

Tài liệu hướng dẫn thực hành Buổi 4

Môn : Lập trình Python

Nội dung chính : Quản lý và sử dụng thư viện trong python

1. Các thư viện hỗ trợ

a. Cách cài đặt thư viện hỗ trợ

Nếu đã cài đặt Conda thì ta mở **anaconda command line** để tiến hành cài đặt các thư viện cần thiết. ta sử dụng cú pháp :

conda install -c anaconda <<tên thư viện cần cài đặt>>

Ví dụ : ta cần cài đặt thư viện numpy :

conda install -c anaconda numpy

Nếu không sử dụng anaconda, ta sử dụng công cụ pip để cài đặt :

pip install <<tên thư viện>>

Ví dụ : cài đặt thư viện matplotlib

pip install matplotlib

Để kiểm tra danh sách những thư viện đã cài đặt, ta dùng lệnh :

pip list

b. Một số thư viện phổ biến trong Python

Numpy : Đây là thư viện hỗ trợ rất mạnh trong việc tính toán trên các ma trận và các dữ liệu lớn, mảng đa chiều. Numpy còn hỗ trợ quản lý các mảng cực kỳ vượt trội, nhanh chóng và hữu hiệu.

Pillow : Đây là thư viện phổ biến được sử dụng để xử lý hình ảnh trong Python. Thư viện này hỗ trợ các module thao tác cơ bản trên hình ảnh, giúp cho việc xử lý hình ảnh trở nên thuận tiện và dễ dàng hơn trong Python.

Pandas : Đây là một thư viện được thiết kế để làm việc dễ dàng và trực quan với dữ liệu có cấu trúc và dữ liệu chuỗi thời gian. Pandas sử dụng 2 cấu trúc chính là Series để xử lý dữ liệu 1 chiều và Dataframe để xử lý dữ liệu 2 chiều. Pandas dùng để xử lý phần lớn các dữ liệu điển hình trong tài chính, thống kê, khoa học và kỹ thuật.

Matplotlib : Đây là thư viện giúp chúng ta vẽ và hiển thị các đồ thị 2D một cách trực quan thường được dùng trong thống kê và khoa học dữ liệu. Matplotlib có thể kết xuất những đồ thị chất lượng một cách dễ dàng và thuận tiện.

Open CV : Open CV là viết tắt của Open Source Computer Vision, là một thư viện hỗ trợ mạnh mẽ trong việc thiết kế và xây dựng các ứng dụng liên quan đến thị giác máy tính. OpenCV cung cấp các module cho phép đọc, xử lý và thay đổi nhiều dữ liệu hình ảnh cùng một lúc.

Tensorflow : Đây là thư viện mã nguồn mở rất dễ học và dễ sử dụng dành cho lĩnh vực Máy học – Machine Learning. Tensorflow cung cấp nhiều API có hiệu suất cao và tự động, giúp người dùng dễ dàng xây dựng các mô hình máy học theo ý muốn của mình.

Keras : Keras là một thư viện về mạng thần kinh sâu mã nguồn mở được viết bằng Python. Keras cung cấp một quyền kiểm tra hiệu quả trên các mạng chi tiết. Nó có khả năng chạy trên nhiều nền tảng khác nhau và đem lại hiệu quả cao.

NLTK : NLTK là viết tắt của Natural Language Toolkit. Đây là một trong những thư viện xử lý ngôn ngữ tự nhiên tốt nhất của Python. Nó là một tập hợp các thư viện xử lý ngôn ngữ các cung cấp các giải pháp xử lý ngôn ngữ số và ký hiệu dành cho tiếng Anh. NLTK cho phép xử lý, phân loại, gán thẻ, mã hóa, phân tích cú pháp và lập luận ngữ nghĩa. NLTK là sự chọn lựa hoàn hảo cho các dự án về xử lý ngôn ngữ tự nhiên.

2. Bài tập

Sinh viên tạo file <<Hoten>>_THBuoi4.py (vd : tranvanan_THBuoi4.py), Nhớ chú thích họ tên và MSSV lên đầu file.

Phần 1 : Thao tác trên mảng với Numpy

Câu 1 : Sử dụng thư viện Numpy, Hãy khởi tạo một mảng một chiều gồm các giá trị : 1, 3, 5, 7, 9 sau đó in mảng đó ra màn hình (Gợi ý : `numpy.array()`)

Câu 2 : Sử dụng thư viện Numpy, Hãy khởi tạo một mảng hai chiều gồm các giá trị của dòng thứ nhất: 1, 3, 5, 7, 9 và dòng thứ hai : 2, 4, 6, 8, 10 sau đó in mảng đó ra màn hình (Gợi ý : `numpy.array()`)

Câu 3 : Sử dụng thư viện Numpy, Hãy khởi tạo một mảng gồm 10 giá trị 0 và sau đó cập nhật giá trị thứ 6 thành số 13. In mảng đó ra màn hình (Gợi ý : `numpy.zeros()`)

Câu 4 : Sử dụng thư viện Numpy, Hãy khởi tạo một mảng hai chiều với kích thước 5x5 mà giá trị các đường viền đều là số 1, giá trị các phần tử còn lại là 0. (Gợi ý : `numpy.ones()`)

Câu 5 : Sử dụng thư viện Numpy, Hãy khởi tạo một mảng hai chiều với kích thước 3x3 mà giá trị các phần tử đều là 1. Sau đó thêm 1 đường viền bao bên ngoài theo cả 2 chiều với toàn giá trị 0 vào mảng vừa rồi. In mảng kết quả ra màn hình (Gợi ý : `numpy.ones()` và `numpy.pad()`)

Câu 6 : Sử dụng thư viện Numpy, Hãy khởi tạo mảng thứ nhất gồm các phần tử : [10, 20, 40, 60, 70] và mảng thứ 2 gồm các phần tử : [10, 30, 50, 60]. Hãy tìm các phần tử chung của 2 mảng và in kết quả ra màn hình (Gợi ý : `numpy.intersect1d()`)

Câu 7 : Sử dụng thư viện Numpy, Hãy khởi tạo mảng A gồm các phần tử toàn số chẵn từ 1 đến 100 (tính luôn số 100). In mảng A ra màn hình. (Gợi ý : `numpy.linspace()`)

Phần 2 : Vẽ đồ thị với Matplotlib

Hướng dẫn : Để có thể vẽ đồ thị với thư viện matplotlib, ta tiến hành import module pyplot của thư viện này.

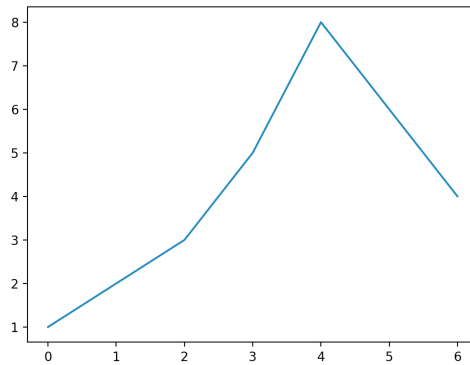
Ví dụ :

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 5, 8, 6, 4]

plt.plot(x)
plt.show()
```

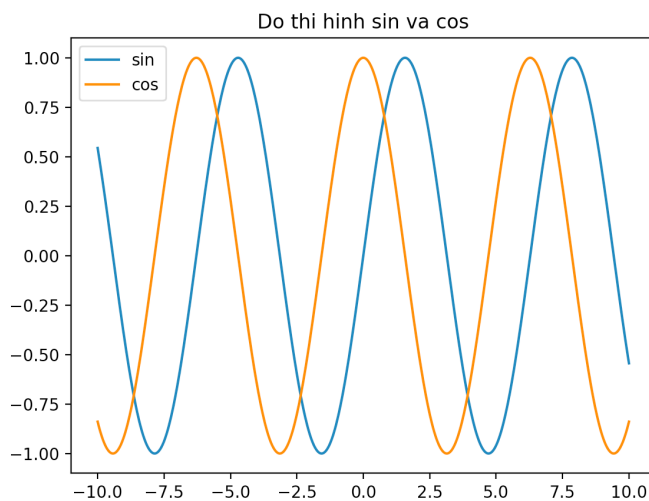
Ta được kết quả như sau :



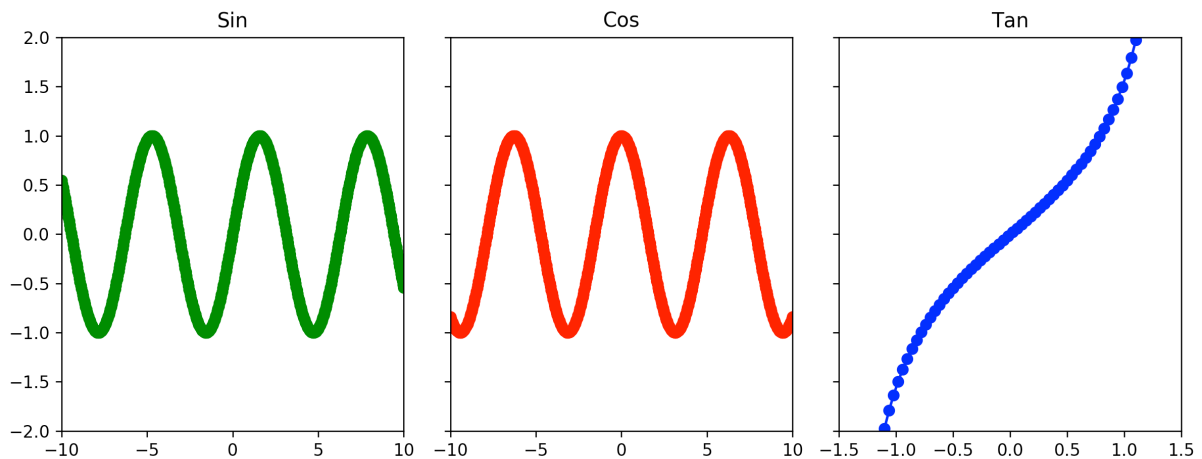
Câu 8 : Sử dụng thư viện Numpy và matplotlib, Hãy vẽ đồ thị hàm số $y = \sin(x)$ với x chạy từ đoạn -10 đến 10. (Gợi ý : `numpy.linspace()` và `plt.scatter()`, `plt.show()`)

Câu 9 : Sử dụng thư viện Numpy và matplotlib, Hãy vẽ đồ thị hàm số $x^3 - 2x^2 + x + 5$ với x chạy từ đoạn -5 đến 5. (Gợi ý : `numpy.linspace()` và `plt.scatter()`, `plt.show()`)

Câu 10 : Sử dụng thư viện Numpy và matplotlib, Hãy vẽ đồ thị hàm số $y = \sin(x)$ và $y = \cos(x)$ trên cùng một hình với x chạy từ đoạn -10 đến 10. Đặt tiêu đề là “Đồ thị hình sin và cos”, gán legend tương ứng và đặt ở vị trí tốt nhất. (Gợi ý : `plt.plot()`, `plt.title()`, `plt.legend()`). Kết quả như hình bên dưới :



Câu 11 : Sử dụng thư viện Numpy và matplotlib, Hãy vẽ 3 đồ thị hàm số $y = \sin(x)$, $y = \cos(x)$ và $y = \tan(x)$ trong cùng 1 ảnh với 3 đồ thị khác nhau. Mỗi ảnh có tiêu đề riêng. Kết quả giống như hình bên dưới. (Gợi ý : `plt.subplots()`, `plt.plot()`, `plt.title()`). Hai đồ thị Cos và Sin thì phần X hiển thị từ -10 đến 10, đồ thị Tan thì hiển thị X từ -1.5 đến 1.5. Cả 3 Đồ thị thì có Y hiển thị trong khoảng từ -2 đến 2 (dùng : `set_xlim()`, `set_ylim()`). Mỗi đồ thị hiển thị bằng một màu khác nhau.



Phần 3 : Xử lý ảnh với Pillow

Hướng dẫn : Ta cần cài đặt thư viện pillow và import module Image vào :

from PIL import Image

Một số hàm phổ biến :

- Image.open(): Mở ảnh
- Image.show() : Hiển thị ảnh
- Image.save() : Lưu ảnh
- Image.crop() : Cắt ảnh
- Image.resize() : Thay đổi kích thước của ảnh
- Image.rotate() : Xoay ảnh
- Image.convert() : Chuyển đổi chế độ màu của ảnh

Câu 12 : Viết chương trình cho phép đọc vào một ảnh, sau đó hiển thị ảnh đó ra màn hình

Câu 13 : Viết chương trình cho phép đọc vào một ảnh, sau đó tạo một box có kích thước 200x200 tại vị trí bắt đầu ở điểm có tọa độ (100, 100). Cắt ảnh với box tương ứng và hiển thị ảnh ra màn hình

Câu 14 : Viết chương trình cho phép đọc vào một ảnh, xoay ảnh đó 180 độ, sau đó lưu ảnh lại với tên là “anhchinh sua.jpg” và hiển thị ảnh mới ra màn hình

Câu 15 : Viết chương trình cho phép chuyển ảnh màu thành ảnh trắng đen, sau đó xoay ảnh trắng đen 90 độ, lưu ảnh lại với tên “anhtrangden.jpg” và hiển thị ảnh mới ra màn hình.