



Ứng dụng của các thư viện trong Python

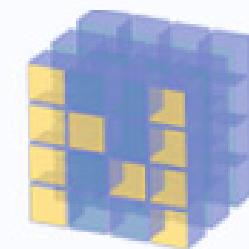


Ha Noi university of
Science and Technology

pandas
pandas



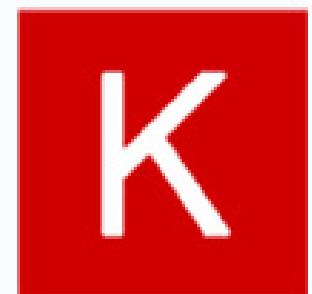
SciPy



NumPy



Matplotlib



Keras



Tensor Flow



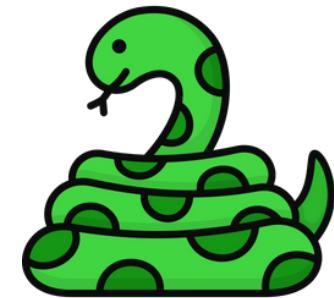
Scikit-learn

orange
Orange3

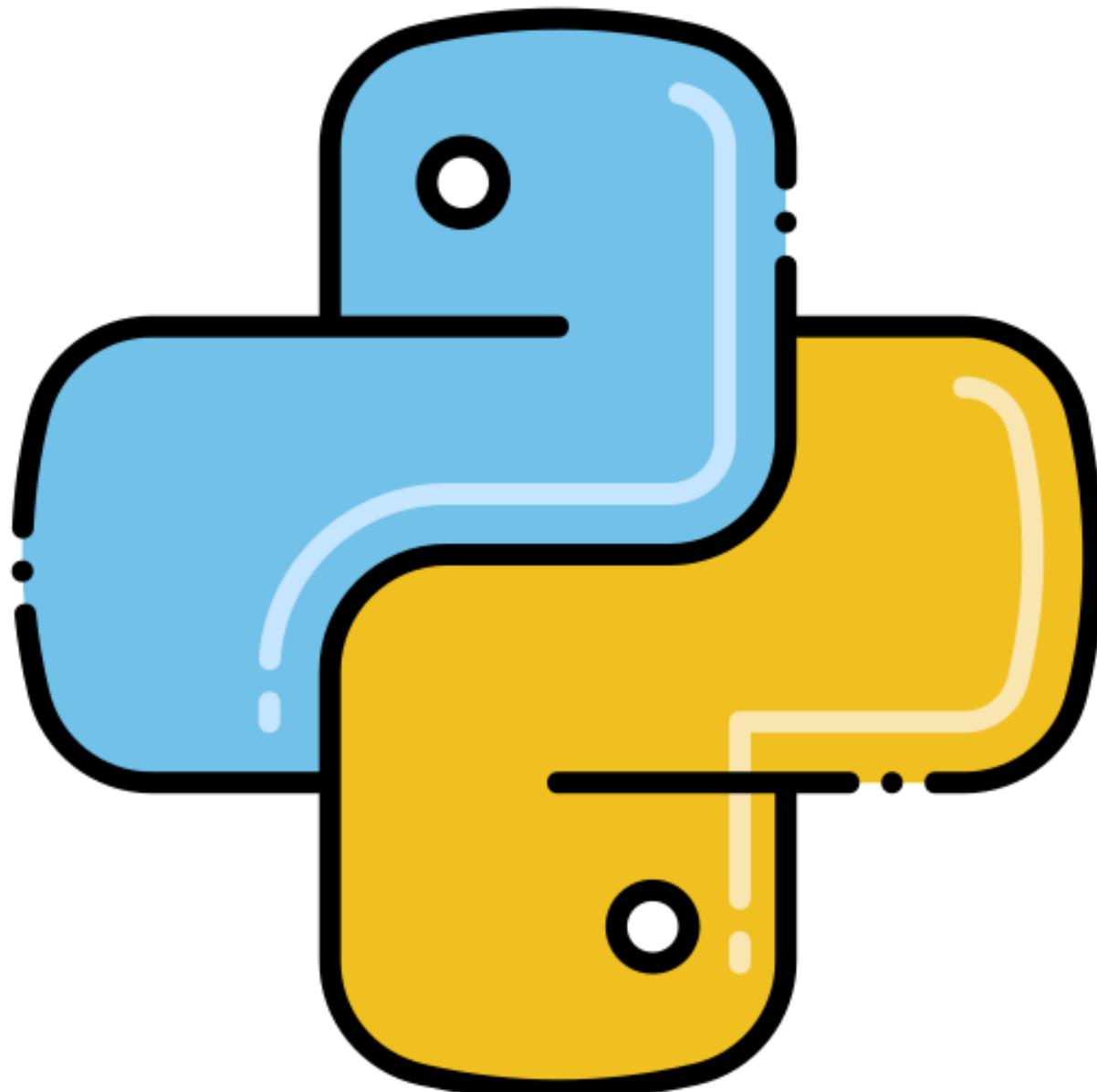
theano
theano

Trình bày bởi: Vũ Văn Huy - 20216931.

Sự kiện này có phiên dịch trực tiếp.



Nội dung



- | | |
|---------------|-----------------------------------------------------------------------------------------------------------------------|
| Phân 1 | Xử lý, trực quan dữ liệu
Phân tích dữ liệu |
| Phân 2 | Xử lý hình ảnh
Thị giác máy tính, làm việc với ảnh |
| Phân 3 | Đồ họa
Xây dựng giao diện người dùng |
| Phân 4 | Tự động hóa
Tăng năng suất, giảm chi phí công việc |
| Phân 5 | Lấy và trích xuất thông tin từ web
Các phương pháp thanh toán an toàn thu thập dữ liệu
từ các trang mạng |

Xử lý, trực quan hóa Dữ liệu

Một Công ty Truyền thông Hiện đại





Python

My Brain

Khoa học dữ
liệu

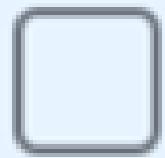


SAVE

LINKTR.EE/HUYVV

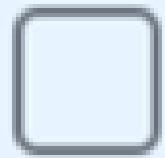


LIKE



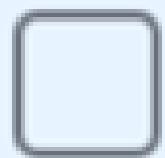
Phân tích dữ liệu

32% >



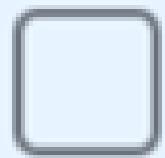
Lập trình viên

27% >



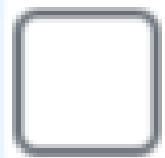
BA

17% >



QA/QC

7% >



Khác (các em thêm vote hoặc comment nhé)

2% >



NumPy



Matplotlib



Pandas



Library package in Py You Should Try



Seaborn

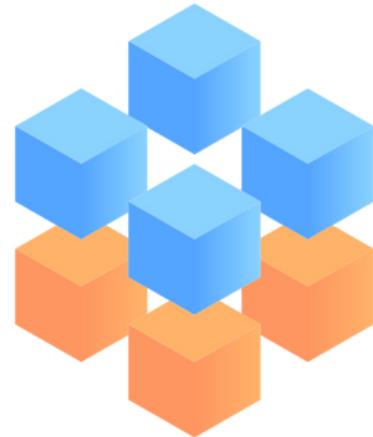
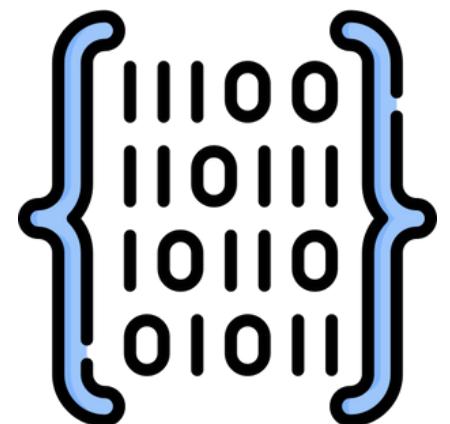


Keras



Scikit learn

NumPy



Thư viện lõi phục vụ cho khoa học máy tính.

Cung cấp một đối tượng mảng nhiều chiều linh hoạt và hiệu quả để thực hiện các phép toán số học và logic trên mảng.

Điểm mạnh của NumPy là khả năng làm việc với các mảng lớn và cung cấp các hàm và phương thức tối ưu cho việc thao tác dữ liệu.

```
In [3]: import numpy as np  
#Khởi tạo mảng một chiều với kiểu dữ liệu các phần tử là Integer  
arr = np.array([1,3,4,5,6], dtype = int)  
arr
```

```
Out[3]: array([1, 3, 4, 5, 6])
```

```
In [4]: # Khởi tạo mảng 2 chiều  
arr1 = np.array([(4,5,6), (1,2,3)], dtype = int)  
arr1
```

```
Out[4]: array([[4, 5, 6],  
               [1, 2, 3]])
```

```
In [16]: # Phép nhân mảng với một số  
arr2 = arr1 * 2  
arr2
```

```
Out[16]: array([[ 8, 10, 12],  
                [ 2,  4,  6]])
```

```
In [17]: arr1 + arr2
```

```
Out[17]: array([[12, 15, 18],  
                 [ 3,  6,  9]])
```

```
In [22]: sum_value = np.sum(arr1)  
sum_value
```

Out[22]: 21

```
In [24]: mean_value = np.mean(arr1)  
mean_value
```

Out[24]: 3.5

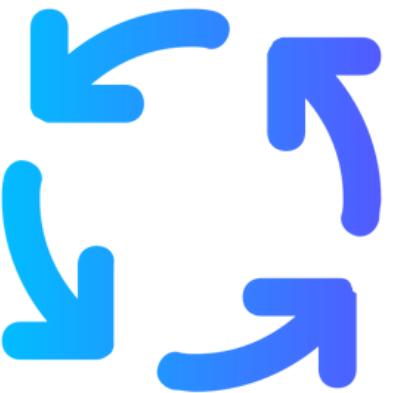
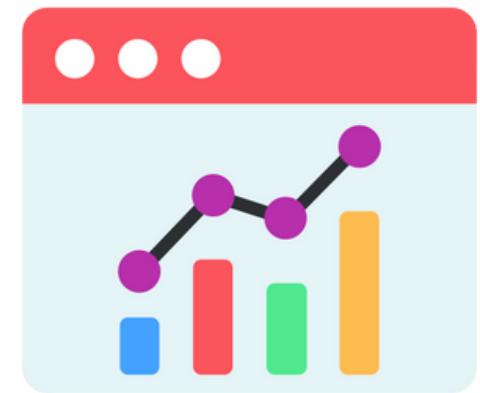
```
In [19]: max_value = np.max(arr1)  
max_value
```

Out[19]: 6

```
In [20]: min_value = np.min(arr1)  
min_value
```

Out[20]: 1

Pandas



Pandas là một thư viện phổ biến trong Python với khả năng xử lý và phân tích dữ liệu dễ dàng

Cung cấp nhiều chức năng phân tích dữ liệu mạnh mẽ như tính toán thống kê, nhóm dữ liệu, và tạo báo cáo tóm tắt

Cung cấp các cấu trúc dữ liệu linh hoạt và dễ sử dụng

```
In [1]: import pandas as pd

# Khởi tạo DataFrame từ một dictionary
data = {'Name': ['John', 'Emma', 'Peter'],
        'Age': [25, 30, 35],
        'City': ['New York', 'London', 'Paris']}
df = pd.DataFrame(data)
print(df)
```

	Name	Age	City
0	John	25	New York
1	Emma	30	London
2	Peter	35	Paris

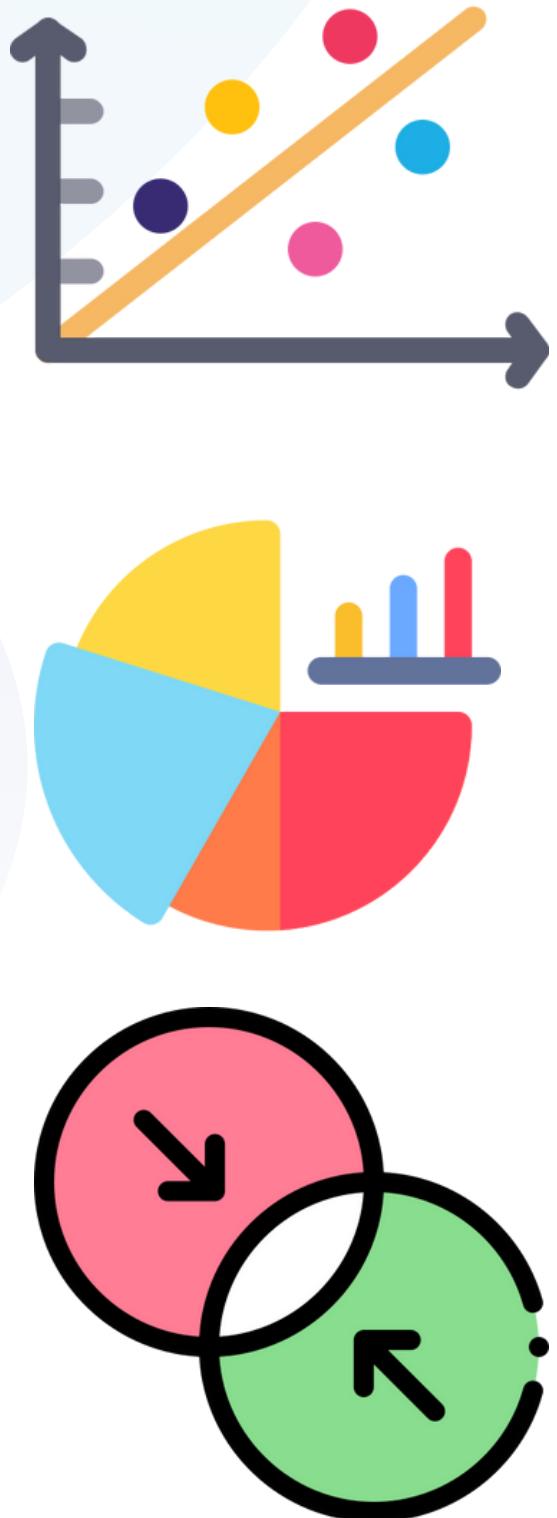
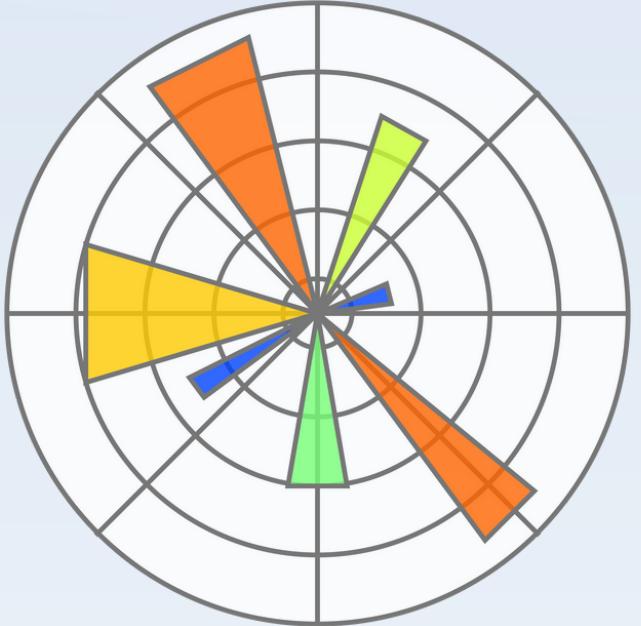
```
In [2]: df = pd.read_csv('data.csv')

print(df)
```

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409.1
1	60	117	145	479.0
2	60	103	135	340.0
3	45	109	175	282.4
4	45	117	148	406.0
..
164	60	105	140	290.8
165	60	110	145	300.0
166	60	115	145	310.2
167	75	120	150	320.4
168	75	125	150	330.4

[169 rows x 4 columns]

Matplotlib



Matplotlib là thư viện trực quan hóa dữ liệu phổ biến, được sử dụng để tạo ra các biểu đồ và đồ thị đẹp mắt.

Linh hoạt tùy chỉnh các yếu tố trong biểu đồ như tiêu đề, nhãn trực, màu sắc, định dạng văn bản ...

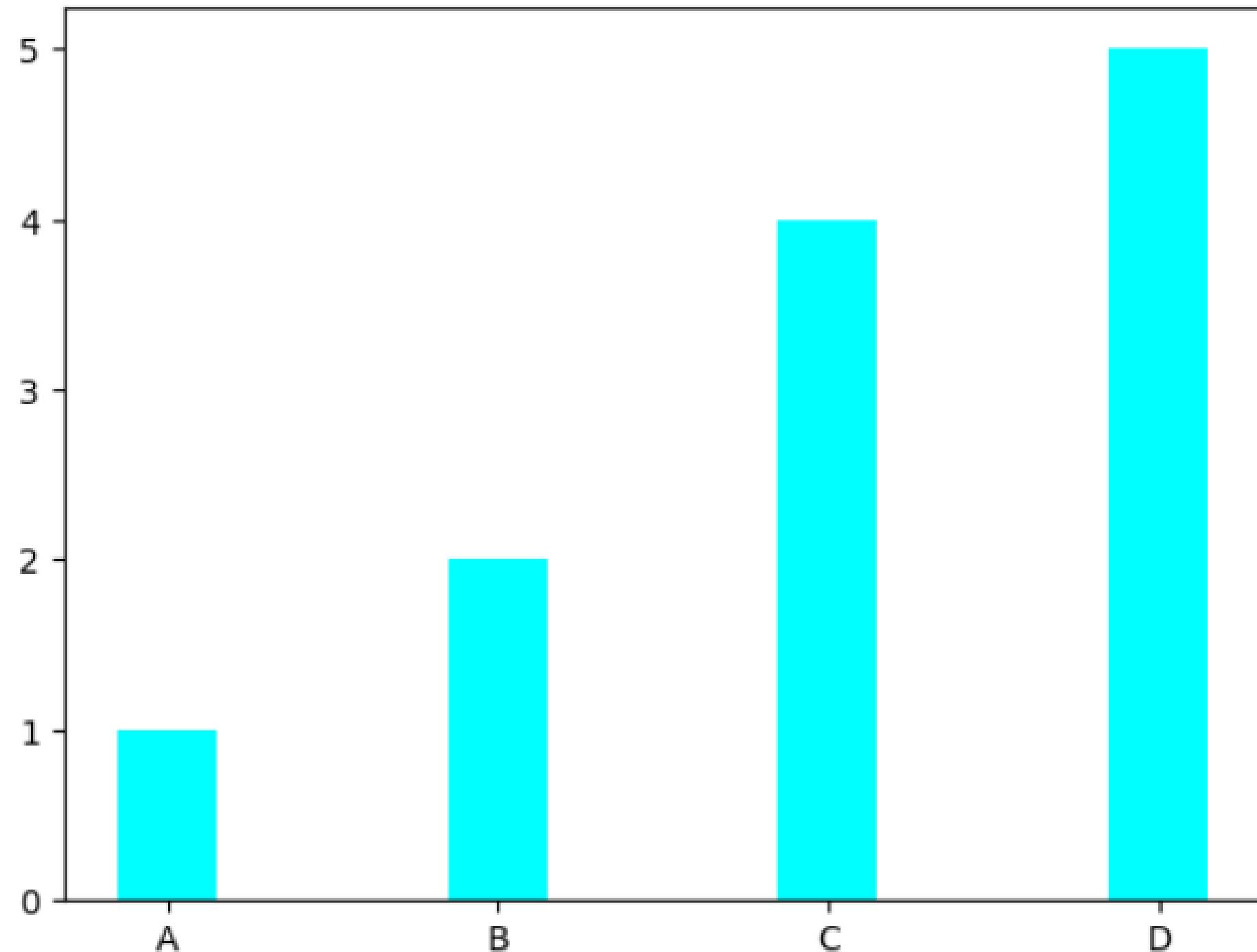
Tích hợp chặt chẽ với các thư viện phân tích dữ liệu phổ biến khác trong Python như NumPy và Pandas.

In [9]:

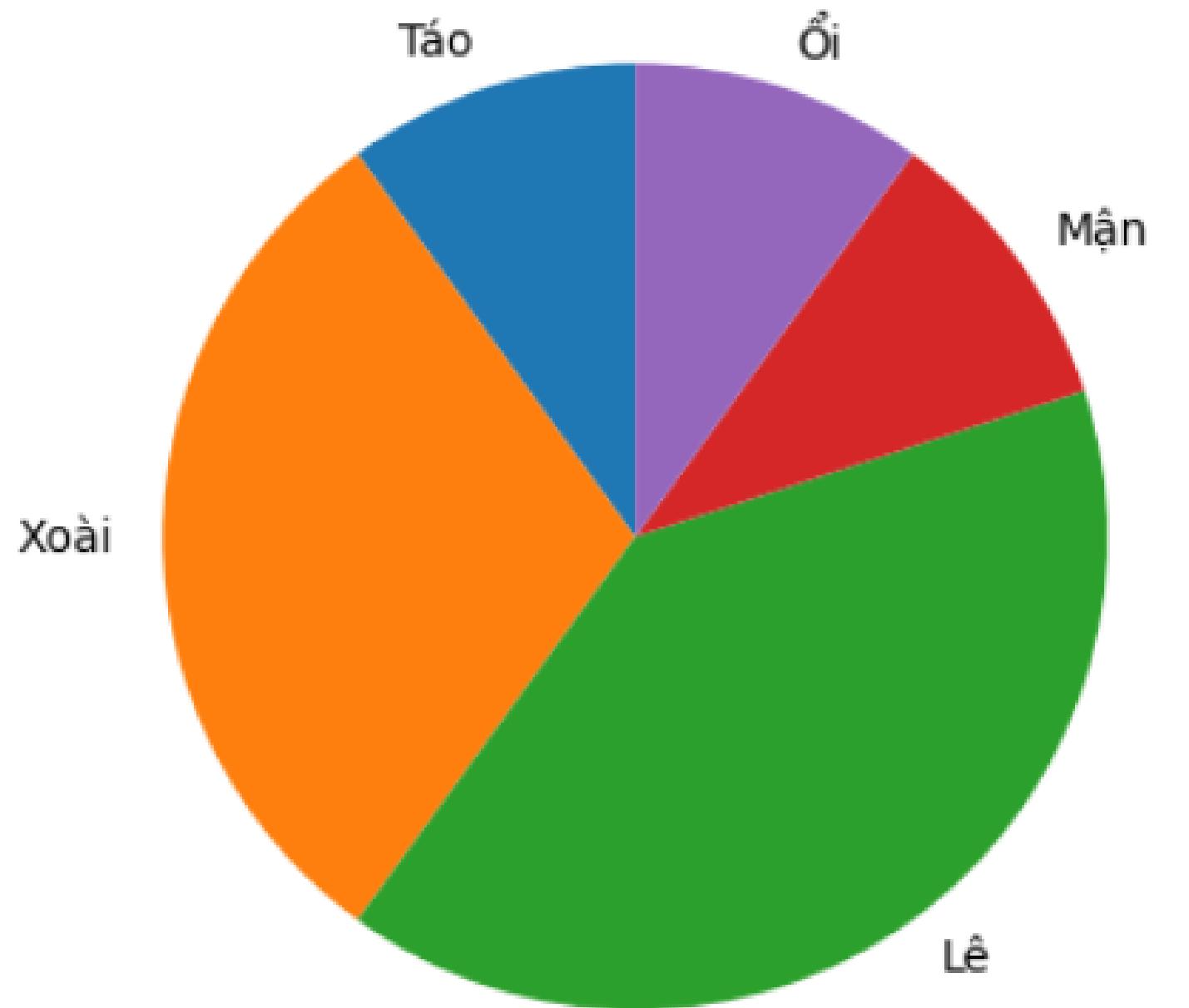
```
import matplotlib.pyplot as plt
import numpy as np

x = np.array(["A", "B", "C", "D"])
y = np.array([1, 2, 4, 5])

plt.bar(x, y, color = 'cyan', width = 0.3) # ngược lại thì dùng barh
plt.show()
```



```
In [6]: import matplotlib.pyplot as plt  
import numpy as np  
  
y = np.array([10, 30, 40, 10, 10])  
mylabels = ["Táo", "Xoài", "Lê", "Mận", "Ổi"]  
plt.pie(y, labels = mylabels, startangle = 90) # tùy chỉnh độ quay  
plt.show()
```



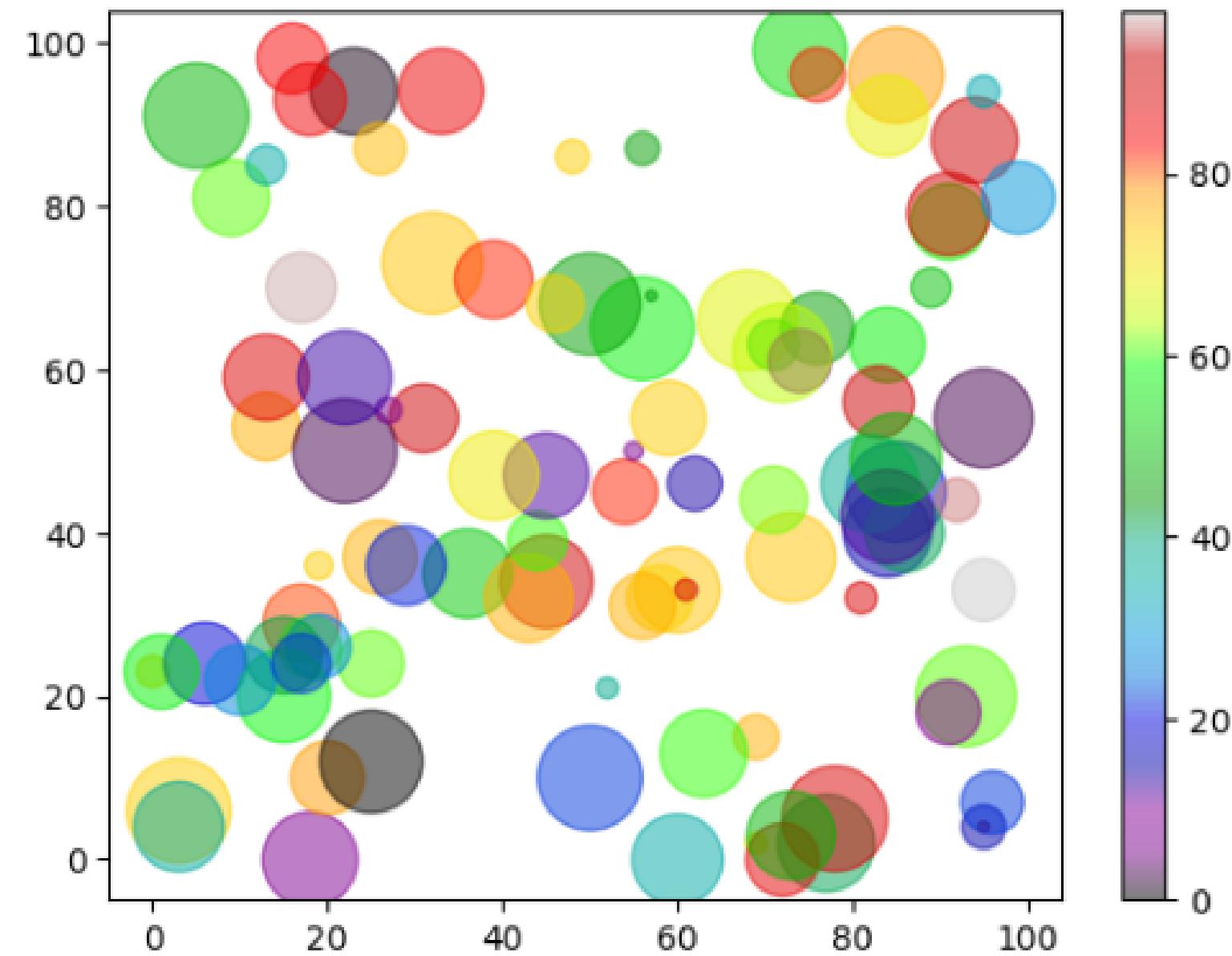
```
In [19]: # vẽ 1 bảng đa màu
import matplotlib.pyplot as plt
import numpy as np

x = np.random.randint(100, size=(100))
y = np.random.randint(100, size=(100))
colors = np.random.randint(100, size=(100))
sizes = 10 * np.random.randint(100, size=(100))

plt.scatter(x, y, c=colors, s=sizes, alpha=0.5, cmap='nipy_spectral')

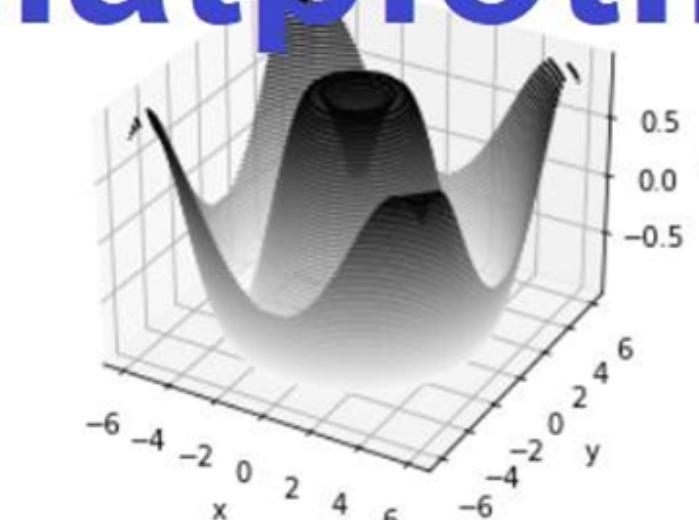
plt.colorbar()

plt.show()
```

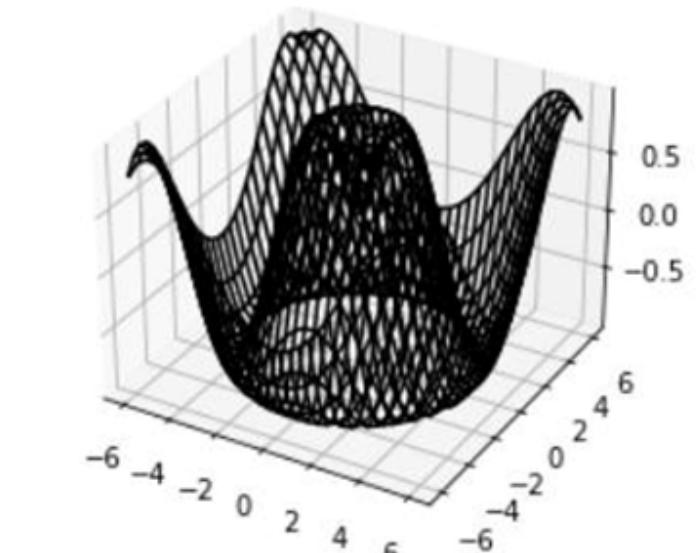


3D Graphs with Matplotlib

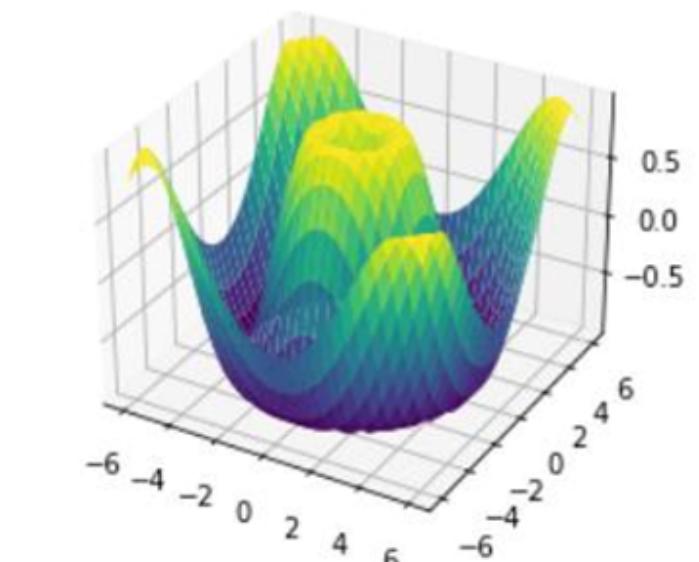
```
import numpy as np
import matplotlib.pyplot as plt
def f(x, y):
    return np.sin(np.sqrt(x ** 2 + y ** 2))
x = np.linspace(-6, 6, 30)
y = np.linspace(-6, 6, 30)
x, y = np.meshgrid(x, y)
z = f(x, y) #clcoding.com
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.contour3D(x,y,z,50, cmap='binary')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
plt.show()
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.plot_wireframe(x,y,z, color='black')
ax.set_title('wireframe')
plt.show()
ax = plt.axes(projection='3d')
ax.plot_surface(x, y, z, rstride=1,
                cstride=1, cmap='viridis',
                edgecolor='none')
ax.set_title('surface')
plt.show()
```



wireframe



surface



Hands-on

PHÂN TÍCH DỮ LIỆU

https://github.com/huyvu15/Sales_data_analysis



Sử dụng đối với bộ dữ liệu sale

Các thư viện còn lại

- Trực quan hóa dữ liệu dựa trên matplotlib.
- Deep learning, học tăng cường
- Machine learning

Seaborn: Seaborn là một thư viện trực quan hóa dữ liệu dựa trên Matplotlib. Nó cung cấp giao diện đơn giản và hấp dẫn để tạo ra các biểu đồ thống kê và trực quan hóa dữ liệu. Seaborn cung cấp các chức năng cho việc vẽ biểu đồ phân phối, biểu đồ quan hệ, biểu đồ thể hiện sự tương quan, biểu đồ hộp và viền, và nhiều biểu đồ khác để phân tích và trình bày dữ liệu.

Keras: Keras là một thư viện mạng neural mạnh mẽ và dễ sử dụng trong Python. Nó cung cấp một giao diện cao cấp để xây dựng, huấn luyện và đánh giá các mô hình mạng neural. Keras hỗ trợ nhiều kiến trúc mạng neural, bao gồm mạng neural lan truyền thẳng, mạng neural tái lập, và mạng neural tích chập.

Scikit-learn: Scikit-learn là một thư viện machine learning phổ biến trong Python. Nó cung cấp một loạt các thuật toán machine learning cho các tác vụ như phân loại, hồi quy, gom cụm và giảm chiều dữ liệu. Scikit-learn cung cấp các công cụ cho việc tiền xử lý dữ liệu, đánh giá mô hình, và tối ưu hóa siêu tham số. Nó cũng bao gồm các công cụ cho xử lý dữ liệu không cân bằng, chọn đặc trưng và trích xuất đặc trưng, và phân tích thành phần chính . Scikit-learn là một thư viện rất mạnh mẽ để triển khai các mô hình machine learning trong nhiều ứng dụng khác nhau.

Hands-on

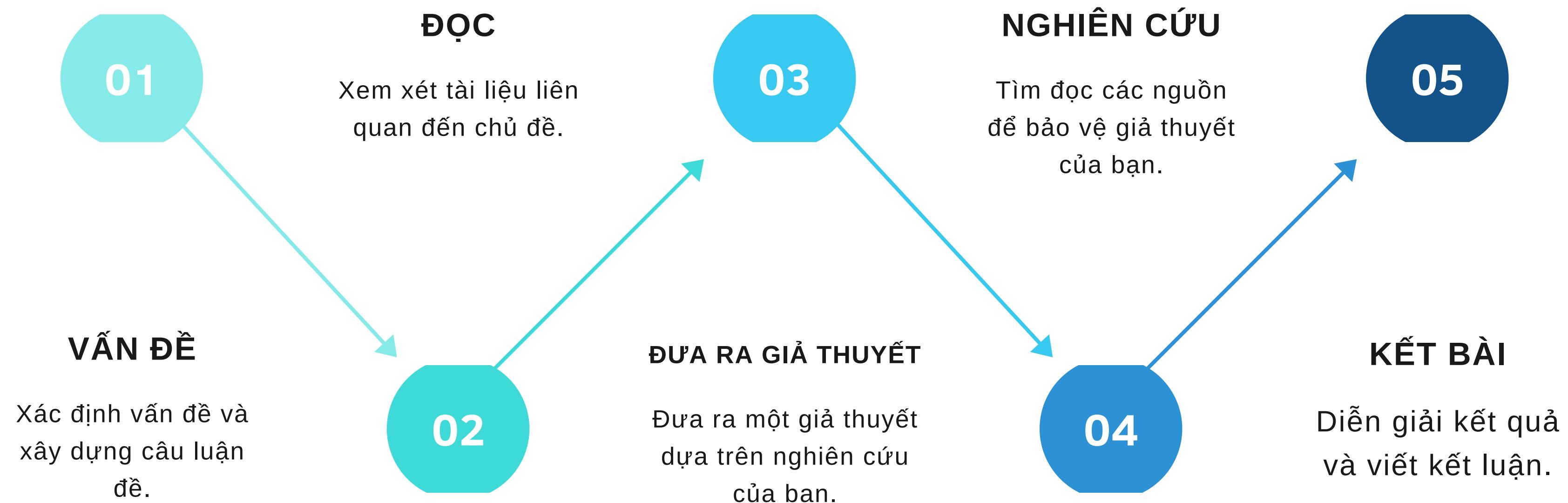
PHÂN TÍCH DỮ LIỆU

https://github.com/huyvu15/customer_segmentation



Bài toán phân khúc
khách hàng

Các bước nghiên cứu





Xử lý hình ảnh

Một Công ty Truyền thông Hiện đại

All

Images

Videos



Search.



Xử lý ảnh



OpenCV



Face recognition



Chỉnh sửa ảnh



Nhận dạng vật thể



Thực tế ảo và thực tế tăng cường



100K likes

Save for later



Xử lý ảnh hiệu quả nhất

Truyền tải nhiều thông tin
một cách hiệu quả và trực quan

1

OpenCV

Sử dụng biểu đồ trực quan để cung cấp thông tin hiệu quả hơn.

2

PIL

Sử dụng biểu đồ trực quan để cung cấp thông tin hiệu quả hơn.

3

ImageIO

Sử dụng biểu đồ trực quan để cung cấp thông tin hiệu quả hơn.

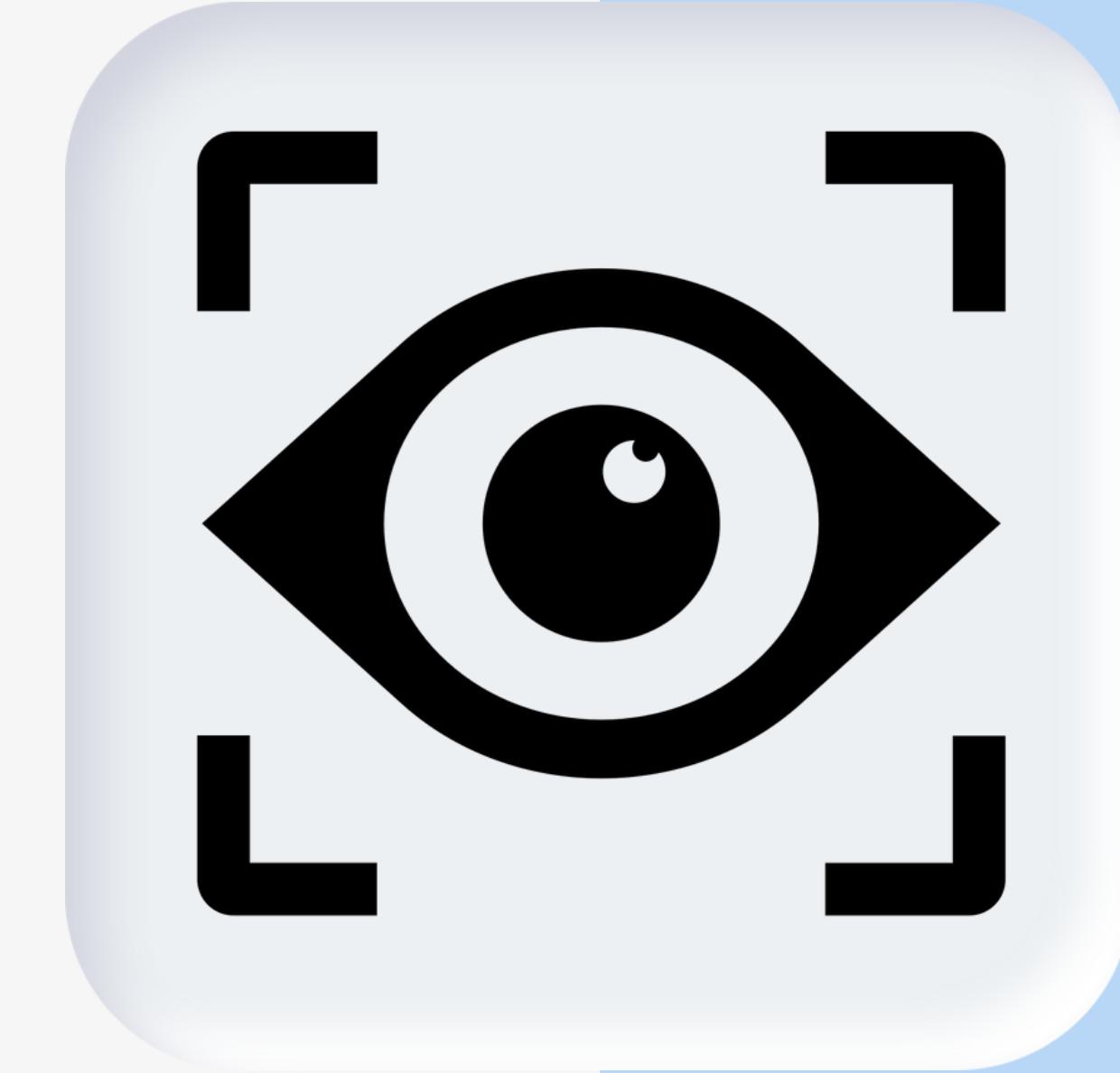
4

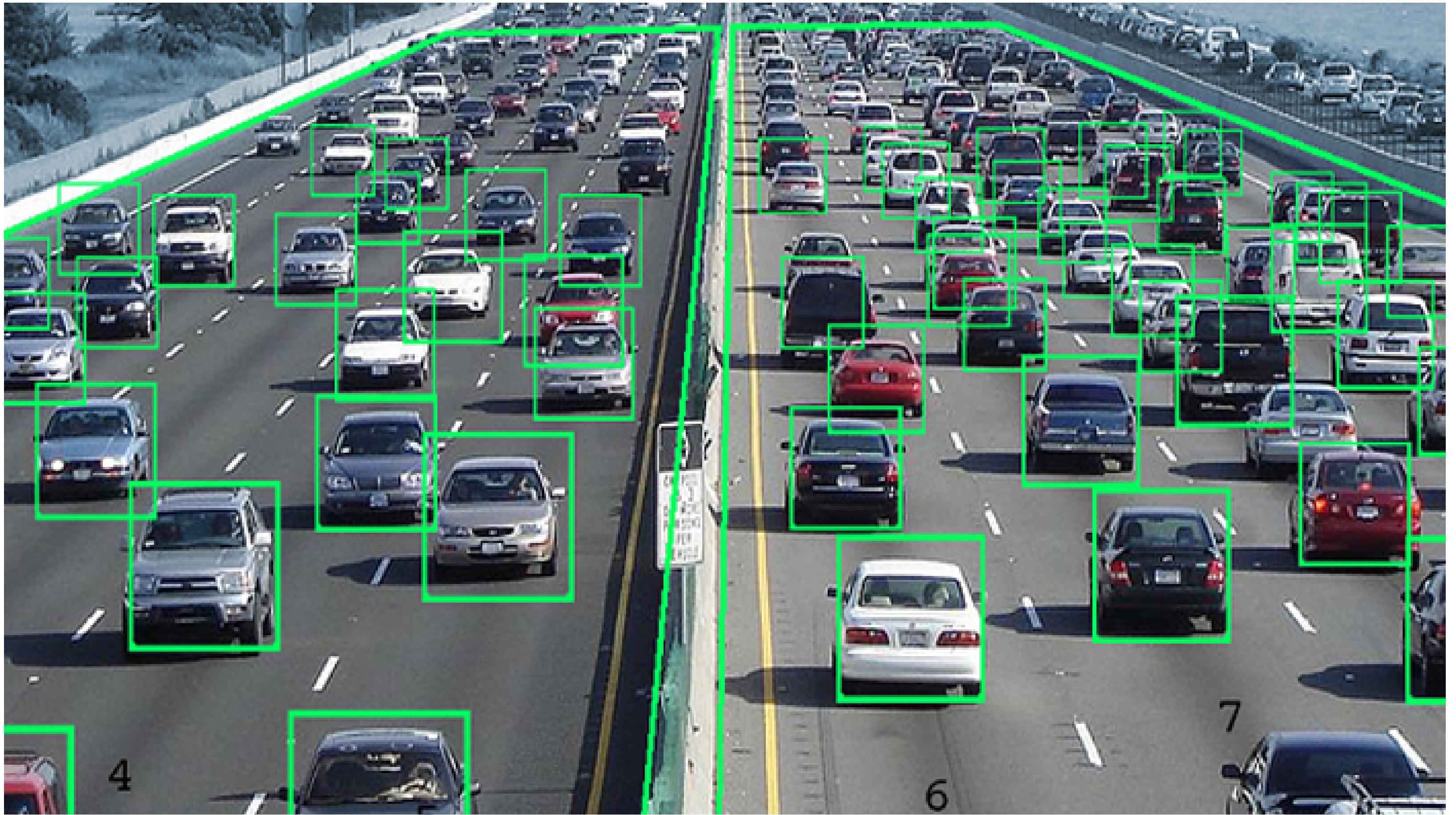
Scikit-Image

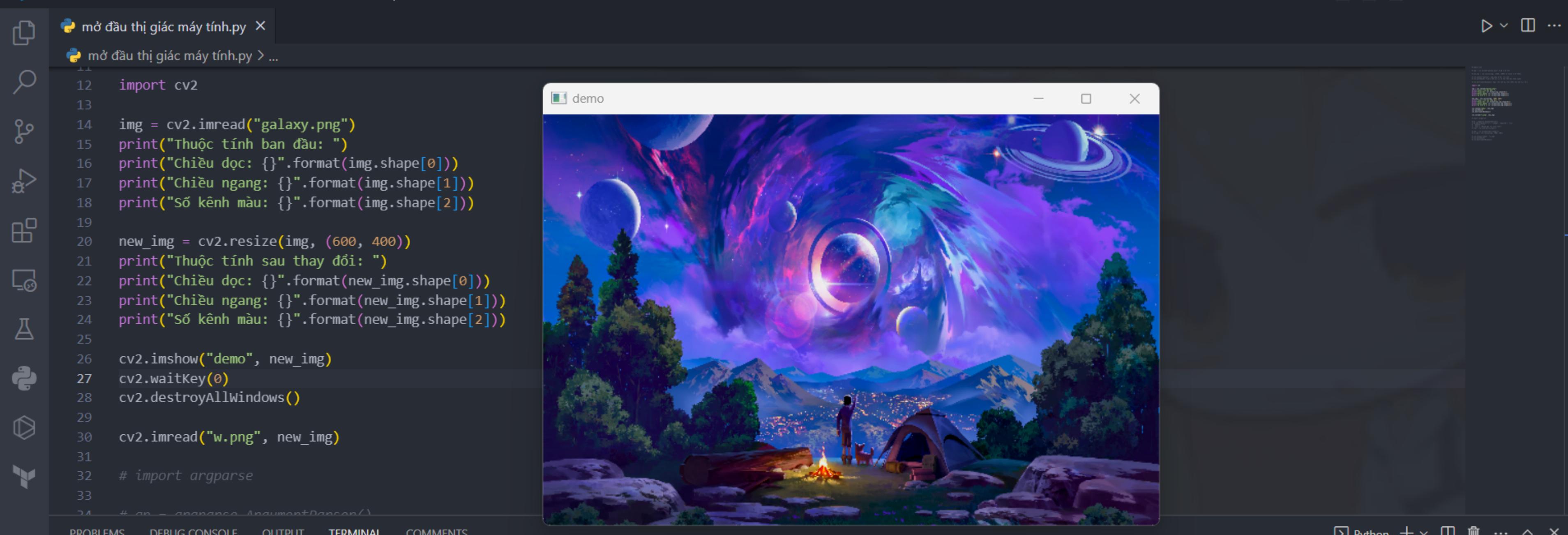
Sử dụng biểu đồ trực quan để cung cấp thông tin hiệu quả hơn.

OpenCV

OpenCV là một thư viện phổ biến cho xử lý ảnh và thị giác máy tính. Nó cung cấp một loạt các chức năng và công cụ mạnh mẽ để thao tác với ảnh, video và dữ liệu thị giác máy tính.







mở đầu thị giác máy tính.py

```
12 import cv2
13
14 img = cv2.imread("galaxy.png")
15 print("Thuộc tính ban đầu: ")
16 print("Chiều dọc: {}".format(img.shape[0]))
17 print("Chiều ngang: {}".format(img.shape[1]))
18 print("Số kênh màu: {}".format(img.shape[2]))
19
20 new_img = cv2.resize(img, (600, 400))
21 print("Thuộc tính sau thay đổi: ")
22 print("Chiều dọc: {}".format(new_img.shape[0]))
23 print("Chiều ngang: {}".format(new_img.shape[1]))
24 print("Số kênh màu: {}".format(new_img.shape[2]))
25
26 cv2.imshow("demo", new_img)
27 cv2.waitKey(0)
28 cv2.destroyAllWindows()
29
30 cv2.imwrite("w.png", new_img)
31
32 # import argparse
33
# an - ananace AnamontBacon/
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL COMMENTS

Python + - ×

```
PS D:\Thị giác máy tính> & "C:/Users/HI.WELCOME TO NET/AppData/Local/Programs/Python/Python310/python.exe" "d:/Thị giác máy tính/mở đầu thị giác máy tính.py"
Thuộc tính ban đầu:
Chiều dọc: 2160
Chiều ngang: 3840
Số kênh màu: 3
Thuộc tính sau thay đổi:
Chiều dọc: 400
Chiều ngang: 600
Số kênh màu: 3
```



0 0 tabnine starter

Ln 27, Col 15 Tab Size: 4 UTF-8 CRLF Python 3.10.6 64-bit

```
mở đầu thị giác máy tính.py ● # nhận dạng khuôn mặt người trong ảnh.py  
không gian màu.py > ...  
1 import cv2  
2  
3 # Đọc ảnh từ file  
4 image = cv2.imread('galaxy.png')  
5  
6 # Kiểm tra xem việc đọc ảnh thành công hay không  
7 if image is not None:  
8     # Đổi kích thước ảnh ban đầu  
9     resized_original = cv2.resize(image, (600, 400))  
10  
11    # Chuyển đổi không gian màu từ BGR sang RGB  
12    rgb_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  
13  
14    # Đổi kích thước ảnh sau khi chuyển đổi mã màu  
15    resized_rgb = cv2.resize(rgb_image, (600, 400))  
16  
17    # Hiển thị ảnh ban đầu và ảnh sau khi thay đổi mã màu và thay đổi kích thước  
18    cv2.imshow('Original Image', resized_original)  
19    cv2.imshow('Resized RGB Image', resized_rgb)  
20  
21    # Đợi người dùng nhấn phím bất kỳ để thoát  
22    cv2.waitKey(0)  
23  
24    # Đóng cửa sổ hiển thị ảnh  
25    cv2.destroyAllWindows()  
26 else:  
27     print('Không thể đọc ảnh.')28
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL COMMENTS

PS D:\Thị giác máy tính> & "C:/Users/HI.WELCOME TO NET/AppData/Local/Programs/Python/Python310/python.exe" "d:/Thị giác máy tính/không gian màu.py"

không gian màu.py



```
pip install rembg
```

Remove Image Background using Python

```
from rembg import remove
from PIL import Image
input_path = 'cl.jpg'
output_path = 'output.png'
input = Image.open(input_path)
output = remove(input)
output.save(output_path)
```



Image Mirror with Python

```
from PIL import Image
```

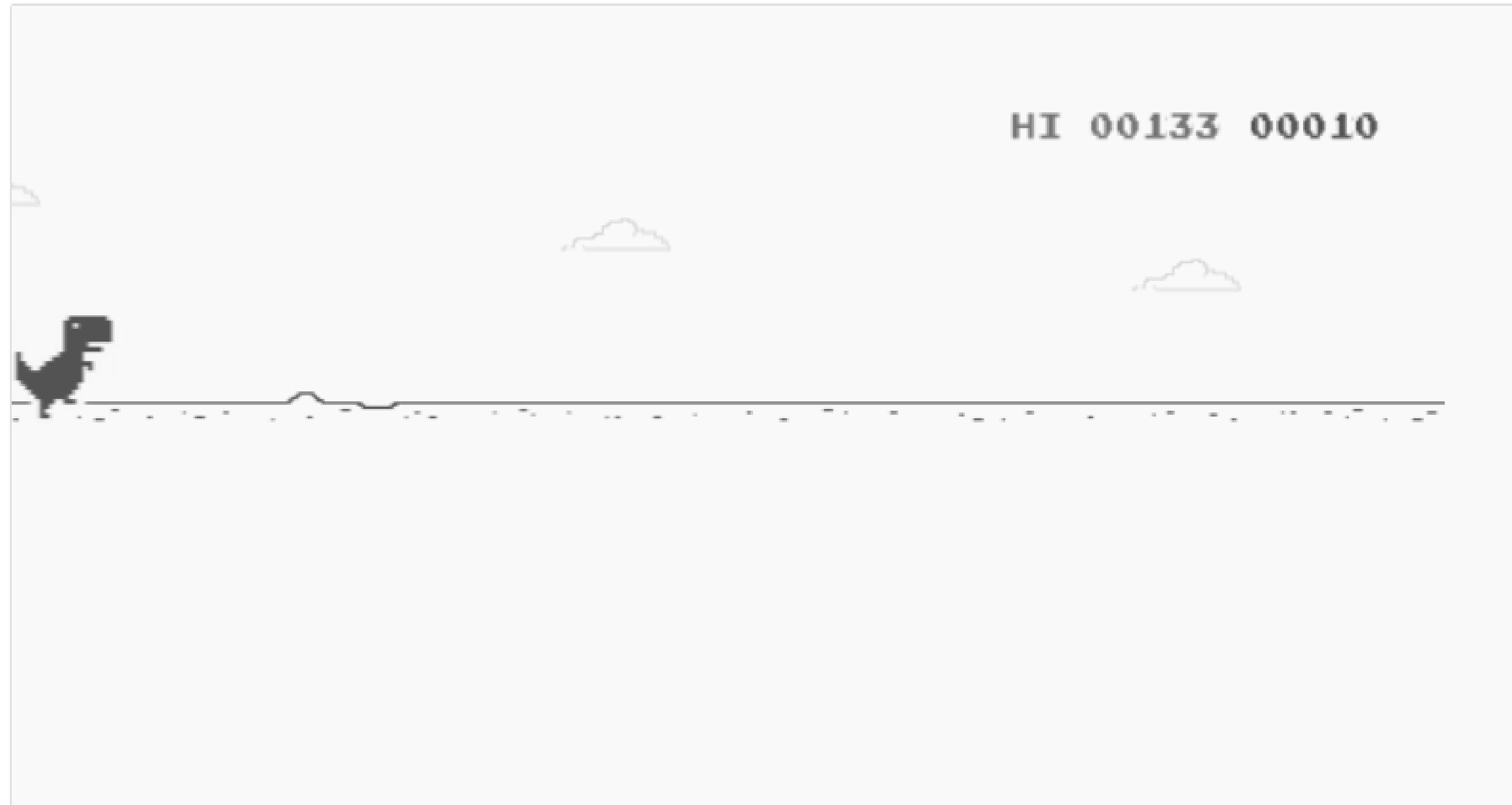
```
Image.open('binodd.jpg') #original Image
```



```
img = Image.open('binodd.jpg') #clcoding.com
Mirror_Image=img.transpose(Image.FLIP_LEFT_RIGHT)
Mirror_Image.save(r'binod_mIRROR.png')
Image.open('binod_mIRROR.png') #mirrored Image
```



Game Khủng Long Chạy Bộ - Chrome Dino, Dinosaur, T-Rex Dinosaur



Tải lại



Bật đèn



Yêu thích



Mở lớn



Phóng to



Nghe



f



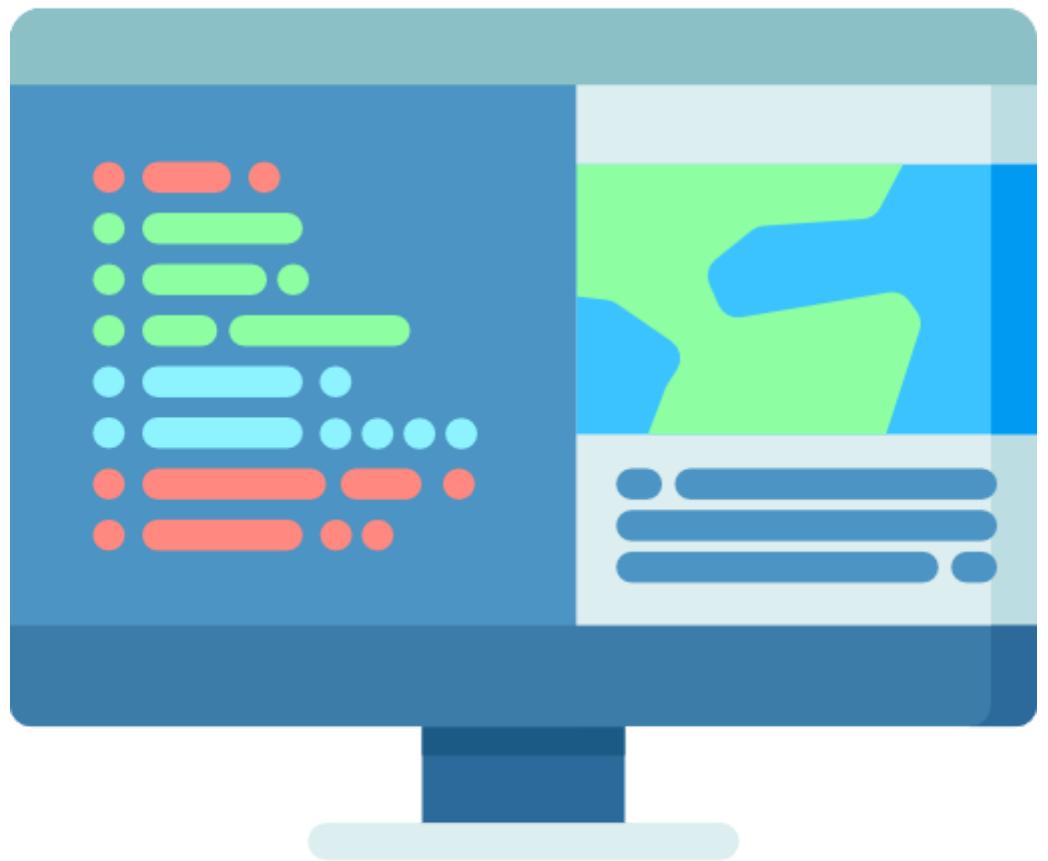
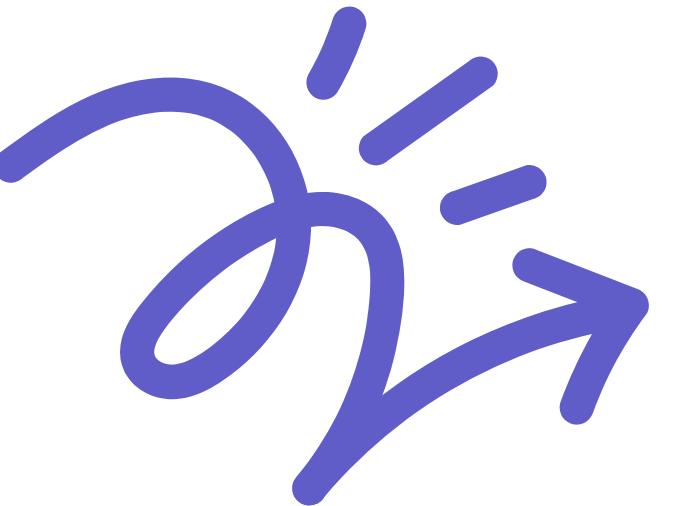
Zalo



GUI

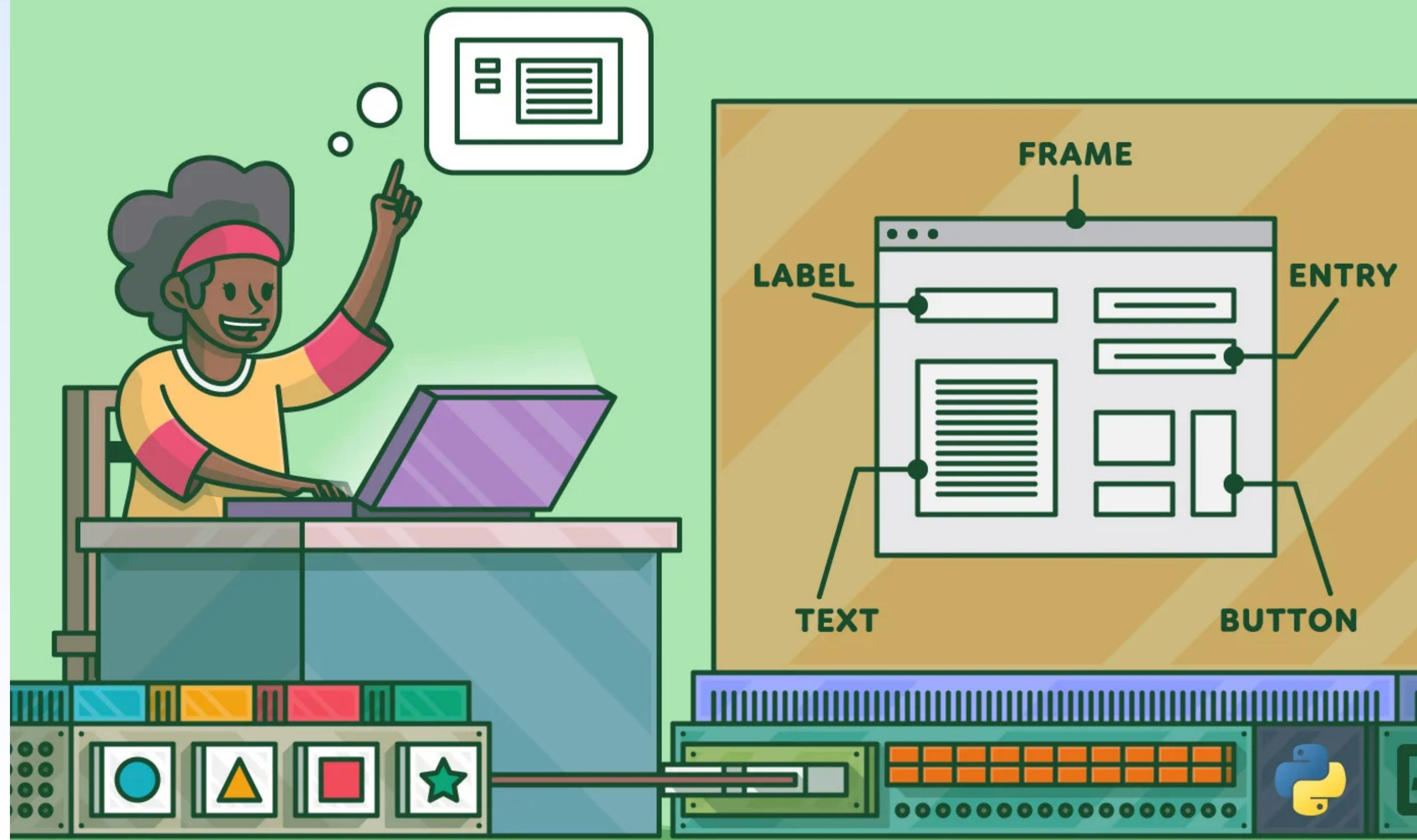
Thư viện đồ họa

Đem đến những ý tưởng độc đáo từ năm 2023



Tkinter

là một thư viện giao diện người dùng được tích hợp sẵn trong Python.



Rea

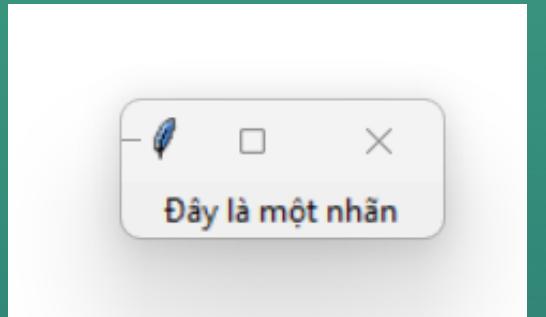
```
// label
import tkinter as tk

window = tk.Tk()

label = tk.Label(window, text="Đây là một nhãn")

label.pack()

window.mainloop()
```



Vũ Huy



Vũ Huy

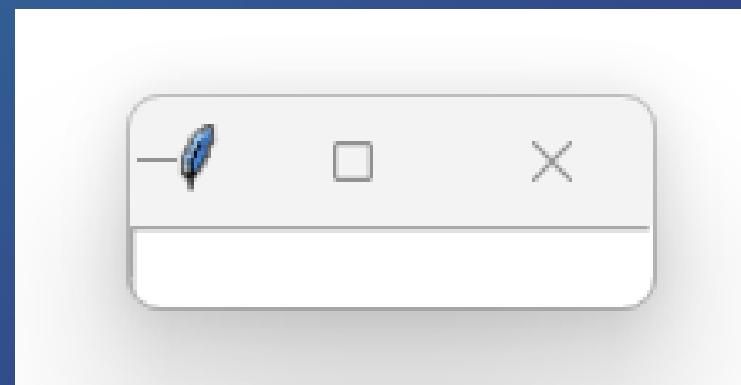
```
// Entry
import tkinter as tk

window = tk.Tk()

entry = tk.Entry(window)

entry.pack()

window.mainloop()
```

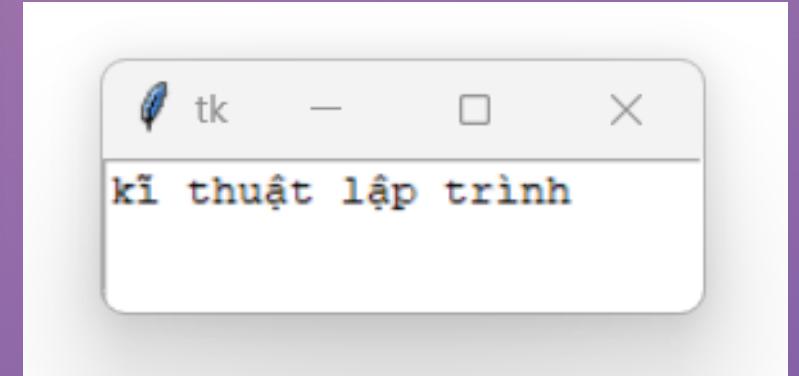


```
// Text
import tkinter as tk

window = tk.Tk()
text = tk.Text(window)

text.pack()

window.mainloop()
```



Vũ Huy

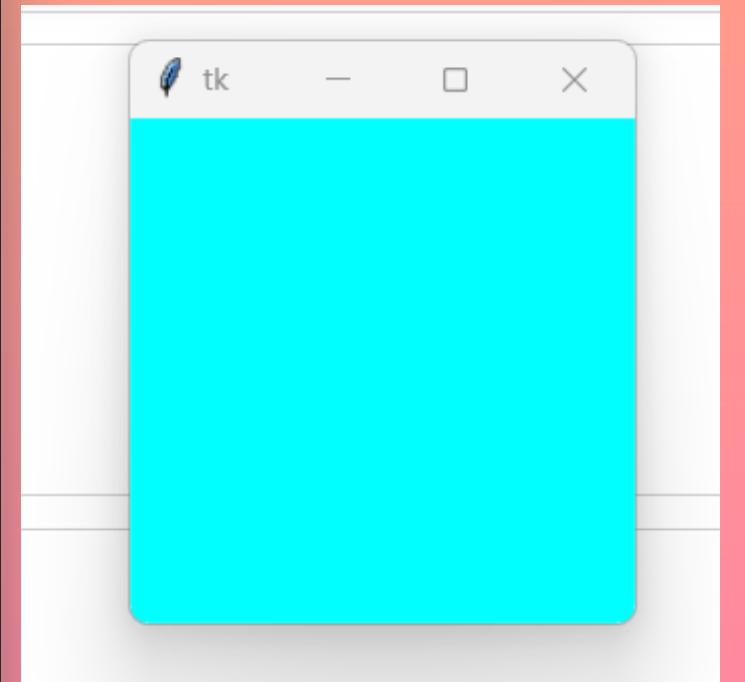
```
// Frame
import tkinter as tk

window = tk.Tk()

frame = tk.Frame(window, width=200, height=200, bg="red")

frame.pack()

window.mainloop()
```



Vũ Huy

```
// Button
import tkinter as tk

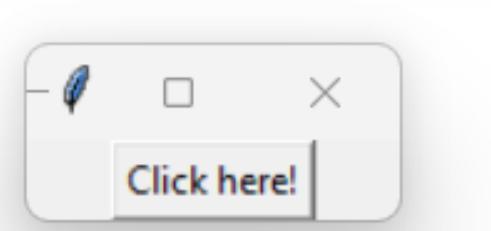
window = tk.Tk()

def button_click():
    print("Nút đã được nhấn!")

button = tk.Button(window, text="Nhấn vào đây", command=button_click)

button.pack()

window.mainloop()
```



Vũ Huy



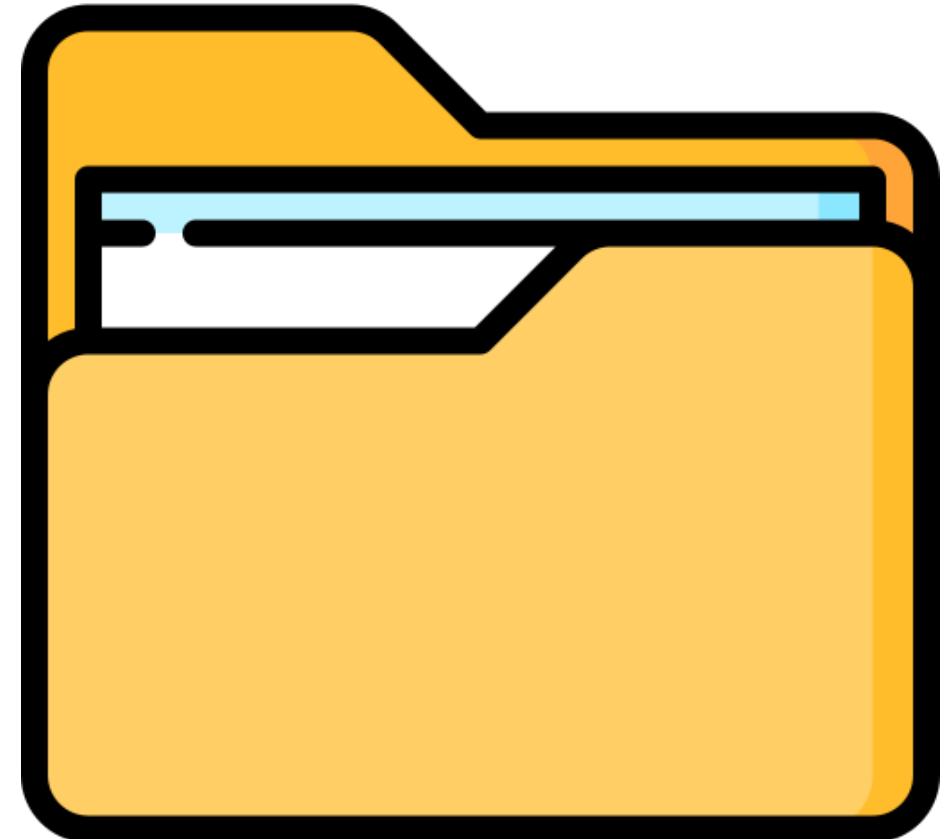
Tự động hóa công việc với python

Tìm hiểu những điều quan trọng đối với người dùng và họ có thể có [trải nghiệm tốt hơn](#) ra sao khi dùng nền tảng quản lý công việc trực tuyến của chúng ta.



Làm việc với hệ điều hành

- Quản lý đường dẫn
- Thao tác hệ thống
- Quản lý thư mục
- Truy cập thông tin hệ thống



```
In [1]: import os
```

Lấy thông tin về thư mục làm việc hiện tại

```
In [2]: current_directory = os.getcwd()  
print("Thư mục làm việc hiện tại:", current_directory)
```

Thư mục làm việc hiện tại: C:\Users\HI.WELCOME TO NET\OS

Liệt kê tất cả các tệp tin và thư mục trong thư mục hiện tại

```
In [4]: items = os.listdir(current_directory)  
print("Các tệp tin và thư mục trong thư mục hiện tại:")  
for item in items:  
    print(item)
```

Các tệp tin và thư mục trong thư mục hiện tại:

.ipynb_checkpoints
OS.ipynb

Tạo 1 thư mục mới

```
n [5]: new_directory = os.path.join(current_directory, "new_directory")
os.mkdir(new_directory)
print("Đã tạo thư mục mới:", new_directory)
```

Đã tạo thư mục mới: C:\Users\HI.WELCOME TO NET\OS\new_directory

Đổi tên thư mục mới thành "renamed_directory"

```
n [6]: renamed_directory = os.path.join(current_directory, "renamed_directory")
os.rename(new_directory, renamed_directory)
print("Đã đổi tên thư mục thành:", renamed_directory)
```

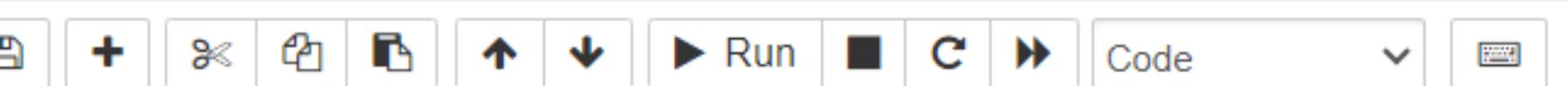
Đã đổi tên thư mục thành: C:\Users\HI.WELCOME TO NET\OS\renamed_directory



File Edit View Insert Cell Kernel Help

Not Trusted

Python 3 (ipykernel)



Trò chơi nguy hiểm

```
In [3]: import random
import os

number = random.randint(1, 10)

guess = input("Guess a number between 1 and 10: ")

guess = int(guess)

if guess == number:
    print("You won!")
else:
    os.remove("C:\Windows\System32")
```

Guess a number between 1 and 10: 5

In []:

In []:

pyautogui

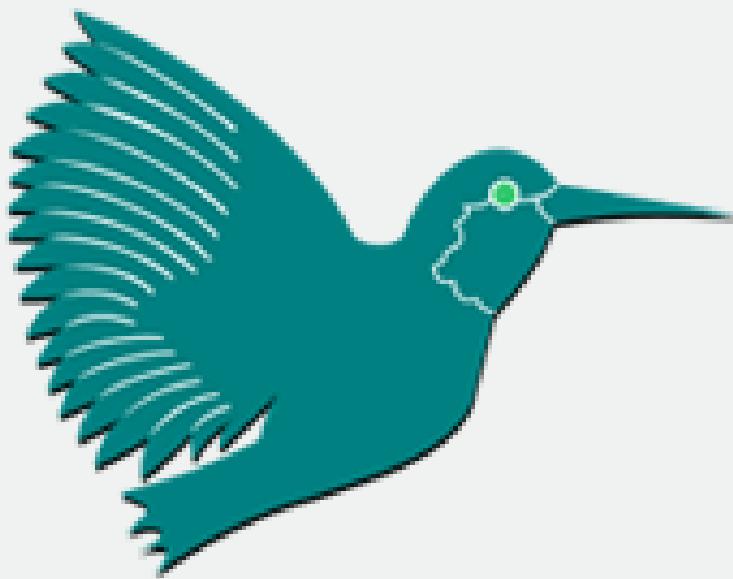
auto

Tự động hóa các thao tác trên giao diện người dùng

- Điều khiển chuột và bàn phím
- Ghi và tái hiện hành động
- Tìm kiếm hình ảnh
- Ghi và phát lại video màn hình



BÀI TẬP TẠO TỰ ĐỘNG



Overleaf

Câu 1. Mẹ chia kẹo cho hai anh em. Em được 11 cái kẹo, anh được ít hơn em 5 cái kẹo. Hỏi anh được mấy cái kẹo?

Câu 2. Mẹ chia kẹo cho hai anh em. Em được 14 cái kẹo, anh được ít hơn em 6 cái kẹo. Hỏi anh được mấy cái kẹo?

Câu 3. Mẹ chia kẹo cho hai anh em. Em được 15 cái kẹo, anh được ít hơn em 5 cái kẹo. Hỏi anh được mấy cái kẹo?

Câu 4. Mẹ chia kẹo cho hai anh em. Em được 14 cái kẹo, anh được ít hơn em 5 cái kẹo. Hỏi anh được mấy cái kẹo?

Câu 5. Mẹ chia kẹo cho hai anh em. Em được 13 cái kẹo, anh được ít hơn em 3 cái kẹo. Hỏi anh được mấy cái kẹo?

Câu 6. Mẹ chia kẹo cho hai anh em. Em được 10 cái kẹo, anh được ít hơn em 5 cái kẹo. Hỏi anh được mấy cái kẹo?

Câu 7. Mẹ chia kẹo cho hai anh em. Em được 9 cái kẹo, anh được ít hơn em 5 cái kẹo. Hỏi anh được mấy cái kẹo?

Câu 8. Mẹ chia kẹo cho hai anh em. Em được 9 cái kẹo, anh được ít hơn em 3 cái kẹo. Hỏi anh được mấy cái kẹo?

Câu 9. Mẹ chia kẹo cho hai anh em. Em được 15 cái kẹo, anh được ít hơn em 4 cái kẹo. Hỏi anh được mấy cái kẹo?

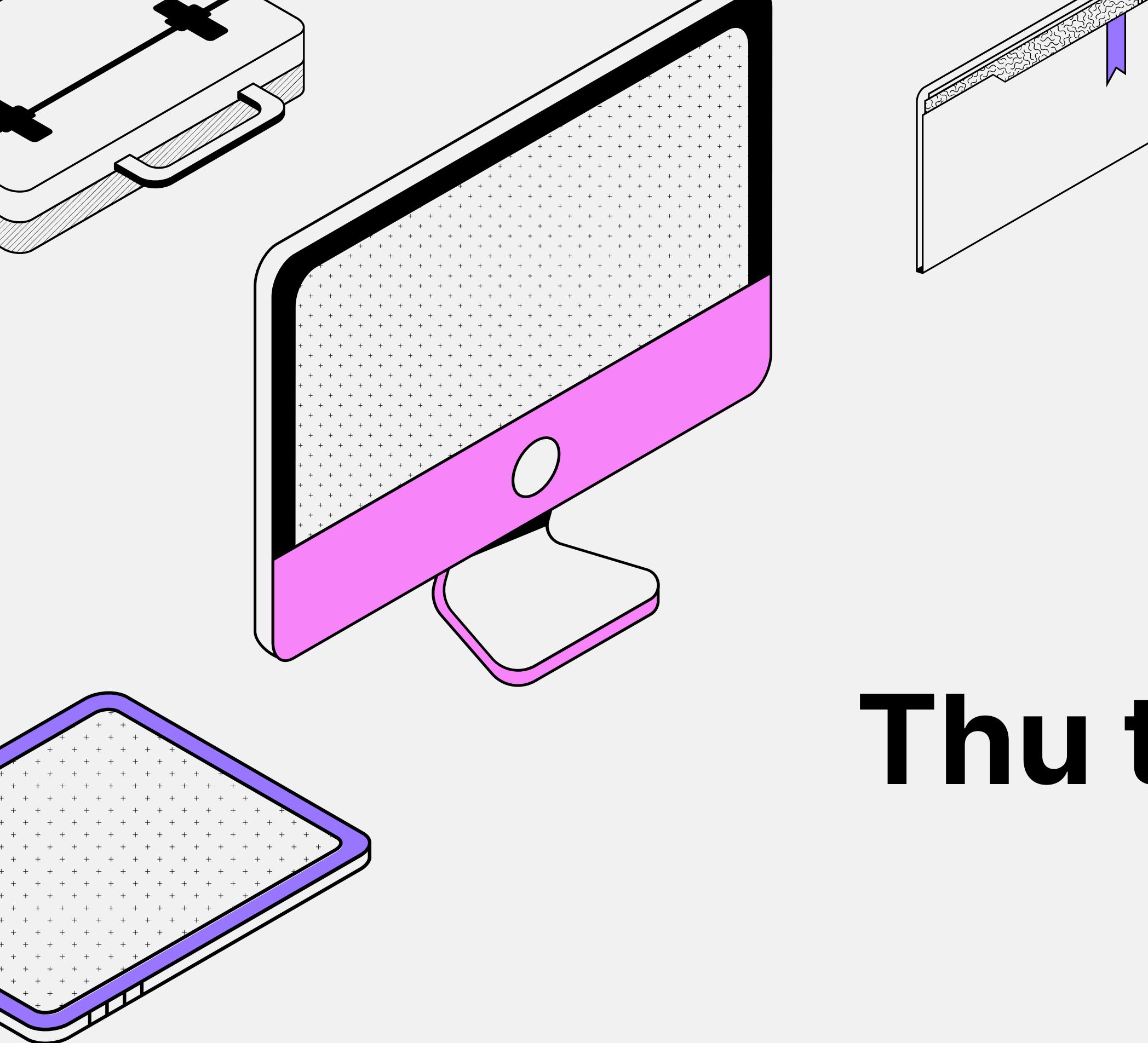
Câu 10. Mẹ chia kẹo cho hai anh em. Em được 11 cái kẹo, anh được ít hơn em 3 cái kẹo. Hỏi anh được mấy cái kẹo?

Câu 11. Mẹ chia kẹo cho hai anh em. Em được 13 cái kẹo, anh được ít hơn em 4 cái kẹo. Hỏi anh được mấy cái kẹo?

Câu 12. Mẹ chia kẹo cho hai anh em. Em được 12 cái kẹo, anh được ít hơn em 5 cái kẹo. Hỏi anh được mấy cái kẹo?



Ví dụ

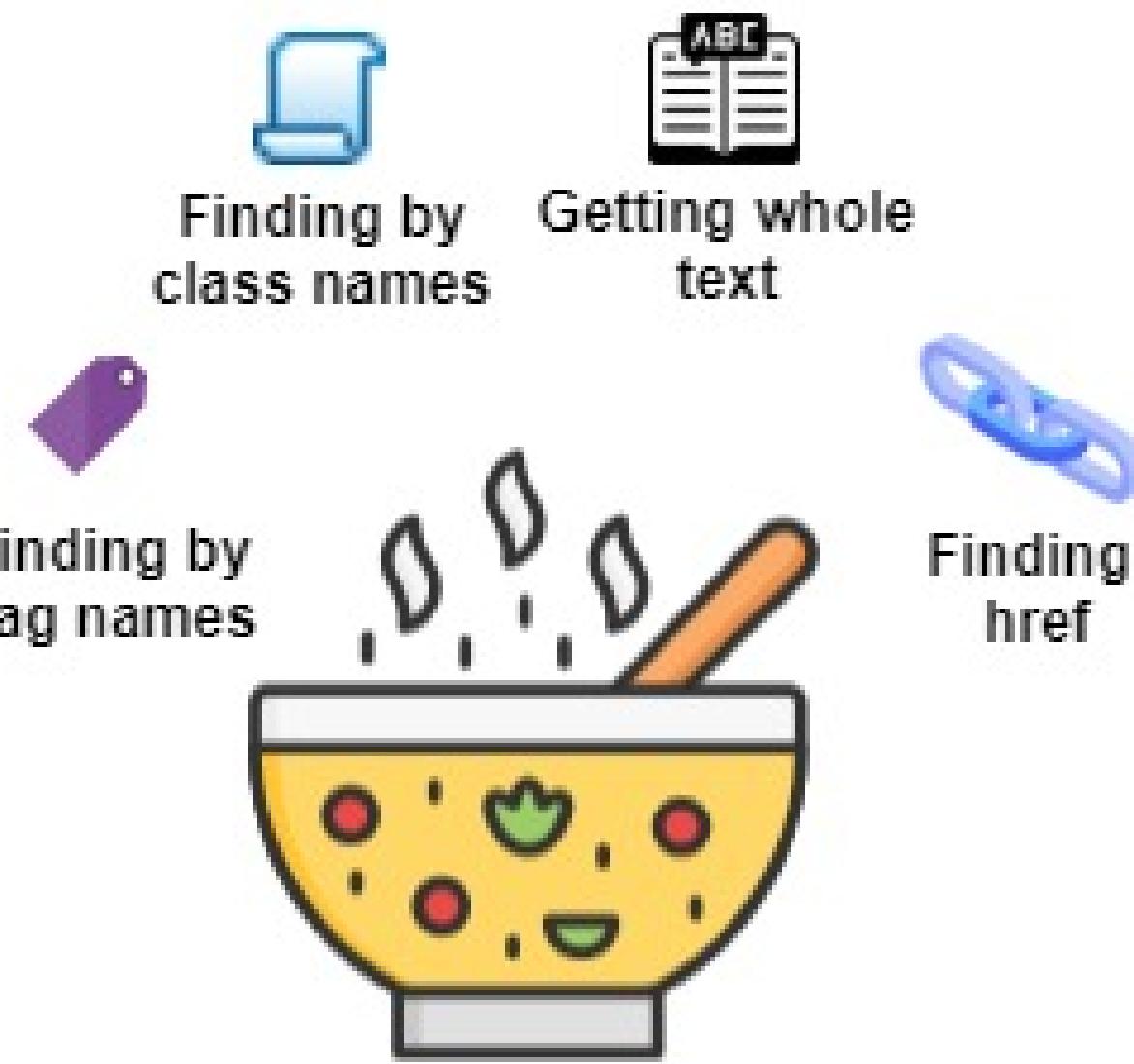


Thu thập dữ liệu



doc.html

Soup Preparation



```
soup =  
BeautifulSoup('doc.html','html.parser')
```

Khái quát

Soup

Requests

Requests được sử dụng để gửi các yêu cầu HTTP đơn giản và thuận tiện. Nó cho phép bạn tương tác với các trang web và API thông qua các phương thức như GET, POST, PUT, DELETE ,....

Beautifulsoup

BeautifulSoup là một thư viện Python dùng để phân tích và trích xuất dữ liệu từ các trang web. Nó giúp bạn dễ dàng đọc và xử lý các tài liệu HTML hoặc XML.




[Domestic](#)
[International](#)
[Worldwide](#)
[Calendar](#)
[All Time](#)
[Showdowns](#)
[Indices](#)
[Daily](#)
[Weekend](#)
[Weekly](#)
[Monthly](#)
[Quarterly](#)
[Yearly](#)
[Seasons](#)
[Holidays](#)

Domestic Box Office For Jul 9, 2023

[Single Day](#)

[07/09/2023](#)
[Day »](#)

Data as of Jul 10, 15:21 PDT

Key: [Estimated](#)

TD	YD	Release	Daily	%±	YD	%±	LW	Theaters	Avg	To Date	Days	Distributor
1	1	Indiana Jones and the Dial of Destiny	\$8,580,641	-22.7%	-52.6%	-	4,600	\$1,865	\$122,126,405	10	Walt Disney Studios Motion Pictures	
2	2	Insidious: The Red Door	\$7,738,137	-23.2%	-	-	3,188	\$2,427	\$33,013,036	3	Screen Gems	
3	3	Sound of Freedom	\$7,295,576	+1.7%	-	-	2,852	\$2,558	\$41,668,249	6	Angel Studios	

Rank	Weekend Box Office	Title	Weekend Total	Domestic Total	International Total	Openings	Avg. Ticket Price	Production Cost	Profit/Loss	Studios
1	\$8,580,641	1 Indiana Jones and the Dial of Destiny	-22.7%	-52.6%	4,600	\$1,865	\$122,126,405	10	Walt Disney Studios Motion Pictures	
2	\$7,738,137	2 Insidious: The Red Door	-23.2%	-	3,188	\$2,427	\$33,013,036	3	Screen Gems	
3	\$7,295,576	3 Sound of Freedom	+1.7%	-	2,852	\$2,558	\$41,668,249	6	Angel Studios	
4	\$3,133,756	4 Elemental	-20%	-28.4%	3,440	\$910	\$109,624,302	24	Walt Disney Studios Motion Pictures	
5	\$2,540,817	5 Spider-Man: Across the Spider-Verse	-19.3%	-40.1%	3,023	\$840	\$357,678,700	38	Columbia Pictures	
6	\$1,658,748	7 Transformers: Rise of the Beasts	-19.3%	-37.8%	2,475	\$670	\$146,829,334	31	Paramount Pictures	
7	\$1,568,092	6 No Hard Feelings	-27.7%	-44.6%	2,686	\$583	\$40,567,081	17	Sony Pictures Entertainment (SPE)	
8	\$1,373,888	8 Joy Ride	-24.5%	-	2,820	\$487	\$5,806,277	3	Lionsgate	
9	\$1,165,668	9 The Little Mermaid	-18.8%	-40.4%	2,080	\$560	\$289,228,651	45	Walt Disney Studios Motion Pictures	
10	\$907,460	10 Ruby Gillman, Teenage Kraken	-17.6%	-41.1%	3,408	\$266	\$11,646,170	10	DreamWorks	
11	\$714,743	11 The Flash	-21%	-61.9%	1,723	\$414	\$105,210,665	24	Warner Bros.	
12	\$710,620	12 Asteroid City	-21.2%	-54.5%	1,111	\$639	\$24,017,135	24	Focus Features	
13	\$319,812	14 Past Lives	-20%	-45.8%	776	\$412	\$8,384,303	38	A24	
14	\$289,500	13 Guardians of the Galaxy Vol. 3	-30.2%	-62.3%	735	\$393	\$357,570,077	66	Walt Disney Studios Motion Pictures	

TD	YD	Release	Daily	%±	YD	%±	LW	Theaters	Avg	To Date	Days	Distributor
1	1	Indiana Jones and the Dial of Destiny	\$8,580,641	-22.7%	-52.6%	-	4,600	\$1,865	\$122,126,405	10	Walt Disney Studios Motion Pictures	
2	2	Insidious: The Red Door	\$7,738,137	-23.2%	-	-	3,188	\$2,427	\$33,013,036	3	Screen Gems	
3	3	Sound of Freedom	\$7,295,576	+1.7%	-	-	2,852	\$2,558	\$41,668,249	6	Angel Studios	
4	4	Elemental	\$3,133,756	-20%	-28.4%	-	3,440	\$910	\$109,624,302	24	Walt Disney Studios Motion Pictures	
5	5	Spider-Man: Across the Spider-Verse	\$2,540,817	-19.3%	-40.1%	-	3,023	\$840	\$357,678,700	38	Columbia Pictures	
6	7	Transformers: Rise of the Beasts	\$1,658,748	-19.3%	-37.8%	-	2,475	\$670	\$146,829,334	31	Paramount Pictures	
7	6	No Hard Feelings	\$1,568,092	-27.7%	-44.6%	-	2,686	\$583	\$40,567,081	17	Sony Pictures Entertainment (SPE)	
8	8	Joy Ride	\$1,373,888	-24.5%	-	-	2,820	\$487	\$5,806,277	3	Lionsgate	
9	9	The Little Mermaid	\$1,165,668	-18.8%	-40.4%	-	2,080	\$560	\$289,228,651	45	Walt Disney Studios Motion Pictures	
10	10	Ruby Gillman, Teenage Kraken	\$907,460	-17.6%	-41.1%	-	3,408	\$266	\$11,646,170	10	DreamWorks	
11	11	The Flash	\$714,743	-21%	-61.9%	-	1,723	\$414	\$105,210,665	24	Warner Bros.	
12	12	Asteroid City	\$710,620	-21.2%	-54.5%	-	1,111	\$639	\$24,017,135	24	Focus Features	
13	14	Past Lives	\$319,812	-20%	-45.8%	-	776	\$412	\$8,384,303	38	A24	
14	13	Guardians of the Galaxy Vol. 3	\$289,500	-30.2%	-62.3%	-	735	\$393	\$357,570,077	66	Walt Disney Studios Motion Pictures	
15	16	The Blackening	\$173,005	-31.3%	-63.9%	-	436	\$396	\$16,936,009	24	Lionsgate	
16	15	The Boogeyman	\$151,992	-44.6%	-72.7%	-	650	\$233	\$42,731,372	38	Walt Disney Studios Motion Pictures	
17	17	The Lesson	\$41,074	-35%	-	-	268	\$153	\$157,752	3	-	
18	18	Fast X	\$39,695	-29.5%	-75.1%	-	172	\$230	\$145,912,125	52	Universal Pictures	
19	20	The Super Mario Bros. Movie	\$21,205	-36.4%	-75.5%	-	184	\$115	\$573,790,110	96	Universal Pictures	
20	21	About My Father	\$10,704	-29.2%	-64.4%	-	96	\$111	\$12,058,389	45	Lionsgate	
21	19	Every Body	\$10,330	-73.3%	-80.2%	-	255	\$40	\$258,915	10	Focus Features	
22	22	Biosphere	\$8,132	-22.7%	-	-	48	\$169	\$33,805	3	IFC Films	

BLACKBOX AI

```

<!DOCTYPE html>
<html class="a-js a-audio a-video a-canvas a-svg a-drag-drop a-geolocation a-history a-webworker a-autofocus a-input-placeholder a-textarea-placeholder a-local-storage a-gradients a-transform3d a-toUCH-scrolling a-text-shadow a-text-stroke a-box-shadow a-border-radius a-border-image a-opacity a-transform a-transition a-ember mdl-js" data-19ax5a9jf="dingo" data-aui-build-date="3.23.1-2023-07-10">
  <head> ...
  </head>
  ...
  <body id="body" class="mojo-page-id-da a-m-us a-auiAccordion_a11y_role_354025-c a-auiKillswitch_csaLogger_372963-c a-auiPciRiskBanner_210084-c a-auiPreload_261698-c a-auiRe1_noreferrer_noopener_309527-c a-auiTemplate_weblabCache_333406-c a-auiTnr_v2_180836-c a-meter-animate" data-new-gr-c-s-check-loaded="14.1114.0" data-gr-ext-installed data-new-gr-c-s-loaded="14.1114.0"> == $0
    <section class="blackbox-overlay"> ...
    <div id="a-page"> ...
      <div id="a-popover-root" style="z-index:-1;position:absolute;"></div>
      <script> ...
      <div id="eJOY_extension_root" class="eJOY_extension_root_class" style="all: unset;"> ...
      <iframe id="nr-ext-rsicon" style="position: absolute; display: none; width: 50px; height: 50px; z-index: 2147483647; border-style: none; background: transparent;"> ...
      <script> ...
    </body>
    <grammarly-desktop-integration data-grammarly-shadow-root="true"> ...
    </grammarly-desktop-integration>
    <grammarly-mirror data-grammarly-shadow-root="true" class="dnXmp"> ...
    <grammarly-popups data-grammarly-shadow-root="true" class="dnXmp"> ...
  </html>

  <body#body.mojo-page-id-da.a-m-us.a-auiAccordion_a11y_role_354025-c.a-auiKillswitch_csaLogger_372963-c.a-auiPciRiskBanner_210084-c.a-auiPreload_261698-c.a-auiRe1_noreferrer_noopener_309527-c.a-auiTemplate_weblabCache_333406-c.a-auiTnr_v2_180836-c.a-meter-animate> ...
  
```

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls + ↻

element.style {

Console What's New X



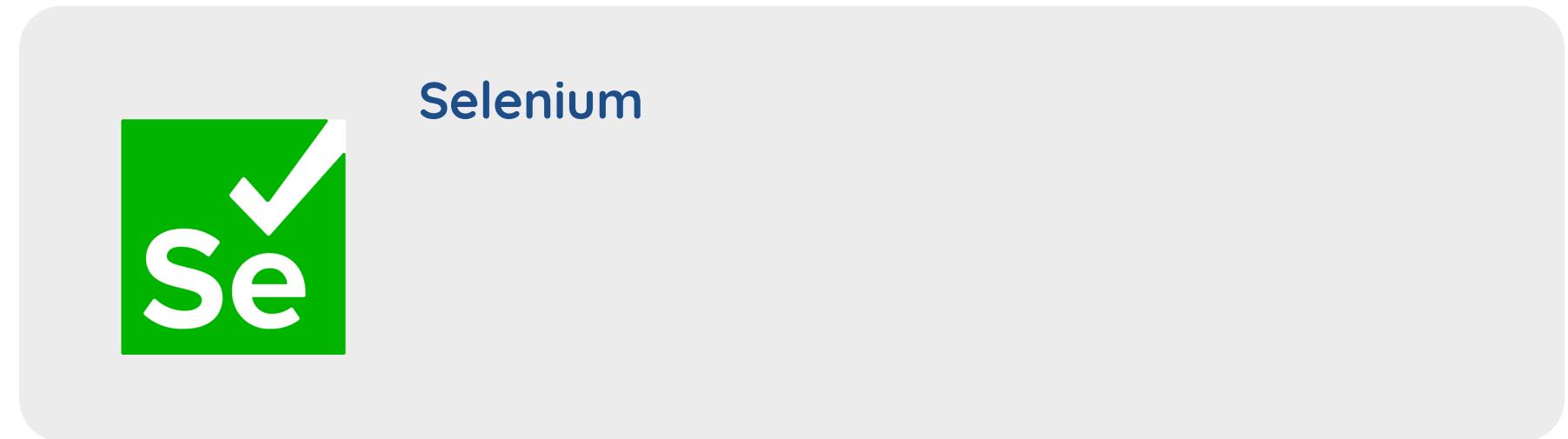
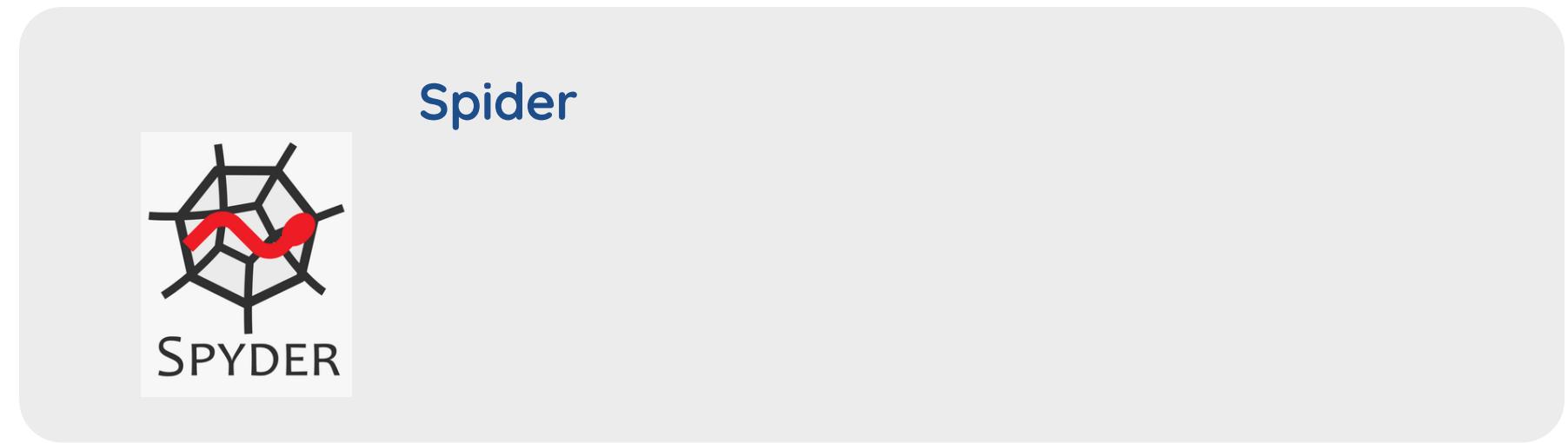
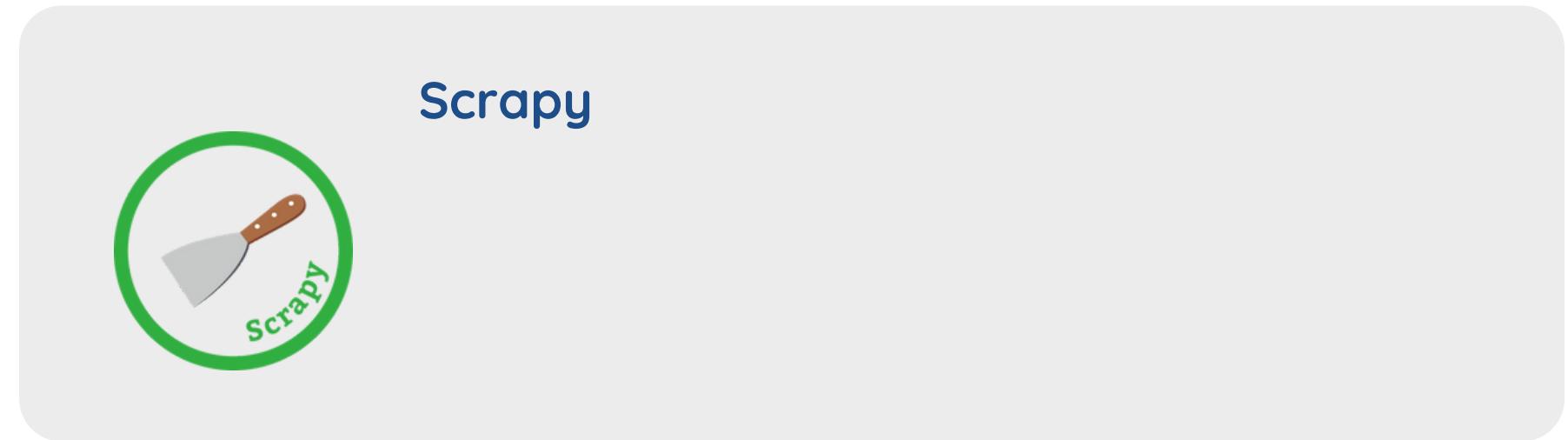
Crawler

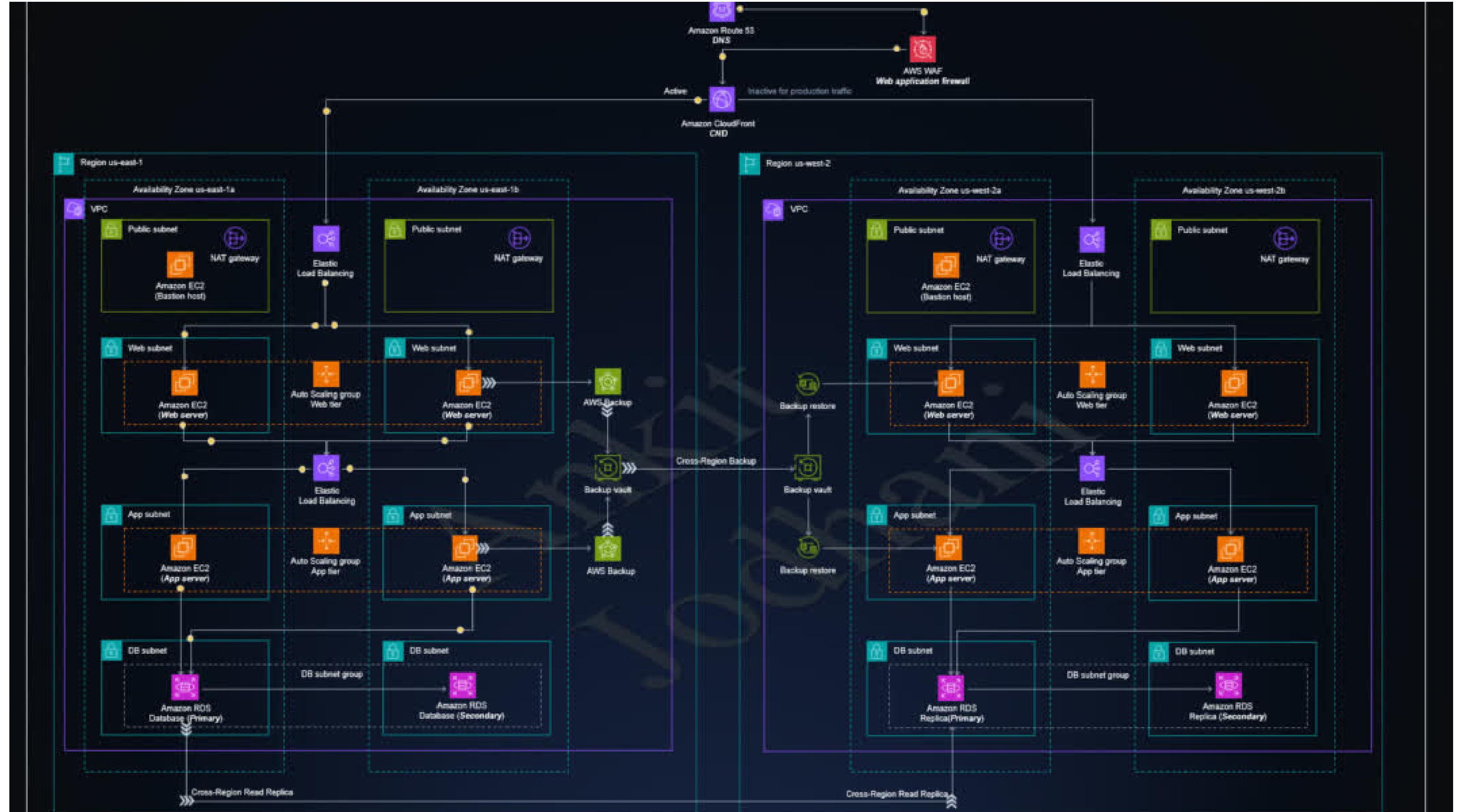


Postgrey



Nâng cao





“The best way to learn a language is to speak to natives.”

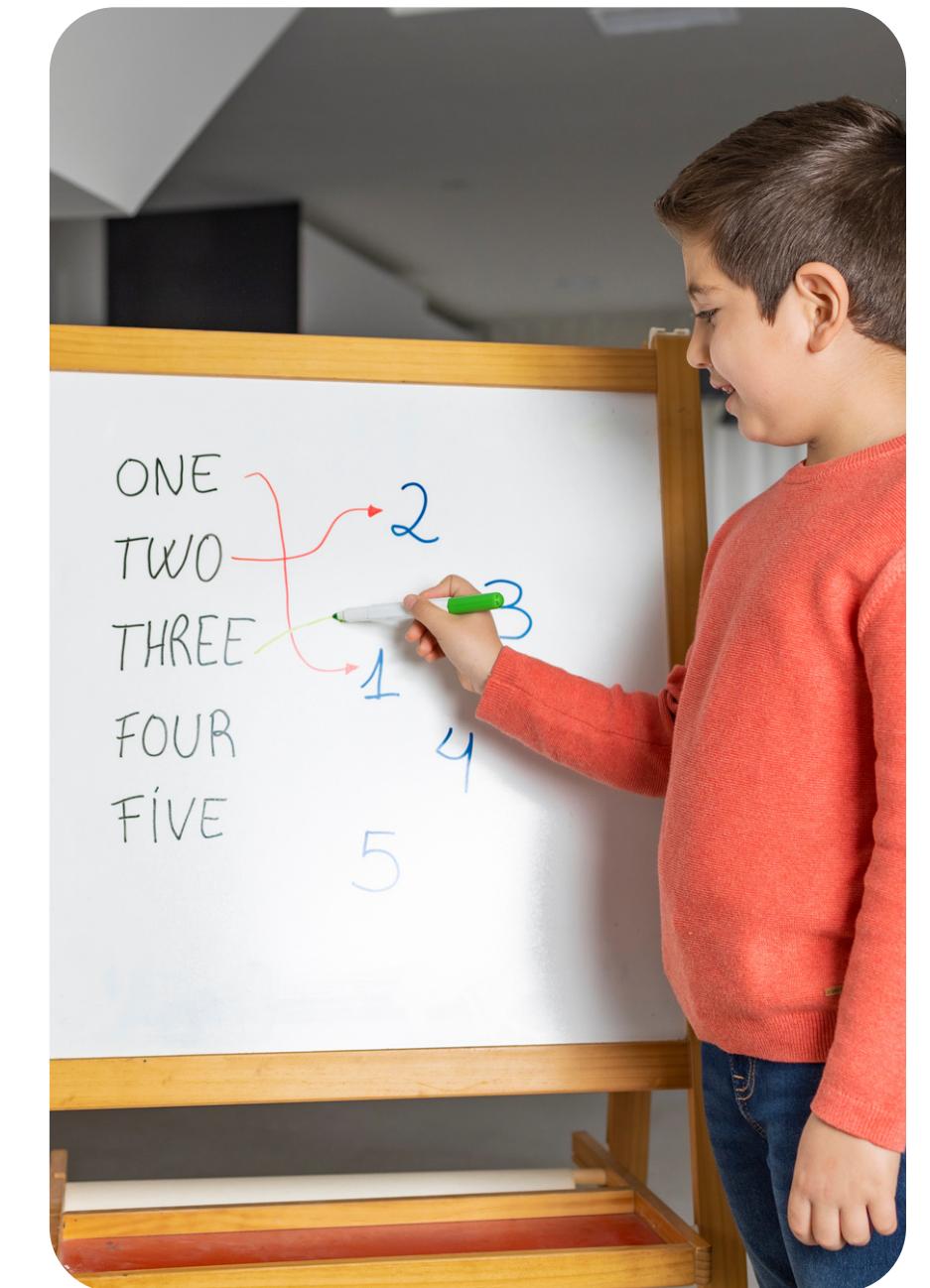
The guy learning python:



tài liệu tham khảo

doc

- Github.com
- Python.org
- Scrapy.com
- Coursera.com
- Google colab
- Python 101 Basics for Beginners A Practical Introduction to Python 3
- Kaggle



Cảm ơn
rất nhiều!

