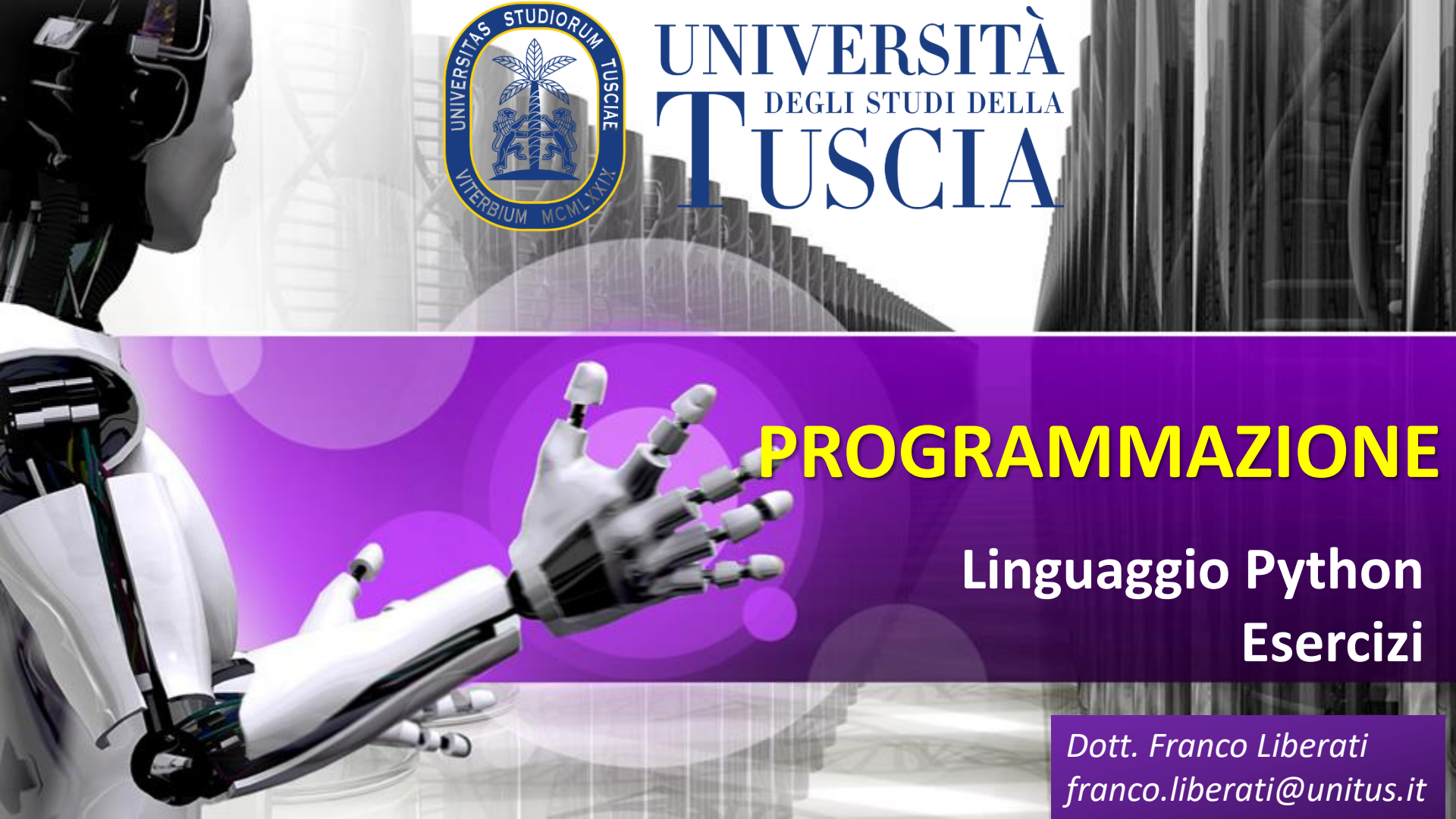


UNIVERSITÀ DEGLI STUDI DELLA TUSCIA

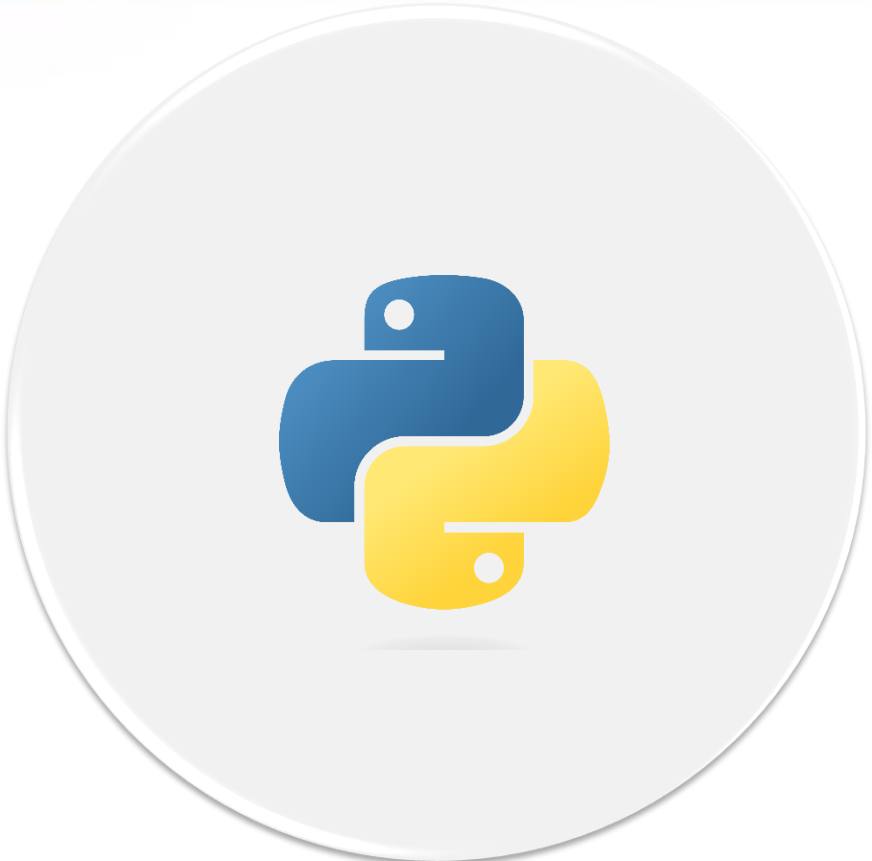
PROGRAMMAZIONE

Linguaggio Python
Esercizi

Dott. Franco Liberati
franco.liberati@unitus.it

LINGUAGGIO PYTHON

Esercizi



LINGUAGGIO PYTHON

Esercizio 0

Scrivere un programma in Python che:

- Prende due liste e crea un dizionario

Usare la funzione ZIP che associa le due liste in una serie di tuple

LINGUAGGIO PYTHON

Esercizio 0

```
nomi = ['Numa','Tullo','Anco']
cognomi = ['Pompilio','Ostilio','Marzio']
dict = { a: b for a, b in zip(nomi, cognomi)}

print(dict)
```

```
{'Numa': 'Pompilio', 'Tullo': 'Ostilio', 'Anco': 'Marzio'}
```

LINGUAGGIO PYTHON

Esercizio 1

Scrivere un programma in Python che:

- prenda una stringa in input da tastiera, rappresentante un nucleotide (A,C,G,T)
- stampa a video il nucleotide complementare

Assicurarsi che il programma funzioni correttamente sia con input maiuscolo che minuscolo

LINGUAGGIO PYTHON

Esercizio 1

```
nucleotide = raw_input('Inserisci un nucleotide (A,C,G,T): ')
 if nucleotide == 'A' or nucleotide == 'a':
                print 'T'
 elif nucleotide == 'C' or nucleotide == 'c':
                print 'G'
 elif nucleotide == 'G' or nucleotide == 'g':
                print 'C'
 elif nucleotide == 'T' or nucleotide == 't':
                print 'A'
```

LINGUAGGIO PYTHON

Esercizio 2

Scrivere un programma in Python che:

- Calcola la somma dei primi 500 numeri naturali (da 0 a 500 escluso)

LINGUAGGIO PYTHON

Esercizio 2

```
n = 0
for i in range(0,500):
        n = n+i
```

Alternativa:

```
sum(range(0,500))
```

LINGUAGGIO PYTHON

Esercizio 3

Scrivere un programma in Python che:

- Data la stringa 'abcdefghi', analizza la stringa carattere per cartere stampandolo a video

  Lettera 1: a
  Lettera 2: b
  ...

- Modificare poi il programma in modo da leggere la stringa da tastiera.

LINGUAGGIO PYTHON

Esercizio 3

```
for i, letter in enumerate('abcdefghi’):
        print 'Lettera %d: %s' % (i+1, letter)
```

Modifica:

```
s = raw_input('Inserisci una stringa: ‘)
for i, letter in enumerate(s):
        print 'Lettera %d: %s' % (i+1, letter)
```

LINGUAGGIO PYTHON

Esercizio 4

Scrivere un programma in Python che:

- stampa la lunghezza delle stringhe fornite dall'utente, finchè l'utente non inserisce la stringa ‘fine'

LINGUAGGIO PYTHON

Esercizio 4

```
while True:
   line = raw_input('Inserisci una stringa: ')
            if line == 'fine':
                        break
            print len(line)
```

LINGUAGGIO PYTHON

Esercizio 5

Scrivere un programma in Python che:

- definisce una funzione che prenda come argomento un nucleotide (A,C,G,T) e restituisce il suo complementare

LINGUAGGIO PYTHON

Esercizio 5

```
def complementare(nucleotide):
   nucleotide = nucleotide.capitalize()
if nucleotide == 'A':
            return 'T'
elif nucleotide == 'C':
            return 'G'
elif nucleotide == 'G':
            return 'C'
elif nucleotide == 'T':
            return 'A'

nucleotide = raw_input('Inserisci un nucleotide (A,C,G,T): ')
print complementare(nucleotide)
```

LINGUAGGIO PYTHON

Esercizio 6

Scrivere un programma in Python che:

- definisce una funzione che prenda come argomento un nucleotide (A,C,G,T) e restituisce il suo complementare
- Definisce una ulteriore funzione filamento_opposto(...) che utilizzi la funzione complementare(...) per ritornare il filamento opposto a quello passato come argomento

Esempio:

```
filamento_opposto('CTAATGT')
 'GATTACA'
```

LINGUAGGIO PYTHON

Esercizio 6

```
def complementare(nucleotide):
                nucleotide = nucleotide.capitalize()
                 if nucleotide == 'A':
                                return 'T'
                 elif nucleotide == 'C':
                                return 'G'
                 elif nucleotide == 'G':
                                return 'C'
                 elif nucleotide == 'T':
                                return 'A'

 def filamento_opposto(filamento):
                opposto = ''
                for nucleotide in filamento:
                                opposto = opposto+complementare(nucleotide)
                return opposto

 nucleotidi = raw_input('Inserisci una sequenza di nucleotidi (A,C,G,T): ')
 print filamento_opposto(nucleotidi))
```

LINGUAGGIO PYTHON

Esercizio 7

Scrivere un programma in Python che:

- dati i due elenchi di numeri sottostanti, crei la matrice dei loro prodotti:

v1: 1,2,3,4,5
v2: 6,7,8,9,10
mat:
1*6 1*7 1*8 ...
2*6 2*7 2*8 ...
…

- Completare il programma con una stampa della matrice riga per riga:
  [6, 7, 8 ...]
  [12, 14, 16 ..]
  …

LINGUAGGIO PYTHON

Esercizio 7

```
v1 = (1,2,3,4,5)
v2 = (6,7,8,9,10)
mat = []
for i, x1 in enumerate(v1):
        mat.append([])
        for x2 in v2:
                mat[i].append(x1*x2)

for row in mat:
        print row
```

LINGUAGGIO PYTHON

Esercizio 8

Scrivere un programma in Python che:

- definisce una funzione square(val,n), che ritorna una matrice quadrata di dimensione n con il valore val in ogni cella

Es.: square (5,3)

5	5	5
5	5	5
5	5	5

LINGUAGGIO PYTHON

Esercizio 8

```
def square(val, n):
        mat = []
        for i in range(0,n):
                row = []
                for j in range(0,n):
                        row.append(val)
        mat.append(row)
        return mat


square(5,3)
```

LINGUAGGIO PYTHON

Esercizio 9

Scrivere un programma in Python che:

- Crea una stringa ripetendo 10,000 volte la stringa 'CGAT'.
- Crea una stringa di 40,000 caratteri scelti a caso dall'alfabeto 'CGAT'.
- Utilizza la funzione compress() del modulo zlib per comprimere le due stringhe.
- Calcolarle lunghezze delle nuove stringhe così ottenute ed i relativi rapporti di compressione

LINGUAGGIO PYTHON

Esercizio 9

```
import random
import zlib

s1 = 'CGAT'*10000
s2 = ''.join([random.choice('CGAT') for i in xrange(10000)])
zs1 = zlib.compress(s1)
zs2 = zlib.compress(s2)
s1l=len(zs1)
s2l=len(zs2)
tc1=len(zs1)/float(s1l)
tc2=len(zs2)/float(s2l)
```

LINGUAGGIO PYTHON

Esercizio 10

Scrivere un programma in Python che:

- definisce una funzione che, ricevendo in ingresso il nome di un file fasta, restituisca una stringa con la sequenza in esso contenuta

```
>gi|251831106|ref|NC_012920.1| Homo sapiens mitochondrion, complete
genome
GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGGTATTTTCGTCTGGGGG
GTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTC
CTGCCTCATCCTATTATTTATCGCACCTACGTTCAATATTACAGGCGAACATACTTACTAAAGTGTGTTA
...
```

LINGUAGGIO PYTHON

Esercizio 10

```
def get_seq(fastafile):
        f = open(fastafile)
        lines = f.readlines()
        f.close()
        lines = [row.strip() for row in lines]
        lines = [row.replace( '\n','') for row in lines if not row.startswith('>')]
        return ''.join(lines)


print get_seq('mitochondrion.fasta')
```


Fine