Currency Converter

Group 1 Project

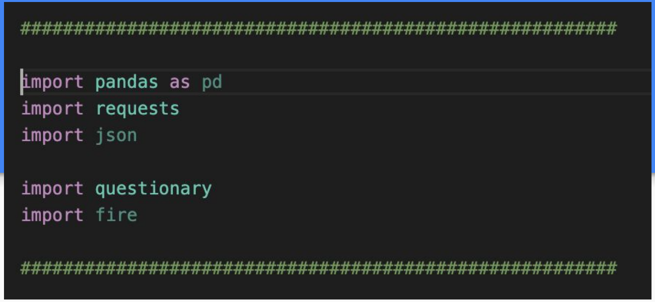
User Experience:

Retrieve current value price of 8 cryptos with 8 fiats, in CLI format.

Executive Summary: Initial Goals

- Create a currency converter between crypto and fiat (5 crypto & 5 fiat).
- User inputs their selection via a dropdown menu.
- Display the value of what one crypto is worth with selected fiat pair.
- User inputs the amount of fiat they'd like to convert into selected crypto to show how much of the particular crypto they will purchase.
- Ask the user if they'd like to display a 180-day historical performance chart of their selections.

The Process


```
###########################################################

import pandas as pd
import requests
import json

import questionary
import fire

###########################################################
```

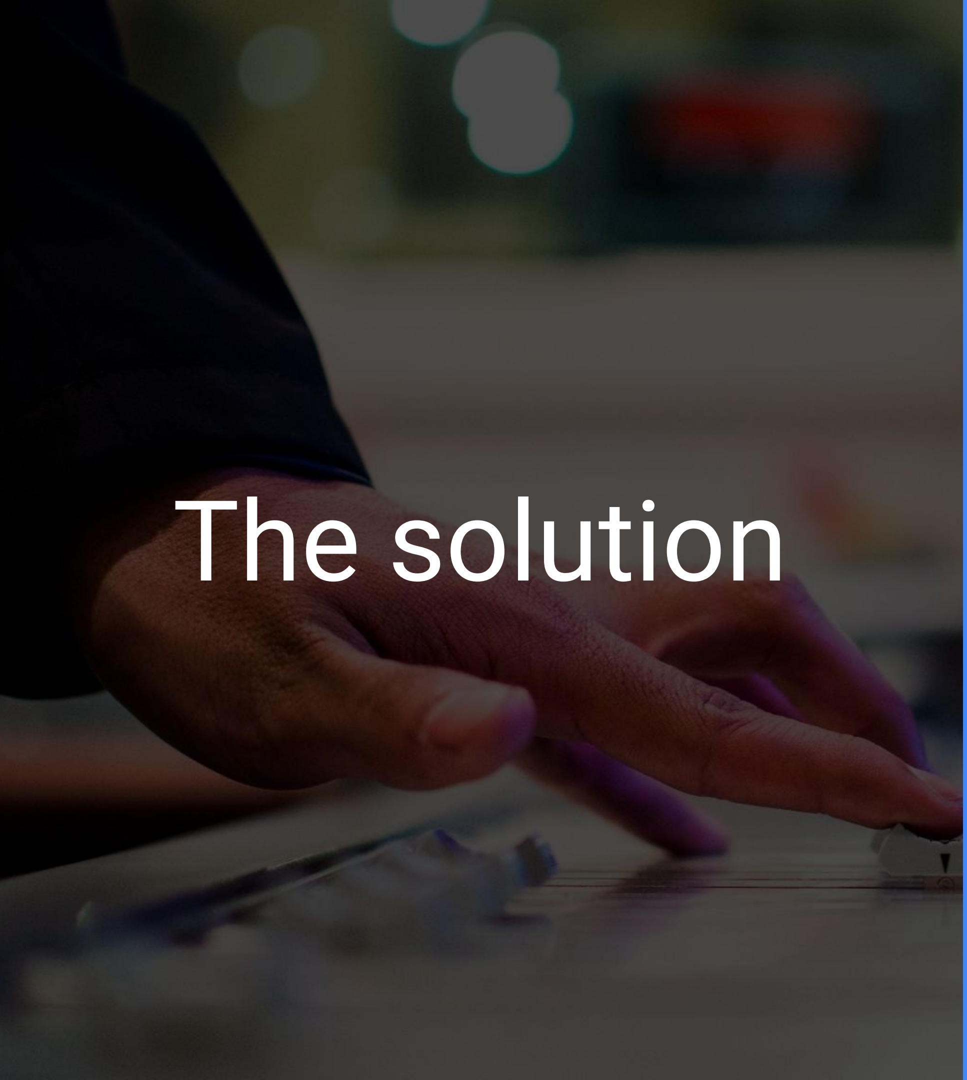
- Import our libraries —> —> —>
- The initial challenge was to connect to an API.
- https://api.alternative.me

> This had multiple fiats and cryptos to choose from, and ended up working nicely for us.

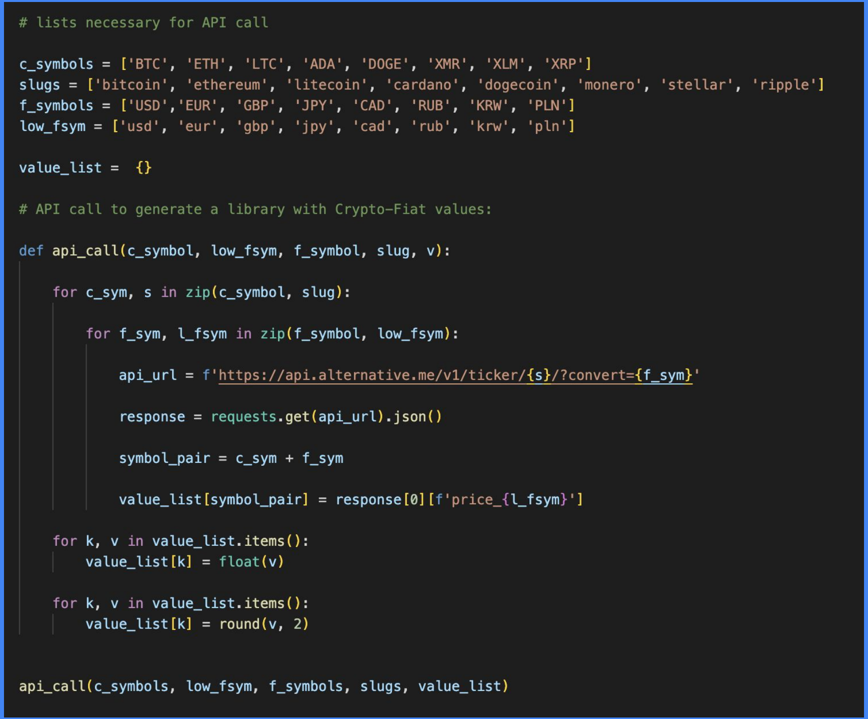
BUT...

Obstacles Faced

- Gave us 418 lines of code!!!
- Basically, each crypto would display 64 scenarios, and with 8 different crypto/fiat pairs
- 64 x 8 = 512

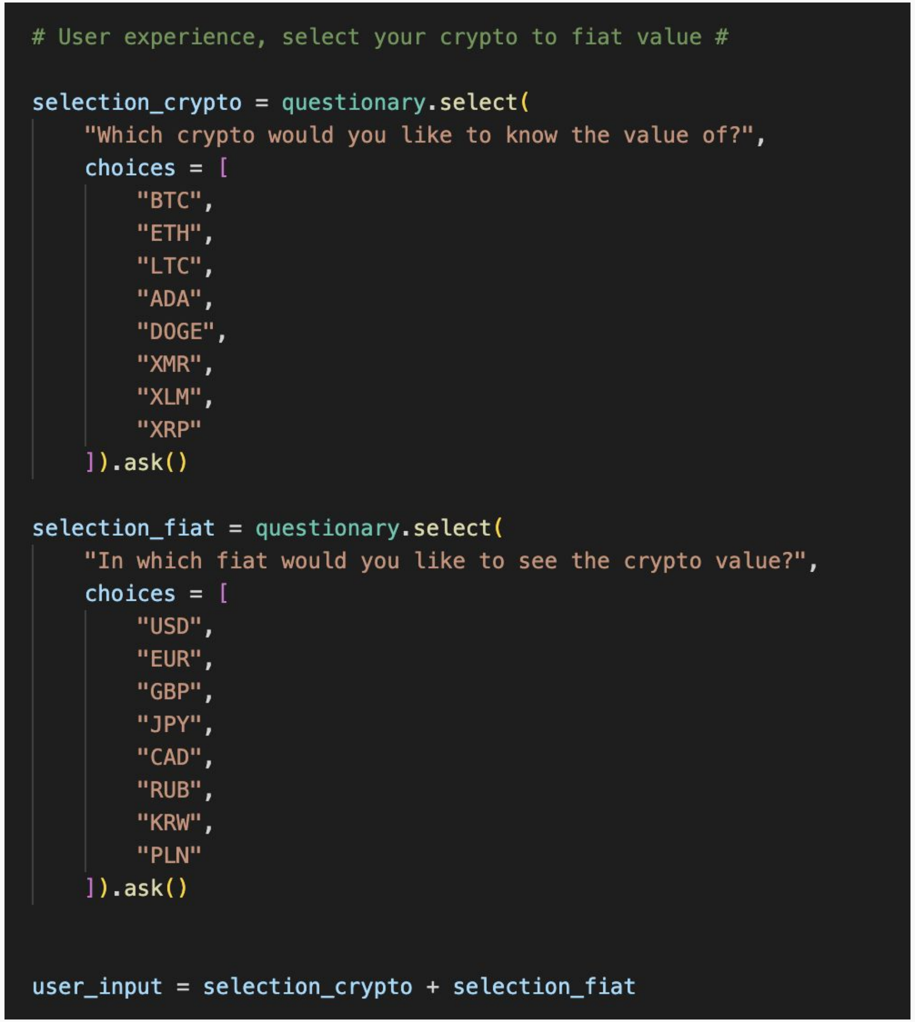


The solution



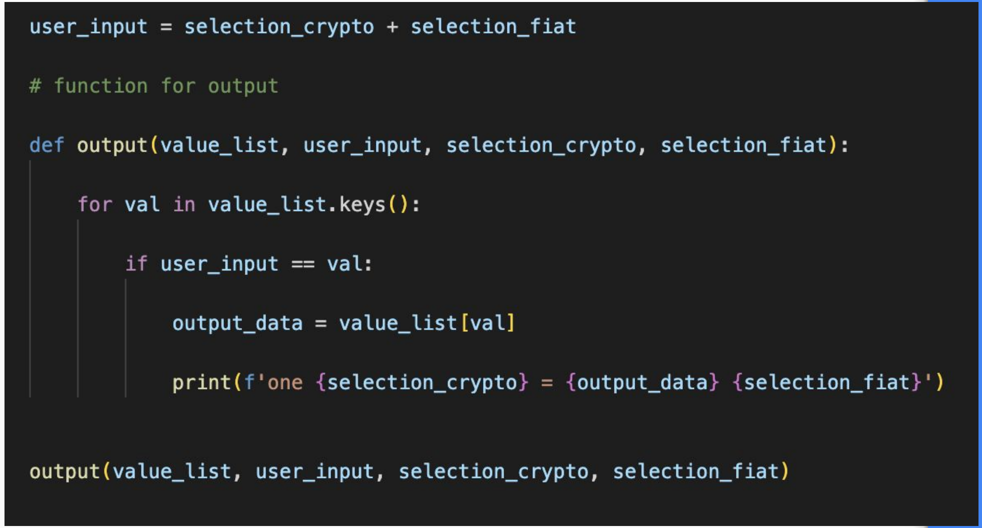
```python
# lists necessary for API call

c_symbols = ['BTC', 'ETH', 'LTC', 'ADA', 'DOGE', 'XMR', 'XLM', 'XRP']
slugs = ['bitcoin', 'ethereum', 'litecoin', 'cardano', 'dogecoin', 'monero', 'stellar', 'ripple']
f_symbols = ['USD','EUR', 'GBP', 'JPY', 'CAD', 'RUB', 'KRW', 'PLN']
low_fsym = ['usd', 'eur', 'gbp', 'jpy', 'cad', 'rub', 'krw', 'pln']

value_list =  {}

# API call to generate a library with Crypto-Fiat values:

def api_call(c_symbol, low_fsym, f_symbol, slug, v):

    for c_sym, s in zip(c_symbol, slug):

        for f_sym, l_fsym in zip(f_symbol, low_fsym):

            api_url = f'https://api.alternative.me/v1/ticker/{s}/?convert={f_sym}'

            response = requests.get(api_url).json()

            symbol_pair = c_sym + f_sym

            value_list[symbol_pair] = response[0][f'price_{l_fsym}']

    for k, v in value_list.items():
        value_list[k] = float(v)

    for k, v in value_list.items():
        value_list[k] = round(v, 2)


api_call(c_symbols, low_fsym, f_symbols, slugs, value_list)
```

Iterate through the API via a for loop, producing a value list dictionary of all 64 crypto/fiat pairs.



```
# User experience, select your crypto to fiat value #

selection_crypto = questionary.select(
    "Which crypto would you like to know the value of?",
    choices = [
        "BTC",
        "ETH",
        "LTC",
        "ADA",
        "DOGE",
        "XMR",
        "XLM",
        "XRP"
    ]).ask()

selection_fiat = questionary.select(
    "In which fiat would you like to see the crypto value?",
    choices = [
        "USD",
        "EUR",
        "GBP",
        "JPY",
        "CAD",
        "RUB",
        "KRW",
        "PLN"
    ]).ask()


user_input = selection_crypto + selection_fiat
```

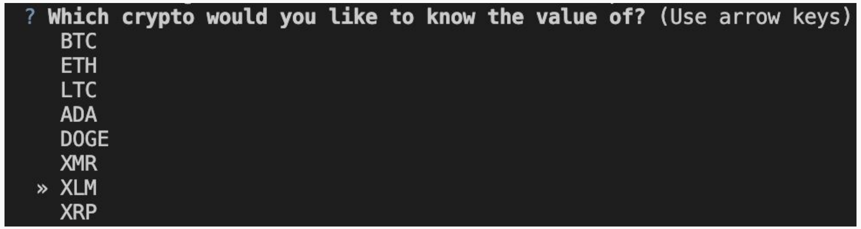
1. Next, we setup the user input with Questionary, asking the user to select a crypto, then select a fiat.



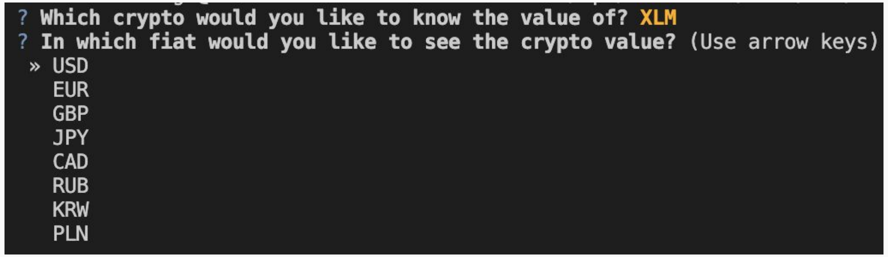
```python
user_input = selection_crypto + selection_fiat

# function for output

def output(value_list, user_input, selection_crypto, selection_fiat):

    for val in value_list.keys():

        if user_input == val:

            output_data = value_list[val]

            print(f'one {selection_crypto} = {output_data} {selection_fiat}')


output(value_list, user_input, selection_crypto, selection_fiat)
```

2. Then, run a function to iterate through the 64 pairs until there's a match with the user's input.
3. Displays the current value.

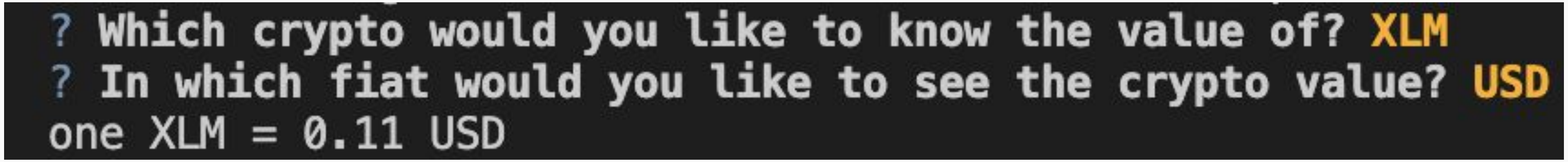
Quick View of the Command Line Interface



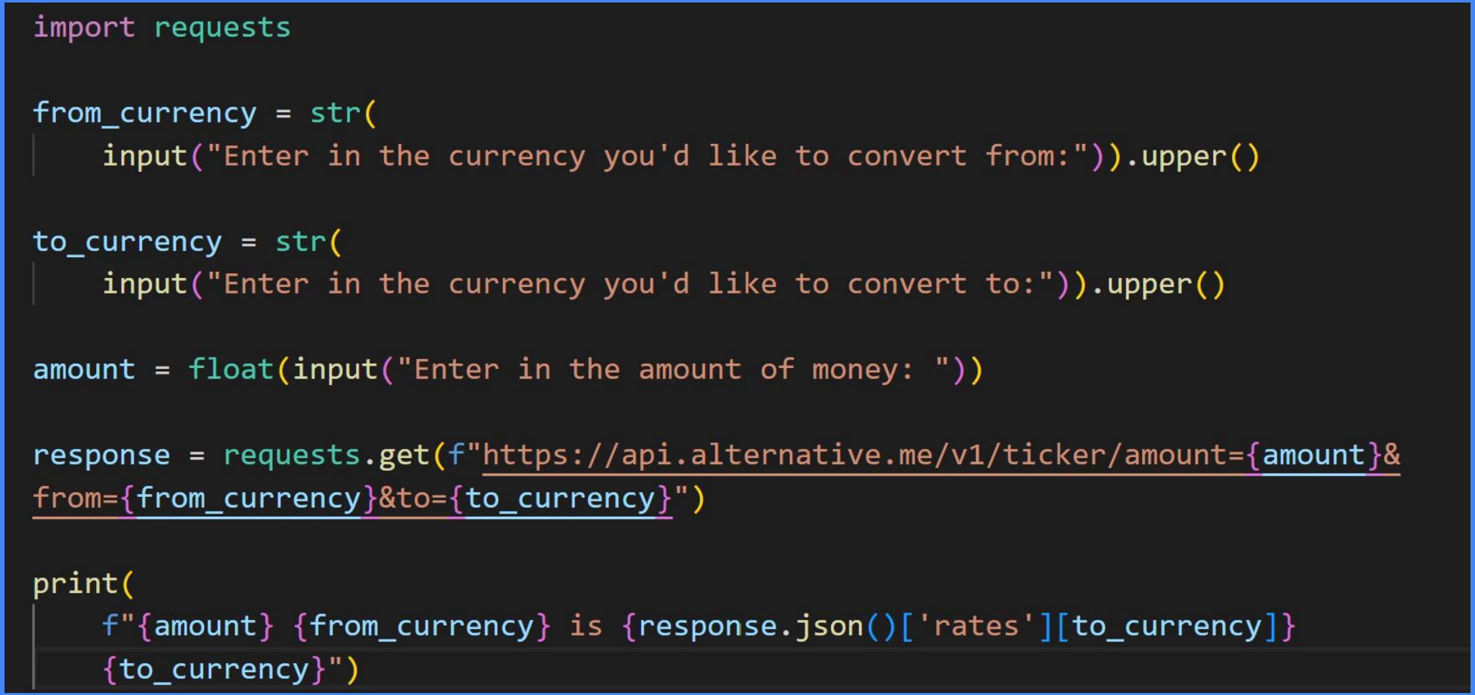
```
? Which crypto would you like to know the value of? (Use arrow keys)
   BTC
   ETH
   LTC
   ADA
   DOGE
   XMR
 » XLM
   XRP
```



```
? Which crypto would you like to know the value of? XLM
? In which fiat would you like to see the crypto value? (Use arrow keys)
 » USD
   EUR
   GBP
   JPY
   CAD
   RUB
   KRW
   PLN
```



```
? Which crypto would you like to know the value of? XLM
? In which fiat would you like to see the crypto value? USD
one XLM = 0.11 USD
```



```python
import requests

from_currency = str(
    input("Enter in the currency you'd like to convert from:")).upper()

to_currency = str(
    input("Enter in the currency you'd like to convert to:")).upper()

amount = float(input("Enter in the amount of money: "))

response = requests.get(f"https://api.alternative.me/v1/ticker/amount={amount}&
from={from_currency}&to={to_currency}")

print(
    f"{amount} {from_currency} is {response.json()['rates'][to_currency]}
    {to_currency}")
```

Finally, our group ran out of time trying to code what would've been the user entering the amount of their desired fiat they would like to convert, displaying the amount of desired crypto they would purchase. The above code was close, but Carl mentioned [“rates”] was where the code would fail.

Achievements

- Completed an idea from zero, all the way to user output.
- Successfully worked with Questionary, pandas, json, and requests libraries.
- Successfully wrote a function to loop through the API to get user results, instead of printing hundreds of results within a dictionary to then extract user input data.
- Able to retrieve live price value of Bitcoin (BTC), Ethereum (ETH), Cardano (ADA), Litecoin (LTC), Dogecoin (DOGE), Monero (XMR), Stellar Lumens (XLM), and Ripple (XRP) paired with a combination of USD, EUR, GBP, JPY, CAD, RUB, KRW, or PLN.

Reflections

- Figure out a way to ask the user for their crypto and fiat input BEFORE gathering data from the API.

    - Once run, the program takes at least 15 seconds to begin because it’s pulling all data from the API.

    - If the user inputs their selections first, then the function would only have to fetch a pair of data.

Moving Forward

- Add to the program where the user has the option to convert from crypto to fiat as well.
- Display historical data from the user’s selections

The team

Carl Mikel

Drew Pagnotta

Dan Poreda