutte le parole e le separo in una lista
6. parole = file.read().split()
7.
8. # Il numero di parole e' la lunghezza della lista
9. print("Parole lette: %d" % len(parole))
10.
11. # Il numero di caratteri e' la lunghezza della stringa ottenuta
12. # concatenando tutte le parole della lista senza spazi
13.
14. print("Numero caratteri: %d" % len("".join(parole)))
15.
16. # La lunghezza media e' il rapporto tra i precedenti valori
17. print("Lunghezza media: %2.f" % (len("".join(parole))/len(parole)))
18. file.close()
```

**14. Scrivere una funzione python che data una lista di interi, restituisca il minimo e il massimo tra gli interi presenti in lista.**

```
# Funzione che calcola sia massimo che minimo scorrendo
# tutti gli elementi di una lista

def min_max(lista):
    if not lista:
        return None
    min = max = lista[0]
    for i in lista[1:]:
        if i < min:
            min = i
        if i > max:
            max = i
    return min, max #Restituisco due valori (come tupla)

print("Minimo %d - Massimo %d" % min_max([1,4,2,5,3,1,2,9,-1]))
```

**15. Scrivere un programma Python 3 che legge un testo (memorizzato nel file testo.txt), ordina le parole lette in base alla loro lunghezza crescente e stampa solo le parole di lunghezza dispari senza le eventuali vocali. Ad esempio, CREMONA diventerebbe CRMN.**

```
# Apro il file in lettura
file = open("testo.txt", "r")

# Recupero dal file la lista delle parole di lunghezza dispari
parole = [p for p in file.read().split() if len(p) % 2 != 0]

# Ordino le parole in ordine crescente di lunghezza
parole.sort(key=len) # key e' la funzione che viene usata
# per la comparison

for p in parole:
    # per ogni parola stampo solo i caratteri che non sono
    # nella stringa di vocali
    for c in p:
        if c not in 'AEIOUaeiou':
            print(c, end="")

print('\n')
file.close()
```

**1. Scrivere una funzione python che generi tutte le possibili combinazioni degli elementi di una lista**

```
def combinazioni(lista):
    if not lista:
        return [lista]
    else:
        lista_combinazioni = combinazioni(lista[1:])
        return [lista[:1] + c for c in lista_combinazioni] + lista_combinazioni

print(combinazioni(['a','b','c']))
```

**16. La seconda funzione prende in ingresso un puntatore a file e restituisce un intero, ovvero il numero di parole palindroma nel file. Il programma principale deve chiedere all'utente di inserire il nome del file, aprire il file, invocare le funzioni definite in precedenza, se e come si ritiene opportuno, e stampare il numero di parole palindroma nel file considerato.**

```
1.
2. # Questa funzione guarda se una parola e' palindroma
3. def palindrome(p):
4.     if len(p) == 1:
5.         return True
6.     else:
7.         if p == p[::-1]:
8.             return True
9.         return False
10.
11. #N.B. se p ='ciao' -> p[::-1] = 'oaic'
12.
13. def pPalindrome(f):
14.     r=0
15.     parole = f.read().split()
16.     for p in parole:
17.         if palindrome(p):
18.             r = r + 1
19.     return r
20.
21.
22. def main():
23.     nf = input("Inserisci nome file: ")
24.     f = open(nf,'r')
25.     print("Il numero di parole palindroma e': ", pPalindrome(f))
26.     f.close()
27.
28. main()
```

**17. ESAME 19/02/18 es. 6.**

Scrivere un programma Python 3 composto da due funzioni ed un programma principale. La prima funzione prende in ingresso un intero e restituisce 1 se il numero è primo, 0 altrimenti. La funzione non può essere ricorsiva. La seconda funzione prende in ingresso un puntatore a un file e restituisce un intero, pari alla quantità di numeri primi contenuti nel file, e tre altri numeri, ovvero i primi tre numeri primi identificati. Il programma principale deve chiedere all'utente di inserire il nome del file, aprire il file in modo appropriato, invocare le funzioni definite in precedenza, se e come si ritiene opportuno, e stampare il numero restituito, ovvero quanti sono i numeri primi nel file considerato, e i primi tre numeri identificati. Non è possibile usare variabili condivise.

```

1. # Funzione che restituisce True se n e' primo, False altrimenti
2. def numeroPrimo(n):
3.     primo = True
4.     if n <=1 or n%2==0:
5.         primo = False
6.     for i in range(2,round(n/2)):
7.         if n % i==0:
8.             primo = False
9.     return primo
10.
11. # Funzione che conta il totale di numeri primi letti e salva
12. # i primi 3 tra questi
13. def numeriPrimiFile(fileNumeri):
14.     # Dichiariamo un dizionario dove il primo elemento e' il totale di numeri
15.     # primi contenuti nel file, mentre gli altri elementi sono i primi 3
16.     # numeri primi letti
17.     dictPrimi = {'Totale':0,'numero1':None,'numero2':None,'numero3':None}
18.     for n in fileNumeri.read().split():
19.         if numeroPrimo(int(n)):
20.             dictPrimi['Totale']+=1 # Incremento il num tot. di numeri primi
21.                                     # letti
22.             # salvo i primi 3 numeri primi letti
23.             if dictPrimi['Totale'] <=3:
24.                 dictPrimi['numero'+str(dictPrimi['Totale'])]=int(n)
25.     return dictPrimi
26.
27.

```



```

28.
29. def main():
30.     # Chiedo all'utente di inserire il nome del file da cui leggere
31.     nomeFile=input('Inserisci il nome del file: ')
32.     fileNumeri = open(nomeFile,'r')
33.     dictPrimi = numeriPrimiFile(fileNumeri)
34.     print('Numero totale di numeri primi: %d' % (dictPrimi['Totale']))
35.     for i in range(1,4):
36.         print('Numero primo %d: %d' % (i,dictPrimi['numero'+str(i)]))
37.
38. main()

```

## 18. Esame 15/07/2019 Esercizio 6

**Scrivere un programma Python 3 che legge il nome di un file (di testo), apre il file e conta il numero di parole che contengono almeno una a ed una l, in qualsiasi ordine. In questo caso, una parola non `e altro che una sequenza di caratteri senza spazi nel mezzo. Si consiglia l'adozione di uno stile di programmazione *alla Python*.**

```

ftesto= open("testo.txt","r")
t= ftesto.read().split()
ftesto.close()
a=0
for p in t :
    if 'a' or 'l' in p :
        print(p) #la stampiamo per vedere quale
        a+=1
print("numero di parole trovate: ",a)

```