

## PYTHON(2)

### Funzione generatrice di password

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 qualora desideri una password più complicata

---

#### RISULTATO DEL CODICE

##### per la password da 8 caratteri

Per eseguire il codice che abbiamo fatto basta digitare 'python password.py'  
Per scegliere la password da 8 caratteri, bisogna digitare l'opzione 1

```
(jessica@kali)-[~]  
$ python passwordpro.py  
Generatore di password casuale  
  
Inserisci la tua scelta:  
Inserisci 1 per scegliere la password corta da 8 caratteri alfanumerici.  
Inserisci 2 per scegliere la password lunga da 20 caratteri alfanumerici.  
1  
Hai selezionato 1. La password per te generata è:  
HyxkTSHO
```

Password casuale da 8 caratteri alfanumerici

#### RISULTATO DEL CODICE

##### per la password da 20 caratteri

Per eseguire il codice che abbiamo fatto basta digitare 'python password.py'  
Per scegliere la password da 20 caratteri, bisogna digitare l'opzione 2

```
(jessica@kali)-[~]  
$ python passwordpro.py  
Generatore di password casuale  
  
Inserisci la tua scelta:  
Inserisci 1 per scegliere la password corta da 8 caratteri alfanumerici.  
Inserisci 2 per scegliere la password lunga da 20 caratteri alfanumerici.  
2  
Hai selezionato 2. La password per te generata è:  
lAdEHrUeIHQxStMDHgrZ
```

Password casuale da 20 caratteri alfanumerici

**CODICE DEL FILE CHE ABBIAMO ESEGUITO**  
in questo caso il file l'ho denominato 'passwordpro.py'

```
GNU nano 7.2 passwordpro.py
import random
import string

def get_string(letter_count, digits_count):
    letters = ''.join((random.choice(string.ascii_letters) for i in range(letter_count)))
    digits = ''.join((random.choice(string.digits) for i in range(digits_count)))

    print("Generatore di password casuale\n")

    pwcorta = 8
    pwlunga = 20
    caratteri = string.ascii_letters + string.digits

    pw1 = ''.join((random.choice(string.ascii_letters) for i in range(pwcorta)))
    pw2 = ''.join((random.choice(string.ascii_letters) for i in range(pwlunga)))

    while True:
        x = int(input("Inserisci la tua scelta:\nInserisci 1 per scegliere la password cort>
        if x == 1:
            print("Hai selezionato 1. La password per te generata è:\n", pw1)
            break
        elif x == 2:
            print("Hai selezionato 2. La password per te generata è:\n", pw2)
            break
        else:
            print("\nErrore, digitare nuovamente tra le scelte proposte")
            break
```

**import random:** per generare la password alfanumerica in maniera casuale

**import string:** Il modulo string è un modulo Python integrato che fornisce varie costanti, funzioni e classi relative alle stringhe. Include un insieme di costanti di stringa, come lettere ASCII, cifre e segni di punteggiatura.

**def get\_string(letter\_count, digit\_count)** = definisco la stringa

- **.join(...)** = la funzione Python join() viene utilizzata per unire tutti gli elementi dall'iterabile e creare una stringa e restituirla come output all'utente. Python join() restituisce una nuova stringa che è la concatenazione delle altre stringhe nell'iterazione specificata
- **'for i in range'** = in Python, la "i" in "for i in range" è una variabile utilizzata per memorizzare il valore dell'iterazione corrente di un ciclo. La variabile "i" può avere qualsiasi nome, ma è comunemente usata come scorciatoia per "indice". La funzione "range" in Python crea una sequenza di numeri, su cui poi il ciclo "for" itera.

**string.ascii\_letters:** una stringa contenente tutte le lettere ASCII (sia minuscole che maiuscole).

**string.digits:** una stringa contenente tutte le cifre ASCII (0-9).

**while True:** = è un costrutto ciclico nel linguaggio di programmazione Python che consente di ripetere all'infinito un blocco di codice. Viene spesso utilizzato insieme a un'istruzione break, che consente di uscire dal ciclo in determinate condizioni.