VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi–590014

A Mini Project Report On

Password Generator Using Python

A study of how to generate a password by applying it to a real-time project which is implemented with the help of python.

Submitted By

HARSHITH N J 1GG20CS008

COURSE COORDINATOR

Dr. Chetan K C

Assistant Professor Dept of CS&E

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT ENGINEERING COLLEGE

B. M. ROAD, RAMANAGARA-562159. 2022-23

INTRODUCTION

Passwords are a means by which a user proves that they are authorized to use a device. It is important that passwords must be long and complex. It should contain at least more than ten characters with a combination of characters such as percent (%), commas(,), and parentheses, as well as lower-case and upper-case alphabets and numbers. Here we will create a random password using Python code.

Example of a weak password :

password123

Example of a strong password :

&gj5hj&*178a1

MODULES NEEDED

- **STRING.ASCII_LETTERS:**

  ASCII is a system that is used to represent characters digitally, every ASCII character has its own unique code. string.ascii_letters is a string constant which contains all the letters in ASCII ranging from A to Z and a to z. Its value is non-locale dependent and it is just a concatenation of ascii_uppercase and ascii_lowercase. Thus it provides us the whole letter set as a string that can be used as desired.

- **STRING.DIGITS:**

  This is a pre-initialized string that contains all the digits in the Arabic numeral system i.e. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. It should be kept in mind that even though these are digits, the type is still a string constant, and all digits are concatenated like this – "0123456789". If we want to access specific numbers then we can do so using slicing.

- **STRING.PUNCTUATION:**

  Apart from letters and digits, python also provides us all the special characters in a pre-initialized string constant. These include various kinds of braces, logical operators, comparison operators, arithmetical operators as well as punctuation marks like commas, inverted commas, periods, exclamations marks, and question marks. The whole string is – !"#$%&'()*+, -./:;<=>?@[\]^_`{|}~

INTRODUCTION TO RANDOM

Random:

- The python random module helps a user to generate pseudo-random numbers.
- Inside the module, there are various functions that just depend on the function "random()".
- This function generates a random float uniformly in the semi-open range [0.0, 1.0) i.e. it generates a decimal number greater than or equal to 0 and strictly less than one.
- Other functions use this number in their own ways.
- These functions can be used for bytes, integers, and sequences. for our task, we are interested in sequences.
- There are functions random. **choices** that take in a sequence as its argument and return a random element from that sequence.

CODE IMPLEMENTATION

First, take the length of the password as input. Then we can display a prompt about the possible list of characters that a user wants to include in the password –

- For including letters in the character set append **string.ascii_letters** in the character list.
- For including letters in the character set append **string.digits** in the character list.
- For including letters in character set append **string.punctuation** in the character list.

Run a for loop till the length of the password and in each iteration choose a random character using **random.choice()** from characterList. Finally, print the password.

PROGARM

```
import string

import random

# Getting password length

length = int(input("Enter password length: "))

print('''Choose character set for password from these :

          1. Digits

          2. Letters

          3. Special characters

          4. Exit''')

characterList = ""

# Getting character set for password

while (True):

    choice = int(input("Pick a number "))

    if (choice == 1):

        # Adding letters to possible characters

        characterList += string.digits

    elif (choice == 2):

        # Adding digits to possible characters

        characterList += string.ascii_letters
```

```
    elif (choice == 3):

        # Adding special characters to possible

        # characters

        characterList += string.punctuation

    elif (choice == 4):

        break

    else:

        print("Please pick a valid option!")

password = []

for i in range(length):
    # Picking a random character from our

    # character list

    randomchar = random.choice(characterList)

    # appending a random character to password

    password.append(randomchar)

# printing password as a string

print("The random password is " + "".join(password))
```

OUTPUT 1: USING ONLY DIGITS

```
C:\Users\Harshith\PycharmProjects\pythonProject3\venv\Scripts\python.exe
"C:\Users\Harshith\PycharmProjects\pythonProject3\Harshith N J\pwd\__init__.py"

Enter password length: 10

Choose character set for password from these :


    1. Digits


    2. Letters


    3. Special characters


    4. Exit

Pick a number 1

Pick a number 4

The random password is 0554734853


Process finished with exit code 0
```

OUTPUT 2: USING ONLY LETTERS

```
Enter password length: 10

Choose character set for password from these :


    1. Digits
```

2. Letters

3. Special characters

4. Exit

Pick a number 2

Pick a number 4

The random password is ApzPMFxVol

Process finished with exit code 0

OUTPUT 3: USING ONLY SPECIAL CHARACTERS

Enter password length: 10

Choose character set for password from these :

1. Digits

2. Letters

3. Special characters

4. Exit

Pick a number 3

Pick a number 4

The random password is }>;[)`#]{>

Process finished with exit code 0

OUTPUT 4: USING ALL THE THREE(DIGITS,LETTERS,SPECIAL CHARACTERS) FUNCTIONS

C:\Users\Harshith\PycharmProjects\pythonProject3\venv\Scripts\python.exe "C:\Users\Harshith\PycharmProjects\pythonProject3\Harshith N J\pwd__init__.py"

Enter password length: 10

Choose character set for password from these :

1. Digits

2. Letters

3. Special characters

4. Exit

Pick a number 1

Pick a number 2

Pick a number 3

Pick a number 4

The random password is >i6<WB0]8'

Process finished with exit code 0

CONCLUSION

Through this project, I've found profound insight into the working of the **password generator** and the function of random password generating process during the sing up to some applications.

Python programming language provided me with the necessary tools to implement the **password generator**, with its rich collection of libraries it simplified the implementation process and helped me concentrate on the actual functioning and understanding of the program easier.

Through this we can understand the process involved password generator. And can interact with the system to obtain the password as required to the application.