

# **Esercizi lezione 5**

## Esercizio

Abbiamo una lista di numeri:

```
numeri = [4, 10, 50, 100, 128, 71, 46]
```

creare una nuova lista i cui elementi siano gli stessi di numeri incrementati di 10, mediante comprehension.

## Esercizio

Abbiamo una lista di numeri:

```
numeri = [4, 10, 50, 100, 128, 71, 46]
```

creare una nuova lista i cui elementi siano gli stessi di numeri ma raddoppiati, mediante comprehension.

## Esercizio

Abbiamo una lista di nomi:

```
nomi = ["Alex", "Bob", "Cindy", "Dan", "Emma", "Faith",  
"Grace", "Henry"]
```

creare una nuova lista i cui elementi siano le iniziali dei nomi, mediante comprehension.

## Esercizio

Abbiamo una lista di nomi:

```
nomi = ["Alex", "Bob", "Cindy", "Dan", "Emma", "Faith",  
"Grace", "Henry"]
```

creare una nuova lista i cui elementi siano le iniziali dei nomi seguite da punto, mediante comprehension.

## Esercizio

Abbiamo una lista di codici fiscali:

```
lista_cf = ["ABCDEF95G01A123B", "GHIJKL91M02A321C",  
"MNOPQR89S03A456D", "STUVWX95Z04A654E", "XYZABC01D05A789F",  
"DEFGHI95J06A987G"]
```

Creare una lista che contenga (per ogni CF) solo i caratteri relativi al nome, mediante comprehension

## Esercizio

Abbiamo una lista di codici fiscali:

```
lista_cf = ["ABCDEF95G01A123B", "GHIJKL91M02A321C",  
"MNOPQR89S03A456D", "STUVWX95Z04A654E", "XYZABC01D05A789F",  
"DEFGHI95J06A987G"]
```

Creare due liste, una che contenga (per ogni CF) solo i caratteri relativi al nome, e una che contenga (per ogni CF) solo i caratteri relativi al cognome; entrambe mediante comprehension

## Esercizio

Abbiamo una lista di codici fiscali:

```
lista_cf = ["ABCDEF95G01A123B", "GHIJKL91M02A321C",  
"MNOPQR89S03A456D", "STUVWX95Z04A654E", "XYZABC01D05A789F",  
"DEFGHI95J06A987G"]
```

Creare una lista che contenga solo i codici fiscali dei nati nel '95, tramite comprehension



## Esercizio

Abbiamo una lista di codici fiscali:

```
lista_cf = ["ABCDEF95G01A123B", "GHIJKL91M02A321C",  
"MNOPQR89S03A456D", "STUVWX95Z04A654E", "XYZABC01D05A789F",  
"DEFGHI95J06A987G"]
```

Creare una lista che contenga gli ultimi cinque caratteri dei soli codici fiscali di persone nate nel '95, tramite comprehension

## Esercizio

Abbiamo una lista di stringhe di prezzi in dollari, che erroneamente sono stati scritti con il simbolo dell'euro:

```
prezzi = ["100 €", "200 €", "500 €", "10 €", "50 €", "70  
€"]
```

cambiare il simbolo dell'euro (€) in quello del dollaro (\$) per ogni stringa nella lista, usando la comprehension.

## Esercizio

Abbiamo una lista di studenti:

```
studenti = ["Alex", "Bob", "Cindy", "Dan", "Emma", "Faith",  
"Grace", "Henry", "Isabelle", "John"]
```

vogliamo dividere gli studenti in due squadre per un campionato di Uno nel seguente modo: la prima metà per un squadra, e la seconda metà per l'altra.

Creiamo due liste per ogni squadra, e alla fine visualizziamole.

## Esercizio

Abbiamo una lista di studenti:

```
studenti = ["Alex", "Bob", "Cindy", "Dan", "Emma", "Faith",  
"Grace", "Henry", "Isabelle", "John"]
```

vogliamo dividere gli studenti in due squadre per un campionato di Uno nel seguente modo: selezioneremo i nomi in posizione pari per un squadra, e i nomi in posizione dispari per l'altra.

Creiamo due liste per ogni squadra, e alla fine visualizziamole.

## Esercizio

Abbiamo una lista con i guadagni degli ultimi 12 mesi (supponiamo da Gennaio a Dicembre):

```
guadagni = [100, 90, 70, 40, 50, 80, 90, 120, 80, 20, 50, 50]
```

dobbiamo confrontare, stampando tutto a video, il guadagno di ogni mese con la media dei guadagni precedenti

Esempio di un possibile output:

Mese 1: 100 €

Mese 2: 90 € (media prec: 100 €)

Mese 3: 70 € (media prec: 95 €)

## Esercizio

Abbiamo una lista con i guadagni degli ultimi 12 mesi (supponiamo da Gennaio a Dicembre):

```
guadagni = [100, 90, 70, 40, 50, 80, 90, 120, 80, 20, 50,  
50]
```

dobbiamo confrontare, stampando tutto a video, il guadagno di ogni mese con la media dei guadagni precedenti, e specificare nell'output se è maggiore o minore

## Esercizio

Scrivere una funzione che, data una lista di numeri, fornisca in output il maggiore di tutti, senza utilizzare la funzione `max()`

## Esercizio

Scrivere una funzione che, data una lista di numeri, fornisca in output il minimo e il massimo (possiamo usare o meno le funzioni `min()` e `max()`)



## Esercizio

Scrivere una funzione che, data una lista di numeri, fornisca in output i due numeri più grandi

## Esercizio

Scrivere una funzione che in input acquisisce una lista di numeri e un numero  $K$ ; in output, dovrà restituire la media di tutti i numeri nella lista maggiori o uguali a  $K$ ; se non ce ne dovesse essere nessuno, dovrà stampare a schermo un messaggio adeguato

## Esercizio

Scrivere una funzione che, data una lista di numeri, come output stamperà lo stesso numero di asterischi su righe diverse, ottenendo una semplice visualizzazione grafica

Esempio, supponendo di chiamare la funzione `aster()`:

```
numeri = [5, 2, 3, 4]
aster(numeri)
```

Output:

```
*****
**
***
****
```

## Esercizio

Abbiamo un testo, ad esempio:

```
racconto = """Lisetta va a passeggio in campagna; è felice di  
raccogliere i fiori che crescono sulle rive, ai bordi della strada.  
Ha già un bel mazzetto di ranuncoli e margherite."""
```

Creiamo una funzione che prenda il testo in input, e in output ci dia un dizionario che contiene ogni tipo di carattere e quante volte appare.

Esempio, se il testo è

```
testo = """Ciao a tutti!"""
```

l'output sarà:

```
{"C": 1, "i": 2, "a": 2, "o": 1, " ": 2, "t": 3, "u": 1, "(": 1}
```

## Esercizio

Abbiamo un testo, ad esempio:

```
racconto = """Lisetta va a passeggio in campagna; è felice di  
raccogliere i fiori che crescono sulle rive, ai bordi della strada.  
Ha già un bel mazzetto di ranuncoli e margherite."""
```

Creiamo una funzione che prenda il testo in input, e in output ci dia un dizionario che contiene ogni tipo di carattere e quante volte appare, esclusi punteggiatura e spazi.

Esempio, se il testo è

```
testo = """Ciao a tutti!"""
```

l'output sarà:

```
{"C": 1, "i": 2, "a": 2, "o": 1, "t": 3, "u": 1}
```

## Esercizio

Abbiamo un testo, ad esempio:

```
racconto = """Lisetta va a passeggio in campagna; è felice di  
raccogliere i fiori che crescono sulle rive, ai bordi della strada.  
Ha già un bel mazzetto di ranuncoli e margherite."""
```

Creiamo una funzione che prenda il testo in input, e in output ci dia un dizionario che contiene ogni tipo di carattere e quante volte appare, senza fare differenza tra maiuscole e minuscole.

## Esercizio 1/2

Andiamo su

[http://www.datiopen.it/it/opendata/Mappa dei pub circoli locali in Italia](http://www.datiopen.it/it/opendata/Mappa_dei_pub_circoli_locali_in_Italia) e scarichiamo la mappa dei pub, circoli e locali in Italia

Nota: per leggerlo nella funzione `open()` dovremo aggiungere il parametro `encoding="latin1"`, ad esempio:

```
f = open(file_path, "r", encoding="latin1")
```

## Esercizio 2/2

Esaminiamo il dataset:

quanti dati ci sono in totale?

quali sono i metadati?

stampiamo il primo elemento

stampiamo l'ultimo elemento

riusciamo a stampare un elemento a caso?

quali sono gli anni di inserimento presenti?

quante attività ci sono nel quadrato di longitudine 9-10 e latitudine 45-46?

quante attività ci sono nella provincia di Vicenza?

quante enoteche ci sono, e come si chiamano?

quante attività ci sono in Lazio e Abruzzo assieme?





**GRAZIE**  
Epicode