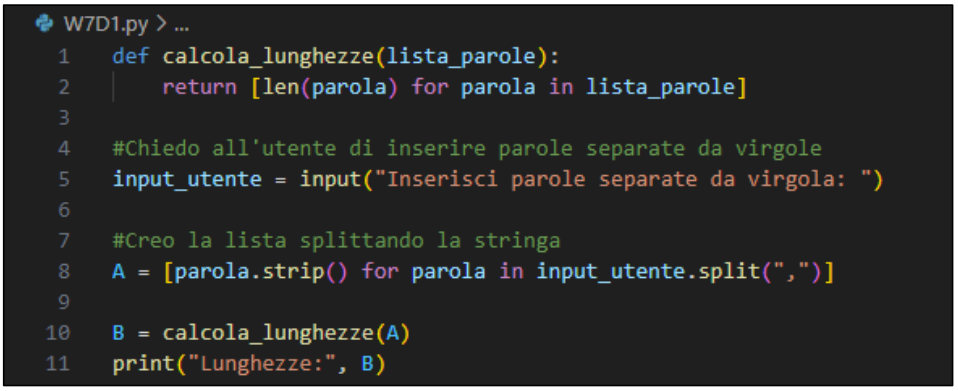
W7D1 - Programmazione per Hacker - Python pt.2

Obiettivo dell'esercitazione

L'esercizio richiede l'implementazione di una funzione in Python che, data una lista di parole (lista A), restituisca una lista di valori interi (lista B) corrispondenti alla lunghezza di ciascuna parola presente in A. Questo tipo di operazione è fondamentale per comprendere la manipolazione di liste e stringhe in Python, nonché l'utilizzo di funzioni e cicli impliciti (list comprehnsion).

Implementazione del Codice

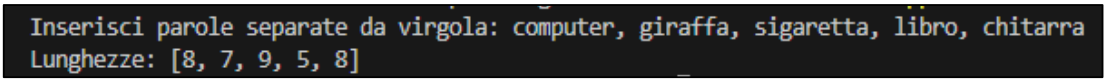


Il codice è strutturato in due parti principali:

1. Definizione della funzione calcola_lunghezze:
   - Parametro: accetta una lista di parole (lista parole);
   - Strumento: utilizza una list comprehension per generare la lista delle lunghezze, applicando la funzione len() a ogni elemento;
   - Efficienza: questo approccio è conciso e ottimizzato, evitando l'uso esplicito di cicli for.

2. Logica di interazione con l'utente:
   - Input: la funzione input() cattura una stringa di parole separate da virgole;
   - Pulizia e divisione: il metodo split(" , ") suddivide la stringa in una lista, mentre strip() elimina spazi bianchi accidentali;
   - Output: la lista risultante (B) viene stampata con un messaggio descrittivo.

Esecuzione del programma



1. L'utente inserisce una stringa di parole separate da virgole: "computer, giraffa, sigaretta, libro, chitarra";
2. Il programma elabora l'input e restituisce la lista delle lunghezze: [8, 7, 9, 5, 8]. Questo output conferma che il codice gestisce correttamente:
    - La divisione della stringa in parole singole;
    - La rimozione degli spazi superflui;
    - Il calcolo accurato della lunghezza di ciascuna parola.

Esercizio Facoltativo

L'esercizio richiede la creazione di un generatore di password in Python con due modalità:

1. Password semplice: stringa alfanumerica di 8 caratteri (lettere e numeri);
2. Password complessa: stringa di 20 caratteri ASCII stampabili, inclusi simboli speciali.

Inoltre, abbiamo incluso un menu interattivo per permettere all'utente di scegliere tra le due opzioni o uscire.

Implementazione del Codice

La funzione genera_password() è il cuore del sistema. Con un semplice parametro complessa, decide se creare una password breve (8 caratteri alfanumerici) o robusta (20 caratteri con simboli). Utilizza intelligentemente i moduli random e string per assicurare casualità e varietà.
L'interfaccia utente, gestita da "menu_generatore_password()", mostra come un menu a scelta multipla possa essere implementato in modo pulito con un ciclo while. Ogni opzione guida l'utente attraverso un percorso chiaro:

1. Scelta 1: password semplice;
2. Scelta 2: password complessa;
3. Scelta 3: uscita dal programma.

Particolarmente interessante è l'uso di "string.ascii_letters + string.digits + string.punctuation" per creare il pool di caratteri complessi, dimostrando come Python renda semplice operazioni che in altri linguaggi sarebbero complesse.

W7D1_facoltativo.py > genera_password

```python
import random
import string

def genera_password(complessa=False):

    if complessa:
        # Genera 20 caratteri ASCII stampabili
        lunghezza = 20
        caratteri = string.ascii_letters + string.digits + string.punctuation
    else:
        # Genera 8 caratteri alfanumerici (lettere + numeri)
        lunghezza = 8
        caratteri = string.ascii_letters + string.digits

    # Combina caratteri casuali
    password = ''.join(random.choice(caratteri) for _ in range(lunghezza))
    return password

def menu_generatore_password():
    #Interfaccia interattiva per generare password

    print("\n=== GENERATORE DI PASSWORD ===")
    print("1. Password semplice (8 caratteri alfanumerici)")
    print("2. Password complessa (20 caratteri ASCII con simboli)")
    print("3. Esci")

    while True:
        scelta = input("\nScegli un'opzione (1-3): ")

        if scelta == '1':
            password = genera_password(complessa=False)
            print(f"\nLa tua password semplice è: {password}")
        elif scelta == '2':
            password = genera_password(complessa=True)
            print(f"\nLa tua password complessa è: {password}")
        elif scelta == '3':
            print("Arrivederci!")
            break
        else:
            print("Scelta non valida. Riprova.")

# Avvia il generatore interattivo
if __name__ == "__main__":
    menu_generatore_password()
```

Esecuzione del programma

```
=== GENERATORE DI PASSWORD ===
1. Password semplice (8 caratteri alfanumerici)
2. Password complessa (20 caratteri ASCII con simboli)
3. Esci

Scegli un'opzione (1-3): 1

La tua password semplice è: noKDaFSm

Scegli un'opzione (1-3): 2

La tua password complessa è: 'yJFRMBn>'%$'7d#HCV3

Scegli un'opzione (1-3): 3
Arrivederci!
```

L'esempio di esecuzione mostra il funzionamento del programma:

1. Menu Iniziale (Opzioni visualizzate correttamente);

2. Scelta 1 (Password semplice): genera una stringa alfanumerica di 8 caratteri;
3. Scelta 2 (Password complessa): genera una stringa di 20 caratteri con simboli;
4. Scelta 3 (Uscita): termina il programma con un messaggio di saluto.