Python Program to generate a Random String

A random refers to the collection of data or information that can be available in any order. The **random** module in python is used to generate random strings. The random string is consisting of numbers, characters and punctuation series that can contain any pattern. The random module contains two methods **random.choice()** and **secrets.choice()**, to generate a secure string. Let's understand how to generate a random string using the random.choice() and secrets.choice() method in python.

Using random.choice()

The **random.choice()** function is used in the python string to generate the sequence of characters and digits that can repeat the string in any order.

Create a program to generate a random string using the random.choices() function.

Random_str.py

```
import string
import random # define the random module
S = 10  # number of characters in the string.
# call random.choices() string module to find the string in Uppercase + numeric data.
ran = ''.join(random.choices(string.ascii_uppercase + string.digits, k = S))
print("The randomly generated string is : " + str(ran)) # print the random data
```

Output:

```
C:\Users\AMIT YADAV\C Program>python -u "c:\Users\AMIT YADAV\C Program\Random_str.py"
The randomly generated string is : QLW3J5T2T8
```

Following are the method used in the random module to generate the random string.

Methods	Description
String.ascii_letters	It returns a random string that contains both uppercase and lowercase characters.
String_ascii_uppercase	It is a random string method that only returns a string in uppercase characters.
String.ascii_lowercase	It is a random string method that returns a string only in lowercase characters.
String.digits	It is a random string method that returns a string with numeric characters.
String.punctuation	It is a random strin characters.

Generate a random string of upper case and l

UprLwr.py

```
# write a program to generate the random string in upper and lower case letters.
import random
import string
def Upper_Lower_string(length): # define the function and pass the length as argument
    # Print the string in Lowercase
    result = ''.join((random.choice(string.ascii_lowercase) for x in range(length))) # run loop until the de
    print(" Random string generated in Lowercase: ", result)

    # Print the string in Uppercase
    result1 = ''.join((random.choice(string.ascii_uppercase) for x in range(length))) # run the loop until
    print(" Random string generated in Uppercase: ", result1)

Upper_Lower_string(10) # define the length
```

Output:

```
C:\Users\AMIT YADAV\C Program>python -u "c:\Users\AMIT YADAV\C Program\UprLwr.py"
 Randomly generated string is:  bmkfugrrak
 Randomly generated string is:  AFRLJVFMYL
```

Random String of Specified Characters

Specific.py

```
# create a program to generate the random string of given letters.
import random
import string
def specific_string(length):
    sample_string = 'pqrstuvwxy' # define the sp
    # define the condition for random string
    result = ''.join((random.choice(sample_string)
    print(" Randomly generated string is: ", resul

specific_string(8) # define the length
```

```
specific_string(10)
```

Output:

```
C:\Users\AMIT YADAV\C Program>python -u "c:\Users\AMIT YADAV\C Program\specific.py"
 Random generated string is:  uxwuxsyv
 Random generated string is:  wxyrxwryrq
```

> Note: The random.choice() method is used in the python program to repeat the same characters strings. If we don't want to display repetitive characters, we should use random.sample() function.

Generate a random string without repeating the same characters

WithoutRepeat.py

```
# create a program to generate a string with or without repeating the characters.
import random
import string
print("Use of random.choice() method")
def specific_string(length):

    letters = string.ascii_lowercase # define the l
     # define the condition for random.choice() 
    result = ''.join((random.choice(letters)) for x 
    print(" Random generated string with repetit
```

```
specific_string(8) # define the length
specific_string(10)



print("") # print the space
print("Use of random.sample() method")
def WithoutRepeat(length):
    letters = string.ascii_lowercase # define the specific string
    # define the condition for random.sample() method
    result1 = ''.join((random.sample(letters, length)))
    print(" Random generated string without repetition: ", result1)

WithoutRepeat(8) # define the length
WithoutRepeat(10)
```

Output:

```
C:\Users\AMIT YADAV\C Program>python -u "c:\Users\AMIT YADAV\C Program\Without_Repeat.py"
Use of random.choice() method
 Random generated string with repetition:  pnpptvfl
 Random generated string with repetition:  purkpjkqqh

Use of random.sample() method
 Random generated string without repetition:  oxbuintl
 Random generated string without repetition:  tywioebxcn
```

As we can see in the above output, the random.sample() method returns a string in which all characters are unique and non-repeating. Whereas, the random.choice() method returns a string that may contain repetitive characters. So, we can say that if we want to generate a unique random string use **random.sample**() method.

Generate a random alphanumeric str

For example, suppose we want a randomly ger
and four digits. We need to define these parame

Let's write a program to generate an alphanume
digits.

fixedString.py

```
import random
import string
def random_string(letter_count, digit_count):
    str1 = ''.join((random.choice(string.ascii_letters) for x in range(letter_count)))
    str1 += ''.join((random.choice(string.digits) for x in range(digit_count)))

    sam_list = list(str1) # it converts the string to list.
    random.shuffle(sam_list) # It uses a random.shuffle() function to shuffle the string.
    final_string = ''.join(sam_list)
    return final_string

# define the length of the letter is eight and digits is four
print("Generated random string of first string is:", random_string(8, 4))

# define the length of the letter is seven and digits is five
print("Generated random string of second string is:", random_string(7, 5))
```

Output:

```
C:\Users\AMIT YADAV\C Program>python -u "c:\Users\AMIT YADAV\C Program\fixedString.py"
Generated random string of first string is: aGPk92Cc1Y1R
Generated random string of second string is: vGl8Ty4032Nn
```

Using secrets.choice()

A secrets.choice() method is used to generate a
a cryptographically random string generator th
results simultaneously using secrets.choice() met

Let's write a program to print a secure random string using the secrets.choice method.

Secret_str.py

```
import random
import string
import secrets # import package
num = 10 # define the length of the string
# define the secrets.choice() method and pass the string.ascii_letters + string.digits as an parameters.
res = ''.join(secrets.choice(string.ascii_letters + string.digits) for x in range(num))

# print the Secure string
print("Secure random string is :"+ str(res))
```

Output:

```
C:\Users\AMIT YADAV\C Program>python -u "c:\Users\AMIT YADAV\C Program\Secret_str.py"
Secure random string is :tRzNJnvw4p
```

Use the different method of the random module to generate a safe random string.

Let's write a program to print secure random strings using different methods of secrets.choice().

Secret.py

```
# write a program to display the different random string method using the secrets.choice().
# imports necessary packages
import random
import string
import secrets
num = 10 # define the length of the string
# define the secrets.choice() method and pass t
res = ''.join(secrets.choice(string.ascii_letters + s
# Print the Secure string with the combination
```

```
print("Secure random string is :"+ str(res))

res = ''.join(secrets.choice(string.ascii_letters) for x in range(num))
# Print the Secure string with the combination of ascii letters
print("Secure random string is :"+ str(res))

res = ''.join(secrets.choice(string.ascii_uppercase) for x in range(num))
# Print the Secure string in Uppercase
print("Secure random string is :"+ str(res))

res = ''.join(secrets.choice(string.ascii_lowercase) for x in range(num))
# Print the Secure string in Lowercase
print("Secure random string is :"+ str(res))

res = ''.join(secrets.choice(string.ascii_letters + string.punctuation) for x in range(num))
# Print the Secure string with the combination of letters and punctuation
print("Secure random string is :"+ str(res))

res = ''.join(secrets.choice(string.digits) for x in range(num))
# Print the Secure string using string.digits
print("Secure random string is :"+ str(res))

res = ''.join(secrets.choice(string.ascii_letters + string.digits + string.punctuation) for x in range(num))
# Print the Secure string with the combonation of letters, digits and punctuation
print("Secure random string is :"+ str(res))
```

Output:

```
C:\Users\AMIT YADAV\C Program>python -
Secure random string is :1koFIfbUyy
Secure random string is :NFkQKqlkVS
Secure random string is :SBWQMHBODN
Secure random string is :gognkskcxo
Secure random string is :qZ,g~!sTvh
Secure random string is :4504475720
Secure random string is :mB6]xY^d|;
```

← Prev Next →

For Videos Join Our Youtube Channel: Join Now

Feedback

- Send your Feedback to feedback@javatpoint.com

Help Others, Please Share

Learn Latest Tutorials

Splunk

SPSS

Tumblr

ReactJS

Regex

Reinforcement Learning

R Programming

RxJS

React Native tutorial

React Native

Python Design Patterns

Python Design Patterns

Python Pillow tutorial

Python Pillow

Python Turtle tutorial

Python Turtle

Keras tutorial

Keras

Preparation

Aptitude

Aptitude

Logical Reasoning

Reasoning

Verbal Ability

Verbal Ability

Interview Questions

Interview Questions

Company Interview Questions

Company Questions

Trending Technologies

Artificial Intelligence

Artificial Intelligence

AWS Tutorial

AWS


Hadoop

ReactJS Tutorial

ReactJS

Data Science Tutorial

Data Science

Angular 7 Tutorial

Angular 7

Blockchain Tutorial

Blockchain


Git

Machine Learning Tutorial

Machine Learning

DevOps Tutorial

DevOps

B.Tech / MCA


DBMS

Data Structures tutorial

Data Structures


DAA

Operating System

Operating System

Computer Network tutorial

Computer Network

Compiler Design tutorial

Compiler Design

Computer Organization and Architecture

Computer Organization

Discrete Mathematics Tutorial

Discrete Mathematics

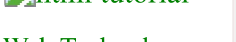

Ethical Hacking

Computer Graphics Tutorial

Computer Graphics

Software Engineering

Software Engineering


Web Technology

Cyber Security tutorial

Cyber Security

Automata Tutorial

Automata


Java

.Net Framework tutorial

.Net

Control Systems tutorial

Control System

Data Mining Tutorial

Data Mining

Data Warehouse Tutorial

Data Warehouse