Currency Converter

November 13th, 2019.

University of St. Gallen

Skills: Programming – introduction level

Philip Reiche: 19-603-737

Content.

- Introduction.
- Main aspects of the code.
- How to run the code.
- Code.
- Sample input and output.

Introduction.

The program allows the user to convert up to four currencies into one. This saves them the time of having to convert each currency individually. For example, if someone wanted to convert USD, EUR and GBP into CHF manually, they would have to convert each currency and then add them all together. In this program the user simply inputs the currencies, corresponding monetary quantities and final currency, and the program will return the sum of all the converted currencies. The program is also able to check the users gains/losses from previous currency exchanges. For example, if the user had converted USD to EUR on a specific date and wanted to know whether or not he would be better off trading the currenices now, he would be able to check that. Lastly, the program can also give the user current spot rates.

All the exchange rates come from https://exchangeratesapi.io/, a currency API which uses the European Central Banks exchange rates which are updated on a daily basis. The program also uses the historical rates up until 2000 which are available from the same API.

Main aspects of the code.

1. The loading of the data from the API. The program uses the requests package in order to obtain the data from a URL and stores them as a json file. This process is also used to obtain the historical rates, the code simply extends the URL by adding the date to it.
2. The while True: loop at the beginning of the code allows the user to repeat the process without having to manually click rerun. While true the program runs all the code and at the end has an if function where the user can either continue using the program or stop using it in which case the program ends.
3. The currency conversion formula is a bit more complicated than a standard conversion since all our exchange rates use USD as the base rate, so if a user wants the EUR/CHF rate, we need use the EUR/USD and CHF/USD

exchange rates. The two exchange rates are then divided in order to get the EUR/CHF rate.

4. To calculate the gains/losses from previous trade we do the same thing as above where we calculate the value of that exchange if it were done today. In addition, we repeat the process but use the historical rates to calculate the value of the trade on a different date. We then calculate the difference between the two values to see if the value was higher/lower if it were made today.

   Note: in the older datasets the base rate is EUR and the data set does not attach a value to it (the value should be 1 since the rate would be EUR/EUR). Hence, there a few extra lines of code which attach the value 1 to the EUR/EUR. In the recent data sets the exchange rate of the base currency is included so we do not have this issue.

How to run the code.

To run the code simply download the CurrencyConverterPython.py file from GitHub and run it using Spyder from the Anaconda-Navigator. To download the file, click on raw, the window should change so that all the code is shown on a white background, then right click and save the file. Downloading the Anaconda Navigator is straightforward, and it comes with the Spyder preinstalled. Then simply run Spyder and open the python file. You can also copy paste the code directly into Spyder.

Code.

```python
1.  #!/usr/bin/env python3
2.  # -*- coding: utf-8 -*-
3.  """
4.  Created on Thu Nov 14 15:57:40 2019
5.
6.  @author: philipreiche
7.  """
8.  #This program uses the requests function to retrieve the data from a url, in our
    case the exchange rate API.
9.
10. import requests
11.
12. #We import the latest currency rates (quoted against the dollar) and store them
    under the name "data".
13.
14. url = "https://api.exchangerate-api.com/v4/latest/USD"
15.
16. response = requests.get(url)
17. data = response.json()
18.
19. #Introduction to the program, what it can do, how to use it, and which currencies
    are supported.
20.
21. print("Welcome to the currency converter!")
22. print("")
23. print("This converter allows you to convert 1 or more currencies.")
24. print("")
25. print("It is able to check if you are better/worse off from previous exchanges.")
26. print("")
27. print("It can also give you the current spot rates.")
28. print("")
29. print("Please keep in mind that everything is case sensitive!")
30. print("")
```

```python
31. currencies = ["AED", "ARS", "AUD", "BGN", "BRL", "BSD", "CAD", "CHF", "CLP", "CNY",
       "COP", "CZK", "DKK", "DOP", "EGP", "EUR", "FJD", "GBP", "GTQ", "HKD", "HRK", "HUF"
       , "IDR", "KRW", "KZT", "MXN", "MYR", "NOK", "NZD", "PAB", "PEN", "PHP", "PKR", "PLN
       ", "PYG", "RON", "RUB", "SAR", "SEK", "SGD", "THB", "TRY", "TWD", "UAH", "USD", "UY
       U", "VND", "ZAR"]
32. print("These are the currencies we support: ",currencies)
33.
34. #Use while True function, which at the end of the program asks the user if they
       want to run the program again.
35.
36. while True:
37.
38. #Ask the user if he would like to convert current currencies, if No the program
       will skip to the next option.
39.
40.     choice = input("Would you like to convert current currencies? (Yes/No) ")
41.
42.     if choice == "Yes":
43.
44.         print("You can convert a maximum of 4 currencies at a time!")
45.
46. #Use the try function so that if there is a ValueError or KeyError the program
       tells the user what went wrong.
47.
48.         try:
49.
50.             n = int(input("How many currencies would you like to convert? "))
51.
52. #Ask the user how many currencies to convert, supports up to 4 currencies.
53. #The process for converting currencies is the same no matter how many currencies
       are to be converted. We simply add more inputs and the conversion formula becomes
       longer, however; they all use the same principal.
54.
55.             if n == 1:
56.
57. #Ask the user for 3 inputs, an input currency,amount, and a final currency that the
        user wants to convert to.
58. #If the user wanted to convert 2 currencies, we would need an additional currency
       input and amount input.
59.
60.                 cur1 = input("Which currency would you like to convert? ")
61.                 amm1 = int(input("How much of this currency would you like to
       convert? "))
62.
63.                 finalcur = input("What currency would you like the convert this
       currency to? ")
64.
65. #We retreive the corresponding exchange rates from "data". First we specify that we
        want to retrieve something from "rates" and then use the user input to specifiy
       the specific rate which we want to retrieve.
66.
67.                 rate1 = (data["rates"][cur1])
68.                 finalrate = (data["rates"][finalcur])
69.
70. #We do the exchange rate calculation using the exchange rates we retrieved and the
       amounts which the user entered.
71. #For multiple currencies we simply repeat the same calculation for the second
       currency.
72. #The dataset uses USD as the base rate, so if the user wants to exchange something
       other than USD we will have to calculate the exchange rates ourselves.
73. #We do this by dividing the initial exchange rate by the final exchange rate, so a
       if the user wants to convert 10 EUR to CHF we would have to calculate how many
       dollars 10 EUR is, and then multiply this by the CHF/USD rate to get the CHF amount
       .
74. #This is what we do in the function below, first we divide the amount by the
       exchange rate and then multiply it be the final exchange rate.
```

```python
75.                print("")
76.                print(amm1, cur1,"=",
77.                     ((amm1/rate1)*finalrate),finalcur,"!")
78.
79.          elif n == 2:
80.
81. #As seen below, if the user wants to convert 2 currencies, we have 2 sets of
    currency/amount inputs.
82. #The conversion formula is the same as above, except we repeat the formula for the
    second currency and get the sum of the two.
83. #The same goes for 3 and 4 formula, we simply add more inputs and extend the
    formula.
84.
85.                cur1 = input("What is the first currency? ")
86.                amm1 = int(input("How much of this currency would you like to
    convert? "))
87.
88.                cur2 = input("What is the second currency? ")
89.                amm2 = int(input("How much of this currency would you like to
    convert? "))
90.
91.                finalcur = input("What currency would you like the convert these
    currencies to? ")
92.
93.                rate1 = (data["rates"][cur1])
94.                rate2 = (data["rates"][cur2])
95.                finalrate = (data["rates"][finalcur])
96.
97.                print("")
98.                print(amm1, cur1, "+", amm2, cur2, "=",
99.                     ((amm1/rate1)*finalrate)+((amm2/rate2)*finalrate),finalcur,"!
    ")
100.
101.              elif n == 3:
102.
103.                   cur1 = input("What is the first currency? ")
104.                   amm1 = int(input("How much of this currency would you like
    to convert? "))
105.
106.                   cur2 = input("What is the second currency? ")
107.                   amm2 = int(input("How much of this currency would you like
    to convert? "))
108.
109.                   cur3 = input("What is the third currency? ")
110.                   amm3 = int(input("How much of this currency would you like
    to convert? "))
111.
112.                   finalcur = input("What currency would you like the convert
    these currencies to? ")
113.
114.                   rate1 = (data["rates"][cur1])
115.                   rate2 = (data["rates"][cur2])
116.                   rate3 = (data["rates"][cur3])
117.                   finalrate = (data["rates"][finalcur])
118.
119.                   print("")
120.                   print(amm1, cur1, "+", amm2, cur2, "+", amm3, cur3, "=",
121.                        ((amm1/rate1)*finalrate)+((amm2/rate2)*finalrate)+((am
    m3/rate3)*finalrate),finalcur,"!")
122.
123.              elif n == 4:
124.
125.                   cur1 = input("What is the first currency? ")
126.                   amm1 = int(input("How much of this currency would you like
    to convert? "))
127.
```

```python
128.                    cur2 = input("What is the second currency? ")
129.                    amm2 = int(input("How much of this currency would you like
     to convert? "))
130.
131.                    cur3 = input("What is the third currency? ")
132.                    amm3 = int(input("How much of this currency would you like
     to convert? "))
133.
134.                    cur4 = input("What is the fourth currency? ")
135.                    amm4 = int(input("How much of this currency would you like
     to convert? "))
136.
137.                    finalcur = input("What currency would you like the convert
     these currencies to? ")
138.
139.                    rate1 = (data["rates"][cur1])
140.                    rate2 = (data["rates"][cur2])
141.                    rate3 = (data["rates"][cur3])
142.                    rate4 = (data["rates"][cur4])
143.                    finalrate = (data["rates"][finalcur])
144.
145.                    print("")
146.                    print(amm1, cur1, "+", amm2, cur2, "+", amm3, cur3, "+", amm
     3, cur4, "=",
147.                          ((amm1/rate1)*finalrate)+((amm2/rate2)*finalrate)+((am
     m3/rate3)*finalrate)
148.                          +((amm4/rate4)*finalrate),finalcur,"!")
149.
150.        #This try function at the beginning ends here where the except (KeyError,
     ValueError) tells the user to input valid currencies or numbers if unsupported
     currencies or words entered instead of numbers since will cause the program to not
     work.
151.
152.             except (KeyError, ValueError):
153.                 print("Please make sure you entered a valid currency or number!"
     )
154.
155.        #Use elif so that if the user enters "No" to the first question they are
     taken here.
156.
157.         elif choice == "No":
158.
159.            choice2 = input("Would you like to check your gains/losses from
     previous exchanges? (Yes/No) ")
160.
161.        #Give them the choice to check gains/losses, uses the same principal as the
     first if function so when they say no they skip this part and are brought to the
     final option without further choices.
162.
163.            if choice2 == "Yes":
164.
165.                try:
166.
167.                    print("You can now check if you
     saved or lost money on previous exchanges. WE DO NOT ADJUST FOR INFLATION!")
168.                    print("Choose any date from 2000 onwards.")
169.                    print("")
170.                    date = input("Please enter a date in the form %Y-%m-
     %d, ex. 2010-12-20. ")
171.
172.        #Ask the user to input the date of their previous exchange in the specific
     format so that we can add their input to the URL in order to retreive the exchange
     rates for that day.
173.        #The data retreival is the same as in the beginning simply the URL has been
     changed and we store the data under dataold.
174.
```

```python
175.                         urlold = "https://api.exchangeratesapi.io/"+date
176.                         response = requests.get(urlold)
177.                         dataold = response.json()
178.
179.                         curold = input("What currency did you exchange? " )
180.                         finalcurold = input("What currency did you exchange it to? "
    )
181.                         ammold = int(input("How much did you exchange? "))
182.
183.        #The older datasets have a different format where the base currency is not
    included, so we have to add the if functions in case the user wants to convert EUR.
     Since in the dataset there will be no value associated with EUR so we need to tell
     the program that the value of the base currency is always 1 since 1 EUR = 1 EUR.

184.
185.                         if finalcurold == "EUR":
186.
187.                             rateold = (dataold["rates"][curold])
188.                             finalrateold = 1
189.
190.                             convertold = ((ammold/rateold)*finalrateold)
191.
192.                             rate = (data["rates"][curold])
193.                             finalrate = (data["rates"][finalcurold])
194.
195.                             convertnew = ((ammold/rate)*finalrate)
196.
197.                             if convertnew > convertold:
198.                                 print("You lost money, had you converted your money
    today you would've gotten",(convertnew-convertold),"more.")
199.
200.                             else:
201.                                 print("You made money, had you converted your money
    today you would've gotten", (convertold-convertnew),"less.")
202.
203.                         elif curold == "EUR":
204.                             rateold = 1
205.                             finalrateold = (dataold["rates"][finalcurold])
206.
207.                             convertold = ((ammold/rateold)*finalrateold)
208.
209.                             rate = (data["rates"][curold])
210.                             finalrate = (data["rates"][finalcurold])
211.
212.                             convertnew = ((ammold/rate)*finalrate)
213.
214.                             if convertnew > convertold:
215.                                 print("You lost money, had you converted your money
    today you would've gotten",(convertnew-convertold),"more.")
216.
217.                             else:
218.                                 print("You made money, had you converted your money
    today you would've gotten", (convertold-convertnew),"less.")
219.
220.        #If the user does not want to exchange from/to EUR we can retreive the
    exchange rates normally.
221.        #Just as before, we go to the list named dataold, specify that we want the
    rates and then specify which rates we want.
222.        #We calculate the outcome of the old conversion the same way in which we
    calculated the conversions before, and we call the "convertold".
223.        #We also need to calculate what the outcome of that conversion would have
    been if it was done today, so we retreive the same currencies but this time using t
    he first data set which included the current exchange rates.
224.        #We calculate the outcome of the conversion and call it "convertnew", we use
     the if function to check if convertnew is more than convertold or vice versa and
    print the difference between the two.
```

```python
225.                           else:
226.                               rateold = (dataold["rates"][curold])
227.                               finalrateold = (dataold["rates"][finalcurold])
228.
229.                               convertold = ((ammold/rateold)*finalrateold)
230.
231.                               rate = (data["rates"][curold])
232.                               finalrate = (data["rates"][finalcurold])
233.
234.                               convertnew = ((ammold/rate)*finalrate)
235.
236.                               if convertnew > convertold:
237.                                   print("You lost money, had you converted your money
      today you would've gotten",(convertnew-convertold),"more.")
238.
239.                               else:
240.                                   print("You made money, had you converted your money
      today you would've gotten", (convertold-convertnew),"less.")
241.
242.          #Again we use a try except function in case an invalid currency was entered,
       an integer was entered instead of a string or vice versa.
243.
244.                       except (ValueError, KeyError):
245.                           print("Make sure you entered a valid currency/number.")
246.
247.          #If the user answered No at choice2 we can deduce that they want to check
      the spot rate since no other options remain.
248.
249.                   else:
250.                       print("")
251.                       print("You can now check the spot rate for a currency.")
252.                       print("")
253.                       base = input("What is the base currency? (ex. EUR): " )
254.                       quote = input("What is the quote currency? (ex. CHF): ")
255.
256.          #Calculating the spot rate is easy, we again retreive the exchange rates
      from the data set and transform them into the spot rate.
257.          #Since the exchange rates all use USD as the base rate, we have to divide th
      e exchange rates corresponding to the currencies which the user wants in the spot
      rate.
258.          #If the user wanted the EUR/CHF rate we would have to calculate: (EUR/USD)/(
      CHF/USD), this gives us the EUR/CHF rate.
259.
260.                       baserate = (data["rates"][base])
261.                       quoterate = (data["rates"][quote])
262.
263.                       print("Currently 1",base,"=",(quoterate/baserate),quote)
264.
265.          #If the user did not enter Yes/No then the first if and elif are not true
      and the user ends up here at the final part of the if/else function. We know that
      he did not enter Yes or No as instructed so we remind him to enter yes or no.
266.
267.               else:
268.                   print("")
269.                   print("Please enter either Yes or No.")
270.
271.          #This corresponds to our first while True loop, if the user completes the
      program they are asked if they want to go again, if they answer "Yes" the loop
      continues, otherwise it finishes.
272.
273.               while True:
274.                   answer = input("Would you like to run it again? (Yes/No): ")
275.                   if answer in ("Yes", "No"):
276.                       break
277.                   print("Invalid input.")
278.               if answer == "Yes":
```

```
279.              continue
280.          else:
281.              print("Goodbye!")
282.              break
```

Sample input and output.
- (everything in bold is the user input)

Welcome to the currency converter!

This converter allows you to convert 1 or more currencies.

It is able to check if you are better/worse off from previous exchanges.

It can also give you the current spot rates.

Please keep in mind that everything is case sensitive!

These are the currencies we support: ['AED', 'ARS', 'AUD', 'BGN', 'BRL', 'BSD', 'CAD', 'CHF', 'CLP', 'CNY', 'COP', 'CZK', 'DKK', 'DOP', 'EGP', 'EUR', 'FJD', 'GBP', 'GTQ', 'HKD', 'HRK', 'HUF', 'IDR', 'KRW', 'KZT', 'MXN', 'MYR', 'NOK', 'NZD', 'PAB', 'PEN', 'PHP', 'PKR', 'PLN', 'PYG', 'RON', 'RUB', 'SAR', 'SEK', 'SGD', 'THB', 'TRY', 'TWD', 'UAH', 'USD', 'UYU', 'VND', 'ZAR']

Would you like to convert current currencies? (Yes/No) *Yes*
You can convert a maximum of 4 currencies at a time!

How many currencies would you like to convert? *3*

What is the first currency? *USD*

How much of this currency would you like to convert? *100*

What is the second currency? *EUR*

How much of this currency would you like to convert? *100*

What is the third currency? *GBP*

How much of this currency would you like to convert? *100*

What currency would you like the convert these currencies to? *CHF*

100 USD + 100 EUR + 100 GBP = 334.7873502746932 CHF!

Would you like to run it again? (Yes/No): *Yes*

Would you like to convert current currencies? (Yes/No) *No*

Would you like to check your gains/losses from previous exchanges? (Yes/No) **Yes**
You can now check if saved or lost money on previous exchanges.
WE DO NOT ADJUST FOR INFLATION!
Choose any date from 2000 onwards.

Please enter a date in the form %Y-%m-%d, ex. 2010-12-20. ***2018-12-31***

What currency did you exchange? ***EUR***

What currency did you exchange it to? ***CHF***

How much did you exchange? ***1000***
You made money, had you converted your money today you would've gotten
38.59484174769909 less.

Would you like to run it again? (Yes/No): ***Yes***

Would you like to convert current currencies? (Yes/No) ***No***

Would you like to check your gains/losses from previous exchanges? (Yes/No) ***No***

You can now check the spot rate for a currency.

What is the base currency? (ex. EUR): ***EUR***

What is the quote currency? (ex. CHF): ***USD***
Currently 1 EUR = 1.1007324273571635 USD

Would you like to run it again? (Yes/No): ***No***
Goodbye!