

# Python: exercises

---

## Esercizi

1. — Costruire una lista di 10 nomi propri e farsi stampare la stringa “Ciao <nome>”
2. — Data la lista composta dai seguenti elementi:  
2 3 4 5 12 13 14 15 0 1 22 23 24 25 32 30 34 35  
scrivere un breve script che, a partire dal terzo elemento, restituisca per ogni elemento la somma dei due elementi precedenti se l'elemento è maggiore di 3 e minore di 20, dei due successivi se è maggiore di 20 e minore di 33, l'elemento stesso altrimenti.
3. — Costruire una lista data dall'unione di due liste:  
la prima contenente 30 elementi che siano tutti numeri pari,  
la seconda contenente 40 elementi che siano tutti numeri dispari.  
Usando il ciclo while, scorrere gli elementi della lista unione delle due liste e farsi stampare “<numero> è pari” finché il numero è pari.
4. — Costruire un dizionario (ad es.: lista della spesa) e stampare le singole coppie chiave/valore.
5. — Costruire una tupla e farsi stampare la coppia indice, elemento servendosi di enumerate.

# Python: exercises

## Esercizi

1. — Costruire una lista di 10 nomi propri e farsi stampare la stringa “Ciao <nome>”

```
>>> my_name_list = ["paolo", "giovanni", "anna", "tom", "gertrude", "anastasia", "giorgia", "luca", "matteo", "marco"]
>>>
>>> for i in range(len(my_name_list)):
...     print("ciao {}".format(my_name_list[i]))
...
ciao paolo
ciao giovanni
ciao anna
ciao tom
ciao gertrude
ciao anastasia
ciao giorgia
ciao luca
ciao matteo
ciao marco
>>>
```

```
>>> for i in my_name_list:
...     print("ciao {}".format(i))
...
ciao paolo
ciao giovanni
ciao anna
ciao tom
ciao gertrude
ciao anastasia
ciao giorgia
ciao luca
ciao matteo
ciao marco
>>>
```



## Esercizi

3. — Costruire una lista data dall'unione di due liste:  
la prima contenente 30 elementi che siano tutti numeri pari,  
la seconda contenente 40 elementi che siano tutti numeri dispari.  
Usando il ciclo while, scorrere gli elementi della lista unione delle due liste  
e farsi stampare “<numero> è pari” finché il numero è pari.

```
>>> lista_pari = [i for i in range(2,62,2)]  
>>> lista_dispari = [i+1 for i in range(2,82,2)]
```

# Python: exercises

## Esercizi

3. — Costruire una lista data dall'unione di due liste:

la prima contenente 30 elementi che siano tutti numeri pari,

la seconda contenente 40 elementi che siano tutti numeri dispari.

Usando il ciclo while, scorrere gli elementi della lista unione delle due liste e farsi stampare “<numero> è pari” finché il numero è pari.

```
>>> lista_pari = [i for i in range(2,62,2)]
>>> lista_dispari = [i+1 for i in range(2,82,2)]
>>> lista_tot = lista_pari + lista_dispari
>>> lista_tot
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81]
>>> j = 0
>>> while lista_tot[j] % 2 == 0:
...     print("{} è pari".format(lista_tot[j]))
...     j = j+1
...
2 è pari
4 è pari
6 è pari
8 è pari
10 è pari
...
56 è pari
58 è pari
60 è pari
>>> print("esco dal ciclo perché {} è il primo numero dispari".format(lista_tot[j]))
esco dal ciclo perché 3 è il primo numero dispari
```

# Python: exercises

## Esercizi

4. — Costruire un dizionario (ad es.: lista della spesa) e stampare le singole coppie chiave/valore.

```
[>>> to_buy = {'apples':10, 'eggs':6, 'lemons':2, 'oranges':8}
[>>> to_buy
{'apples': 10, 'eggs': 6, 'lemons': 2, 'oranges': 8}
[>>> to_buy['apples']
10
[>>> for key in to_buy:
[...     if key == 'apples':
[...         print('I have to buy: {} {}'.format(to_buy[key], key))
[...     else:
[...         print('and {} {}'.format(to_buy[key], key))
[...
I have to buy: 10 apples
and 6 eggs
and 2 lemons
and 8 oranges
```

# Python: exercises

## Esercizi

5. — Costruire una tupla e farsi stampare la coppia indice, elemento servendosi di enumerate.

```
>>> my_tuple = ('a', 'b', [1, 2, 3], 'one', 2.5, 'three')
>>> type(my_tuple)
<class 'tuple'>
>>> for i, element in enumerate(my_tuple):
...     print('Element {} is {}'.format(i, element))
...
Element 0 is a
Element 1 is b
Element 2 is [1, 2, 3]
Element 3 is one
Element 4 is 2.5
Element 5 is three
```