



CONTEXT

Hil I'd like to buy a hoodie.
How much are you selling it for?

Each hoodie costs \$100 regardless of size. Shipping fee for international orders is \$10.

How much is that in Philippine Pesos? Could you compute it for me?

Hold on. I would need to check online and compute that for you. I'll get back to you.

with the current restrictions in place, many have resorted to online shopping and selling – both locally and, more importantly, internationally. Most business transactions, corporate affairs, and meetings happen online as well. With each country having their own currency, it would be hard for people from different nations to get a hold on the value of a certain amount without performing conversions first. Having a bot perform this through the messaging platform itself would definitely be a more efficient and transparent way to conduct business.

Hi! I'd like to buy a hoodie.
How much are you selling it for?

Each hoodie costs \$100 regardless of size. Shipping fee for international orders is \$10.

How much is that in Philippine Pesos? Could you compute it for me?

Hold on. I would need to check online and compute that for you. I'll get back to you.

SELLER FROM USA

BUYER FROM PH

SIGNIFICANCE

ACCESIBILITY

The bot was programmed to be available for usage at any time of the day.

CONVENIENCE

The bot is able to convert prices quickly and provide graphs to aid users in interpreting data.

ACCURACY

Data is kept up-to-date ensuring quality data.

MULTIFUNCTIONALITY

The bot is able to perform multiple functions to avoid the hassle of individually searching and computing for each function online.

An enhanced
currency converter
in the comfort of a
messaging platform
(Telegram)

GOALS

- Efficient transactions during foreign negotiations in the comfort of one's own chat box
- Provide a one-stop bot for providing info on currency conversions and graphs of conversion rates
- Promote transparency in currency conversion calculations during international trade and negotiations

NOTABLE BOT FEATURES

- * Bot is hosted on Heroku thus works 24/7
 - * Bot can be called into telegram chats * Currency rates are updated daily

/convert

can convert desired amount to any currency using data of the day

Types:

- one-to-one conversion
- one-to-many conversion

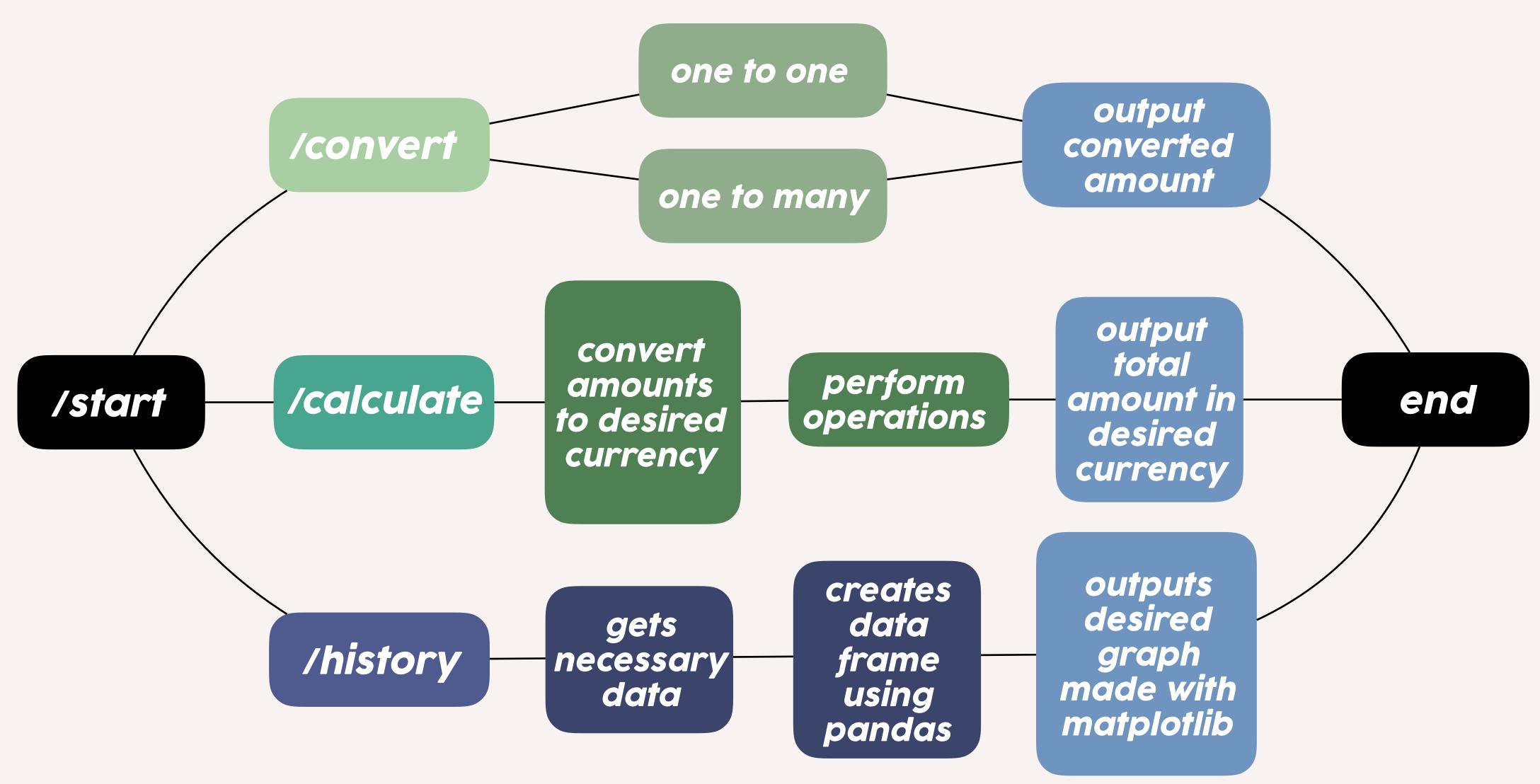
/calculate

adds/subtracts amounts of different currencies and returns total in desired currency

/history

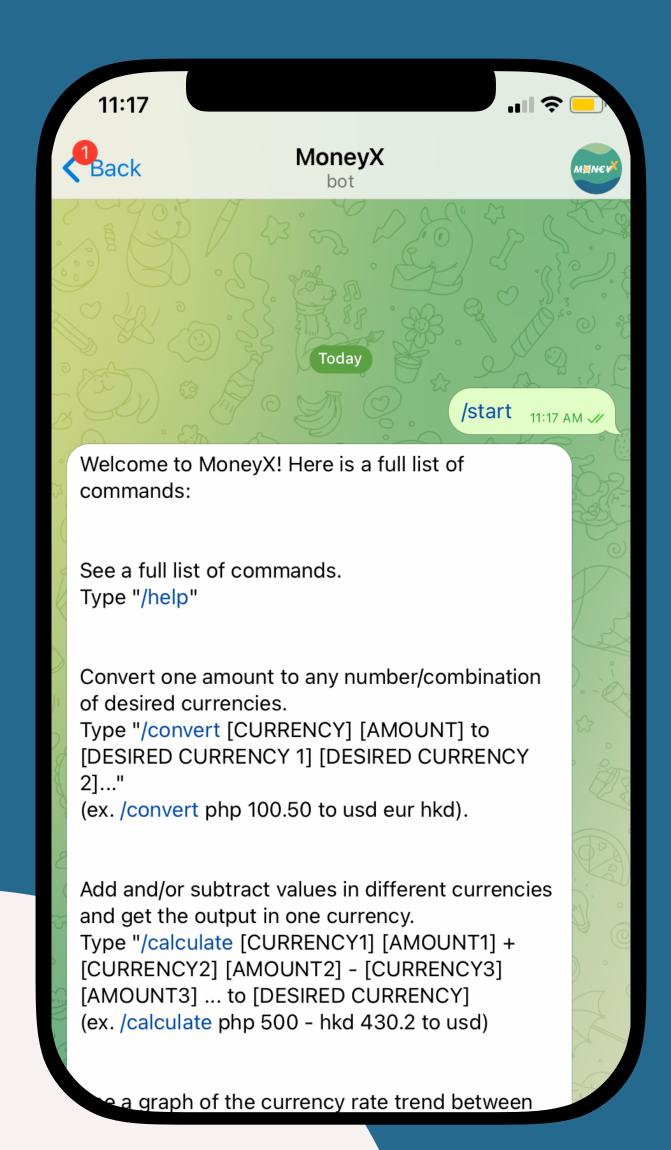
returns a 30-day graph of desired currency with respect to another currency

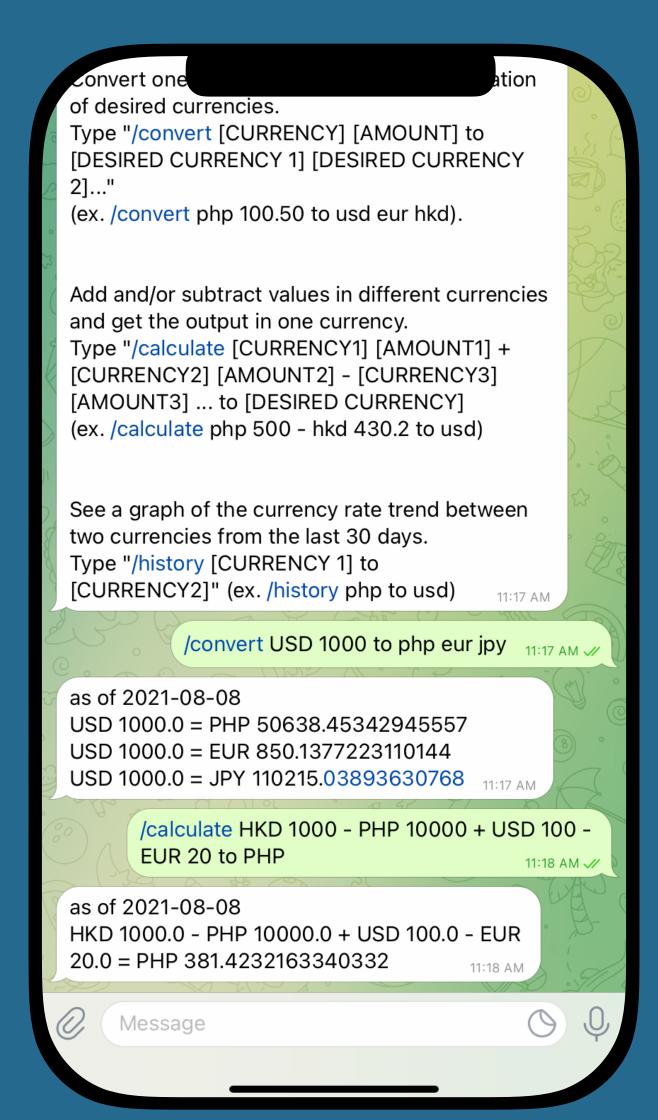
PROGRAM FLOWCHART

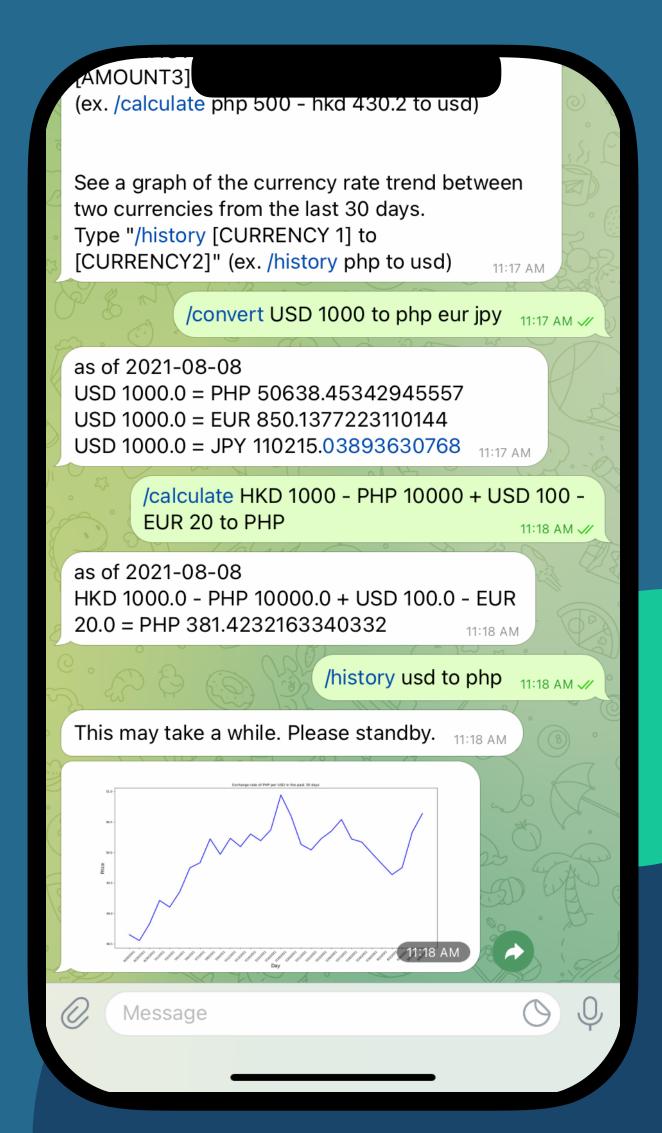


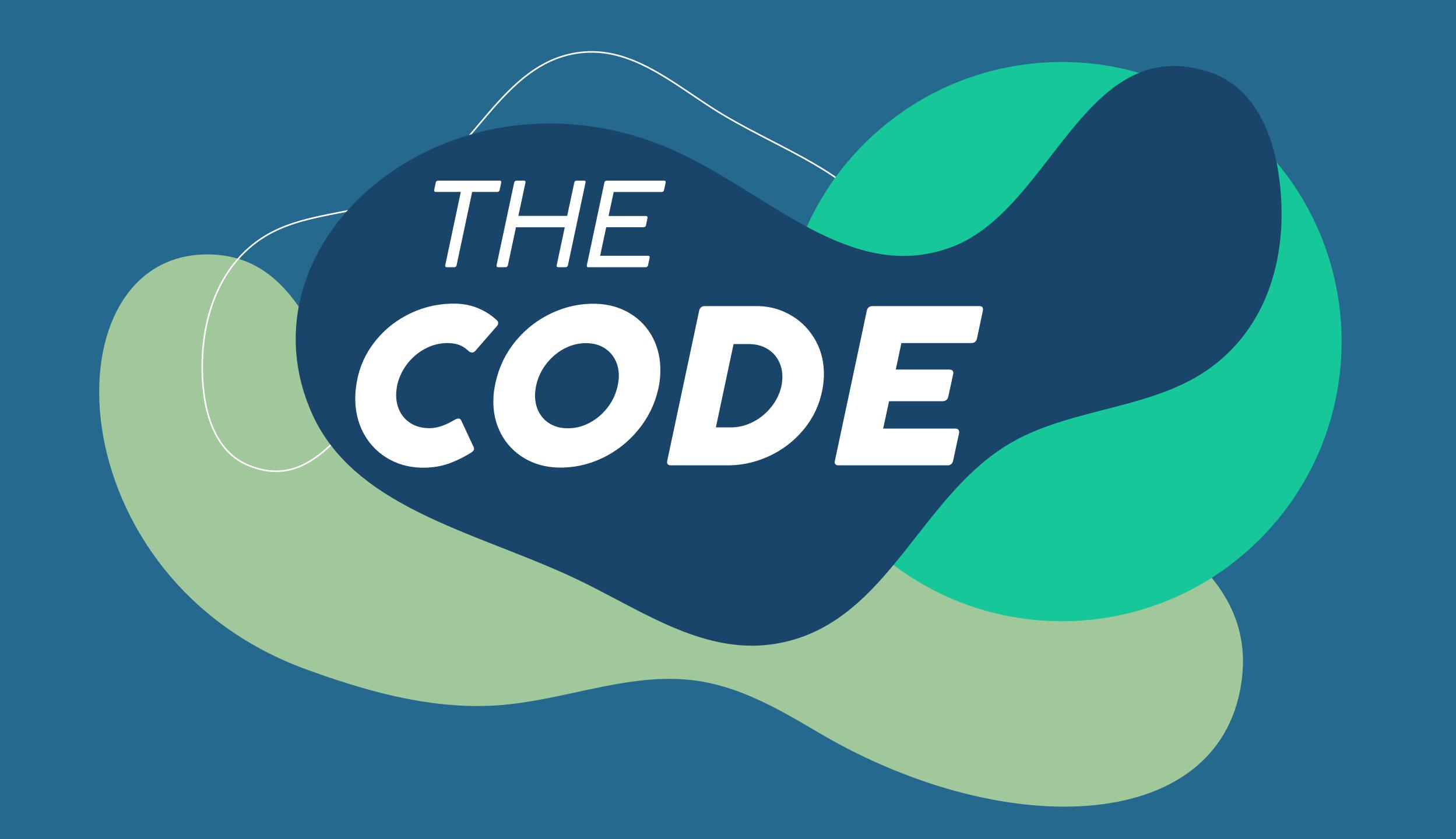
^{*} did not include /help since it just prints out an options menu

SAMPLE TELEGRAM BOT SESSION









PROCFILE

1 worker: python itmgt2021bot.py

Tells heroku what kind of project it's going to be Tells heroku what commands to run in startup

REQUIREMENTS. TXT

```
1 python-telegram-bot
```

2 requests

3 pandas

4 matplotlib

5

6 bob-telegram-tools

7 lxml

Tells heroku what modules it needs to install with pip

IMPORTING LIBRARIES AND INITIALIZATION

```
import logging
    import requests
    from telegram import Update, ForceReply
    from telegram.ext import Updater, CommandHandler, MessageHandler, Filters, CallbackContext
 5
    import pandas as pd
    import matplotlib.pyplot as plt
    import subprocess
    from bob_telegram_tools.bot import TelegramBot
10
     logging.basicConfig(format='%(asctime)s - %(name)s - %(levelname)s - %(message)s', level=logging.INFO)
    logger = logging.getLogger(__name__)
13
```

Istart function

```
def start(update: Update, context: CallbackContext) -> None:
14
15
        update.message.reply_text('''
        Welcome to MoneyX! Here is a full list of commands:\n
16
17
    See a full list of commands.\t
18
    Type "/help"\n
19
20
    Convert one amount to any number/combination of desired currencies.\t
    Type "/convert [CURRENCY] [AMOUNT] to [DESIRED CURRENCY 1] [DESIRED CURRENCY 2]..."\t
     (ex. /convert php 100.50 to usd eur hkd).\n
23
24
    Add and/or subtract values in different currencies and get the output in one currency.\t
    Type "/calculate [CURRENCY1] [AMOUNT1] + [CURRENCY2] [AMOUNT2] - [CURRENCY3] [AMOUNT3] ... to [DESIRED CURRENCY]\t
26
     (ex. /calculate php 500 - hkd 430.2 to usd)\n
27
28
    See a graph of the currency rate trend between two currencies from the last 30 days.\t
    Type "/history [CURRENCY 1] to [CURRENCY2]" (ex. /history php to usd)
        ''')
```

/convert function

```
def convert(update: Update, context: CallbackContext) -> None:
         r = requests.get('http://api.exchangeratesapi.io/v1/latest?access_key=9a223be759130d6d46862c6665f63926')
34
35
36
        try:
            currency1_name = context.args[0].upper()
37
            currency1_rate = r.json()['rates'][currency1_name]
38
            currency1_value = float(context.args[1])
39
40
            currency2_name = context.args[3:]
41
             result_formatted = 'as of ' + r.json()['date']
42
            for i in range(len(currency2_name)):
43
                currency2_name_indiv = currency2_name[i].upper()
44
                currency2_rate = r.json()['rates'][currency2_name_indiv]
45
                currency2_value = (currency1_value/currency1_rate) * currency2_rate
46
47
                 result_formatted += '\n' + currency1_name + ' ' + str(currency1_value) + ' = ' + currency2_name_indiv + ' ' + str(currency2_value)
48
49
            update.message.reply_text(result_formatted)
50
51
         except Exception as e:
            print(e)
            update.message.reply_text('Error. Please try again. Enter /help to see a list of commands.')
53
54
```

/calculate function part 1

```
def calculate(update: Update, context: CallbackContext) -> None:
        r = requests.get('http://api.exchangeratesapi.io/v1/latest?access_key=9a223be759130d6d46862c6665f63926')
56
57
58
        try:
59
            # first conversion
            currency1_name = context.args[0].upper()
60
            currency1_rate = r.json()['rates'][currency1_name]
61
            currency1_value = float(context.args[1])
62
63
            currency2_name = context.args[-1].upper()
64
            currency2_rate = r.json()['rates'][currency2_name]
65
             currency2_value = (currency1_value/currency1_rate) * currency2_rate
66
             result_value = currency2_value
67
             result_formatted = 'as of ' + r.json()['date']
68
             result_formatted += '\n' + currency1_name + ' ' + str(currency1_value)
69
70
            # second conversion
71
            try:
                currency_other = context.args[2:-2]
                currency_other_num = int(len(currency_other) / 3)
74
                for i in range(currency_other_num):
75
                     currency3_name = context.args[1 + (i * 3) + 2].upper()
76
                     currency3_rate = r.json()['rates'][currency3_name]
77
                     currency3_operator = context.args[1 + (i * 3) + 1]
                     currency3_absvalue = float(context.args[1 + (i * 3) + 3])
                     currency3_value = float(currency3_operator + str(currency3_absvalue))
80
```

81

/calculate function part 2

```
currency4_name = context.args[-1].upper()
82
                     currency4_rate = r.json()['rates'][currency4_name]
83
                     currency4_value = (currency3_value/currency3_rate) * currency4_rate
84
85
                     result_value += currency4_value
86
                     result_formatted += ' ' + currency3_operator + ' ' + currency3_name + ' ' + str(currency3_absvalue)
87
88
                 result_formatted += ' = ' + currency4_name + ' ' + str(result_value)
89
90
91
            except:
                 result_formatted += ' = ' + currency2_name + ' ' + str(currency2_value)
92
93
            update.message.reply_text(result_formatted)
94
95
        except:
            update.message.reply_text('Error. Please try again. Enter /help to see a list of commands.')
96
97
```

/history function

```
def history(update: Update, context: CallbackContext) -> None:
         chat_id = update.message.chat_id
 99
         bot = TelegramBot("1930727729:AAEEsKKfPyVlW-Ea2UNGBTzkugMXLeBIB3E", chat_id)
100
101
102
         try:
103
             #ex. USD to PHP
104
             currency1_name = context.args[0].upper()
             currency2_name = context.args[2].upper()
105
             base_url="https://www.exchange-rates.org/history/"
106
             extension=f"{currency2_name}/{currency1_name}/T"
107
             url=f"{base_url}{extension}"
108
109
             dfs = pd.read_html(url)
110
111
112
             date=dfs[0][0][0:30]
113
             price=dfs[0][2][0:30].str.rstrip(currency2_name).astype('float')
114
115
             fig=plt.figure(figsize=(20, 10), dpi=80)
116
             plt.plot(date,price,"b",linewidth=2.50)
             plt.gca().invert_xaxis()
117
             plt.xlabel("Day",fontsize=16)
118
             plt.ylabel("Price", fontsize=16)
119
120
             plt.xticks(rotation=45)
             plt.title(f"Exchange rate of {currency2_name} per {currency1_name} in the past 30 days")
121
122
             update.message.reply_text('This may take a while. Please standby.')
123
124
             bot.send_plot(plt)
125
126
         except:
             update.message.reply_text('Error. Please try again. Enter /help to see a list of commands.')
127
128
```

/help function

```
129
     def help(update: Update, context: CallbackContext) -> None:
         update.message.reply_text('''
130
         Here is a full list of commands:\n
131
132
     Convert one amount to any number/combination of desired currencies.\t
133
     Type "/convert [CURRENCY] [AMOUNT] to [DESIRED CURRENCY 1] [DESIRED CURRENCY 2]..."\t
134
      (ex. /convert php 100.50 to usd eur hkd).\n
135
136
     Add and/or subtract values in different currencies and get the output in one currency.\t
137
     Type "/calculate [CURRENCY1] [AMOUNT1] + [CURRENCY2] [AMOUNT2] - [CURRENCY3] [AMOUNT3] ... to [DESIRED CURRENCY]\t
138
      (ex. /calculate php 500 - hkd 430.2 to usd)\n
139
140
     See a graph of the currency rate trend between two currencies from the last 30 days.\t
141
     Type "/history [CURRENCY 1] to [CURRENCY2]" (ex. /history php to usd)
142
          ''')
143
144
```

echo and main function

```
def echo(update: Update, context: CallbackContext) -> None:
145
          update.message.reply_text(update.message.text)
146
147
     def main() -> None:
148
149
          updater = Updater("1930727729:AAEEsKKfPyVlW-Ea2UNGBTzkugMXLeBIB3E")
          dispatcher = updater.dispatcher
150
151
          dispatcher.add_handler(CommandHandler("start", start))
152
          dispatcher.add_handler(CommandHandler("help", help))
153
          dispatcher.add_handler(CommandHandler("convert", convert))
154
155
          dispatcher.add_handler(CommandHandler("calculate", calculate))
          dispatcher.add_handler(CommandHandler("history", history))
156
157
158
          updater.start_polling()
159
160
          updater.idle()
161
162
163
164
165 if __name__ == '__main__':
166
         main()
```

POSSIBLE EXTENSIONS/UPDATES

- include cryptocurrencyto-cryptocurrency converter
- include cryptocurrencyto-regular currency conversions
- Incorporate a converter for foreign stock prices to desired currency