#flipkart price tracking

Price Scrapper

We'll just take a product and if the price falls below a threshold, we'll set it to send an email to us. You could also do things like writing in a db or storing it to a CSV file.

```
import requests # used to fetch websites
import smtplib # used to send emails
from bs4 import BeautifulSoup #used to parse websites
import time #To continuously check for price drop

#   http://price-tracker-in.herokuapp.com/

#URL : the link to the product on flipkart
#header: headers for the http request. You can get your user gaent by googleing - "My user agent"


#URL = 'https://www.flipkart.com/apple-iphone-11-black-64-gb/p/itm0f37c2240b217?pid=MOBFKCTSVZAXUHGR&lid=LSTMOBFKCTSVZAXUHGREPBFGI'

headers = { 'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36' }
print('-'*70)
URL=str(input('Enter the url of the product: '))
print('-'*70)
recieverMail=input('Enter your email for reminders : ')



def check_price():
    '''Function called when there is a price check to be made '''

    #Loads the HTML ans stores in page
    page = requests.get(URL, headers=headers)

    #Enables use to parse the HTMl through html parser
    soup = BeautifulSoup(page.content, 'html.parser')

    #Gets the title of the product by looking for  tag in the HTML code with the classname "_35KyD6"
    title = soup.find("span", {"class": "B_NuCI"}).get_text()


    #Gets the price of the product by looking for
#  tag in the HTML code with the classname "_35KyD6"
    # [1:] is used to truncate the '₹' symbol and replace method to eradicate any commas if present
    price = float(soup.find("div", {"class": "_30jeq3 _16Jk6d"}).get_text()[1:].replace(',',''))


    #print('-'*70)
    #recieverMail=input('Enter your email for reminders :')
    print("Current Price is : ",price) #prints the price

    urprice=int(input('Enter the Price, at which you want : '))


    if( price < urprice ): #If the price falls below threshold, send an email
```

```
        sendmail()


# https://myaccount.google.com/security

def sendmail():
    '''Function called when the email needs to be sent '''

    # Defines an SMTP client session with the host name, here being, smtp.gmail.com as we will
be using Gmail
    # to send our emails. 587 is the port number for Gmail's TLS
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo() #Extended HELO used by email server to identify itself
    server.starttls() #Put the SMTP connection in TLS mode. All SMTP commands that follow will
be encrypted
    server.ehlo()

    #Your email and app password. Follow the steps in the readme file to get your app password
    server.login('s.webscraping@gmail.com', 'nibbrafyvaqjdzim')

    subject = 'Hey! Price fell down' #Subject of the email
    body = 'Check the link ' + URL #Body of the email

    msg = f"Subject: {subject}\n\n{body}" #Aggregation

    server.sendmail('s.webscraping@gmail.com', recieverMail, msg) #Sending the email


    print('Email has send to {}'.format(recieverMail))

    server.quit() #Closing the connection



while(True):
    check_price()
    time.sleep(60*60) #Checks price every 60*60 seconds i.e, every hour
```