

vncoder.vn

Bài 22: Matplotlib - Numpy trong Python

4-5 phút

Matplotlib là một thư viện vẽ đồ thị cho Python. Nó được sử dụng cùng với NumPy để cung cấp môi trường thay thế nguồn mở hiệu quả cho MatLab. Nó cũng có thể được sử dụng với các bộ công cụ đồ họa như PyQt và wxPython.

Matplotlib lần đầu tiên được viết bởi John D. Hunter. Kể từ năm 2012, Michael Droettboom là nhà phát triển chính. Hiện tại, Matplotlib ver. 1.5.1 là phiên bản ổn định có sẵn. Gói này có sẵn ở dạng phân phối nhị phân cũng như ở dạng mã nguồn trên www.matplotlib.org.

Thông thường, package được nhập vào Python như sau:

```
from matplotlib import pyplot as plt
```

Ở đây **pyplot ()** là hàm quan trọng nhất trong thư viện matplotlib, được sử dụng để vẽ dữ liệu 2D. Tập lệnh sau vẽ phương trình $y = 2x + 5$

Ví dụ :

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

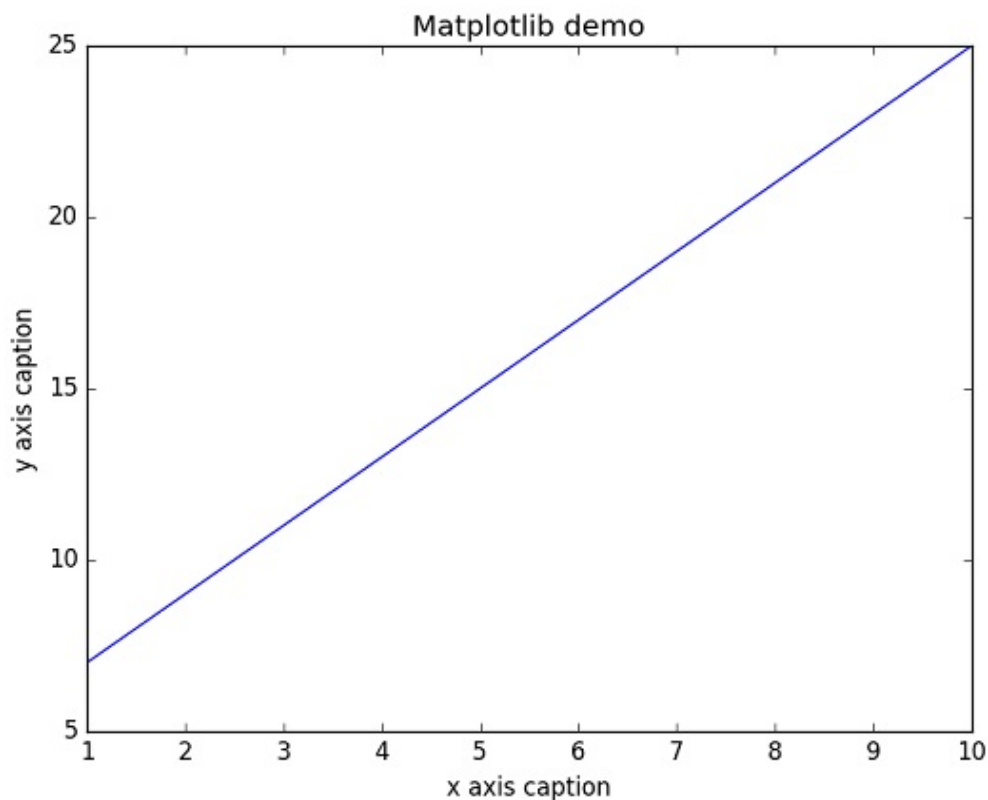
x = np.arange(1,11)
y = 2 * x + 5
plt.title("Matplotlib demo")
```

```
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x, y)
plt.show()
```

Đối tượng ndarray x được tạo từ hàm **np.arange ()** như các giá trị trên trục x. Các giá trị tương ứng trên trục y được lưu trữ trong một đối tượng ndarray khác y. Các giá trị này được vẽ bằng cách sử dụng hàm **plot ()** của mô-đun con pyplot trong package matplotlib.

Biểu diễn đồ thị bằng hàm show ().

Kết quả như sau :



Thay vì biểu đồ tuyến tính, các giá trị có thể được hiển thị riêng biệt bằng cách thêm chuỗi định dạng vào hàm plot (). Các ký tự định dạng sau có thể được sử dụng.

1. '-' : Kiểu đường liền nét
2. '-.': Kiểu đường đứt nét

3. '-.' : Kiểu đường dấu gạch ngang
4. ':': Kiểu đường chấm
5. '.' : Điểm đánh dấu
6. ',': Điểm đánh dấu pixel
7. 'o': Điểm đánh dấu vòng tròn
8. 'v': Điểm đánh dấu tam giác_dưới
9. '^': Điểm đánh dấu tam giác_trên
10. '<': Điểm đánh dấu tam giác_trái
11. '>': Điểm đánh dấu tam giác_phải

Các chữ viết tắt màu sau đây :

Character	Color
'b'	Blue
'g'	Green
'r'	Red
'c'	Cyan
'm'	Magenta
'y'	Yellow
'k'	Black
'w'	White

Để hiển thị các vòng tròn đại diện cho các điểm, thay vì dòng trong ví dụ trên, ta có thể sử dụng “ob” làm chuỗi định dạng trong hàm plot ().

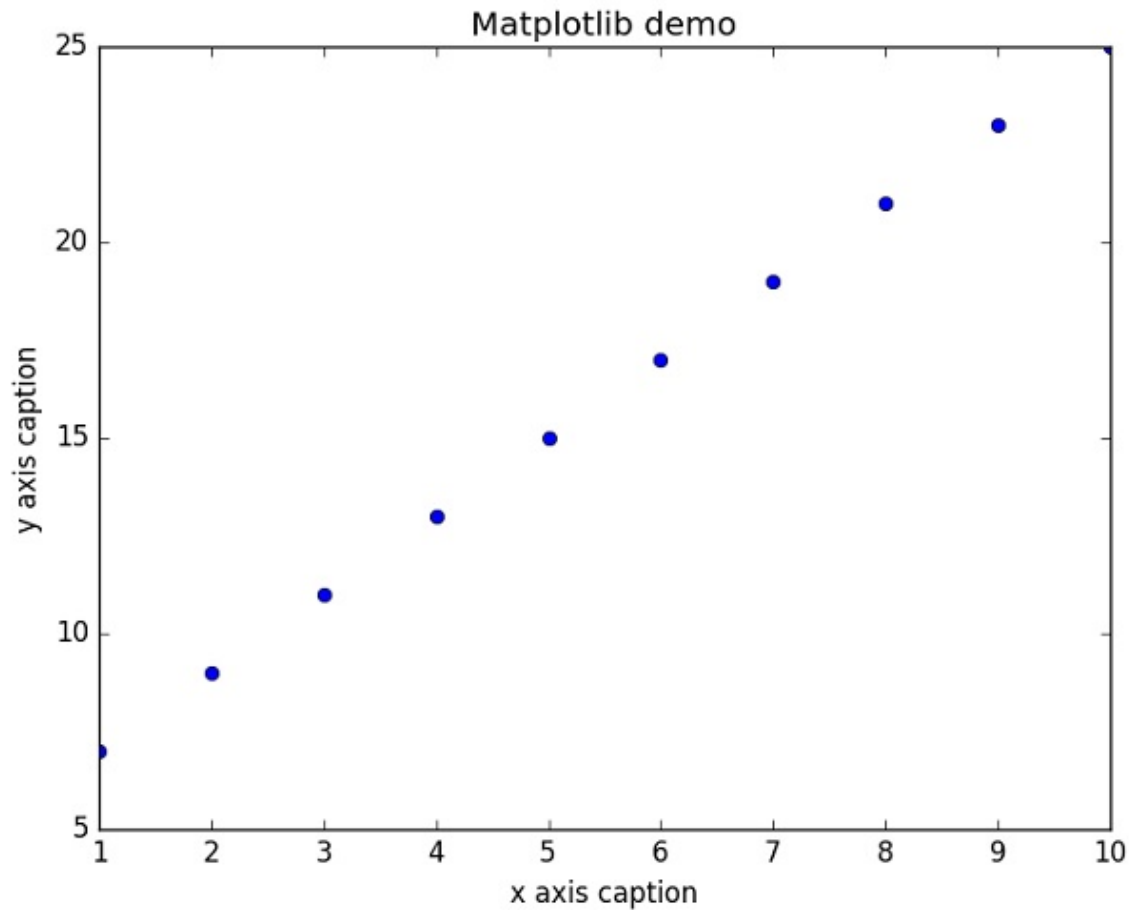
Ví dụ :

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

x = np.arange(1,11)
y = 2 * x + 5
plt.title("Matplotlib demo")
```

```
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x, y, "ob")
plt.show()
```

Kết quả :

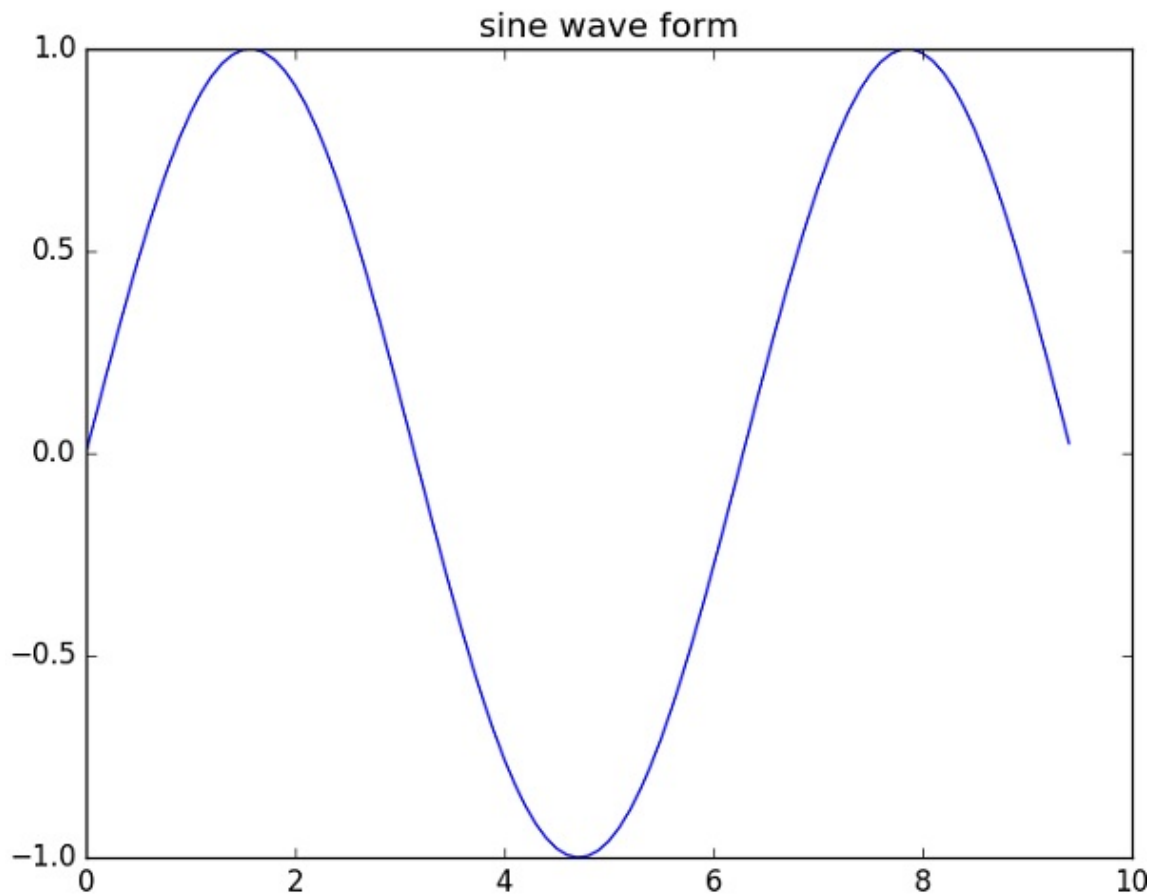


1. Sóng hình sin :

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

# Compute the x and y coordinates for points on
a sine curve
x = np.arange(0, 3 * np.pi, 0.1)
y = np.sin(x)
plt.title("sine wave form")
```

```
# Plot the points using matplotlib
plt.plot(x, y)
plt.show()
```



2. subplot()

Hàm `subplot()` cho phép bạn vẽ những thứ khác nhau trong cùng một hình. Trong ví dụ sau, các giá trị sin và cosine được vẽ trong cùng biểu đồ.

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

# Compute the x and y coordinates for points on
sine and cosine curves
x = np.arange(0, 3 * np.pi, 0.1)
y_sin = np.sin(x)
y_cos = np.cos(x)
```

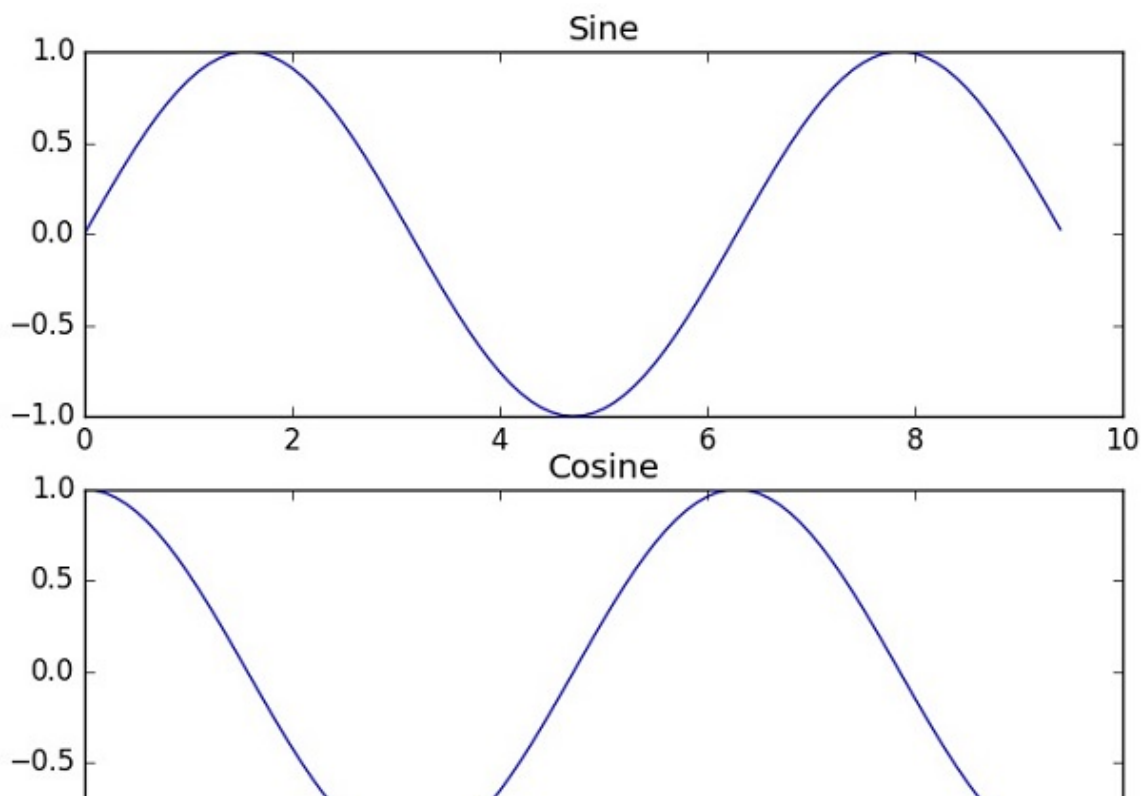
```
# Set up a subplot grid that has height 2 and
width 1,
# and set the first such subplot as active.
plt.subplot(2, 1, 1)

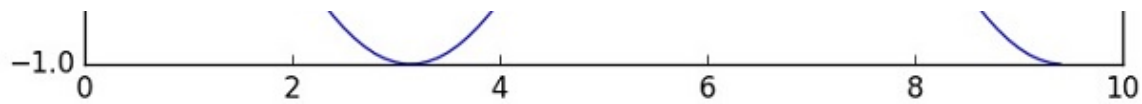
# Make the first plot
plt.plot(x, y_sin)
plt.title('Sine')

# Set the second subplot as active, and make
the second plot.
plt.subplot(2, 1, 2)
plt.plot(x, y_cos)
plt.title('Cosine')

# Show the figure.
plt.show()
```

Kết quả :





3. bar()

pyplot submodule cung cấp hàm `bar ()` để tạo đồ thị hình cột.

Ví dụ sau tạo biểu đồ thanh của hai tập hợp các mảng `x` và `y`.

```
from matplotlib import pyplot as plt
x = [5,8,10]
y = [12,16,6]

x2 = [6,9,11]
y2 = [6,15,7]
plt.bar(x, y, align = 'center')
plt.bar(x2, y2, color = 'g', align = 'center')
plt.title('Bar graph')
plt.ylabel('Y axis')
plt.xlabel('X axis')

plt.show()
```

Kết quả :

