GABRIELE ROMANATO

Python: estrarre elementi casuali da una lista

In questo articolo vedremo come estrarre elementi casuali da una lista in Python fornendo un esempio concreto.

Supponiamo di voler creare un'applicazione che generi un'identità casuale di un utente soffermadoci al solo nome e cognome.

Per farlo abbiamo bisogno di due file testuali contenenti i nomi e i cognomi più frequenti sul territorio italiano. Questi file sono formattati in modo tale che ciascuna riga contenga una sola voce.

Il primo passo da effettuare è leggere il contenuto di tali file restituendo una lista contenente le voci relative.

```
import random
import os

def get_file_data_list(filename=None):
    if filename is None:
        return []
    if not os.path.exists(filename):
        return []
    with open(filename, 'r') as f:
        return f.readlines()
```

La nostra funzione restituirà una lista vuota nel caso in cui non venga passato alcun argomento o se l'argomento passato non sia un file esistente. Quindi il file viene letto e viene restituita la lista delle righe con il metodo `readlines()`.

A questo punto per estrarre valori casuali dalle due liste dobbiamo usare il metodo `random.choice()` prestando attenzione al fatto che la lista passata come argomento non sia vuota. In tal caso, infatti, l'invocazione di tale metodo solleverebbe un errore.

```
def create_person_info():
    firstnames = get_file_data_list('./nomi.txt')
    lastnames = get_file_data_list('./cognomi.txt')
    if len(firstnames) == 0 or len(lastnames) == 0:
        return ''
    firstname = random.choice(firstnames).replace('\n',
    '').capitalize()
    lastname = random.choice(lastnames).replace('\n',
    '').capitalize()
    return f'{firstname} {lastname}'
```

La funzione restituisce una stringa contenente il nome e il cognome formattati eliminando eventuali interruzioni di riga presenti e trasformando la prima lettera in maiuscolo o una stringa vuota nel caso in cui una delle liste non abbia elementi. Volendo possiamo loggare tale eccezione prima di restituire la sopracitata stringa vuota.

Quindi possiamo usare il nostro codice come segue:

```
def main():
    print(create_person_info())

if __name__ == '__main__':
    main()
```