1. Panoramica

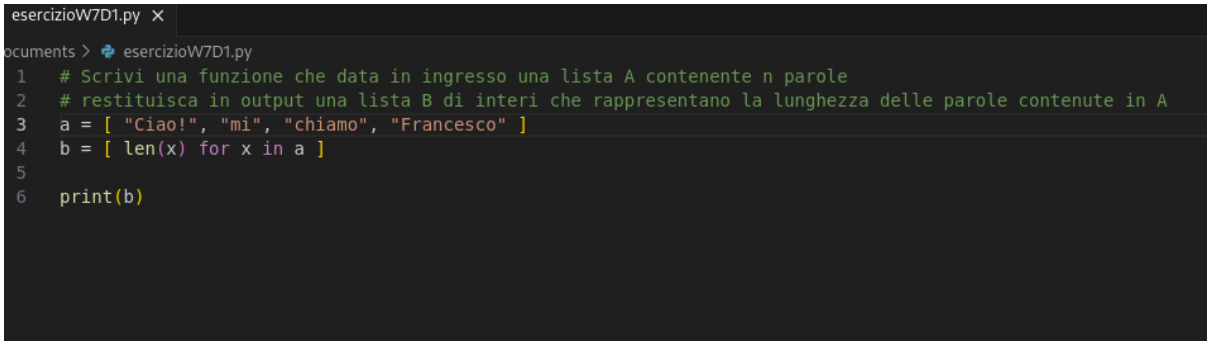
Questo esercizio è stato pensato per mettere in pratica concetti fondamentali di Python, come la gestione delle liste e la creazione di funzioni personalizzate. È diviso in due sezioni: una per analizzare parole, l'altra per generare password.

2. Sezione 1 – Analisi delle parole

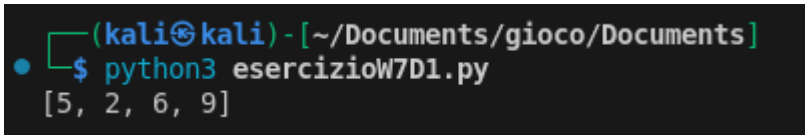
2.1 Scopo
Partendo da una lista di parole, si vuole ottenere una nuova lista che contenga il numero di lettere di ciascuna parola.

2.2 Implementazione
Nel file Python è stata creata la lista: a = ["Ciao!", "mi", "chiamo", "Francesco"]
Poi, ho generato la lista delle lunghezze: b = [len(x) for x in a]

Infine, ho mostrato il risultato con print



2.3 Output
Il terminale ha restituito: [5, 2, 6, 9]
che rappresenta la lunghezza di ogni parola.

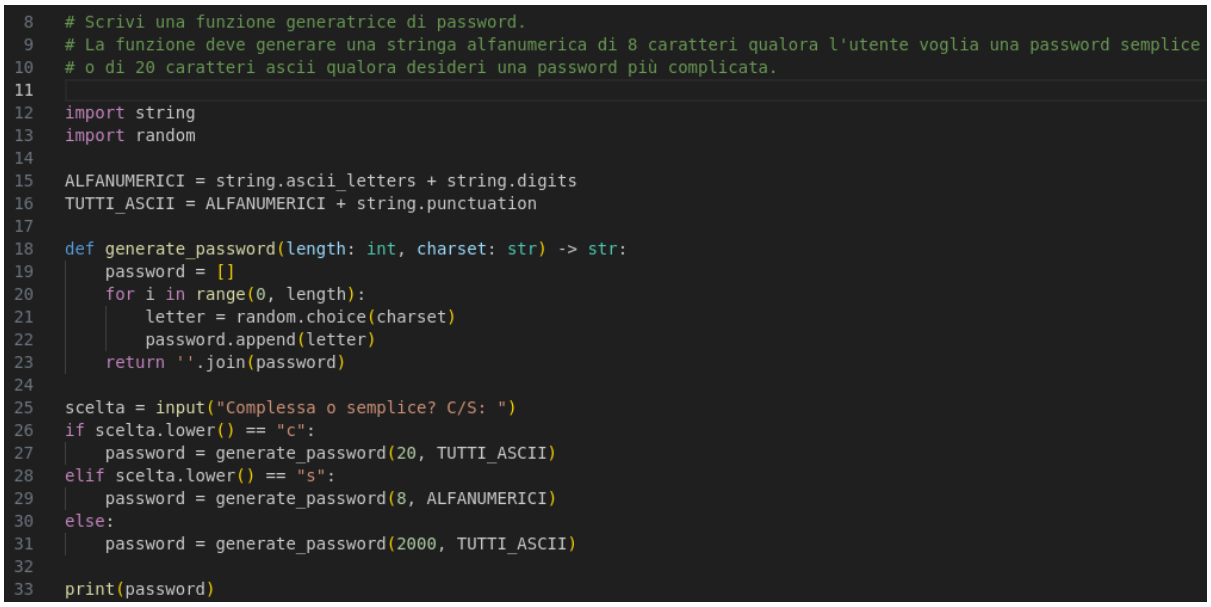


3. Sezione 2 – Creazione di password

3.1 Scopo
Realizzare una funzione che produca password casuali, con due livelli di complessità: base e avanzata.

3.2 Codice
Sono state definite due stringhe di caratteri:
ALFANUMERICI = string.ascii_letters + string.digits
TUTTI_ASCII = ALFANUMERICI + string.punctuation

La funzione “generate_password” costruisce la password scegliendo caratteri casuali



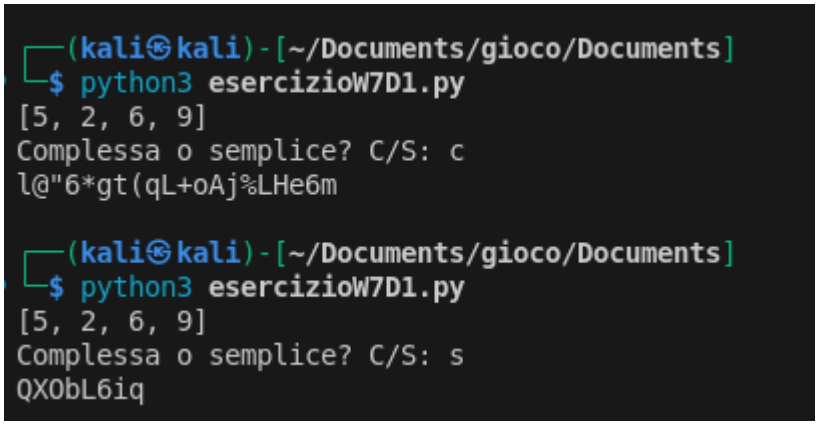
```python
# Scrivi una funzione generatrice di password.
# La funzione deve generare una stringa alfanumerica di 8 caratteri qualora l'utente voglia una password semplice
# o di 20 caratteri ascii qualora desideri una password più complicata.

import string
import random

ALFANUMERICI = string.ascii_letters + string.digits
TUTTI_ASCII = ALFANUMERICI + string.punctuation

def generate_password(length: int, charset: str) -> str:
    password = []
    for i in range(0, length):
        letter = random.choice(charset)
        password.append(letter)
    return ''.join(password)

scelta = input("Complessa o semplice? C/S: ")
if scelta.lower() == "c":
    password = generate_password(20, TUTTI_ASCII)
elif scelta.lower() == "s":
    password = generate_password(8, ALFANUMERICI)
else:
    password = generate_password(2000, TUTTI_ASCII)

print(password)
```

3.3 Input dell’utente
L’utente inserisce la preferenza: scelta = input("Complessa o semplice? C/S: ")

A seconda della risposta:
genera una password di 8 caratteri
ne genera una di 20
qualsiasi altra scelta produce una password lunghissima da 2000 caratteri

3.4 Esecuzione

Il file è stato lanciato da terminale e la password è stata stampata a schermo a seconda della scelta



4. Riflessione finale

Questo esercizio ha aiutato a comprendere meglio come usare le funzioni, come gestire input e output, e come sfruttare librerie Python per creare strumenti utili come un generatore di password.