



FONDAMENTI DI INFORMATICA

<http://people.unica.it/gianlucamarcialis>

A.A. 2018/2019

Docente: **Gian Luca Marcialis**

PYTHON: ESERCIZI DI PROGRAMMAZIONE PARTE 2

Scrittura «formattata» in Python

➤ Provare sull'interprete la seguente:

```
valore=12.547389
s="%15.2f\n" % valore
print(s)
```

- Abbiamo ottenuto una stampa «formattata», dove il luogo e lo spazio per la stampa dei valori è predefinito
 - 15 è il numero totale di caratteri dove inserire il numero allineato a destra
 - 2 sono le cifre significative da visualizzare nella parte frazionaria
 - f indica che vogliamo stampare un numero floating point
 - 16 è il numero totale di caratteri presenti nella stringa (incluso "\n")

Più valori in formattazione

Valore da stampare	Codice stringa
Intero	%d
Float	%f
Stringa	%s

➤ Una stringa formattata più complessa:

```
valore=12.547389
nome= "pippo"
A=10
s="%15.2f%10s%4d\n" % (valore, nome, A)
print(s)
XXXXXXXXXX12.55XXXXXpippoXX10\n #X==spazio
```

Lettura/scrittura su file formattato

- Scrivere un programma che legge un file (nome acquisito da tastiera) di **numeri reali** e li scrive in un altro file (nome acquisito da tastiera) allineati a destra e seguiti dalla loro somma e dalla loro media aritmetica.
- Formattazione dei singoli valori:
 - 15 caratteri totali per ciascuno, con 2 cifre in parte frazionaria
 - Un solo valore per riga
- Righe relative a somma e media
 - Un solo valore per riga
 - Valori allineati ai precedenti, preceduti nella stessa riga dall'indicazione Total: per la somma e Average: per la media

Soluzione

```
#Chiede all'utente i nomi dei file di input e
di output
inputFileName=input("Inserire il nome del file
di input:")
outputFileName=input("Inserire il nome del file
dove scrivere il risultato:")

#Apre il file di input e di output
inFile=open(inputFileName,"r")
outFile=open(outputFileName,"w")
```

```
#Legge i dati e scrive il file di output
total=0.0
count=0
```

```
line=inFile.readline()
while line != "":
    valore=float(line)
    outFile.write("%15.2f\n" % valore)
    total=total+valore
    count = count + 1
    line=inFile.readline()
```

```
#Scrive il totale e il valore medio
outFile.write("%15s\n" % "-----")
outFile.write("Total: %8.2f\n" % total)

avg=total/count
outFile.write("Average: %6.2f\n" % avg)

#Chiude il file
inFile.close()
outFile.close()
```

Esercizio

- Implementare un programma Python che legga da file «persone.txt» una sequenza di coppie nome/età relativi rispettivamente a dei nomi di persone ed alla loro età, ciascuna su righe separate del file
- Scriva su un file «classi.txt», per ciascuna riga, il nome letto e l'attribuzione «Bambino», «Giovane», «Adulto», «Anziano», in funzione dei seguenti intervalli di età:
 - 0-14 Bambino
 - 15-25 Giovane
 - 26-65 Adulto
 - 66-. Anziano
- Il file prodotto deve presentare le coppie nome/classe perfettamente incolonnate a destra concendendo un campo di 20 caratteri al nome e 10 alla classe

Esempio

➤ File «persone.txt»

Giovanni 78

Filippo 17

Alessandra 40

➤ File «classi.txt»

Giovanni	Anziano
Filippo	Giovane
Alessandra	Adulto

Soluzione

```
fin=open("persone.txt","r")
fout=open("classi.txt","w")
```

```
#Ciclo di lettura del file
#inserire qui il codice
#della slide seguente
```

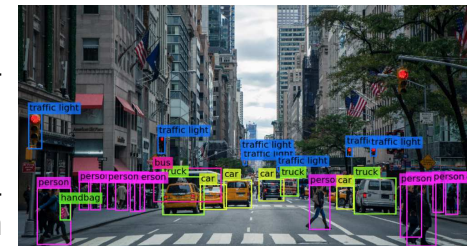
```
fin.close()
fout.close()
```

Ciclo di lettura del file

```
riga=fin.readline()
while riga!="":
    nome, eta=riga.split()
    eta=int(eta)
    if eta<15:
        classe="Bambino"
    elif eta<26:
        classe="Giovane"
    elif eta<66:
        classe="Adulto"
    else:
        classe="Anziano"
    s="%20s%10s\n" % (nome, classe)
    fout.write(s)
```

Esercizio

- Un sistema di rilevamento di oggetti in movimento ha registrato in ciascuna riga di un file «dati.txt» una coppia di valori numerici reali relativi a larghezza l ed altezza h dell'oggetto visto, in cm.



- Si vuole implementare un sistema che a partire dalle due misure fornite per ciascun oggetto, scriva su un file «riconoscimento.txt» se l'oggetto è presumibilmente una persona o un'autovettura o sconosciuto. In particolare:
 - Se $l \leq 50$ e $50 < h \leq 200$, l'oggetto è riconosciuto come persona
 - Se $l > 150$ e $100 < h \leq 180$, l'oggetto è riconosciuto come autovettura
 - Altrimenti, l'oggetto è sconosciuto

Esempio

➤ File «dati.txt»

```
45 160
400 100
200 10
```

➤ File «rilevamento.txt»

```
Persona
Autovettura
Sconosciuto
```

Soluzione

```
fin=open("dati.txt","r")
fout=open("riconoscimento.txt","w")

riga=fin.readline()
while riga!="":
    l, h=riga.split()
    l=int(l)
    h=int(h)
    if l<=50 and h>=50 and h<=200:
        classe="Persona\n"
    elif l>=150 and h>=100 and h<=150:
        classe="Autovettura\n"
    else:
        classe="Sconosciuto\n"
    fout.write(classe)

fin.close()
fout.close()
```

Esercizio

➤ Scrivere un programma Python che legga contemporaneamente i file «dati.txt» e «riconoscimento.txt» dell'esercizio precedente e memorizzi i dati in un dizionario le cui tre chiavi sono «Autovettura», «Persona», «Sconosciuto»

➤ Ad ogni chiave è associata una lista di tuple contenenti ciascuna larghezza ed altezza corrispondenti agli oggetti delle tre classi estratti dai due file

➤ Esempio. Il dizionario ottenuto dal caso precedente è:

```
{"Autovettura": [(400, 100)], "Persona":
[(45, 160)], "Sconosciuto": [(200, 10)]}
```

Soluzione

```
fd=open("dati.txt","r")
fc=open("riconoscimento.txt","r")

d={"Autovettura":[], "Persona":[], "Sconosciuto":[]}
rd=fd.readline()
rc=fc.readline()
while rd!="":
    rd=rd.split()
    t=(int(rd[0]), int(rd[1]))
    d[rc]=d[rc]+[t]
    rd=fd.readline()
    rc=fc.readline()

fd.close()
fc.close()
```

Per saperne di più...

- K.A. Lambert, *Programmazione in Python*, Apogeo (Maggioli), 2012.
- C. Horstmann, R.D. Nicaise, *Concetti di informatica e fondamenti di Python*, Apogeo (Maggioli), 2014.