

W7D1-Pratica facoltativo

Nell' esercizio di oggi ci è richiesto di creare programma con python che possa generare password semplici e complesse a richiesta dell' utilizzatore:

Inizio importando due librerie fornite da python, `import random import string` la prima fa scelte casuali la seconda contiene stringhe di numeri lettere e caratteri vari.

```
#importiamo librerie preimpostate di python
✓ import random # serve per generare numeri o fare scelte casuali
    import string # contiene costanti utili come tutte le lettere o i numeri
```

Successivamente definisco la funzione che si occuperà di creare le password :

```
def gen_psw():
    print("\n Benvenuto in Generatore di password\n")
    print("\nPuoi scegliere tra due livelli di complessità\n")
#prepariamo dei set di caratteri che sommiamo per generare delle password
    ascii_chars = string.digits + string.ascii_letters + string.punctuation
    alphanum_chars = string.digits + string.ascii_letters
# creo un ciclo di controllo, questo serve a fare in modo che una volta cr
```

nello specifico:

1. `def` = definisco la funzione e la chiamo `gen_psw`
2. `print` = scrivo a schermo
3. `print` = scrivo a schermo
4. imposto il sistema di scelta dei caratteri della prima password (quella complessa) quindi la chiamo `ascii_char` e gli dico : crea una password con tutti i numeri + tutte le lettere + tutti i simboli
5. faccio lo stesso con la seconda password (quella semplice) la chiamo `alphanum_char` (caratteri alfanumerici) e gli dico : crea una password con tutti i numeri + tutte le lettere

Qui creo un ciclo `while` indendendo al suo interno un ciclo `if- elif-else` :

```
12 ✓     while True:
13         scelta = input("Scegli S se vuoi una password semplice o C se vuoi una password complessa: ").upper() #uso upper per non avere problemi con maiuscole e minuscole
14     #dentro if definisco la psw complessa, definisco la lunghezza in caratteri e do il tipo che ho creato sopra grazie alla variabile tipo
15     if scelta == "C":
16
17         lunghezza = 20
18         tipo = ascii_chars
19         break
20     elif scelta == "S": #dentro elif definisco le psw semplici
21
22         lunghezza = 8
23         tipo = alphanum_chars
24         break
25     else : #uso else per prevenire errori di battitura
26         print("Devi scegliere tra S oppure C")
```

Lo faccio perchè in questo modo l' utilizzatore può scegliere solo tra S e C (minuscole o maiuscole grazie al comando `upper` alla fine di riga 13), se non digita uno dei quattro caratteri il programma riparte all' infinito.

Definisco quindi all'interno di `If` la scelta della password complessa, richiamo il sistema `ascii_chars` e la lunghezza della password espressa in caratteri;

Faccio lo stesso dentro `elif` per la scelta della password semplice richiamando `alphanum_chars` ed infine su `else` imposto l' ipotetico errore dell' utilizzatore.

```
28     psw = "" #preparo il 'contenitore' vuoto della password
29     counter = 0 #definisce che la password parte da 0 caratteri
30
```

Qui preparo un contenitore vuoto e lo chiamo `psw` e definisco che la creazione della password parte da 0 caratteri, con questo contenitore e questo 'start' creo un ciclo `while`

```
31     while counter < lunghezza:
32         char = random.choice(tipo) #pesca un carattere a caso dal set scelto (semplice o complesso)
33         psw += char #lo aggiunge al 'contenitore vuoto'
34         counter += 1 #aggiunge 1 per ogni carattere fino a fermarsi alla lunghezza stabilita (8 o 20)
```

Questo ciclo pesca dalla libreria `random` un carattere la aggiunge al contenitore password e si ferma quando il contatore arriva alla lunghezza definita nei cicli `if-elif`, nell' immagine sotto oltre al `print` della password generata a schermo possiamo vedere la funzione "main" del programma, infatti quest'ultima parte è formata da un ciclo `while` al cui interno troviamo la funzione `creata gen_psw` e l'istruzione di ripetere il programma, a meno che l' utilizzatore non decida di uscire.

Il comando `lower` è il contrario di `upper` quindi accetta anche S invece di s.

```
    print(f"la password generata è: {psw}") #stampa a video la password
    #questa di fatto è la funzione principale richiama la funzione che ho definito in gen_psw e si occupa di far ricominciare
    ~ while True:
        gen_psw()

        ripetere = input("\n Vuoi generare un' altra password? (s/n): ").lower() #lower fa lo stesso lavoro di upper ma al contrario
        ~ if ripetere != "s":
            | print("Arrivederci!")
            | break
    ##### avrei potuto riproporre il concetto di if-elif-else per fare scegliere solo s o n, invece di fatto in questo modo qualche volta mi dava errori
```

Il programma in esecuzione:

```
[kali㉿192kali:~]
$ ./bin/python "/home/kali/Desktop/esercizi_epicode/password.py"
Benvenuto in Generatore di password

Puoi scegliere tra due livelli di complessità

Scegli S se vuoi una password semplice o C se vuoi una password complessa: s
la password generata è: sgovWwao

Vuoi generare un' altra password? (s/n): s

Benvenuto in Generatore di password

Puoi scegliere tra due livelli di complessità

Scegli S se vuoi una password semplice o C se vuoi una password complessa: c
la password generata è: RMVZ.RW.,z?6Ia]f&I5

Vuoi generare un' altra password? (s/n): s

Benvenuto in Generatore di password

Puoi scegliere tra due livelli di complessità

Scegli S se vuoi una password semplice o C se vuoi una password complessa: r
Devi scegliere tra S oppure C
Scegli S se vuoi una password semplice o C se vuoi una password complessa: S
la password generata è: ARALKfFU

Vuoi generare un' altra password? (s/n): n
Arrivederci!
```

Simone Lauria