This Project is Created By Jyotirmay Chowdhury.

https://jyotirmaychowdhury.pages.dev/

```
import string
import random

characters = list(string.ascii_letters + string.digits + " !@#$%^&*()")

def generate_password():
    password_length = int(input("How long would you like your password
to be? "))

    random.shuffle(characters)

    password = []

    for x in range(password_length):
        password.append(random.choice(characters))

    random.shuffle(password)

    password = "" .join(password)
    print(password)

option = input("Do you want to generate a password? (Yes/No): ")

if option == "Yes":
    generate_password()
elif option == "No":
    print("Program ended")
    quit()
else:
    print("Invalid input, please input Yes or No")
    quit()
```

This Project is Created By Jyotirmay Chowdhury.

https://jyotirmaychowdhury.pages.dev/

This Python code generates a random password based on user preferences and character choices.

Here's a step-by-step explanation of what each part of the code does:

1. `import string` and `import random`: These lines import the `string` module, which provides a collection of characters, and the `random` module, which is used to generate random values.

2. `characters`: This variable is a list that contains a combination of characters, including uppercase and lowercase letters, digits, and special symbols like "!@#$%^&*()".

3. `def generate_password():`: This line defines a function named `generate_password` that will generate a random password.

4. Inside the `generate_password` function:

    ○ `password_length = int(input("How long would you like your password to be? "))`: This line prompts the user to enter the desired length for the password and converts the input to an integer, storing it in the `password_length` variable.

    ○ `random.shuffle(characters)`: This shuffles (randomly reorders) the characters in the `characters` list. This step ensures that the password will contain random characters.

    ○ `password = []`: This initializes an empty list called `password` to store the characters of the password.

    ○ A `for` loop (`for x in range(password_length)`) is used to generate the password:

        ■ `password.append(random.choice(characters))`: In each iteration, a random character is chosen from the `characters` list and added to the `password` list.

    ○ `random.shuffle(password)`: After all characters are selected, the `password` list is shuffled again to mix up the characters further.

  - `password = "".join(password)`: The characters in the `password` list are joined together as a single string, creating the final password.

  - `print(password)`: The generated password is printed to the screen.

5. `option = input("Do you want to generate a password? (Yes/No): ")`: This line prompts the user to choose whether they want to generate a password by typing "Yes" or "No" and stores their response in the `option` variable.

6. A conditional statement is used to determine what action to take based on the user's input:

   - If the user inputs "Yes":
     - `generate_password()`: The `generate_password` function is called to generate a password.
   - If the user inputs "No":
     - `"Program ended"` is printed, and the `quit()` function is called to exit the program.
   - If the user inputs anything other than "Yes" or "No":
     - `"Invalid input, please input Yes or No"` is printed, and the `quit()` function is called to exit the program.

In summary, this code allows the user to generate a random password of their desired length and character choices. It asks the user for input, generates the password, and prints it to the screen. The user can choose whether or not to generate a password, and the program handles invalid inputs.