

GABRIELE ROMANATO

Menu

Python: creare stringhe casuali

In Python possiamo creare stringhe casuali sfruttando le caratteristiche del suo modulo core relativo.

Il modulo `string`, come ci riferisce la documentazione, contiene "a collection of string constants". Tra queste costanti troviamo:

- `ascii_lowercase`: caratteri ASCII minuscoli
- `ascii_uppercase`: caratteri ASCII maiuscoli
- `digits`: caratteri numerici da 0 a 9
- `punctuation`: simboli di punteggiatura.

Ciascuno di queste costanti è una stringa. Ad esempio:

```
import string

print(string.digits)

# 0123456789
```

Possiamo in tal senso creare una funzione di utility che, data la lunghezza della stringa di output e data una stringa contenente l'insieme di caratteri da utilizzare, produca una stringa di caratteri casuali.

Come procediamo? Abbiamo la lunghezza finale della stringa di output della nostra funzione, quindi possiamo utilizzare la funzione `range()` per effettuare l'iterazione che produrrà il numero finale di caratteri richiesti. Poichè ad ogni step dell'iterazione dobbiamo estrarre un carattere casuale dalla stringa che contiene l'insieme di caratteri, possiamo utilizzare il metodo `choice()` del modulo `random`. Ciò è possibile perché Python tratta le stringhe come liste di caratteri. A questo punto per generare la stringa finale dobbiamo unire la sequenza di caratteri ottenuti utilizzando il metodo `join()` su una stringa base vuota.

```
import string
import random

def create_random_string(size=4, chars=string.digits):
    return ''.join(random.choice(chars) for _ in range(size))
```

La variabile usata nel loop `for` non è significativa e non verrà usata in seguito, quindi per convenzione usiamo sempre `_` come nome.

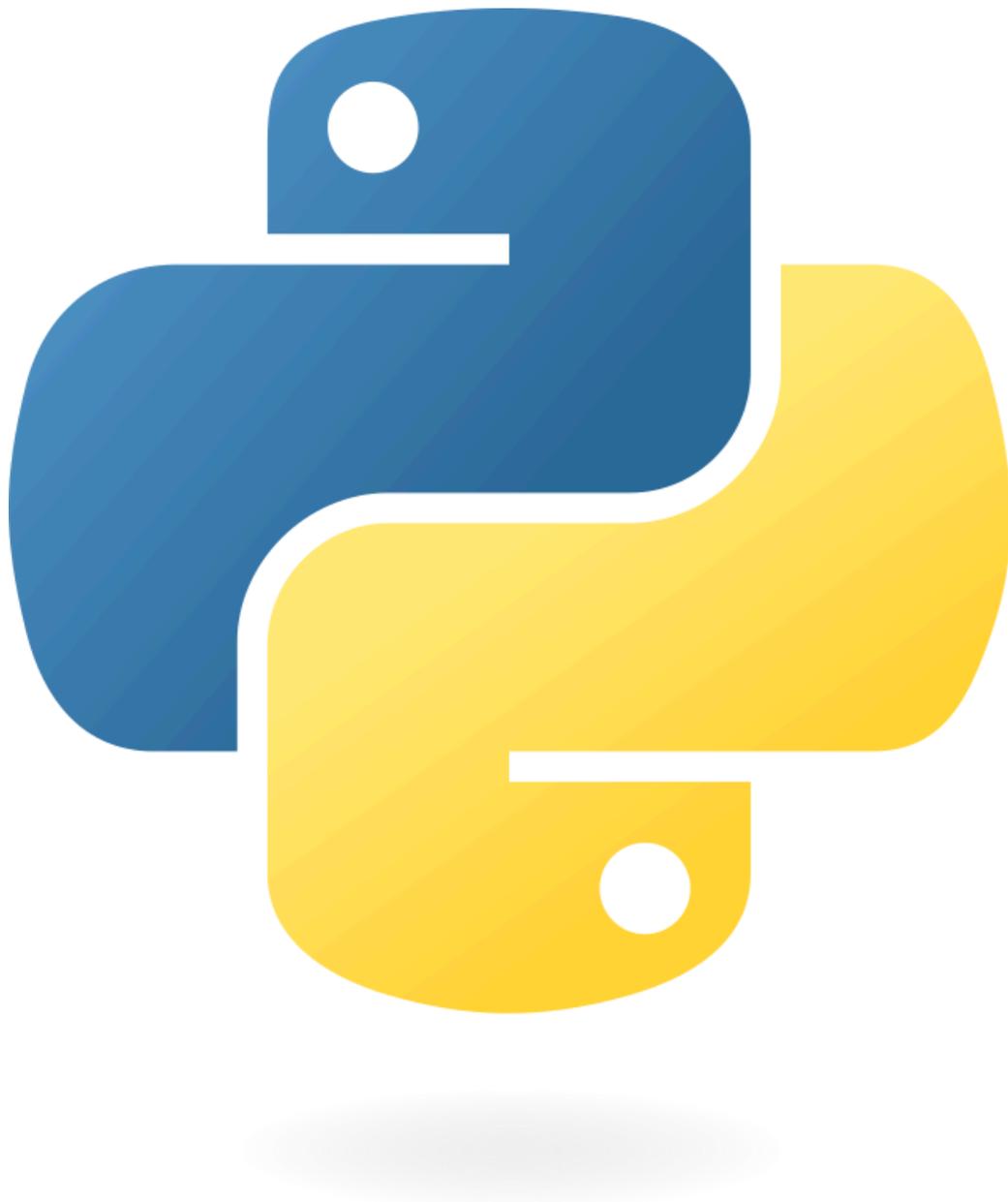
Esempio d'uso predefinito:

```
print(create_random_string()) # 6605
```

Possiamo modificare i parametri passati alla funzione specificando un numero e un tipo di caratteri diversi:

```
all_chars = string.digits + string.ascii_lowercase + string.ascii_uppercase +  
string.punctuation  
  
print(create_random_string(size=20, chars=all_chars))  
  
# `!f9gUyR+J@3o10x&CBW
```

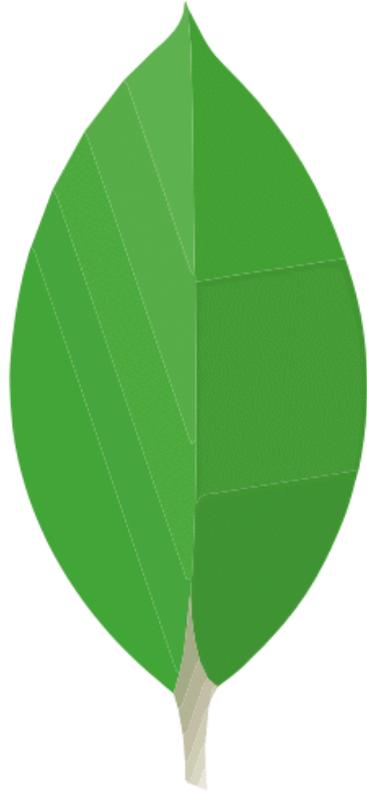
Applicazioni Correlate



-

Python Placeholder Image

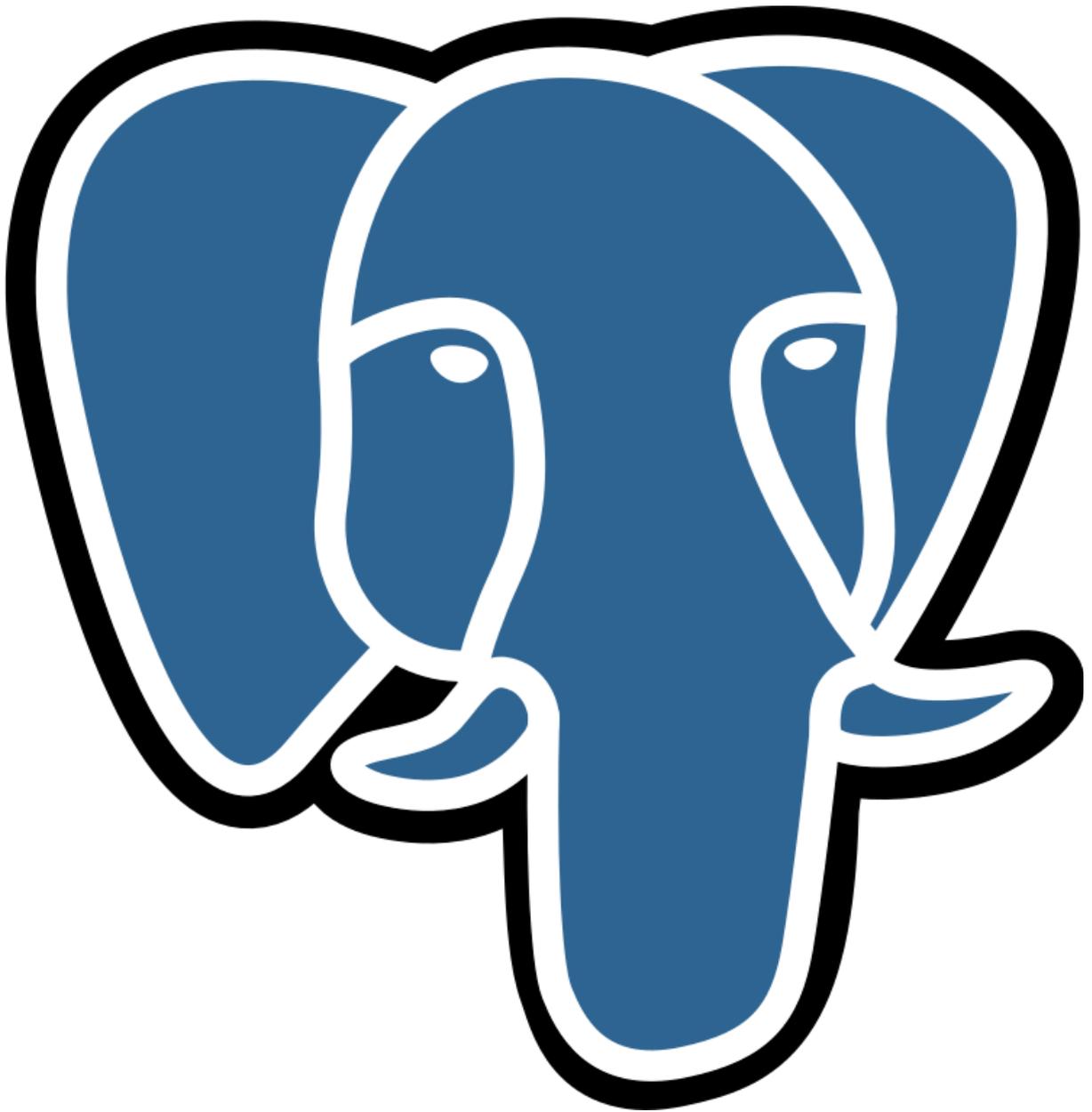
Applicazione sviluppata in Python con Flask per la creazione di immagini segnaposto.
Docker Docker Compose Python Flask JavaScript



-

Python MongoDB App

Applicazione basata su MongoDB e sviluppata in Python e Flask.
Docker Docker Compose Python Flask MongoDB



•

Python PostgreSQL App

Applicazione basata su PostgreSQL con Python e Flask.

Docker Docker Compose Python Flask