**CURRENCY CONVERTER USING PYTHON**

MOTION CUTS INTERNSHIP

PYTHON PROGRAMMING INTERNSHIP

WEEK 2

SUBMITTED BY:

BANUPRAKASH

COLLEGE NAME: DAYANANDA SAGAR COLLEGE OF ENGINEERING

CONTENTS

* 1. INTRODUCTION
  2. KEY COMPONENTS
  3. ALGORITHM
  4. PROGRAM
  5. APPLICATIONS
  6. ADVANTAGES
  7. CONCLUSION

**INTRODUCTION**

A currency converter is a program that allows you to convert one currency into another, based on real-time exchange rates. Currency converters are useful for travelers, businesses engaged in international trade, and anyone dealing with currencies from different countries.

In Python, you can build a currency converter by accessing exchange rate data through APIs provided by financial institutions or currency exchange data providers. These APIs offer real-time or historical exchange rate information, which you can use to convert between different currencies.

* API Integration: Choose a currency exchange rate API to fetch real-time exchange rate data. Popular choices include Open Exchange Rates, ExchangeRates API, or Forex Python library. You may need to sign up for an API key to access the data.
* User Input: Your program should prompt the user to input the following information:
* The amount to convert
* The source currency (e.g., USD, EUR)
* The target currency (e.g., GBP, JPY)
* API Requests: Use Python's requests library to send HTTP requests to the chosen API, providing the source and target currency codes. Retrieve the exchange rate for the conversion.
* Currency Conversion: Multiply the amount to convert by the exchange rate to calculate the converted amount in the target currency.
* Display Result: Present the user with the converted amount and currency.
* Error Handling: Implement error handling to account for cases where the API request fails or the provided currency codes are invalid.

**ALGORITHM :**

1. Import the necessary library, requests, for making HTTP requests to an external API.

2. Define a constant variable API\_KEY with your Open Exchange Rates API key. You need to sign up and replace 'YOUR\_API\_KEY' with your actual API key.

3. Create a function get\_latest\_exchange\_rates(base\_currency):

Construct the API URL to get the latest exchange rates for a given base\_currency.

Send a GET request to the API using the constructed URL and the API key as a query parameter.

Parse the JSON response and extract the exchange rate data.

Return the exchange rate data in the form of a dictionary.

4. Create a function convert\_currency(amount, from\_currency, to\_currency):

Call the get\_latest\_exchange\_rates function to get the latest exchange rates for the from\_currency.

Check if the to\_currency is present in the rates dictionary obtained.

If the conversion is possible, calculate the converted amount using the exchange rate and return it.

If the conversion is not possible (e.g., if the to\_currency is not in the rates), return None.

5. In the main program:

Start an infinite loop to keep the program running until the user decides to exit.

Prompt the user for input:

Enter the amount to convert (numeric value).

Enter the source currency (e.g., USD) and target currency (e.g., EUR).

Call the convert\_currency function to perform the conversion.

Display the converted amount if the conversion is successful.

Handle exceptions for invalid input (e.g., non-numeric input) and allow the user to exit using a keyboard interrupt (Ctrl+C).

PROGRAM :

import requests

# Replace 'YOUR\_API\_KEY' with your actual API key from Open Exchange Rates

API\_KEY = 'YOUR\_API\_KEY'

# Function to get the latest exchange rates

def get\_latest\_exchange\_rates(base\_currency):

url = f"https://open.er-api.com/v6/latest/{base\_currency}.json"

params = {

"apikey": API\_KEY }

response = requests.get(url, params=params)

data = response.json()

return data["rates"]

# Function to convert currency

def convert\_currency(amount, from\_currency, to\_currency):

rates = get\_latest\_exchange\_rates(from\_currency)

if to\_currency in rates:

return amount \* rates[to\_currency]

else:

return None

# Main program

if \_\_name\_\_ == "\_\_main\_\_":

while True:

try:

amount = float(input("Enter the amount to convert: "))

from\_currency = input("Enter the source currency (e.g., USD): ").upper()

to\_currency = input("Enter the target currency (e.g., EUR): ").upper()

converted\_amount = convert\_currency(amount, from\_currency, to\_currency)

if converted\_amount is not None:

print(f"{amount} {from\_currency} is equal to {converted\_amount} {to\_currency}")

else:

print("Currency conversion not available for the specified currencies.")

except ValueError:

print("Invalid input. Please enter a valid numeric amount.")

except KeyboardInterrupt:

print("Conversion tool exited.")

break

**APPLICATIONS**

1. Travel Planning: Travelers can use the program to convert their home currency to the local currency of the destination they are visiting. This helps them budget for their trip and make informed financial decisions.
2. Online Shopping: When shopping online from international websites, customers can use the program to convert prices from the currency displayed on the website to their own currency. This allows them to understand the actual cost of the items they want to purchase.
3. Financial Analysis: Analysts and investors can use the program to convert financial data in different currencies to a common currency for comparative analysis, such as evaluating the performance of global stocks, bonds, or commodities.
4. Currency Exchange Services: Currency exchange services, such as banks and currency exchange kiosks, can use such tools to provide accurate and up-to-date currency exchange rates to customers looking to convert physical cash.
5. Cryptocurrency Transactions: Cryptocurrency users can convert their cryptocurrencies into fiat currencies by using this program to determine the equivalent value of their digital assets.
6. Education: Educational institutions can use the program to teach students about currency conversion, exchange rates, and global economics, allowing students to gain practical knowledge and skills.
7. Budgeting and Personal Finance: Individuals can use the program for personal budgeting and financial planning, helping them keep track of their expenses and income across different currencies, especially in the case of frequent international travelers
8. Travel Expense Tracking: Business travelers can track their expenses in various currencies during trips and easily convert these expenses into a common reporting currency for reimbursement purposes.
9. Tourism and Hospitality: Hotels, restaurants, and tourism-related businesses can offer currency conversion services to international visitors to enhance the customer experience.

**ADVANTAGES**

1. Currency Conversion Accuracy: The program provides accurate real-time exchange rates, ensuring that users can convert currency with precision, minimizing potential errors.
2. Real-Time Data: It relies on real-time exchange rate data from sources like the Open Exchange Rates API, which means users can access the most up-to-date conversion rates.
3. Customization: The program can be customized and extended with additional features, such as historical exchange rate tracking, multiple language support, and graphical user interfaces for enhanced user experience.
4. Versatility: It can be used in various scenarios, from personal finance to international business, catering to a wide range of users and applications.
5. Budgeting and Financial Planning: Users can effectively plan and manage their budgets, especially when dealing with multiple currencies due to international travel or financial investments.
6. Cross-Border Trading: Traders and investors in the forex and cryptocurrency markets can use the program to calculate profits and losses, making trading decisions based on accurate conversion rates.
7. Enhanced Travel Experience: Travelers can have a more enjoyable and stress-free trip by easily converting their home currency to the local currency of their destination.
8. Currency Exchange Services: Currency exchange providers can use the program to offer accurate and reliable currency exchange services to customers, enhancing their reputation and trustworthiness.

**CONCLUSION**

In conclusion, the Currency Converter project using Python has successfully demonstrated the power of programming in simplifying real-world financial tasks. This project not only showcases the versatility of the Python language but also serves as a practical tool for individuals and businesses to quickly and accurately convert currencies. By leveraging APIs, data parsing, and user-friendly interfaces, we have created a valuable resource that streamlines the process of currency conversion, making it more accessible and efficient for users around the world. Additionally, this project offers opportunities for further enhancements and customization, such as adding more currencies or integrating with additional financial services, making it a valuable foundation for future development.