

Report sulla Soluzione: Generatore di Password

Introduzione: L'obiettivo dell'esercizio era sviluppare una funzione in Python che generasse password in base alla preferenza dell'utente: una password semplice di 8 caratteri alfanumerici o una password più complicata di 20 caratteri ASCII.

Procedura: Per raggiungere questo obiettivo, ho creato una funzione denominata **genera_password**. Questa funzione prende in input il livello di complessità desiderato dall'utente (semplice o complicato) e restituisce una password di lunghezza appropriata.

```
main.py  [Icons] Save Run Shell  Cle
1 import random
2 import string
3
4 def genera_password(scomplessita):
5     lunghezza = 8 if scomplessita == 'semplice' else 20
6
7     caratteri_semplici = string.ascii_letters + string.digits
8     caratteri_complessi = string.ascii_letters + string.digits
9         + string.punctuation
10
11     caratteri_usati = caratteri_semplici if scomplessita ==
12         'semplice' else caratteri_complessi
13
14     password_generata = ''.join(random.choice(caratteri_usati)
15         for _ in range(lunghezza))
16     return password_generata
17
18 # Esempio di utilizzo
19 scomplessita_utente = input("Vuoi una password semplice o
20     complicata? (semplice/complicata): ").lower()
```

```
Vuoi una password semplice o complicata? (semplice/complicata):
semplice
La tua nuova password è: IgdVd9aR
>
```

```
main.py  [Icons] Save Run Shell
4 def genera_password(scomplessita):
5     lunghezza = 8 if scomplessita == 'semplice' else 20
6
7     caratteri_semplici = string.ascii_letters + string.digits
8     caratteri_complessi = string.ascii_letters + string.digits
9         + string.punctuation
10
11     caratteri_usati = caratteri_semplici if scomplessita ==
12         'semplice' else caratteri_complessi
13
14     password_generata = ''.join(random.choice(caratteri_usati)
15         for _ in range(lunghezza))
16     return password_generata
17
18 # Esempio di utilizzo
19 scomplessita_utente = input("Vuoi una password semplice o
20     complicata? (semplice/complicata): ").lower()
21 nuova_password = genera_password(scomplessita_utente)
22 print(f"La tua nuova password è: {nuova_password}")
23
```

```
Vuoi una password semplice o complicata? (semplice/complicata):
complicata
La tua nuova password è: WCT9Mv9xw6{XW`zCGm&a
>
```

Dettagli della Soluzione:

1. Ho utilizzato la libreria **random** per la generazione casuale dei caratteri.
2. Ho definito due insiemi di caratteri: uno per le password semplici (**caratteri_semplici**) contenente lettere e numeri, e un altro per le password complesse (**caratteri_complessi**) che include anche caratteri di punteggiatura.
3. In base alla preferenza dell'utente, ho selezionato l'insieme di caratteri appropriato.
4. Ho utilizzato la funzione **random.choice** per selezionare casualmente caratteri dall'insieme scelto e comporre la password della lunghezza desiderata.
5. Infine, ho restituito la password generata.

Esempio di Utilizzo: Nel codice di esempio, ho acquisito la preferenza dell'utente tramite l'input, chiamato la funzione **genera_password** con questa preferenza e stampato la password risultante.

Conclusioni: La soluzione proposta offre un modo flessibile per generare password di diversi livelli di complessità in base alle esigenze dell'utente. Il codice è chiaro, modulare e aderisce alle best practice di Python. L'utente può facilmente ottenere una password sicura con la lunghezza desiderata selezionando il livello di complessità appropriato.