

CODING A DISCORD BOT FOR PRICE TRACKING



PRESENTATION BY: MONICA FIDALGO

SKILLS

BeautifulSoup



Web
Scraping,
API
Calling



Python,
Discord.py



Web
Hosting,
Tasks &
Schedulers

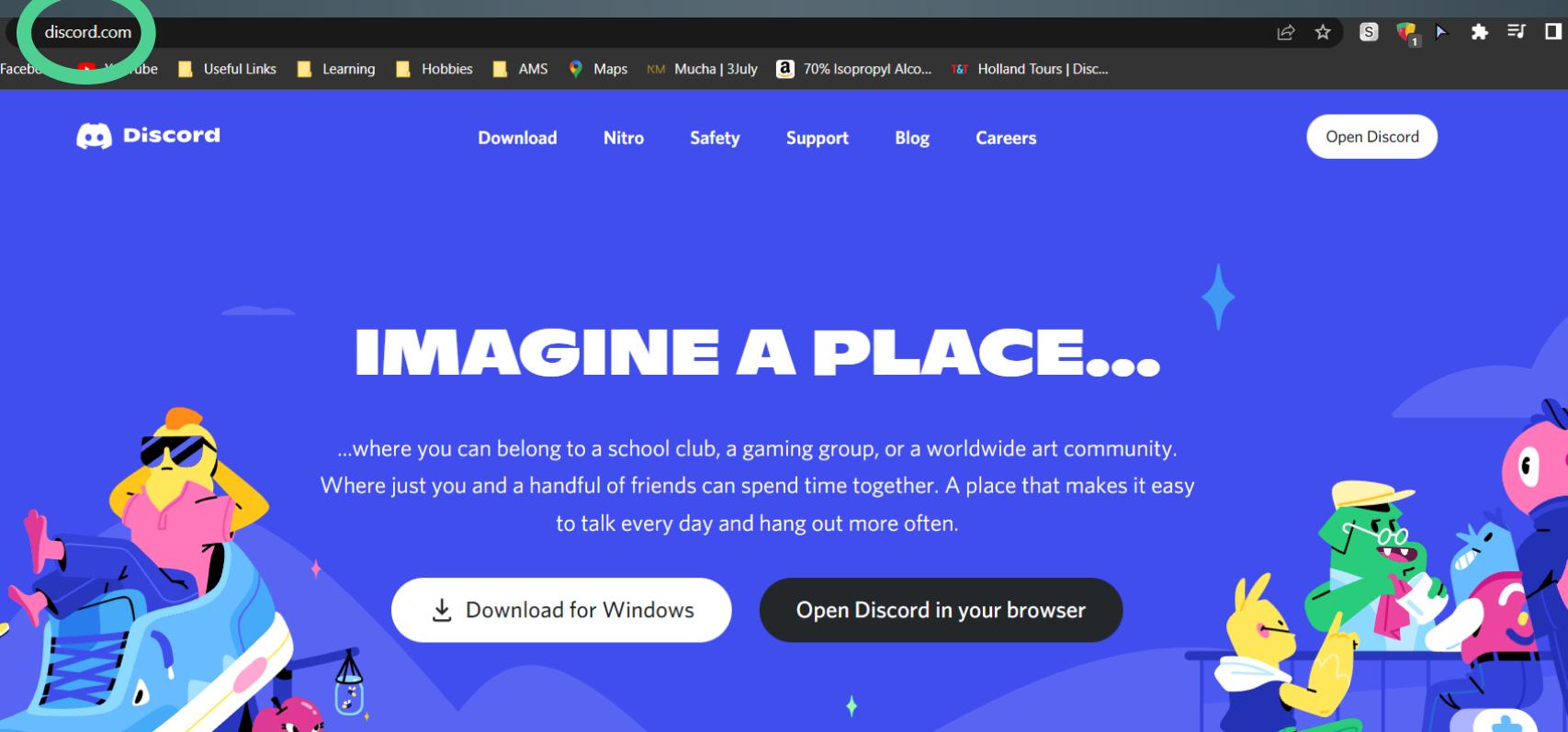


pandas

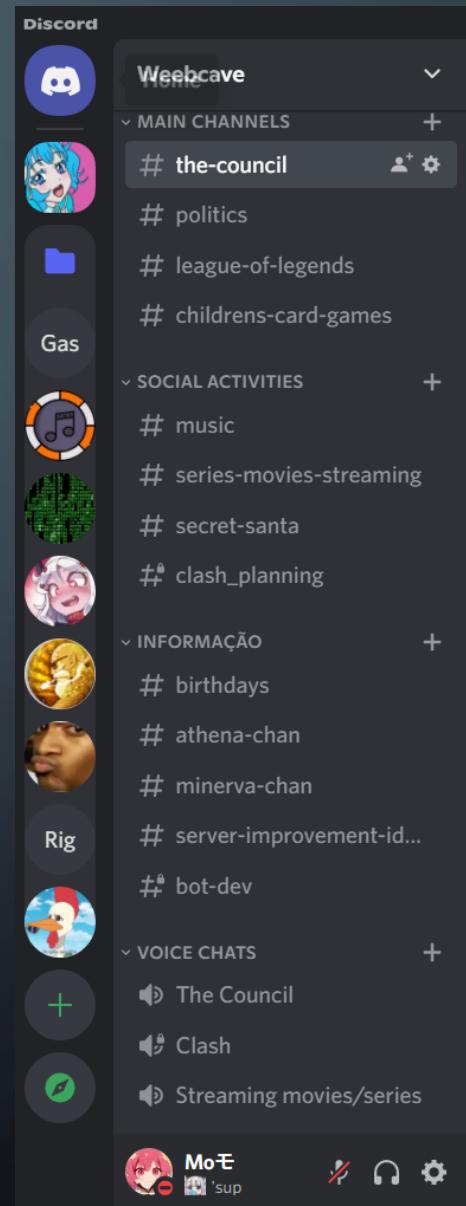
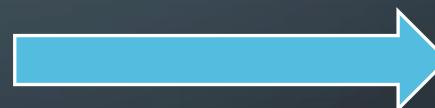
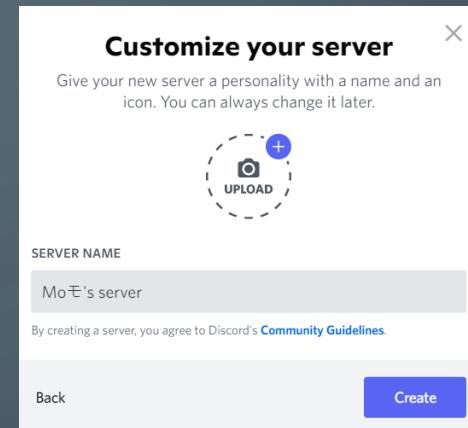
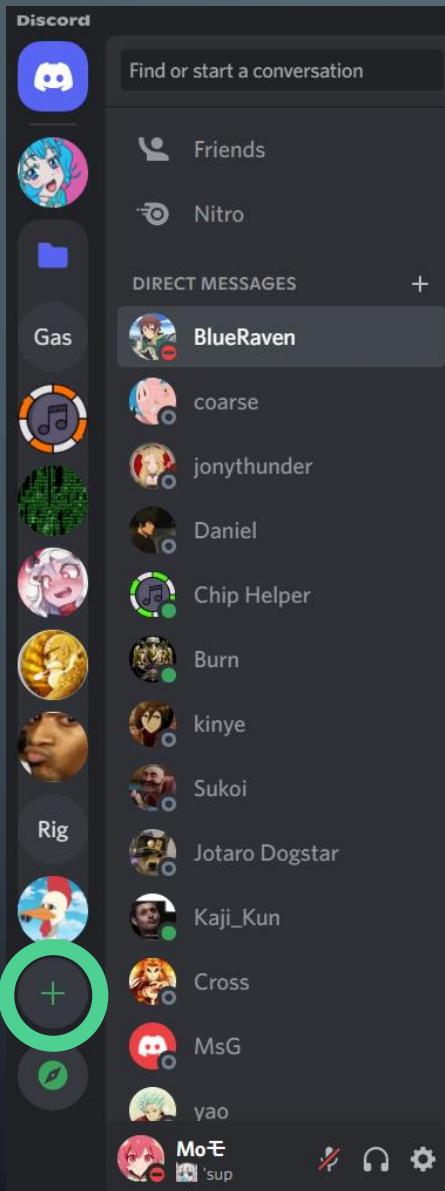
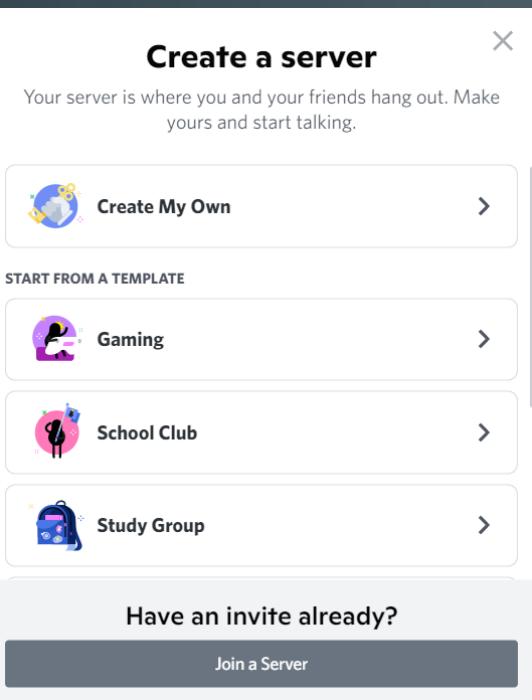
WHAT IS DISCORD?



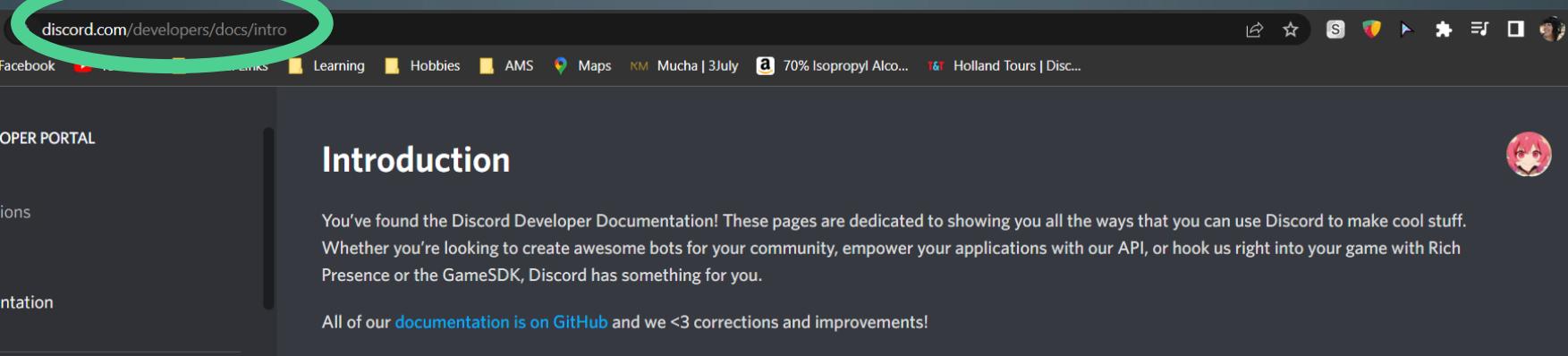
DOWNLOAD DISCORD AND CREATE YOUR ACCOUNT



CREATE YOUR SERVER



GO TO THE DEVELOPER'S PORTAL



A screenshot of the Discord Developer Portal. The top navigation bar has a 'DEVELOPER PORTAL' icon, 'Applications' (which is highlighted in blue), 'Teams', and 'Documentation'. To the right is a 'New Application' button and a user profile icon. The main content area is titled 'Applications' with a large blue downward arrow pointing towards the 'My Applications' section. Below the title, text encourages users to find a feature set for their game in the Game SDK and sign up for Server Commerce. The 'Sort By:' dropdown is set to 'Date Created'. At the bottom, there are 'Small' and 'Large' grid options. The 'My Applications' section shows two cards: 'LadyBots' (small icon) and 'LadyBots #2' (large icon).

CREATE AN APPLICATION WITH A BOT

CREATE AN APPLICATION

Are you a game dev? We may already have your app in our database. Reach out to our [Dev Support](#) for more info and to claim your game!

NAME *

By creating an API application, you agree to the Discord API [Terms of Service](#)

[Cancel](#) [Create](#)

[Back to Applications](#)

SELECTED APP
trackerbot

SETTINGS

- General Information
- OAuth2
- Bot**
- Rich Presence
- App Testers

Bot

Bring your app to life on Discord with a Bot user. Be a part of chat in your users' servers and interact with them directly.

[Learn more about bot users](#)

Build-A-Bot

Bring your app to life by adding a bot user. This action is irreversible (because robots are too cool to destroy).

[Add Bot](#)

Build-A-Bot

Bring your app to life by adding a bot user. This action is irreversible (because robots are too cool to destroy).

ICON

USERNAME
Pheme

TOKEN
For security purposes, tokens can only be viewed once, when created. If you forget or lose access to your token, please regenerate a new one.

OTgwNzU2NDQxNTkxODQ4OTgx.GCS_fU.wbgnU1oLvyiwLYdCDIij0wbTQZ7kes-dW52-q0

[Copy](#) [Reset Token](#)

The Token is your Bot's "password". Anyone with the Token can access your Bot, so keep it a secret or reset it if you think it has been compromised!

#2568

SELECT YOUR BOT'S PERMISSIONS

The screenshot shows the OAuth2 URL Generator page from the Discord developer portal. The left sidebar shows the selected app is "trackerbot". The main area has a heading "OAuth2 URL Generator" and a sub-section "SCOPES" where the "bot" scope is checked. Below this are three sections: "GENERAL PERMISSIONS", "TEXT PERMISSIONS", and "VOICE PERMISSIONS", each containing several checkboxes. At the bottom, there is a "GENERATED URL" field with a blue "Copy" button highlighted with a green circle and arrow.

Our bot will need text permissions and will need to be able to read messages and view channels.

OAuth2 URL Generator

Generate an invite link for your application by picking the scopes and permissions it needs to function. Then, share the URL to others!

SCOPES

GENERAL PERMISSIONS

- Administrator
- View Audit Log
- Manage Server
- Manage Roles
- Manage Channels
- Kick Members
- Ban Members
- Create Instant Invite
- Change Nickname
- Manage Nicknames
- Manage Emojis and Stickers
- Manage Webhooks
- Read Messages/View Channels
- Manage Events
- Moderate Members

TEXT PERMISSIONS

- Send Messages
- Create Public Threads
- Create Private Threads
- Send Messages in Threads
- Send TTS Messages
- Manage Messages
- Manage Threads
- Embed Links
- Attach Files
- Read Message History
- Mention Everyone
- Use External Emojis
- Use External Stickers
- Add Reactions
- Use Slash Commands

VOICE PERMISSIONS

- Connect
- Speak
- Mute Members
- Deafen Members
- Move Members
- Use Voice Activity
- Priority Speaker

GENERATED URL

Copy

Save this URL: it will allow you to invite your Bot to your Server!

USE THE URL YOU SAVED TO INVITE YOUR BOT TO THE SERVER

An external application
trackerbot BOT
Connect to Discord
Signed in as Moモ#3520 Not you?

ADD TO SERVER:
Select a server

This requires you to have Manage Server permission in this server.

Cancel Continue

The bot is in our server! It is online because we did not start developing it yet!

Pheme#2568 BOT

ROLE
● trackerbot +

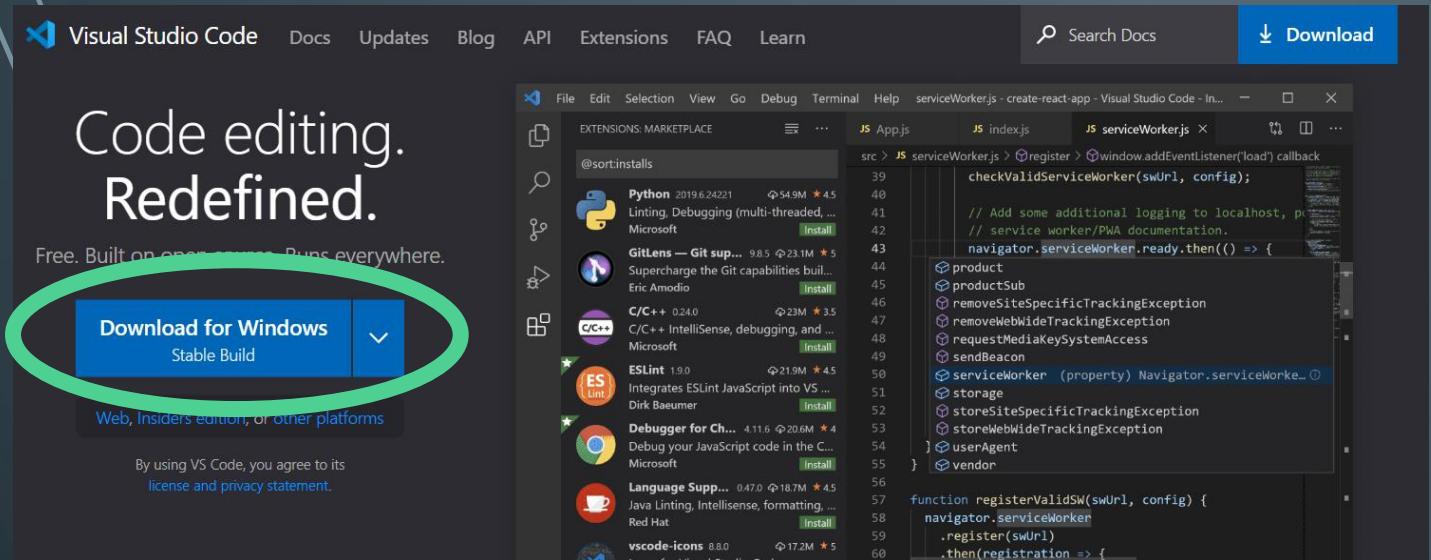
NOTE
Click to add a note

Message @Pheme

OFFLINE — 17

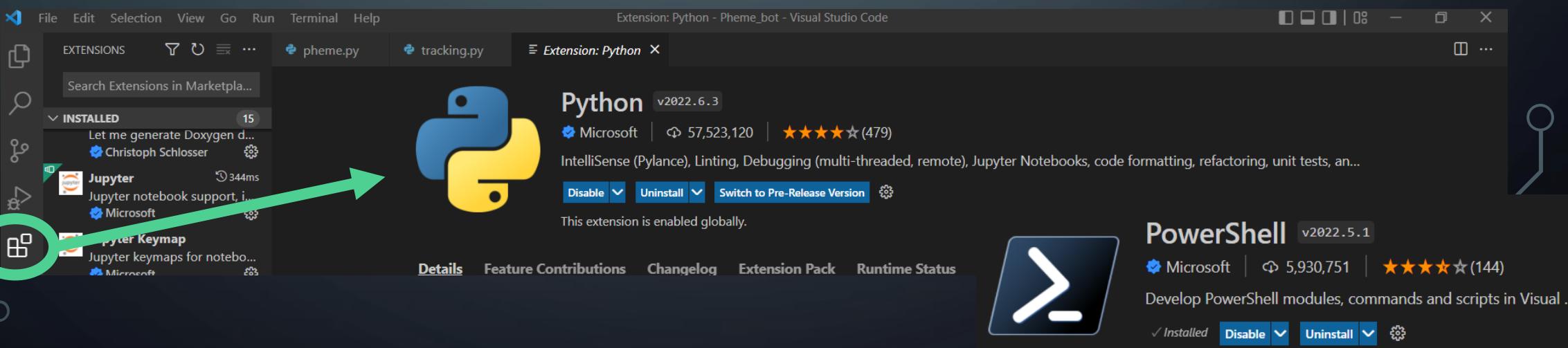
- AfarisPT
- Aiamuhabida
- Bacanos
- Burn
- coarse
- consumivel
- Daniel
- jonythunder
- MaDcHIT
- Ovelhinha

DEVELOP YOUR BOT LOCALLY WITH VS CODE



VS Code also has a Mac Version, but this tutorial will be for Windows.

After you've installed VS Code, go to the extensions on the left and download Python + PowerShell.



CREATE A FOLDER FOR YOUR DISCORD BOT OPEN IT IN VS CODE



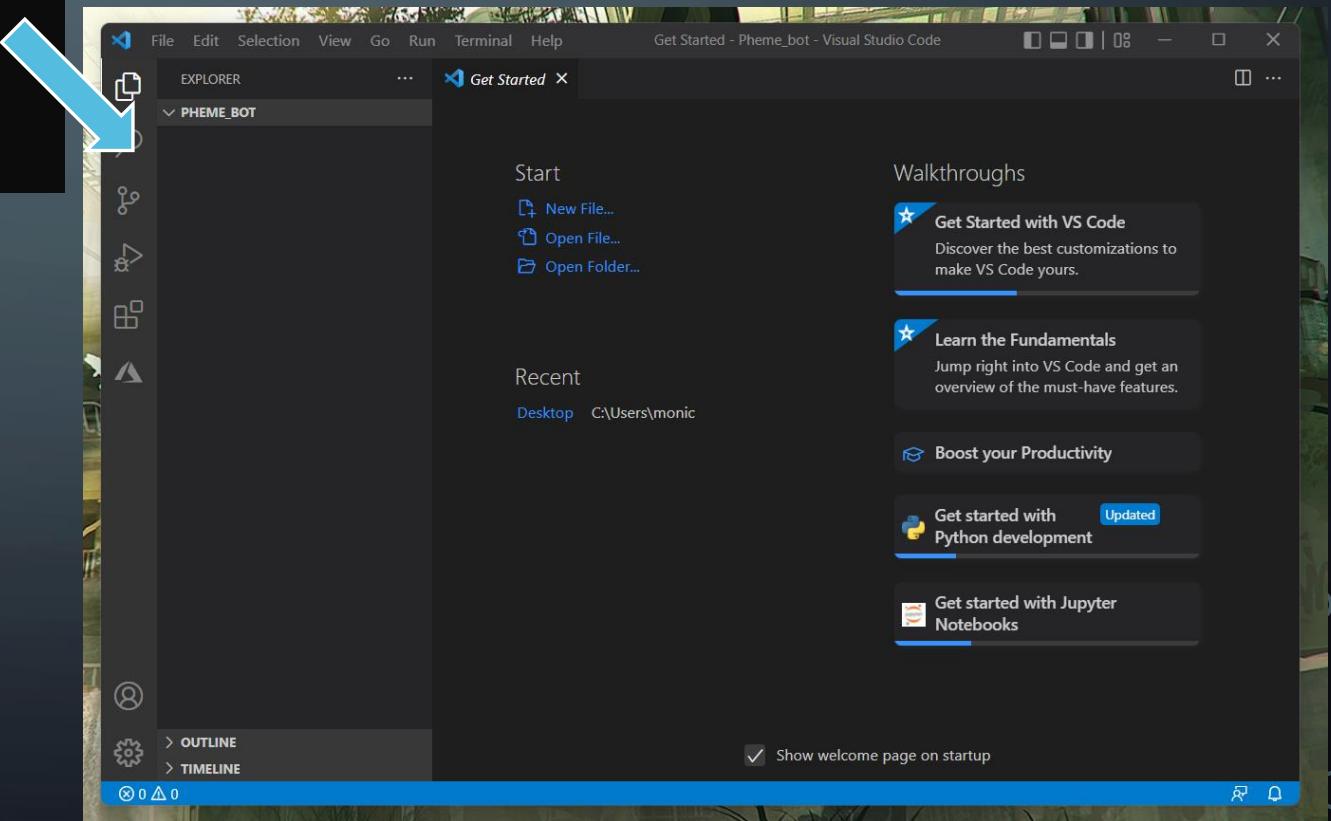
```
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\monic>cd Desktop

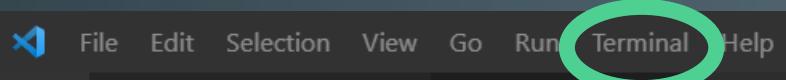
C:\Users\monic\Desktop>cd Pheme_bot

C:\Users\monic\Desktop\Pheme_bot>code .

C:\Users\monic\Desktop\Pheme_bot>
```

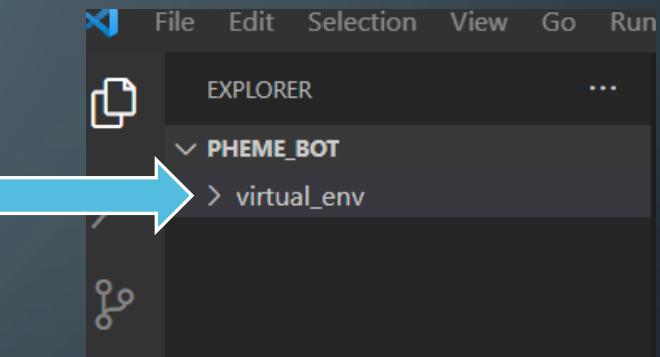


CREATE A VIRTUAL ENVIRONMENT FOR THE BOT



Click "terminal">>"new terminal" to open the console

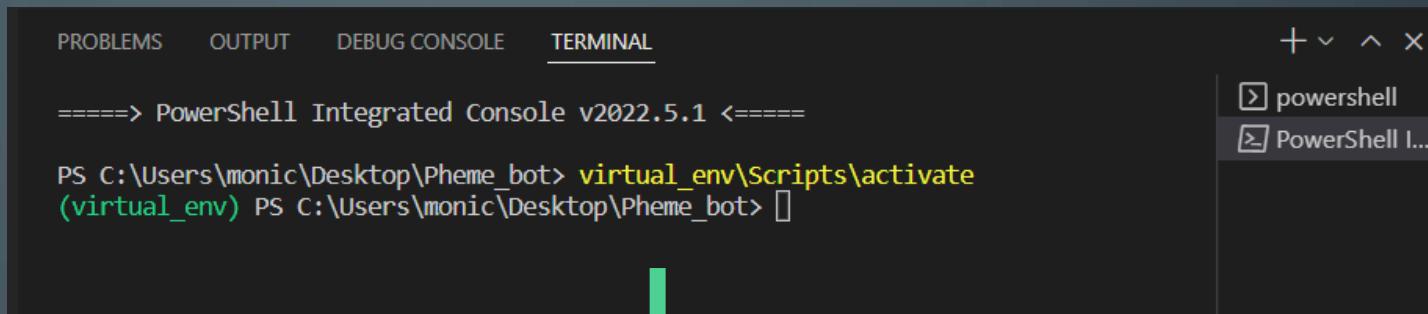
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
=====> PowerShell Integrated Console v2022.5.1 <=====
PS C:\Users\monic\Desktop\Pheme_bot> virtual_env\Scripts\activate
(virtual_env) PS C:\Users\monic\Desktop\Pheme_bot> []
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
(virtual_env) PS C:\Users\monic\Desktop\Pheme_bot> pip install discord
Collecting discord
  Using cached discord-1.7.3-py3-none-any.whl (1.1 kB)
Collecting discord.py>=1.7.3
  Using cached discord.py-1.7.3-py3-none-any.whl (786 kB)
Collecting aiohttp<3.8.0,>=3.6.0
  Using cached aiohttp-3.7.4.post0-cp39-cp39-win_amd64.whl (634 kB)
Collecting typing-extensions>=3.6.5
  Using cached typing_extensions-4.2.0-py3-none-any.whl (24 kB)
Collecting multidict<7.0,>=4.5
  Using cached multidict-6.0.2-cp39-cp39-win_amd64.whl (28 kB)
Collecting attrs>=17.3.0
  Using cached attrs-21.4.0-nv2.nv3-none-any.whl (60 kB)
```

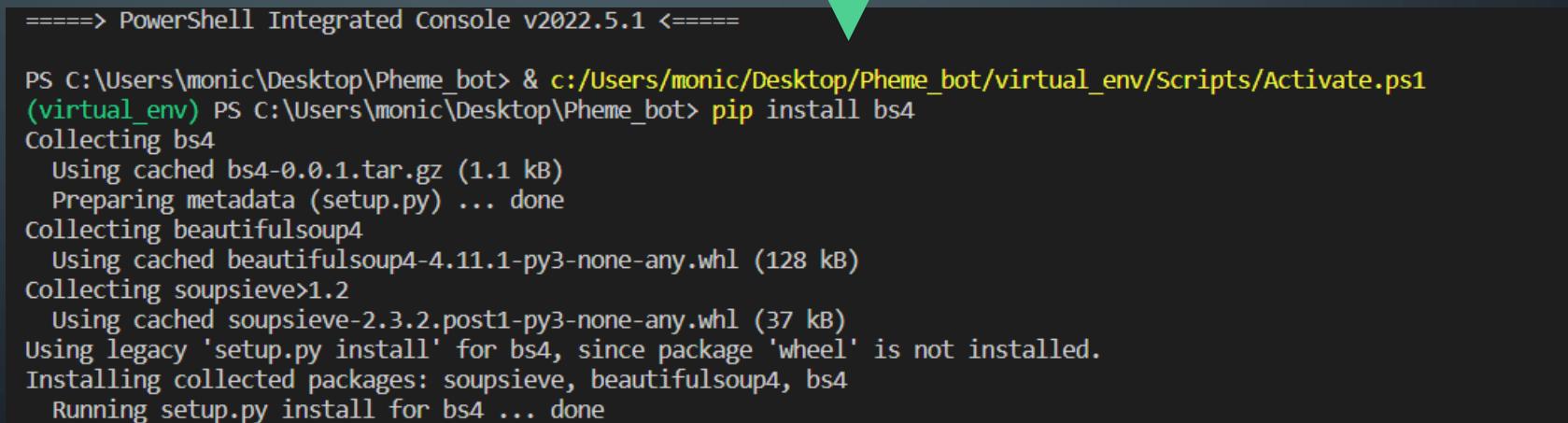
YOU NEED TO OPEN POWERSHELL INTEGRATED CONSOLE TO INSTALL PACKAGES IN THE VENV

If “new terminal” doesn’t open powershell integrated console, try this button:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
=====> PowerShell Integrated Console v2022.5.1 <=====  
PS C:\Users\monic\Desktop\Pheme_bot> virtual_env\Scripts\activate  
(virtual_env) PS C:\Users\monic\Desktop\Pheme_bot> 
```

Install all the packages your bot will need IN THE VIRTUAL ENVIRONMENT



```
=====> PowerShell Integrated Console v2022.5.1 <=====  
PS C:\Users\monic\Desktop\Pheme_bot> & c:/Users/monic/Desktop/Pheme_bot/virtual_env/Scripts/Activate.ps1  
(virtual_env) PS C:\Users\monic\Desktop\Pheme_bot> pip install bs4  
Collecting bs4  
  Using cached bs4-0.0.1.tar.gz (1.1 kB)  
  Preparing metadata (setup.py) ... done  
Collecting beautifulsoup4  
  Using cached beautifulsoup4-4.11.1-py3-none-any.whl (128 kB)  
Collecting soupsieve>1.2  
  Using cached soupsieve-2.3.2.post1-py3-none-any.whl (37 kB)  
Using legacy 'setup.py install' for bs4, since package 'wheel' is not installed.  
Installing collected packages: soupsieve, beautifulsoup4, bs4  
  Running setup.py install for bs4 ... done
```

pip install: bs4, pandas, pillow, requests, IPython,

START DEVELOPING YOUR BOT

```
import discord

pheme = discord.Client() #Create connection to Discord
TOKEN = XXXXXXXXXX

#when bot is ready
@pheme.event
async def on_ready():
    print('Pheme is online.')

#Upon receiving a message
@pheme.event
async def on_message(message):

    #these variables shorten the code
    reply = message.channel
    msg = message.content
    member = message.author

    #If message is from self, ignore
    if member == pheme.user:
        return

    #If message is from Athena or Minerva (bots), ignore
    bot_ids = {
        "Athena": 716649964230541343,
        "Minerva": 803248289146470442
    }
    if member.id == bot_ids["Athena"]:
        return
    if member.id == bot_ids["Minerva"]:
        return

    #Process message
```

Since I have other bots made by me in this server, I want Pheme to ignore any messages from them or they can get stuck in an infinite loop of replying to each other forever (yes, it happened to me before)

(Continuation)

```
#Info
if msg.startswith('info Pheme'):
    await reply.send('Hello. I am Pheme, the Goddess of rumour,')

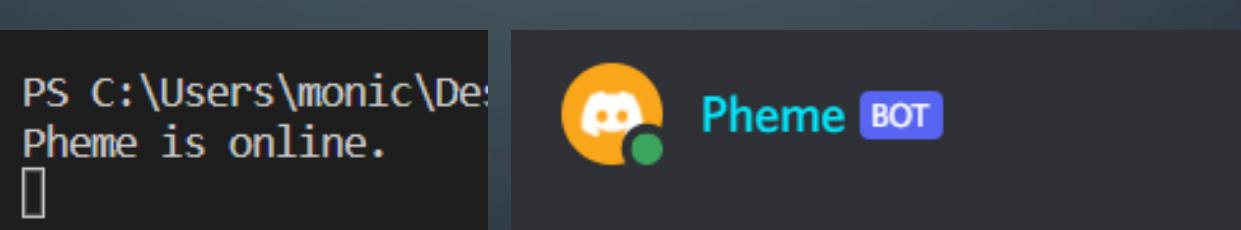
#Shut down
if msg.startswith('terminate Pheme'):
    await reply.send('Going to sleep...')
    await pheme.logout()

# Start bot #####
pheme.run(TOKEN)
```

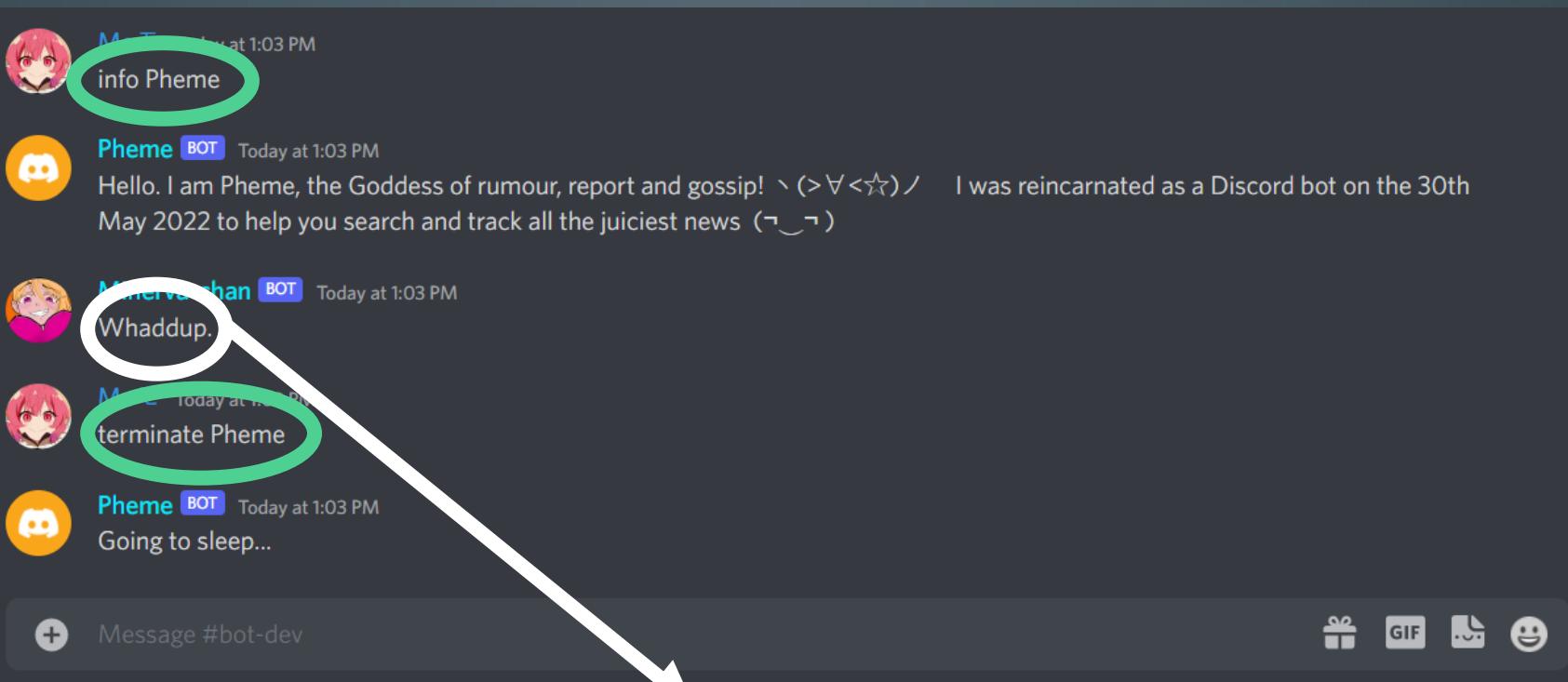
Ideally, you will want to hide this Token as an environment variable. Learn how in the References.

PHEME IS ONLINE!

A screenshot of a terminal window from a code editor. The window title bar shows 'pheme_for_tutorial.py X tracking.py Extension: Python'. Below the title bar, the file path 'D: > Benchtime > pheme_for_tutorial.py > ...' is visible. The terminal tab is selected, showing the following text:
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
PS C:\Users\monic\Desktop\Pheme_bot> & c:/Users/monic/Desktop/Pheme_bot/virtual_env/Scripts/python.exe c:/Users/monic/Desktop/Pheme_bot/pheme.py
Pheme is online.
[]



BUT SHE ONLY HAS 2 FEATURES!



Minerva (one of my other bots)
replies to Pheme because she
was not updated to ignore this
new bot.

THESE ARE ALL THE FEATURES WE WANT TO DEVELOP FOR PHEME

Command:

To show card art:

show + tcg_name+ cardname
ex: show ygo Traptrix Sera
accepted tcg_names are ygo, mtg

Command:

To search for anime/manga status (how many chpt/episodes are out):

search + type + name
ex: search anime summer time
accepted types are anime, manga

Command:

To TRACK a price (games/cards) or status (anime/manga) use the track command:

track + category + name
track manga kaiju 8
accepted categories are anime, manga, digital, physical, ygo, mtg, pkmn

Command:

To search for card prices:

search + tcg_name + cardname
ex: search ygo Traptrix Sera
accepted tcg_names are ygo, mtg, pkmn

To search for game prices:

search + format + game name
ex: search digital gwent
accepted formats are digital, physical

Scheduled task:

Once a day, check if the price/status of any item being tracked decreased/changed and send a message about it in the respective channel.

Scheduled task:

Once a day, check if it's any member's birthday and wish him/her a Happy Birthday

WE WILL DEVELOP THE FUNCTIONS IN SEPARATE FILES, AND THEN IMPORT THEM IN THE MAIN FILE

The screenshot shows a code editor with the following structure:

- EXPLORER** pane:
 - PHEME_BOT** folder
 - __pycache__
 - virtual_env
 - birthday_tracker.py** (highlighted with a green border)
 - pheme.py** (highlighted with a green border)
 - price_tracker.csv** and **status_tracker.csv** (highlighted with a pink border)
 - tracking.py
- pheme.py** (active tab):

```
1 import discord
2 from discord.ext import tasks, commands
3 import os #Import token (on .env file)
#Import custom functions:
4 from birthday_tracker import birthday
5 from tracking import track, price_decrease, status_change, mtg_art, ygo_art, cardprices, gameprices, manga_anime
6 #For ygo card art
7 from PIL import Image
8 from io import BytesIO
9 from IPython import display
```
- tracking.py** (another tab):

```
1 # Import
2 from bs4 import BeautifulSoup
3 import requests
4 from os.path import exists
5 import pandas as pd
6 from datetime import datetime
7 import json
8 from PIL import Image
9 from io import BytesIO
10 from IPython import display
```

Annotations with arrows point to specific parts of the code:

- A green arrow points from the **birthday_tracker.py** file in the Explorer to the **birthday** import statement in the **pheme.py** code.
- A green arrow points from the **tracking.py** tab in the code editor to the **tracking** import statement in the **pheme.py** code.
- A pink arrow points from the **price_tracker.csv** and **status_tracker.csv** files in the Explorer to the **price_decrease** and **status_change** imports in the **pheme.py** code.
- A pink arrow points from the **tracking.py** tab in the code editor to the **mtg_art**, **ygo_art**, **cardprices**, **gameprices**, and **manga_anime** imports in the **pheme.py** code.
- A green arrow points from the **tracking.py** tab in the code editor to the **Image** import statement in the **pheme.py** code.
- A green arrow points from the **tracking.py** tab in the code editor to the **BytesIO** import statement in the **pheme.py** code.
- A green arrow points from the **tracking.py** tab in the code editor to the **display** import statement in the **pheme.py** code.
- A pink arrow points from the **tracking.py** tab in the code editor to the **IPython** import statement in the **pheme.py** code.
- A pink arrow points from the **tracking.py** tab in the code editor to the **display** import statement in the **tracking.py** code.
- A pink arrow points from the **tracking.py** tab in the code editor to the **display** import statement in the **tracking.py** code.

Main file (check the imports above)

Files with functions/features (check the imports bellow and on the right)

These files are byproducts of the features, and are used to track the prices/statuses

SHOW COMMANDS: API CALLING

The image shows two screenshots of API documentation side-by-side.

Left Screenshot (Yugioh Prices API):

- Header:** Yugioh Prices by Brandon Smith • yugiohprices
- Download API Blueprint** button
- INTRODUCTION:** Getting Card and Card Set Images
- REFERENCE:** Checking Card Prices, Getting Set Data, Rising and Falling and Top 100 Lists, Getting Card Data, Getting Card and Card Set Images
- Content:** "Retrieve the image for a specific card using its name" and "Retrieve the image for a specific card set using its name". Each has a "Retrieve the image for a specific card using its name" link.

Right Screenshot (Scryfall API):

- Header:** scryfall.com/docs/api
- Search Bar:** Search for Magic cards
- CORE DOCS:** Overview, Request Formats, Authentication, Colors and Costs, Card Imagery, Layouts and Faces, Frames, Languages, CORS and CSP
- TYPES & METHODS:** (Listed below)
- Table:** Lists parameters for the search API.

PARAMETER	TYPE	ATN	DETAILS
exact	String	0	The exact card name to search for, case insensitive.
fuzzy	String	0	A fuzzy card name to search for.
set	String	0	A set code to limit the search to one set.
format	String	0	The data format to return: json , text , or image . Defaults to json .
face	String	0	If using the image format and this parameter has the value back , the back face of the card will be returned. Will return a 422 if this card has no back face.
version	String	0	The image version to return when using the image format: small , normal , large , png , art_crop , or border_crop . Defaults to large .
pretty	Boolean	0	If true, the returned JSON will be prettified. Avoid using for production code.
- Example Request:** "Retrieve Austere Command using the URL https://api.scryfall.com/cards/named?fuzzy=aust+com"

Yu-Gi-Oh! Cards:
This API receives the exact card name and returns an image object

Magic the Gathering Cards:
This API searches by keywords and returns a dictionary with information about the card whose name best matches your search. From this dictionary, we must extract the card image URL.

THESE FUNCTIONS RECEIVE THE SEARCH KEYWORDS AND RETURN A CARD IMAGE OR IMAGE URL

```
pheme.py pheme_forTutorial.py tracking.py x
tracking.py > ygo_art

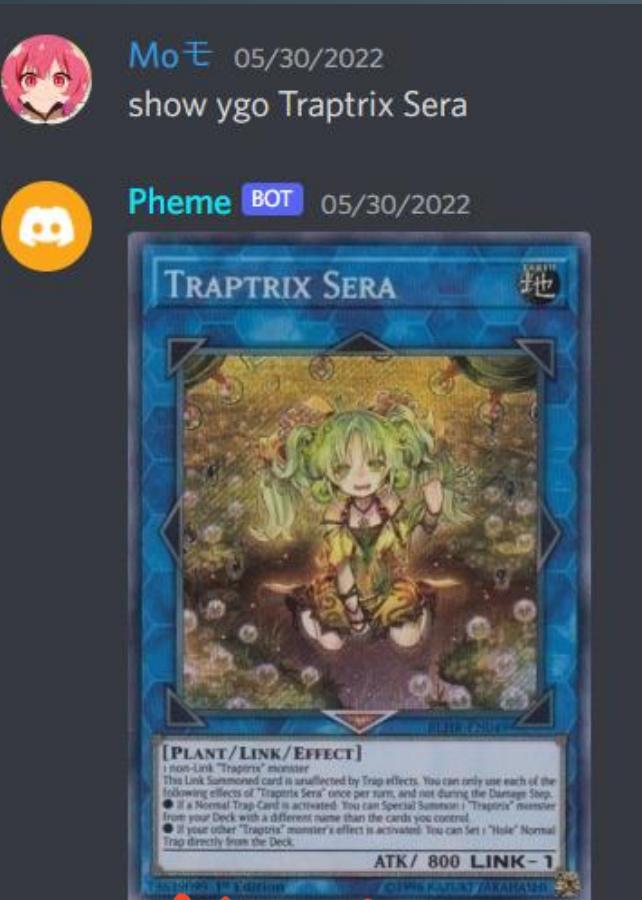
370
371 # GET CARD ART
372
373 def mtg_art(search):
374
375     call_url = "https://api.scryfall.com/cards/named?fuzzy="+search.replace(" ","+")
376     call_card = requests.get(call_url)
377     card_dict = json.loads(call_card.text)
378
379     if call_card.status_code == 404:
380         img = 'Ambiguous name "'+search+'". Add more words to refine your search.'
381     else:
382         img = card_dict['image_uris']['normal']
383
384     return img
385
386 def ygo_art(search):
387
388     call_url = "http://yugiohprices.com/api/card_image/"+search
389     call_card = requests.get(call_url)
390
391     if call_card.status_code == 404:
392         img = 'Ambiguous name "'+search+'". Add more words to refine your search.'
393     else:
394         img = Image.open(BytesIO(call_card.content))
395
396     return img
```

Don't forget to import these functions in the main file...

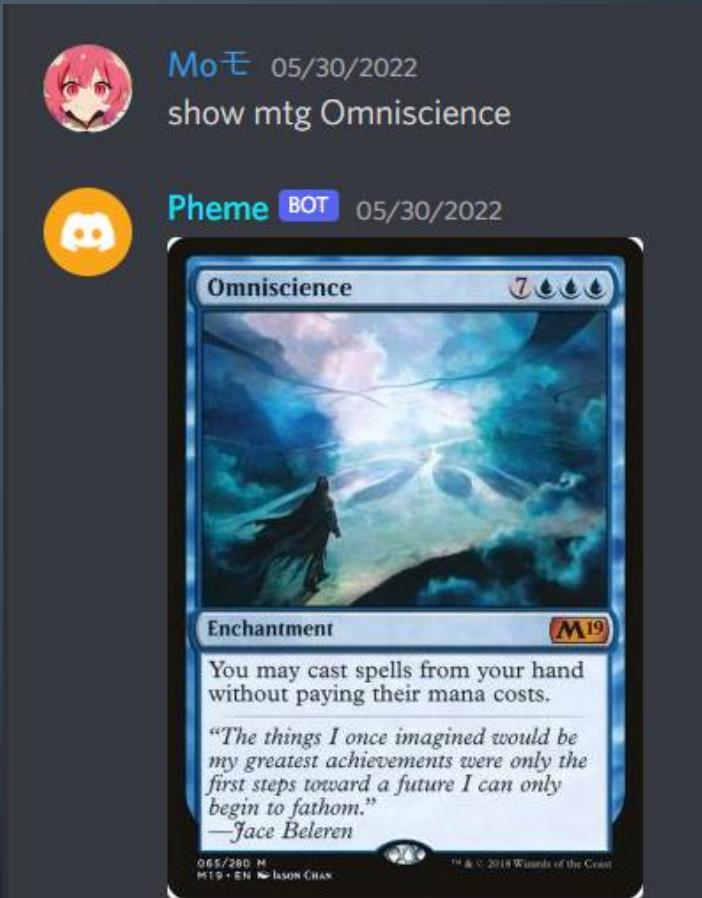
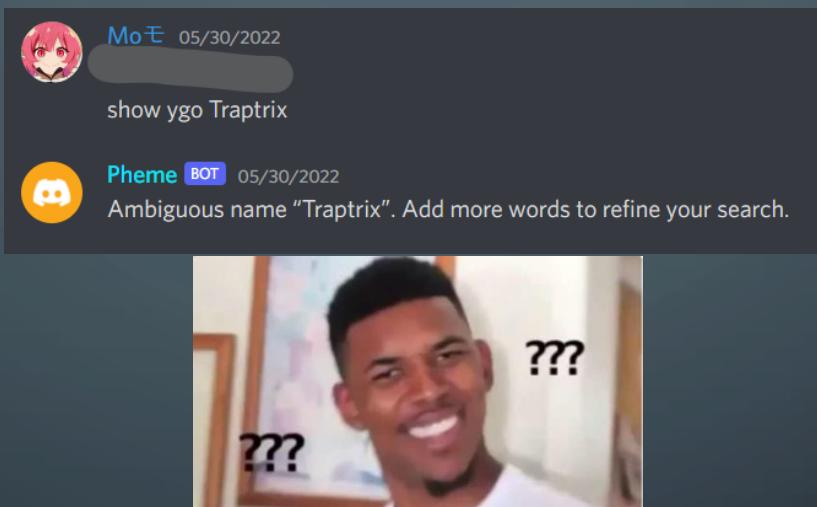
```
pheme.py x pheme_forTutorial.py tracking.py
pheme.py > ...
119 #Show
120 if msg.startswith('show mtg'):
121     search = msg.split('show mtg')[1]
122     img = mtg_art(search)
123     await reply.send(img)
124
125 if msg.startswith('show ygo'):
126     search = msg.split('show ygo')[1]
127     img = ygo_art(search)
128     bytes = BytesIO()
129     img.save(bytes,format='PNG')
130     bytes.seek(0)
131     dfile = discord.File(bytes,filename='img.png')
132     await reply.send(file=dfile)
```

Commands for Pheme. If she reads the keywords, she will call the functions and return the images.

TESTING TIME!



YU-GI-OH!



MAGIC
The Gathering®

SEARCH GAME PRICES WITH WEB SCRAPING

```
def gameprices(type,search):  
  
    if type == 'physical':  
        #NEDGAME  
        search_url = 'https://www.nedgame.nl/zoek/zoek:' + search.replace(" ","_") + '/&sorteer=relevantie'  
        html_search = requests.get(search_url).text  
        soup = BeautifulSoup(html_search,'html.parser')  
        products = soup.find_all("div", attrs={"class": "productshopHeader"})  
        results = []  
        for p in products:  
            title = p.find('div', attrs={'class':'title'}).text.split("\n\n")[1]  
            all_prices = p.find('div', attrs={'class':'buy'})  
            (states,prices) = (all_prices.find_all('div', attrs=['class':'staat']), all_prices.find_all('div', attrs=["class":"currentprice"]))  
            price_info = []  
            for s,pr in zip(states,prices):  
                state = (s.text.replace("Nieuw", "New").replace("Gebruikt", "Used")+' : '+pr.text)  
                price_info.append(state)  
            results.append((title, price_info))  
        return results[:10]  
  
    elif type == 'digital':  
        #STEAM  
        search_url = 'https://store.steampowered.com/search/?term=' + search.replace(" ","+")  
        html_search = requests.get(search_url).text  
        soup = BeautifulSoup(html_search,'html.parser')  
        products = soup.find_all("div", attrs={"class": "responsive_search_name_combined"})  
        results = []  
        for p in products:  
            title = p.find("span", attrs={"class": "title"}).text  
            try:  
                price = p.find("div", attrs={"class": "col search_price responsive_secondrow"}).text.split("\r\n")[1].strip()  
                results.append((title, price))  
            except:  
                pass  
        return results[:10]
```

To help understand which classes you should search with `.find()` and `.find_all()`,
right-click + inspect the web pages



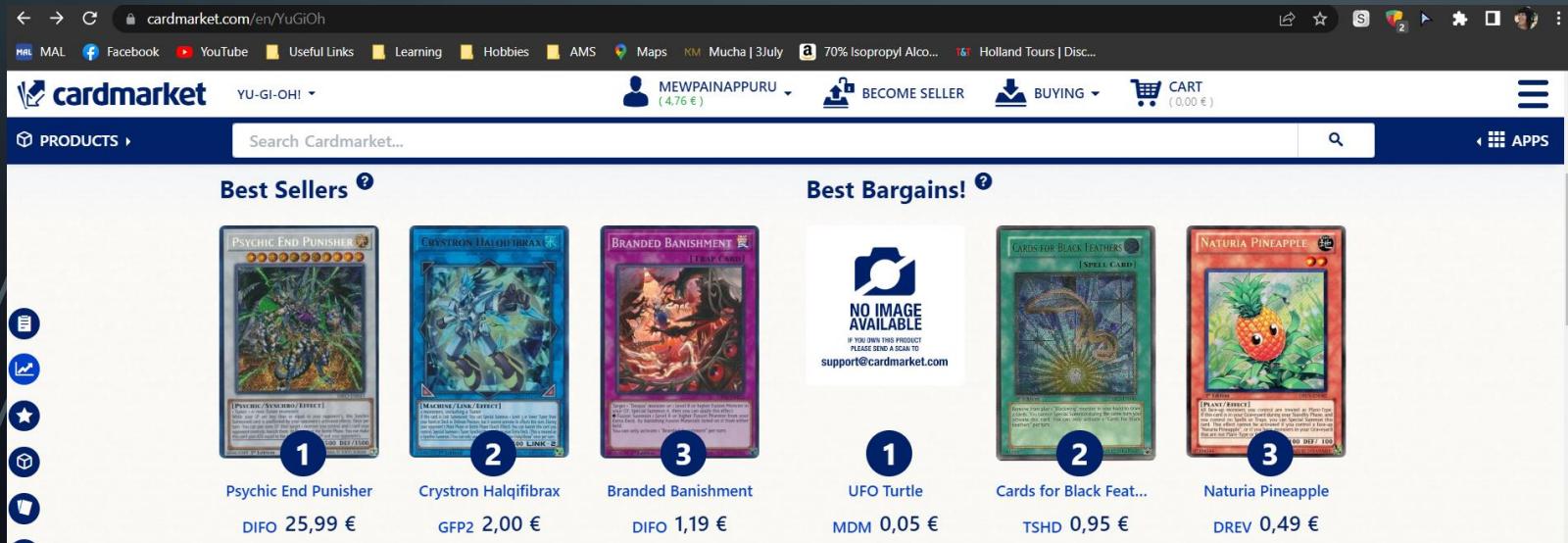
SEARCH CARD PRICES WITH WEB SCRAPING

```
def cardprices(tcg,search):
    results = []
    #CARDMARKET top 15 results
    if tcg == 'mtg':
        search_url = 'https://www.cardmarket.com/en/Magic/Products/Singles?idCategory=1&idExpansion=0&searchString=' + search.replace(" ", "+") + '&onlyAvailable=on&idRarity=0&sortBy=price_asc&perSite=20'
    elif tcg == 'ygo':
        tcg_search = 'YugiOh'
        search_url = 'https://www.cardmarket.com/en/YuGiOh/Products/Singles?idCategory=5&idExpansion=0&searchString=' + search.replace(" ", "+") + '&onlyAvailable=on&idRarity=0&sortBy=price_asc&perSite=20'
    elif tcg == 'pkmn':
        search_url = 'https://www.cardmarket.com/en/Pokemon/Products/Singles?idExpansion=0&searchString=' + search.replace(" ", "+") + '&onlyAvailable=on&idRarity=0&sortBy=price_asc&perSite=30'

    html_search = requests.get(search_url).text
    soup = BeautifulSoup(html_search,'html.parser')
    (cardnames,expansions,prices) = (soup.find_all("div", attrs={"class": "col-10 col-md-8 px-2 flex-column align-items-start justify-content-center"}),soup.find_all("div", attrs={"class": "col-icon small"}),soup.find_all("div", attrs={"class": "col-price pr-sm-2"}))

    for n,e,p in zip(cardnames,expansions,prices):
        name = n.text
        price = p.text
        if (name != 'Name') and (price != "From"):
            expansion = str(e).split('title=')[1].split('"')[1]
            results.append((name,expansion,price))

    return results
```



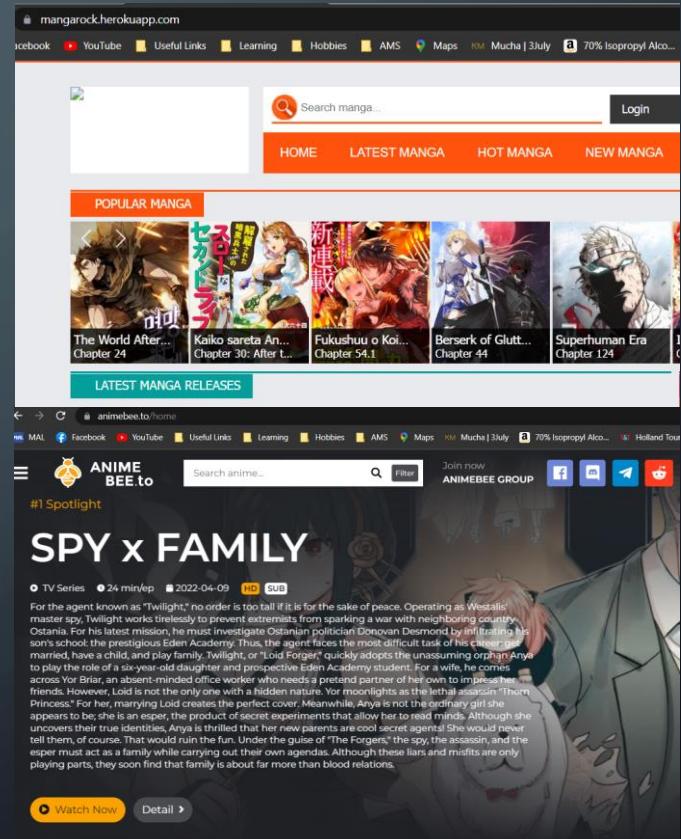
To help understand which classes you should search with .find() and .find_all(), right-click + inspect the web pages

SEARCH ANIME/MANGA STATUS WITH WEB SCRAPING

```
def manga_anime(type, search):
    if type == 'manga':
        search_url = 'https://mangarock.herokuapp.com/search/story/' + search.replace(" ", "_")
        html_search = requests.get(search_url).text
        soup = BeautifulSoup(html_search, 'html.parser')
        series = soup.find_all("div", attrs={"class": "story_item"})
        results = []
        for s in series:
            title = s.find("h3", attrs={"class": "story_name"}).text.split("\n")[1]
            chpt = s.find("em", attrs={"class": "story_chapter"}).text.split('\n')[2].strip()
            results.append((title, chpt))
        return results

    if type == 'anime': #top 4 results
        search_url = 'https://animebee.to/search?keyword=' + search.replace(" ", "+")
        html_search = requests.get(search_url).text
        soup = BeautifulSoup(html_search, 'html.parser')
        series = soup.find_all("div", attrs={"class": "flw-item flw-item-big"})
        results = []
        for s in series:
            title = s.find("h3", attrs={"class": "film-name"}).text.split("\n")[1]
            ep = s.find("div", attrs={"class": "tick-item tick-eps"}).text
            results.append((title, ep))
        return results
```

To help understand which classes you should search with `.find()` and `.find_all()`,
right-click + inspect the web pages



PASSING THESE FUNCTIONS AS COMMANDS TO PHEME...

```
tcgs = ['ygo', 'pkmn', 'mtg']
games = ['physical', 'digital']
animanga = ['anime', 'manga']
categories = tcgs + games + animanga
```

```
#Search
if msg.startswith('search'):
    category = msg.split(" ")[1]
    if category not in categories:
        m = 'Please choose a valid category between "ygo" (cards), "pkmn" (cards), "mtg" (cards), "physical" (games), "digital" (games), "anime" or "manga"'
        await reply.send(m)
    else:
        search = " ".join(msg.split(" ")[2:])
        if category in games:
            m_list = gameprices(category,search)
        elif category in tcgs:
            m_list = cardprices(category,search)
        else:
            m_list = manga_anime(category,search)
        if len(m_list) == 0:
            m = 'No results found for this search.'
            await reply.send(m)
        else:
            for m in m_list:
                await reply.send(m)
```

TESTING TIME!

Mo毛 05/30/2022
search snorlax



Pheme BOT 05/30/2022
Please choose a valid category between "ygo" (cards), "pkmn" (cards), "mtg" (cards), "physical" (games), "digital" (games), "anime" or "manga"

Mo毛 05/30/2022
search ygo Ash blossom

Pheme BOT 05/30/2022
('Ash Blossom & Joyous Spring (V.1 - Super Rare)', 'Prismatic Art Collection', '12,00 €')
('Ash Blossom & Joyous Spring', 'Structure Deck: Soulburner', '16,50 €')
('Ash Blossom & Joyous Spring', 'Legendary Collection Kaiba Mega Pack', '18,00 €')
('Ash Blossom & Joyous Spring', 'Duel Devastator', '18,00 €')
('Ash Blossom & Joyous Spring', 'Shadows in Valhalla', '19,49 €')
('Ash Blossom & Joyous Spring', 'Duel Power', '19,50 €')
('Ash Blossom & Joyous Spring (V.2 - Premium Gold Rare)', 'Maximum Gold', '19,60 €')
('Ash Blossom & Joyous Spring (V.1 - Premium Gold Rare)', 'Maximum Gold', '19,99 €')
('V.2 - Secret Rare)', 'Prismatic Art Collection', '40,00 €')
('V.3 - Secret Rare)', 'Prismatic Art Collection', '45,00 €')
'Maximum Crisis', '50,00 €')



Mo毛 05/30/2022
search physical cyberpunk

Pheme BOT 05/30/2022
('Sense - A Cyberpunk Ghost Story', ['New: € 44,99'])
('Cyberpunk 2077 Day One Edition + Keychain', ['New: € 39,99', 'Used: € 29,99'])
('Cyberpunk 2077 Day One Edition + Keychain', ['New: € 55,-'])
('Cyberpunk 2077 Day One Edition + Keychain', ['New: € 39,99'])
('Cyberpunk 2077 Nendoroid - Male V', ['New: € 48,-'])
('Cyberpunk 2077 Nendoroid - Female V', ['New: € 48,-'])
('Cyberpunk 2077: Johnny Silverhand Nendoroid', ['New: € 58,-'])
('Cyberpunk 2077 - Metro Life Puzzle (1000pcs)', ['Pre-Order: € 24,99'])
('Cyberpunk 2077 G. Takemura PVC Statue', ['New: € 44,99'])
('Cyberpunk 2077 Deluxe Nendoroid - Female V', ['New: € 78,-'])

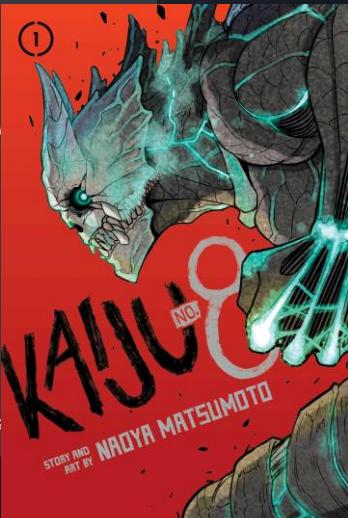


Mo毛 Today at 2:39 PM
search manga kaiju

Pheme BOT Today at 2:39 PM
('100-Nin No Eiyuu O Sodateta Saikyou Yogensha Wa, Boukensha Ni Natte Mo Sekaijuu No Deshi Kara Shitayo', 'Chapter 40')

Minerva-chan BOT Today at 2:39 PM
Oh, hello there.

Pheme BOT Today at 2:39 PM
('Kaiju No. 8', 'Chapter 62')
('Otome Monster Caramelize', 'Chapter 25')
('Hakaijuu', 'Vol.19 Chapter 74: Last Words')
('Saikyou No Mamono Ni Narumichi Wo Tadoru Ore, Isekaijuu De Zamaa Wo Shikkou Suru', 'Chapter 12: Epis...')



FUNNY INTERACTIONS

('Kaijuu Iro No Shima', 'Chapter 11')

('Kaijuu No Tokage (Fukuchi Kamio)', 'Chapter 7')



Minerva-chan BOT Today at 2:39 PM

Oh, hello there.



Pheme BOT Today at 2:39 PM

('Kaijuu No Kodomo', "Vol.5 Extra: A Traveler's Tale")

('Kaijuu No Shiiku lin', 'Chapter 13.6: Side Story 2: No Change On Thw Cherry Blossom Front [End]')

('Mahou Shoujo Of The End Vs Hakaijuu Vs Versus Earth', 'Ch.4 : The Final Chapter Of Shock!![End]')

('Sekaiju No Meikyuu li ~Rikka No Shoujo~', 'Vol.2 Ch.12')

('Kaijuu No Buki Shokunin', 'Chapter 1: Chapter 1')

('Dark Cherry To Shoujo A', "Vol.1 Chapter 9: The Girls' Party")

('Etrian Odyssey li', 'Vol.1 Ch.5 : The Second Stratum')

('Kaijuu No Tail', 'Vol.1 Ch.1 : Monster Daughter')



BlueRaven Today at 2:39 PM

Minerva trying to make herself noticed against the more advanced bot



@**BlueRaven** Minerva trying to make herself noticed against the more advanced bot

Minerva-chan BOT Today at 2:39 PM

Yes, my little pogchamp?



BlueRaven Today at 2:40 PM

stfu



Minerva-chan BOT Today at 2:40 PM

(¬_¬;)

TRACKING FEATURE

- Now we want to be able to save the current status/price of an item, to later compare with a future status and warn the user of any changes.
- This feature requires that we save the information of any item we want to track to a file, and regularly go through the items in the file and search for any changes online.
- We will need 1) A command to 'start' tracking an item (add it to our file) and 2) An automatic task that searches for changes every x hours.

TRACK FUNCTION

```
#START TRACKING

def track(cat,name):

    tcgs = [ 'ygo','pkmn','mtg']
    games = [ 'physical','digital']
    animanga = [ 'anime','manga']

    try:
        if cat in tcgs:
            results = cardprices(cat,name)
            data = []
            for r in results:
                if r[0] == name:
                    data.append(r)
            (log,current,previous) = save_price(data[0],cat)

        elif cat in games:
            results = gameprices(cat,name)
            data = results[0]
            (log,current,previous) = save_price(data,cat)

        elif cat in animanga:
            results = manga_anime(cat,name)
            data = results[0]
            (log,current,previous) = save_status(data,cat)

        return (log,current,previous)

    except:
        return ('Your search is too ambiguous. If you are trying to track a card, you must match the name on the card exactly.',0,0)
```

The diagram illustrates the flow of data and logic in the `track` function. It uses arrows to point from specific code snippets to explanatory text.

- A green arrow points from the line `results = cardprices(cat,name)` to the text "Search functions created previously".
- A green arrow points from the line `(log,current,previous) = save_price(data[0],cat)` to the text "Save functions: they save the first search result (the best match) to our tracking files".
- A blue arrow points from the line `return (log,current,previous)` to the text "The function returns the log message from the save function, the current price/status and the previous price/status of the item".

SAVE PRICE FUNCTION (1 / 3)

Data is the output from the search functions.

Only cards have expansions.

```
def save_price(data,category):
    tcgs = ['ygo','pkmn','mtg']
    games = ['physical','digital']

    if category in tcgs:
        name = data[0]
        expansion = data[1]
        lowest_price = float(data[2].split(' €')[0].replace(',','.').replace('.','')) # Extracting the price as float from the string (output of the search functions). If "free", price is 0.0EUR.

    elif category in games:
        name = data[0]
        expansion = ''
        if category == 'physical':
            if len(data[1]) ==1:
                data[1][0] = data[1][0].replace('Pre-Order','New')
                lowest_price = float(data[1][0].split("New: € ")[1].replace(',','.').replace('-','00')) # Extracting the price as float from the string (output of the search functions). If "free", price is 0.0EUR.
            elif len(data[1]) ==2:
                lowest_price = float(data[1][1].split("Used: € ")[1].replace(',','.').replace('-','00')) # Extracting the price as float from the string (output of the search functions). If "free", price is 0.0EUR.
            else:
                print("Cannot track this game because it does not have a price.")
                return #to exit function
        elif category == 'digital':
            if data[1] == 'Free To Play':
                lowest_price = 0.0
            else:
                lowest_price = float(data[1].split('€')[0].replace(',','.').replace('.','')) # Extracting the price as float from the string (output of the search functions). If "free", price is 0.0EUR.
```

Extracting the price as float from the string (output of the search functions). If “free”, price is 0.0EUR.

SAVE PRICE FUNCTION (2/3)

```
#Create file for the first time if it doesnt exist, get data from file
if exists("price_tracker.csv") == False:
    #print("File doesnt exist in directory. Creating file for the first time...")
    file = open("price_tracker.csv","w")
    file.write("Name|Category|Lowest_Price|Expansion|DateChecked")
    file.close()

#Open file as df
df_prices = pd.read_csv("price_tracker.csv",delimiter="|")

#Check if cardname exists in file
df_check = df_prices[df_prices['Name']==name]

if df_check.empty:
    previous = 999999
    date_time_previous = ''
else:
    previous = df_check['Lowest_Price'].iloc[0]
    date_time_previous = df_check['DateChecked'].iloc[0]

#get datetime (now)
now = datetime.now() # current date and time
date_time = now.strftime("%m/%d/%Y, %H:%M:%S")

#create new df w/update
update = {'Name': name, 'Category': category, 'Lowest_Price': lowest_price, 'Expansion': expansion, 'DateChecked':date_time}
df_update = pd.DataFrame(update, index=[0])
```

Create tracking file for the first time if it doesn't exist.
Else, open it and check if the item we asked the Bot to track is already being tracked.

SAVE PRICE FUNCTION (3/3)

```
#Update file based on the newest price search
# Update df
if previous == 999999: #Insert the new row from df_update in df_prices
    df_prices = pd.concat([df_prices, df_update])
else: #Replace the whole row in df_prices with the new row in df_update
    df_prices.update(df_prices[['Name']].merge(df_update, 'left'))

# Save changes to file
df_prices.to_csv("price_tracker.csv","|",index=False)

if category in tcgs:
    if previous == 999999:
        msg = "The lowest price for " + name + " is " + str(lowest_price) +'€ and corresponds to the expansion '+expansion+'. You never saved prices for this card before."
    else:
        msg = "The lowest average price for " + name + " is " + str(lowest_price) +'€ and corresponds to the expansion '+expansion+'. The previous time you tracked, the lowest price was ' + str(previous) +'€'

elif category in games:
    if previous == 999999:
        msg = "The lowest price for " + name + " is " + str(lowest_price) +'€. You never saved prices for this game before.'
    else:
        msg = "The lowest average price for " + name + " is " + str(lowest_price) +'€. The previous time you tracked, on '+date_time_previous+' , the lowest price was ' + str(previous) +'€'

return (msg,lowest_price,previous)
```

If it's the first time we track the item:
add it to the file as a new row.
Else: update the record with the latest
search results.

Then, generate log messages for each scenario.

SAVE STATUS FUNCTION

- The logic is the same, but it saves the items in a different file.

```
def save_status(data,category):  
  
    name = data[0]  
    status = data[1]  
  
    #Create file for the first time if it doesnt exist, get data from file  
    if exists("status_tracker.csv") == False:  
        #print("File doesnt exist in directory. Creating file for the first time...")  
        file = open("status_tracker.csv","w")  
        file.write("Name|Category|Status|DateChecked")  
        file.close()  
  
    #Open file as df  
    df_status = pd.read_csv("status_tracker.csv",delimiter="|")  
  
    #Check if cardname exists in file  
    df_check = df_status[df_status['Name']==name]  
  
    if df_check.empty:  
        previous = 'N/A'  
        date_time_previous = ''  
    else:  
        previous = df_check['Status'].iloc[0]  
        date_time_previous = df_check['DateChecked'].iloc[0]  
  
    #Get datetime (now)  
    now = datetime.now() # current date and time  
    date_time = now.strftime("%m/%d/%Y, %H:%M:%S")  
  
    #Create new df w/update  
    update = {'Name': name, 'Category': category, 'Status': status, 'DateChecked':date_time}  
    df_update = pd.DataFrame(update, index=[0])
```

I will skip the extra comments, since it is exactly the same thinking process...

```
#Update file based on the newest price search  
# Update df  
if previous == 'N/A': #Insert the new row from df_update in df_prices  
    df_status = pd.concat([df_status, df_update])  
else: #Replace the whole row in df_prices with the new row in df_update  
    df_status.update(df_status[['Name']].merge(df_update, 'left'))  
  
# Save changes to file  
df_status.to_csv("status_tracker.csv","|",index=False)  
  
if previous == 'N/A':  
    msg = "The current status for " + name + " is " + status +'. You never saved the status of this series before.'  
else:  
    msg = "The current status for " + name + " is " + status +'. The previous time you tracked, on '+date_time_previous+', the status was ' + previous+'.  
  
return (msg,status,previous)
```

TIME TO ADD IT AS A COMMAND FOR PHEME AND TEST!

```
#Track
if msg.startswith('track'):
    category = msg.split(" ")[1]
    if category not in categories:
        m = 'Please choose a valid category between "ygo" (cards), "pkmn" (cards), "mtg" (cards), "physical" (games), "digital" (games), "anime" or "manga"'
    else:
        search = " ".join(msg.split(" ")[2:])
        m = track(category,search)[0]
    await reply.send(m)
```



 Mo毛 05/30/2022
track digital witcher 3

 Pheme BOT 05/30/2022
The lowest price for The Witcher® 3: Wild Hunt is 29.99€. You never saved prices for this game before.

 Mo毛 Today at 3:13 PM
track Traptrix Sera

 Pheme BOT Today at 3:13 PM
Please choose a valid category between "ygo" (cards), "pkmn" (cards), "mtg" (cards), "physical" (games), "digital" (games), "anime" or "manga"

 Mo毛 Today at 3:13 PM
track ygo Traptrix

 Pheme BOT Today at 3:13 PM
Your search is too ambiguous.

 Mo毛 05/30/2022
track ygo Traptrix Sera

 Pheme BOT 05/30/2022
The lowest average price for Traptrix Sera is 11.0€ and corresponds to the expansion Battles of Legend: Hero's Revenge. The previous time you tracked, on 05/30/2022, 16:06:40, the lowest price was 11.0€.

TASKS: AUTOMATICALLY SEARCH FOR CHANGES EVERY DAY

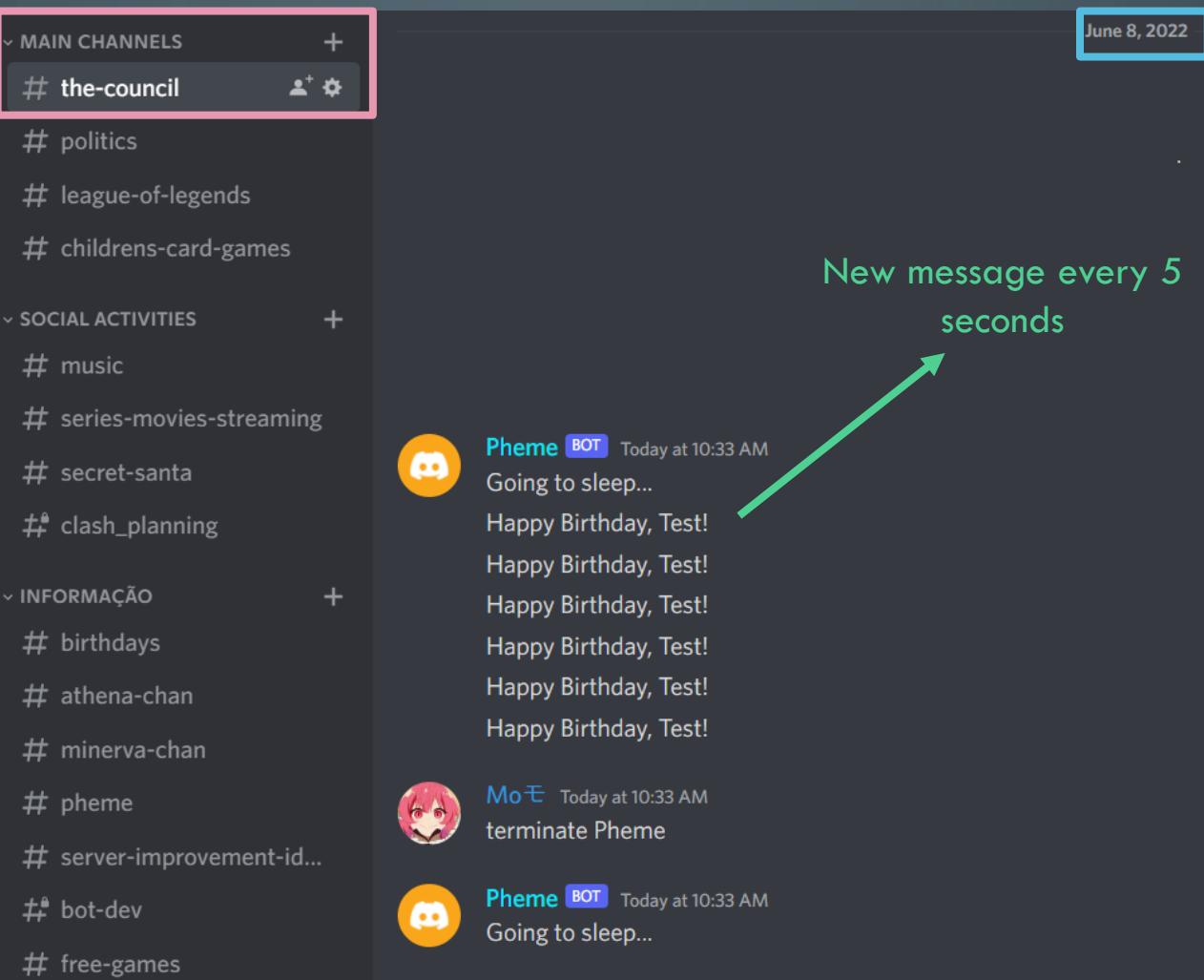
```
File: birthday_tracker.py
1  from datetime import datetime
2
3  def birthday():
4
5      bdays = [
6          ("Burn", '24/07', 228494443668439040),
7          ("Momo", '20/11', 233955951323906048),
8          ("Sukoi", '06/09', 234424901157650432),
9          ("BlueRaven", '06/11', 233993911624794123),
10         ("Daniel", '29/06', 243214755785867264),
11         ("Coarse", '17/04', 319503417783615498),
12         ("Test", '08/06', 319503417783615498)
13     ]
14
15     now = datetime.now()
16     today = now.strftime('%d/%m')
17
18     msg_list = []
19
20     for user,date,id in bdays:
21         if date == today:
22             msg = 'Happy Birthday, ' + user + '!'
23             msg_list.append(msg)
24
25     return msg_list
```

Collecting the name of the channels where Pheme has to reply after the task is over.

```
#Import custom functions:
from birthday_tracker import birthday
```

```
File: pheme.py
21  @pheme.event
22  async def on_ready():
23      print('Pheme is online.')
24      botdev_channel = pheme.get_channel(804400647235043379)
25      tcg_channel = pheme.get_channel(858715632965779526)
26      main_channel = pheme.get_channel(234416220705783808)
27
28  #Birthday tracker
29  @tasks.loop(seconds=5) → Loop time
30  async def birthday_loop(): → Function that is going
31      reply_list = birthday() → to loop every X
32      if len(reply_list) != 0: → seconds
33          for m in reply_list:
34              await main_channel.send(m)
35
36  birthday_loop.start() → Start the loop
```

TESTING BIRTHDAY FEATURE



CHECK PRICE DECREASE FUNCTION

```
def price_decrease(cat):

    df_prices = pd.read_csv("price_tracker.csv",delimiter="|")
    df_cat = df_prices[df_prices['Category']==cat]
    names = list(df_cat['Name'].unique())
    msg_list = []
    for n in names:
        try:
            resultn = track(cat,n)
            if resultn[1] < resultn[2]:
                msg = "The price of "+n+" DECREASED from "+ str(resultn[2]) + "€ to "+str(resultn[1])+'€.'
                msg_list.append(msg)
        except:
            pass
    return msg_list
```

CHECK STATUS CHANGE FUNCTION

```
def status_change(cat):

    type = 'episode' if cat == 'anime' else 'chapter'
    df_prices = pd.read_csv("status_tracker.csv",delimiter="|")
    df_cat = df_prices[df_prices['Category']==cat]
    names = list(df_cat['Name'].unique())
    msg_list = []
    for n in names:
        try:
            resultn = track(cat,n)
            if resultn[1] != resultn[2]:
                msg = "There is a new "+type+" of "+n+"! The status CHANGED from "+ str(resultn[2]) + " to "+str(resultn[1])+'.'
                msg_list.append(msg)
        except:
            pass

    return msg_list
```

ADD IT AS A TASK FOR PHEME

```
@tasks.loop(hours=10)
async def trackers_and_checkers():
    for cat in games:
        m_list = price_decrease(cat)
        if len(m_list)==0:
            m = 'No price decrease found in any product from the ' +cat+ ' category.'
            await main_channel.send(m)
        else:
            for m in m_list:
                await main_channel.send(m)
    for cat in tcgs:
        m_list = price_decrease(cat)
        if len(m_list)==0:
            m = 'No price decrease found in any product from the ' +cat+ ' category.'
            await tcg_channel.send(m)
        else:
            for m in m_list:
                await tcg_channel.send(m)
    for cat in animanga:
        m_list = status_change(cat)
        if len(m_list)==0:
            m = 'No status change detected in any series from the ' +cat+ ' category.'
            await main_channel.send(m)
        else:
            for m in m_list:
                await main_channel.send(m)
trackers_and_checkers.start()
```

UPON TURNING ON PHEME, SHE AUTOMATICALLY SENDS MESSAGES TO THE CORRECT CHANNELS

childrens-card-games



Pheme BOT Today at 11:00 AM

The price of Traptrix Sera DECREASED from 12.0€ to 9.99€.

The price of Klefki (FFI 73) DECREASED from 2.0€ to 0.02€.

The price of Ancient Copper Dragon DECREASED from 25.0€ to 24.95€.

the-council



Pheme BOT Today at 10:56 AM

No price decrease found in any product from the physical category.

The price of Cyberpunk 2077 DECREASED from 60.0€ to 59.99€.

There is a new episode of Ya Boy Kongming!! The status CHANGED from EP 6/12 to EP 7/12.

There is a new chapter of Kaiju No. 8! The status CHANGED from Chapter 61 to Chapter 62.

REMINDER TO INCREASE THE TASK LOOP TIME BEFORE YOU START TESTING OTHER FEATURES

the-council

 Pheme BOT Today at 3:59 PM
No price decrease found in any product from the physical category.
No price decrease found in any product from the digital category.
No status change detected in any series from the anime category.
No status change detected in any series from the manga category.
No price decrease found in any product from the physical category.
No price decrease found in any product from the digital category.
No status change detected in any series from the anime category.
No status change detected in any series from the manga category.
No price decrease found in any product from the physical category.
No price decrease found in any product from the digital category.
No status change detected in any series from the anime category.
No status change detected in any series from the manga category.
No price decrease found in any product from the physical category.
No price decrease found in any product from the digital category.
No status change detected in any series from the anime category.

 Burn Today at 3:59 PM


 Pheme BOT Today at 3:59 PM
No status change detected in any series from the manga category.
No price decrease found in any product from the physical category.
No price decrease found in any product from the digital category.

 BlueRaven Today at 3:59 PM
wow, this bot is deranged

 Pheme BOT Today at 3:59 PM
No status change detected in any series from the anime category.
No status change detected in any series from the manga category.
No price decrease found in any product from the physical category.
No price decrease found in any product from the digital category.
No status change detected in any series from the anime category.
No status change detected in any series from the manga category.
No price decrease found in any product from the physical category.
No price decrease found in any product from the digital category.
No status change detected in any series from the anime category.
No status change detected in any series from the manga category.
No price decrease found in any product from the physical category.
No price decrease found in any product from the digital category.
No status change detected in any series from the anime category.

No status change detected in any series from the manga category.

REMINDER TO INCREASE THE TASK LOOP TIME BEFORE YOU START TESTING OTHER FEATURES

 Daniel Today at 4:00 PM
This is what an apocalypse in production looks like
 2

 Pheme BOT Today at 4:00 PM
No status change detected in any series from the manga category.

 Daniel Today at 4:00 PM


 Pheme BOT Today at 4:00 PM
No price decrease found in any product from the physical category.
No price decrease found in any product from the digital category.
No status change detected in any series from the anime category.
No status change detected in any series from the manga category.
No price decrease found in any product from the physical category.
No price decrease found in any product from the digital category.
No status change detected in any series from the anime category.

 Mo毛 Today at 4:01 PM
 GIF

+ Message #the-council

 Pheme BOT Today at 4:01 PM
No status change detected in any series from the manga category.

 Mo毛 Today at 4:01 PM
terminate Pheme

 Pheme BOT Today at 4:01 PM
No price decrease found in any product from the physical category.
No price decrease found in any product from the digital category.
Going to sleep...

 @Pheme Going to sleep...
Burn Today at 4:01 PM

 GIF
NIGHT NIGHT

2

MY HUSBAND REMINDING ME THAT THE CLIENT IS ALWAYS RIGHT...

- BlueRaven Today at 11:03 AM
Track physical Resident Evil Village
- Mo毛泽 Today at 11:03 AM
- BlueRaven Today at 11:04 AM
Track physical Resident Evil 8
- Mo毛泽 Today at 11:10 AM
track physical Resident Evil 8
- Pheme BOT Today at 11:10 AM
Your search is too ambiguous. If you are trying to track a card, you must match the name on the card exactly.
- Mo毛泽 Today at 11:10 AM
mas tu usaste Track
em vez de track
so its your bad
- BlueRaven Today at 11:11 AM
maybe she should not be case sensitive
 1
dont blame the user
- “but you used Track instead of track”

MAKING ALL COMMANDS CASE SENSITIVE...

```
#Info  
if msg.lower().startswith('info pheme'): ...  
  
#Track  
if msg.lower().startswith('track'):  
    category = msg.lower().split(" ")[1]
```

```
#Search  
if msg.lower().startswith('search'):  
    category = msg.lower().split(" ")[1]
```

```
#Show  
if msg.lower().startswith('show mtg'):  
    search = msg[9:]
```

```
if msg.lower().startswith('show ygo'):  
    search = msg[9:]
```

```
#Shut down  
if msg.lower().startswith('terminate pheme'): ...
```

TEST CASE SENSITIVITY IN ALL FEATURES

 Moモ Today at 11:32 AM
TRACK PhysiCAL Resident Evil Village

 Pheme BOT Today at 11:32 AM
The lowest average price for Resident Evil VIII Village is 33.0€. The previous time you tracked, on 06/08/2022, 11:31:44, the lowest price was 33.0€.

 Moモ Today at 11:32 AM
SeaRCH DIGITaL WitchER 3

 Pheme BOT Today at 11:32 AM
('The Witcher® 3: Wild Hunt', '29,99€')
('The Witcher 3: Wild Hunt - Blood and Wine', '19,99€')
('The Witcher 3: Wild Hunt - Game of the Year Edition', '49,99€')
('GWENT: The Witcher Card Game', 'Free To Play')
('The Witcher 3: Wild Hunt - Hearts of Stone', '9,99€')

 Moモ Today at 11:39 AM
searchH MTG cut a Deal

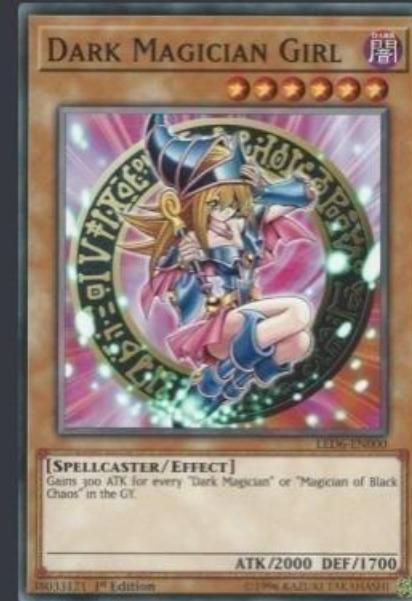
 Pheme BOT Today at 11:39 AM
('Cut a Deal', "Commander Legends: Battle for Baldur's Gate", '0,07 €')

 Moモ Today at 11:39 AM
TRACK Anime Summer TIME

 Pheme BOT Today at 11:39 AM
The current status for Summer Time Rendering is EP 5/25. You never saved the status of this series before.

 Moモ Today at 11:25 AM
ShOW YgO Dark MAGician GIRL

 Pheme BOT Today at 11:25 AM



 Moモ Today at 11:40 AM
TERMINATE PHEME

 Pheme BOT Today at 11:40 AM
Going to sleep...

WEB HOSTING

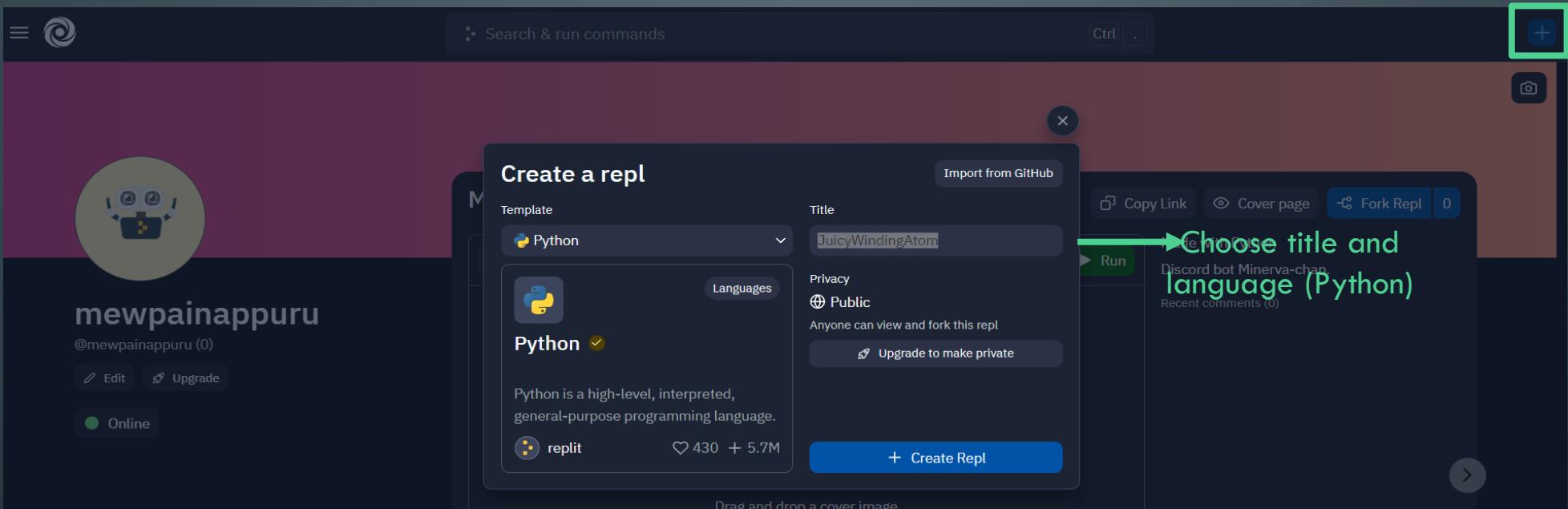
We need a platform to host our bot's code, such as:



And another platform to monitor and ping our bot so it stays online at all times:



WEB HOSTING PHEME



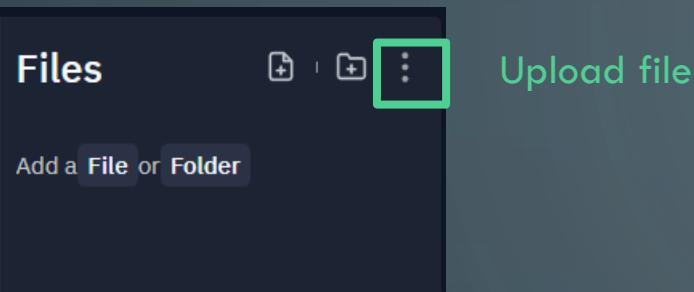
Click to create new
project

```
main.py x
1 Not sure what to do? Run some examples (start typing to dismiss)

Pheme is empty
```

The screenshot shows a code editor window with a single file named 'main.py'. The file contains a single line of text: '1 Not sure what to do? Run some examples (start typing to dismiss)'. Below the code editor, the text 'Pheme is empty' is displayed. The interface includes standard code editor features like a file tree on the left and a console/shell tab on the right.

WEB HOSTING PHEME (FOR FREE)



Name	Date modified	Type	Size
__pycache__	08/06/2022 10:54	File folder	
virtual_env	08/06/2022 10:32	File folder	
birthday_tracker	08/06/2022 10:41	Python Source File	1 KB
pHEME	08/06/2022 11:59	Python Source File	6 KB
price_tracker	08/06/2022 11:40	Microsoft Excel Co...	1 KB
status_tracker	08/06/2022 11:40	Microsoft Excel Co...	1 KB
tracking	30/05/2022 16:39	Python Source File	16 KB

A screenshot of a code editor showing the content of the 'pheme.py' file. The file is part of a project structure shown in the sidebar, where it is currently selected. The code itself is as follows:

```
1 import discord
2 from discord.ext import tasks, commands
3 import os #Import token (on .env file)
4 #Import custom functions:
5 from birthday_tracker import birthday
6 from tracking import track, price_decrease, status_change,
7 mtg_art, ygo_art, cardprices, gameprices, manga_anime
8 #For ygo card art
9 from PIL import Image
10 from io import BytesIO
11 from IPython import display
12 pheme = discord.Client() #Create connection to Discord
```

HIDE TOKEN AS ENV VARIABLE

The terminal window displays the following Python code:

```
pheme.py ×  
1 import discord  
2 from discord.ext import tasks, commands  
3 import os #Import token (on .env file)  
  
# Start bot #####  
pheme.run(os.getenv('TOKEN'))
```

The code imports discord, discord.ext, and os. It uses os.getenv('TOKEN') to retrieve the token from the environment variable TOKEN.

Below the terminal, there are two screenshots of a file explorer interface. The first screenshot shows a file named "pheme.py" selected, and the second screenshot shows a "Secrets" page where the token is stored under the key "TOKEN".

TURN PHEME ON THROUGH REPLIT



Files main.py

```
1 import discord
2 from discord.ext import tasks, commands
3 import os #Import token (on .env file)
```

First, rename
pheme.py to main.py

**If replit is unable to automatically
install some of the modules...**

`ModuleNotFoundError: No module named 'PIL'`

Then use

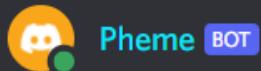
```
from os import system  
system('pip install Pillow')
```

Before

```
from PIL import Image
```

```
Console Shell

Looking in indexes: https://package-proxy.replit.com/pypi/simple/
Collecting Pillow
  Downloading https://package-proxy.replit.com/pypi/packages/a3/51/ea188
62c6a1d8a34aeee643545b6b69bbaff67a1430360d2081833526504/Pillow-9.1.1-cp3
8-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1 MB)
|██████████| 3.1 MB 5.4 MB/s
WARNING: pip is using a content-addressable pool to install files from.
This experimental feature is enabled through --use-feature=content-addre
ssable-pool and it is not ready for production.
Installing collected packages: Pillow
Successfully installed Pillow-9.1.1
Pheme is online.
```



SETTING UP A WEB SERVER FOR PHEME

PHEME is now online, but only until you close your repl.it browser tab...

To keep it running for longer in the background (even after you close your browser) you need to create a web server.

If you do, PHEME will keep running until 1h after her last request, and then go to sleep.

```
keep_running.py x
1 from flask import Flask
2 from threading import Thread
3
4 app = Flask('')
5
6 @app.route('/')
7 def home():
8     return "PHEME is running!"
9
10 def run():
11     app.run(host='0.0.0.0', port=8080)
12
13 def keep_running():
14     t = Thread(target=run)
15     t.start()
```

Web server code

Import server

```
main.py x
1 import discord
2 from discord.ext import tasks, commands
3 import os #Import token (on .env file)
4 from os import system
5 system('pip install Pillow')
6 #Import custom functions:
7 from birthday_tracker import birthday
8 from tracking import track, price_decrease, status_change,
mtg_art, ygo_art, cardprices, gameprices, manga_anime
9 #For ygo card art
10 from PIL import Image
11 from io import BytesIO
12 from IPython import display
13
14 from keep_running import keep_running #Server
```

Call server

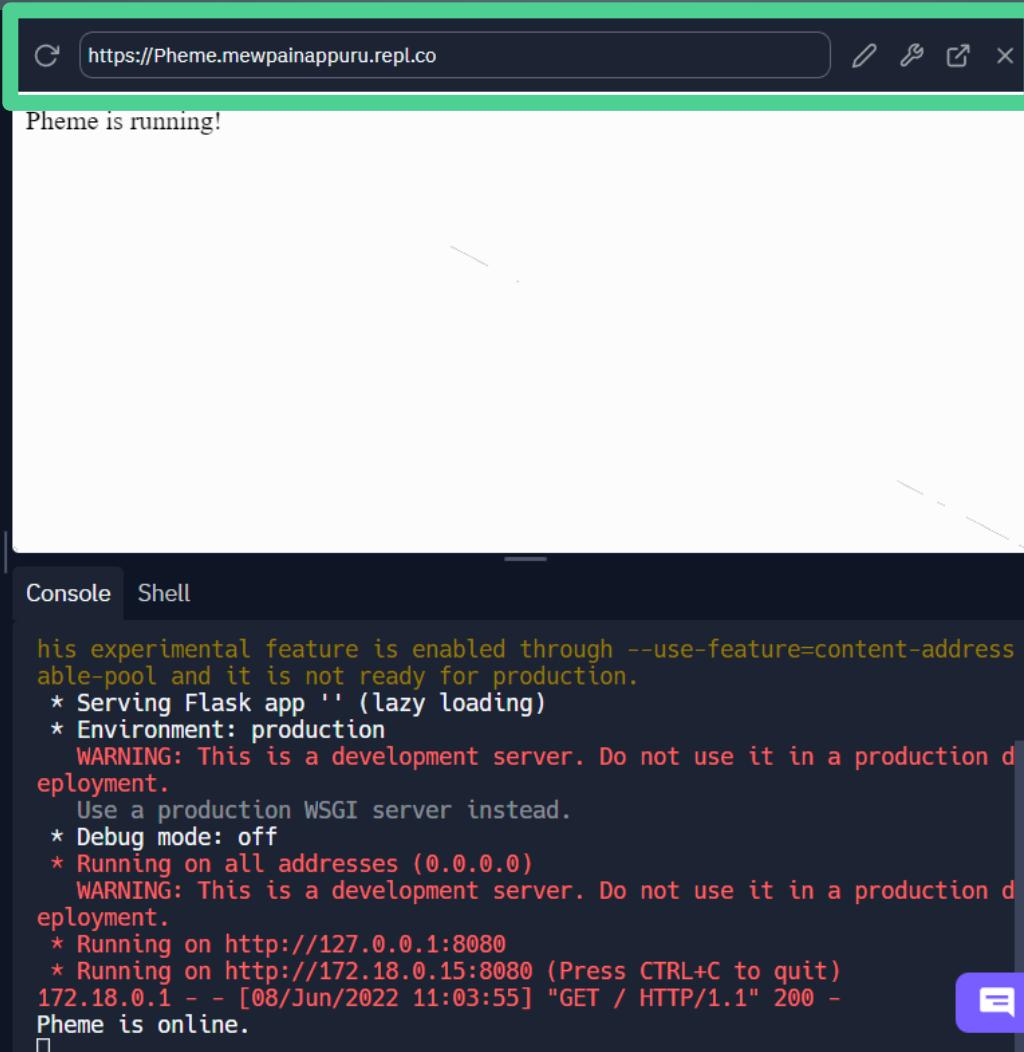
```
main.py x
136
137 #Shut down
138 if msg.lower().startswith('terminate pHEME'):
139     await reply.send('Going to sleep...')
140     await pHEME.logout()
141
142 ######
143 keep_running()
144 ######
145 pHEME.run(os.getenv('TOKEN'))
```

GET THE WEB SERVER URL

Upon hitting

▶ Run

You will see a new window (the web server) above the console:



The screenshot shows a terminal window with two tabs: "Console" and "Shell". Above the terminal, a web browser window is open, displaying a green header bar with the URL <https://Pheme.mewpainappuru.repl.co>. The main content area of the browser says "Pheme is running!". Below the browser is the terminal window. The terminal output is as follows:

```
his experimental feature is enabled through --use-feature=content-addressable-pool and it is not ready for production.
 * Serving Flask app '' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on all addresses (0.0.0.0)
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://127.0.0.1:8080
 * Running on http://172.18.0.15:8080 (Press CTRL+C to quit)
172.18.0.1 - - [08/Jun/2022 11:03:55] "GET / HTTP/1.1" 200 -
Pheme is online.
```

Copy the server URL
(this is the URL we need
to ping from uptime
robot)

PINGING THE WEB SERVER WITH UPTIME ROBOT

The screenshot illustrates the process of setting up a new monitoring task on the UptimeRobot platform.

Left Panel (Dashboard):

- The header shows "UptimeRobot" and "Upgrade" buttons.
- A green button labeled "+ Add New Monitor" is highlighted with a pink box.
- The main area displays two active monitors:
 - "Minerva Bot" with 100.00% status.
 - "Athena Bot" with 100.00% status.
- Below the monitors, a "Sort Monitors" dropdown and a "Last 24 Hours" timeline are visible.

Middle Panel (New Monitor Setup):

- A large blue arrow points from the "+ Add New Monitor" button on the dashboard to the "New Monitor" setup page.
- The "Monitor Information" section includes:
 - "Monitor Type": HTTP(s)
 - "Friendly Name": Pheme Bot
 - "URL (or IP)": https://Pheme.mewpainappuru.rep (highlighted with a pink box)
 - "Monitoring Interval": every 5 minutes (Recommended: 1 minute (Available in PRO))
 - "Monitor Timeout": in 30 seconds
 - "Monitor SSL errors": Recommended (Available in PRO)
 - "Enable SSL expiry reminders": Recommended (Available in PRO)
- "HTTP Method" options: HEAD (selected), GET, POST, or PUT, PATCH, DELETE, OPTIONS.
- A note at the bottom states: "PRO Available only in the PRO plan. Upgrade".
- An "Advanced Settings (Optional) show/hide" link is present.

Right Panel (Alert and Maintenance):

- A blue arrow points from the "Create Monitor" button back to the dashboard.
- "Select 'Alert Contacts To Notify'":
 - Selected: 0 of 1 (hide)
 - Type: Alert Contact
 - monica_fidalgo@outlook...
 - New alert contacts can be defined from the "My Settings" page.
- "Select 'Maintenance Windows'":
 - This is a PRO Plan feature for auto-disabling/enabling monitoring on pre-defined periods. Upgrade

Bottom Right (Buttons):

- "Close"
- "Create Monitor" (highlighted with a pink box)

PHEME IS NOW ONLINE “FOREVER” LET’S START TRACKING STUFF!

Mo毛 Today at 1:16 PM
track anime summer time

Pheme BOT Today at 1:16 PM
The current status for Summer Time Rendering is EP 5/25. You never saved the status of this series before.

Mo毛 Today at 1:17 PM
track manga kaiju 8

Pheme BOT Today at 1:17 PM
The current status for Kaiju No. 8 is Chapter 62. You never saved the status of this series before.

Mo毛 Today at 1:17 PM
track manga spy family

Pheme BOT Today at 1:17 PM
The current status for Spy X Family is Chapter 62.5. You never saved the status of this series before.

Mo毛 Today at 1:17 PM
track anime ya boy

Pheme BOT Today at 1:17 PM
The current status for Ya Boy Kongming! is EP 7/12. You never saved the status of this series before.



Files

main.py

birthday_tracker.py

keep_running.py

price_tracker.csv

status_tracker.csv

tracking.py

Tracking files created!!

Let’s look inside...

Files

main.py

birthday_tracker.py

keep_running.py

price_tracker.csv

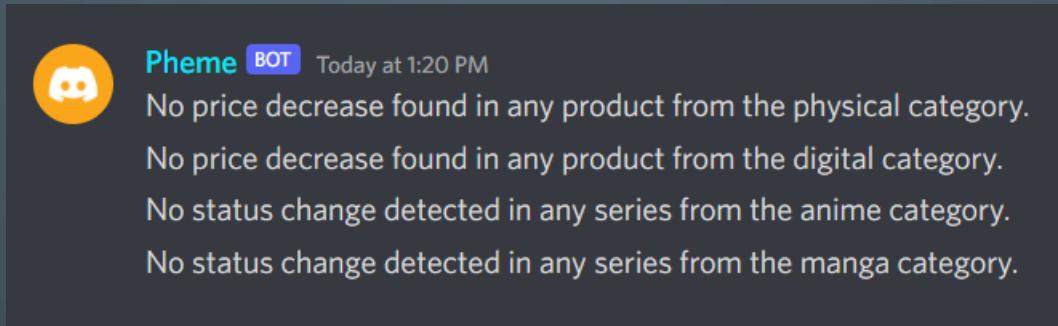
status_tracker.csv

tracking.py

status_tracker.csv

Name	Category	Status	DateChecked
Summer Time Rendering	anime	EP 5/25	06/08/2022, 11:21:00
Kaiju No. 8	manga	Chapter 62	06/08/2022, 11:21:02
Spy X Family	manga	Chapter 62.5	06/08/2022, 11:21:03
Ya Boy Kongming!	anime	EP 7/12	06/08/2022, 11:21:02

AFTER ADDING ENTRIES TO OUR TRACKING FILES, PHEME WRITES TO OUR CHANNELS!



**She can only start the task of searching for changes after the tracking files are created.
That's why she only wrote on the channels after we used the “track” command to start tracking some
items (and not upon being turned on).**

We can keep adding items to our files with the track command.

**Every 10h, she will loop through the files again, search for changes and write to the
channels.**

REFERENCES

- <https://yugiohprices.docs.apiary.io/#>
- <https://scryfall.com/docs/api>
- <https://www.youtube.com/watch?v=SPTfmiYiuok>
- <https://dev.to/maxhumber/how-to-build-a-price-monitoring-slack-bot-13fd>