Python Random Password Generator

By Conner Maris

Dec. 2023

Executive Summary:

The Python Random Password Generator is a simple, yet effective Python script designed to enhance security by creating strong and unpredictable passwords. Developed using Python's **random** and **string** modules, the program generates passwords of varying lengths, incorporating a mix of uppercase and lowercase letters, digits, and punctuation symbols.

Import Modules:

Before we start defining functions, it is important to import the necessary modules to make this program work.

The **random** module helps a user by generating pseudo-random numbers. In this program we will use **random.choice** to choose an option from a predefined **alphabet**.

The **string** module is used to access string constants. In this case we will be using **string.ascii_letters**, **string.digits**, and **string.punctuation**. These will help us create an **alphabet** from which the **random.choice** method will choose from.

```
1   import random
2   import string
```

Define Function:

Now that the proper modules are included, we can begin defining the function that will generate our passwords.

The function will be called **generate_password** and it will take an optional argument, **length**, which is set to 10 by default.

```
4   def generate_password(length: int = 10):
```

Create Alphabet:

Within the **generate_password** function we will need to create a string that will hold all the characters that our function can use to create our password.

The **alphabet** variable will contain everything from **string.ascii_letters**, **string.digits**, and **string.punctuation** all concatenated together.

```
4   def generate_password(length: int = 10):
5       alphabet = string.ascii_letters + string.digits + string.punctuation
```

Generate Password:

Now that the **alphabet** has been created, the **password** can be generated.

The **password** variable is created and given an empty placeholder. The **random.choice** method is used with a **for** loop so that the **random.choice** method will choose a character from **alphabet** the number of times we specified. The **join** method is used to join all the elements into one string.

```
4   def generate_password(length: int = 10):
5       alphabet = string.ascii_letters + string.digits + string.punctuation
6       password = ''.join(random.choice(alphabet) for i in range(length))
```

Return Password:

The last part of the **generate_password** function returns the password that we just generated. Below is the entire function we just created.

```
4   def generate_password(length: int = 10):
5       alphabet = string.ascii_letters + string.digits + string.punctuation
6       password = ''.join(random.choice(alphabet) for i in range(length))
7       return password
```

Call Function:

The **generate_password** function has all the parts we need, but now the script needs to call the function.

We will call the function and assign it to a variable called **password**. For now, the function will only be called with the default **length** of 10.

```
9   password = generate_password()
```

Print Password:

Now that the function has done its job and the result is in the **password** variable, but we also need to **print** that variable.

We will use an **f-string** so that an expression can be embedded into our string literal. This gives us an easy way to include the **password** variable in the text that is output to the terminal.

```
10    print(f"Generated password: {password}")
```

Finished Script:

Here's our finished script! Right now, our function only takes the default length of 10, but the script can easily be modified to accept user input that will change the length of the generated password.

```
1   import random
2   import string
3
4   def generate_password(length: int = 10):
5       alphabet = string.ascii_letters + string.digits + string.punctuation
6       password = ''.join(random.choice(alphabet) for i in range(length))
7       return password
8
9   password = generate_password()
10  print(f"Generated password: {password}")
```