How does the provided Python script use the forex_python library to perform currency conversion, and what mechanisms are in place to handle potential errors during the conversion process?

In [12]:

```
pip install forex-python
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: forex-python in c:\users\nishi\appdata\roaming\python\python310\site-packages (1.8)
Requirement already satisfied: requests in c:\users\nishi\appdata\roaming\python\python310\site-packages (from forex-python) (2.31.0)
Requirement already satisfied: simplejson in c:\users\nishi\appdata\roaming\python\python310\site-packages (from forex-python) (3.19.2)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests->forex-python) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests->forex-python) (2022.12.7)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests->forex-python) (2.0.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests->forex-python) (1.26.14)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:

```python
from forex_python.converter import CurrencyRates

def currency_converter(amount, from_currency, to_currency):
    c = CurrencyRates()

    try:
        # Convert the amount from the source currency to the target currency
        converted_amount = c.convert(from_currency, to_currency, amount)

        # Display the result
        print(f"{amount} {from_currency} is equal to {converted_amount:.2f} {to_currency}")
    except:
        print("Error: Invalid currency code or conversion not supported.")

if __name__ == "__main__":
    # Get user input
    amount = float(input("Enter the amount to convert: "))
    from_currency = input("Enter the source currency code (e.g., USD): ").upper()
    to_currency = input("Enter the target currency code (e.g., EUR): ").upper()

    # Perform the conversion
    currency_converter(amount, from_currency, to_currency)
```

```
Enter the amount to convert: 100
Enter the source currency code (e.g., USD): USD
Enter the target currency code (e.g., EUR): EUR
100.0 USD is equal to 93.72 EUR
```

How does the program handle user input errors and what security measures could be incorporated to enhance password security in the random password generator project?

In [6]:

```
pip install requests
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: requests in c:\users\nishi\appdata\roaming\python\python310\site-packages (2.31.0)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests) (1.26.14)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests) (2022.12.7)
Note: you may need to restart the kernel to use updated packages.
```

In [11]:

```python
import random
import string

def generate_password(length, include_numbers=False, include_special_chars=False):
    characters = string.ascii_letters
    if include_numbers:
        characters += string.digits
    if include_special_chars:
        characters += string.punctuation

    if length < 1:
        print("Password length should be at least 1.")
        return

    password = ''.join(random.choice(characters) for _ in range(length))
    return password

def main():
    print("Random Password Generator")

    length = int(input("Enter the length of the password: "))
    include_numbers = input("Include numbers? (y/n): ").lower() == 'y'
    include_special_chars = input("Include special characters? (y/n): ").lower() == 'y'

    password = generate_password(length, include_numbers, include_special_chars)

    if password:
        print("Generated Password:", password)

if __name__ == "__main__":
    main()
```

```
Random Password Generator
Enter the length of the password: 89
Include numbers? (y/n): n
Include special characters? (y/n): n
Generated Password: EwPArComczKqAHaqkztqfFwVrciiiwPkaDYlZYFpQvSfKpwDDlkdwoUvDPFxiWXZFjmPFCyLTMLnsGSbNOWDonHlC
```

In [ ]:

In [ ]: