Lab 14: Mike Menendez Cassandra Chupacabra

Summary:

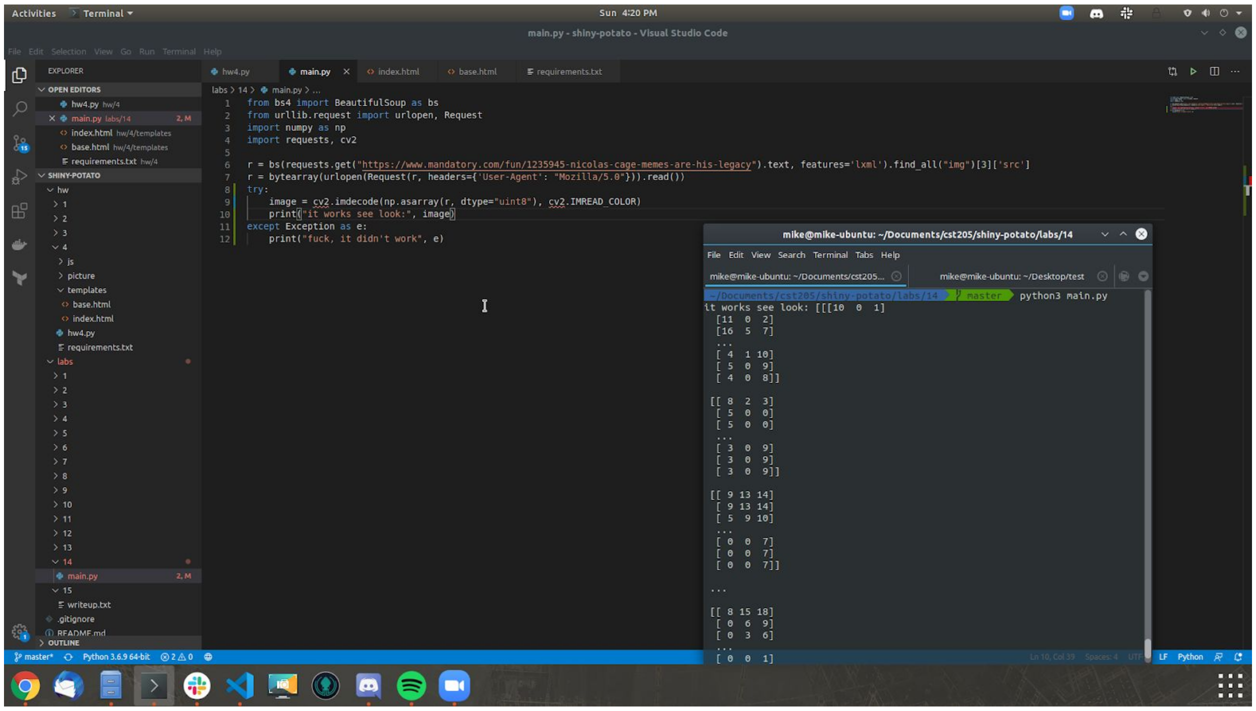
This lab was a little difficult as we were pulling our image from a CDN which was protecting itself from web scraping. The first attempt to circumvent the issue was to use the requests library and to do a simple get request. While doing this we attempted to use PIL to stream the image in it's raw format and then to convert it into a numpy array. This did not work when we got to the open cv portion as PIL and open cv use different internal representations of the image with numpy as the driver. This caused us to get the buffer size assertion error from the compiled so that opencv ships with.

We then switched our strategy from using a direct stream to PIL and switched back to the regular url open methodology which forced us to solve the initial 403 error. The way we solved this issue was to set the user agent which allowed us to get the image. The other error we ran into was related to the default ssl package that is bundled with MacOs however, it worked perfectly fine on Ubuntu so we ran it on my desktop instead of our laptops.

In regards to how the code provided works, it opens the content of the web page's dom using url open which turns it into a graph like structure that then is easily parsed. After the content (which in this case is literally only an image due to how read() works), it is then coerced into a bytearray to a which works because it strips the metadata out and allows for it to become an array of arrays for each pixel. After this is done, opencv takes the data and does it's open internal processing to map the pixel data into arrays in the BGR format.

Task1: we installed it

Task2 and 3:



```python
from bs4 import BeautifulSoup as bs
from urllib.request import urlopen, Request
import numpy as np
import requests, cv2

r = bs(requests.get("https://www.mandatory.com/fun/1235945-nicolas-cage-memes-are-his-legacy").text, features='lxml').find_all("img")[3]['src']
r = bytearray(urlopen(Request(r, headers={'User-Agent': "Mozilla/5.0"})).read())
try:
    image = cv2.imdecode(np.asarray(r, dtype="uint8"), cv2.IMREAD_COLOR)
    print("it works see look:", image)
except Exception as e:
    print("fuck, it didn't work", e)
```

```
~/Documents/cst205/shiny-potato/labs/14  master  python3 main.py
it works see look: [[[10  0  1]
  [11  0  2]
  [16  5  7]
  ...
  [ 4  1 10]
  [ 5  0  9]
  [ 4  0  8]]

 [[ 8  2  3]
  [ 5  0  0]
  [ 5  0  0]
  ...
  [ 3  0  9]
  [ 3  0  9]
  [ 3  0  9]]

 [[ 9 13 14]
  [ 9 13 14]
  [ 5  9 10]
  ...
  [ 0  0  7]
  [ 0  0  7]
  [ 0  0  7]]

 ...

 [[ 8 15 18]
  [ 0  6  9]
  [ 0  3  6]
  ...
  [ 0  0  1]
```