Password Generator Project: A Journey of Learning and Code

In this blog post, we'll explore the process of creating a password generator project using Python. This project aims to generate strong, unique passwords by combining letters, numbers, and symbols.

Learning Process

To achieve this goal, I had to learn and apply the following concepts:

Python basics: Understanding the syntax, data types, and control structures in Python.

Random module: Learning how to use the random module to generate random numbers and choices.

Lists: Understanding how to create, manipulate, and shuffle lists in Python.

String concatenation: Learning how to concatenate strings to form the final password.

The Code

Here's the code for the password generator project:

import random

letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

numbers = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']

symbols = ['!', '#', '$', '%', '&', '(', ')', '\*', '+']

nr\_letters = random.randint(8, 10)

nr\_symbols = random.randint(2, 4)

nr\_numbers = random.randint(2, 4)

password\_list = []

for char in range(nr\_letters):

password\_list.append(random.choice(letters))

for char in range(nr\_symbols):

password\_list += random.choice(symbols)

for char in range(nr\_numbers):

password\_list += random.choice(numbers)

random.shuffle(password\_list)

password = ""

for char in password\_list:

password += char

print(f"Your password is: {password}")

Explanation

We import the random module to generate random numbers and choices.

We define lists of letters, numbers, and symbols to use in the password.

We generate random numbers for the number of letters, symbols, and numbers to include in the password.

We create an empty list password\_list to store the password characters.

We use for loops to append random choices from the letters, symbols, and numbers lists to password\_list.

We shuffle password\_list to ensure randomness.

We concatenate the characters in password\_list to form the final password.

We print the generated password.

Conclusion

Creating this password generator project helped me learn and apply various Python concepts. By following this code and explanation, you can generate strong, unique passwords for your own use. Feel free to modify and improve the code to suit your needs!

GitHub Repository

You can find the code for this project in my GitHub repository: [insert link]