Create a Random Password

Generator using Python

A Course Based Project Submitted in Partial Fulfilment of the Requirements for the Award of the degree of

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

(CYBER SECURITY)



Under the Guidance of

Dr. E.Lalitha

(Assistant prof. Dept of CSE-CYS,DS&(AI&DS))

Submitted by:

21071A6235 – Manem Sai Surya Lochana 21071A6236 – Maya Varsha

CERTIFICATE

This is to Certify that Manem Sai Surya Lochana (21071A6235), Maya Varsha (21071A6236) of CSE-CYS Department of VNRVJIET have successfully completed their project work entitled “***Create a Random Password using Python***” in partial fulfilment of the requirements for the award of the Bachelor of Technology degree during the Academic year 2022-2023.

Project Guide: Head of Department:

Dr. E.LALITHA Dr. M. RAJA SEKHAR

Assistant prof.& Internal Prof. and Head

Dept. of CSE-CYS, DS &(AI&DS) Dept. of CSECYS,DS&(AI&DS)

VNRVJIET VNRVJIET

ACKNOWLEDGEMNET

Behind every achievement lies the heartfelt gratitude to those who activated in completing the project. To them we lay the words of gratitude within us.

We are indebted to our venerable principal Dr. C.D. NAIDU for this inflicting devotion, which lead us to complete this project. The support, encouragement given by him and his motivation led us to complete the project.

We express our sincere thanks to internal guide Dr. E.LALITHA and also Head of the Department Dr. M. RAJA SHEKHAR for having provided us with a lot of facilities to undertake the project work and guide us to complete the project.

We take the opportunity to express thanks to our faculty of the Dept. of COMPUTER SCIENCE AND ENGINEERING-CYBER SECURITY and remaining members of our college VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY who extended their valuable support in helping us to complete the project in time.

Maya Varsha(21071A6236)

Manem Sai Surya Lochana(21071A6235)

ABSTARCT

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.

Python is a versatile programming language that you can use across applications in automation, web development, data analysis, and more.

You can use Python to automate real-world tasks such as monitoring websites, sending emails, and generating passwords. In this tutorial, you’ll learn how to create a secure and random password generator in Python.

## Why Create a Random Password Generator ?

A good and strong password is one of the most essential things in today's world where our data and security is our prime concern. People usually set common passwords that can be easily used by hackers to obtain our data.

A better way to learn any programming language is by learning by doing. Specifically developing a good starter project like the random password generator in Python helps brush up on the fundamentals. This project also helps us to understand the usage of various built-in and external modules of the Python programming language.

## 3 Reasons Why You Should Code a Password Generator in Python

## 1. Automate the password generation process

On a typical day in your life, you’ll visit numerous websites – from e-commerce websites to learning and coding platforms. Most of the websites require you to log in with credentials. This is a repetitive task that you can automate using a Python script.

### 2. Generate Secure Passwords

Coming up with strong passwords that meet security constraints can be challenging; trivial passwords that are not strong are susceptible to security attacks.

You can use Python to programmatically generate **secure** passwords. This lets you run the script to generate passwords as needed—without worrying about the password being secure.

### 3. Add a Practical Project to Your Portfolio

If you’re a beginner Python programmer, you should build projects to hone your skills and showcase your programming expertise through a portfolio.

In addition to a simple Python script, you can customize it further, and publish it as a PyPI package. You can also add a graphical user interface using libraries such as Tkinter, and much more!

This way, you can code your own password generator—to securely generate and manage your passwords—across the various sites.

**Modules needed:**

For accessing string constants, the ones we would need are :

[**string.ascii\_letters**](https://www.geeksforgeeks.org/python-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**](https://www.geeksforgeeks.org/python-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**:

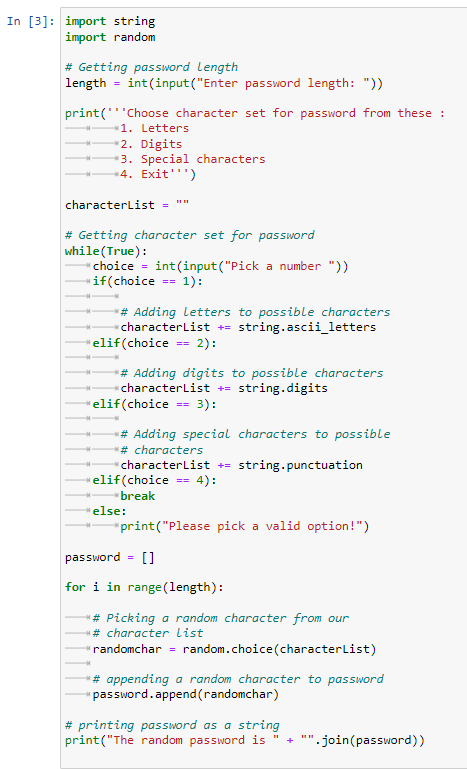
Apart from letters and digits, python also provides us all the special characters in a pre-initialized string constant. These include various kinds of braces, logical operators, comparison operators, arithmetical operators as well as punctuation marks like commas, inverted commas, periods, exclamations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

**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:****

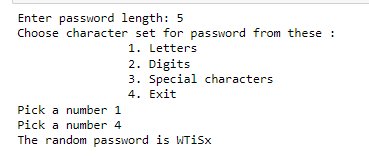
First, take the length of the password as input. Then we can display a prompt about the possible list of characters that a user wants to include in the password –

* 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.

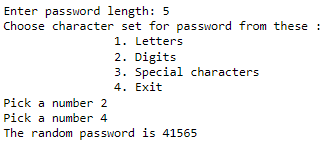


Output:

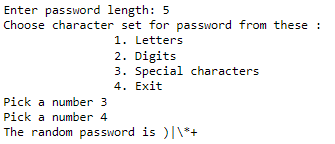
1. If the choice is letters:



2. If the choice is digits:



3. If the choice is special characters:



**CONCLUSION:**

* The random password generator in Python is a program that will generate strong random passwords of the specified length using alphabets, numbers, and symbols.
* We first create a string or list consisting of all the alphabets, numbers, and symbols.
* We will use a loop ranging between 0 and the length of the password and in every iteration we will randomly choose a letter from the set of letters and store them into a resultant password.
* We can also generate multiple random passwords by using an infinite loop that will take the length of the required password and generate the password.
* The **choice()** method selects a random value (or element) from the provided sequence. Now this sequence can be any sequential data type like strings, lists, tuples, or any other sequence of numbers or objects.
* The **sample()** method is used to obtain a sequence of randomly selected values from the provided input sequence. The sample() method return a list of unique choices.
* We can use the sample() method in place of the choice() method to generate a stronger password having no repetition of digits.
* We can use the **Tkinter** module which is an external module to create a GUI for the random password generator.