

Contents

Matplotlib Tutorial.....	1
Matplotlib Getting Started.....	1
Matplotlib Pyplot	1
Matplotlib Plotting	1
Vẽ biểu đồ điểm x và điểm y	1
Nhiều điểm.....	2
Matplotlib Markers	2
Điểm đánh dấu Matplotlib.....	2
Tham chiếu dòng: Tham chiếu màu:	2
Kích thước điểm đánh dấu và Màu đánh dấu	3
Matplotlib Line	3
Line Styles	3
Line Color	4
Line Width	4
Matplotlib Labels and Title	4
Create Labels for a Plot	4
Create a Title for a Plot	4
Matplotlib Adding Grid Lines (lưới)	4
Matplotlib Subplots	4
Matplotlib Scatter :biểu đồ phân tán.....	5
Matplotlib Bars	5
Matplotlib Histograms Biểu đồ Matplotlib	6
Matplotlib Pie Charts: biểu đồ bánh	6

Matplotlib Tutorial

Matplotlib là gì?

Matplotlib là một thư viện vẽ đồ thị cấp thấp trong python, nó phục vụ như một tiện ích trực quan hóa. Matplotlib được tạo ra bởi John D. Hunter.

Matplotlib là mã nguồn mở và chúng ta có thể sử dụng nó một cách tự do.

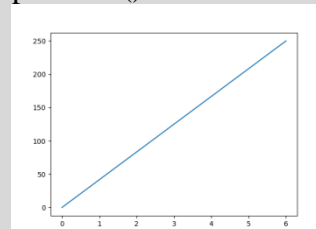
Matplotlib Getting Started

```
#Cài đặt : C:\Users\Your Name>pip install matplotlib
#Sau khi Matplotlib được cài đặt, hãy nhập nó vào các
ứng dụng của bạn bằng cách thêm câu lệnh:import
module
import matplotlib
#kiểm tra phiên bản
print(matplotlib.__version__)
3.0.3
```

Matplotlib Pyplot

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
# Xem biểu đồ trong Jupyter Notebook
%matplotlib inline
print("Setup Complete")
Setup Complete

#Hầu hết các tiện ích Matplotlib nằm dưới pyplotmô-
đun con và thường được nhập dưới pltbí danh: import
matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([0, 6]) #ngang
ypoints = np.array([0, 250]) #dọc
plt.plot(xpoints, ypoints)
plt.show()
```



Matplotlib Plotting

Vẽ biểu đồ điểm x và điểm y

Các **plot()** chức năng được sử dụng để vẽ điểm (marker) trong một sơ đồ.

Theo mặc định, **plot()** hàm vẽ một đường thẳng từ điểm này đến điểm khác.

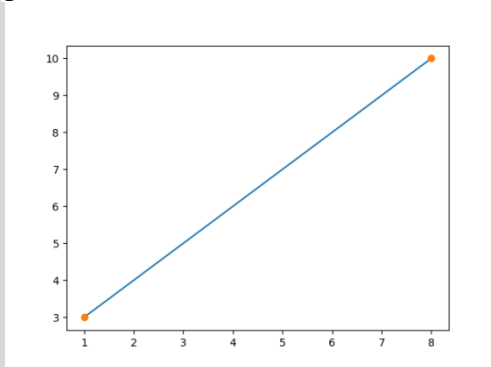
Hàm nhận các tham số để xác định các điểm trong biểu đồ.

Tham số 1 là một mảng chứa các điểm trên trục x .

Tham số 2 là một mảng chứa các điểm trên trục y .

Nếu chúng ta cần vẽ một dòng từ (1, 3) đến (8, 10), chúng ta phải chuyển hai mảng [1, 8] và [3, 10] cho hàm plot.

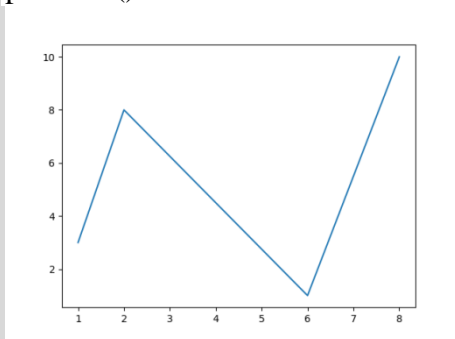
```
from __future__ import division, print_function,
unicode_literals
import numpy as np
import matplotlib.pyplot as plt
xpoints = np.array([1, 8])
ypoints = np.array([3, 10])
#vẽ đường
plt.plot(xpoints, ypoints)
#Đề vẽ các điểm đánh dấu, sử dụng tham số ký hiệu
chuỗi phím tắt 'o', có nghĩa là 'vòng'.
plt.plot(xpoints, ypoints, "o")
plt.show()
```



Nhiều điểm

#Bạn có thể vẽ bao nhiêu điểm tùy thích, chỉ cần đảm bảo rằng bạn có cùng số điểm trong cả hai trục.

```
from __future__ import division, print_function,
unicode_literals
import numpy as np
import matplotlib.pyplot as plt
xpoints = np.array([1, 2, 6, 8])
ypoints = np.array([3, 8, 1, 10])
plt.plot(xpoints, ypoints)
plt.show()
```

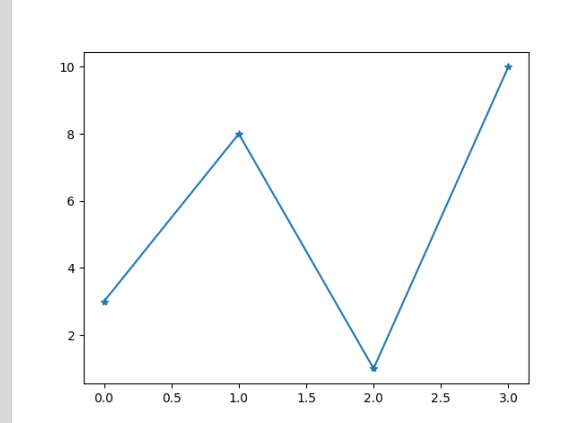


Matplotlib Markers

Điểm đánh dấu Matplotlib

To support both python 2 and python 3
from __future__ import division, print_function,
unicode_literals
import numpy as np

```
import matplotlib.pyplot as plt
ypoints = np.array([3, 8, 1, 10])# => xpoints mặc định
từ 0 đến 4
#Điểm đánh dấu :Bạn có thể sử dụng đối số từ khóa
marker để nhấn mạnh từng điểm bằng một điểm đánh
dấu được chỉ định:
plt.plot(ypoints, marker = '*')
plt.show()
```



Còn có thể có nhiều ký tự khác:

```
'o' '*' '.' ',' 'x' 'X' '+' 'P' 's' 'D'
'd' 'p' 'H' 'h' 'v' '^' '<' '>' '1'
'2' '3' '4' '|' ' _ '
```

Tham chiếu dòng: Tham chiếu màu:

Bạn cũng có thể sử dụng tham số *ký hiệu chuỗi phím tắt* để chỉ định điểm đánh dấu.

Tham số này cũng được gọi là **fmt** và được viết với cú pháp sau:

marker|line|color

```
from __future__ import division, print_function,
unicode_literals
```

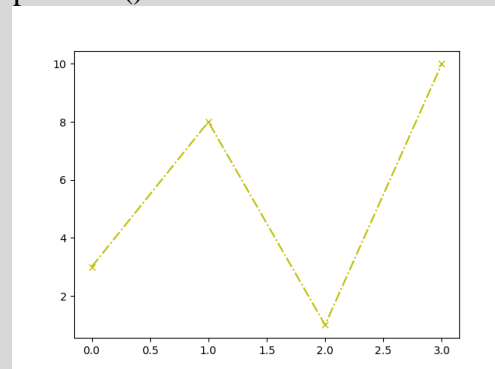
```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
ypoints = np.array([3, 8, 1, 10])
```

```
plt.plot(ypoints, 'x-.y') # đánh dấu x , dòng kẻ - . , và
có màu vàng
```

```
plt.show()
```

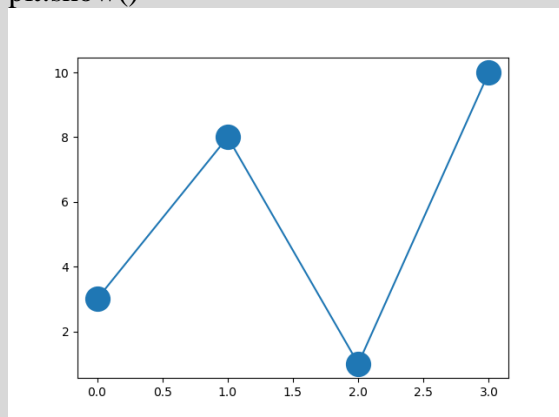


Tham chiếu dòng: `Line Syntax: '-
' : ' ' _ _ ' ' _ . '`

Tham chiếu màu: `Color
Syntax: 'r' 'g' 'b' 'c' 'm' 'y' 'k' '
w'`

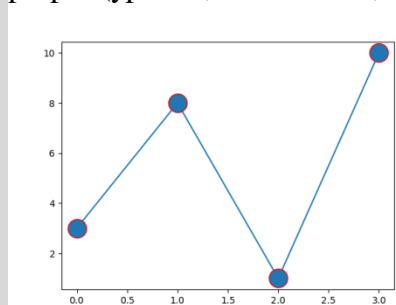
Kích thước điểm đánh dấu và Màu đánh dấu
Bạn có thể sử dụng đối số từ khóa `markersize` hoặc phiên bản ngắn hơn, `ms` để đặt kích thước của các điểm đánh dấu:

```
from __future__ import division, print_function,  
unicode_literals  
import numpy as np  
import matplotlib.pyplot as plt  
ypoints = np.array([3, 8, 1, 10])  
plt.plot(ypoints, marker = 'o', ms = 20)  
plt.show()
```



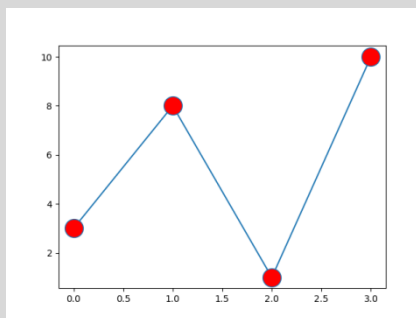
Bạn có thể sử dụng đối số từ khóa `markeredgecolor` hoặc đối số ngắn hơn `mec` để đặt màu cho *cạnh* của các điểm đánh dấu:

```
plt.plot(ypoints, marker = 'o', ms = 20, mec = 'r')
```



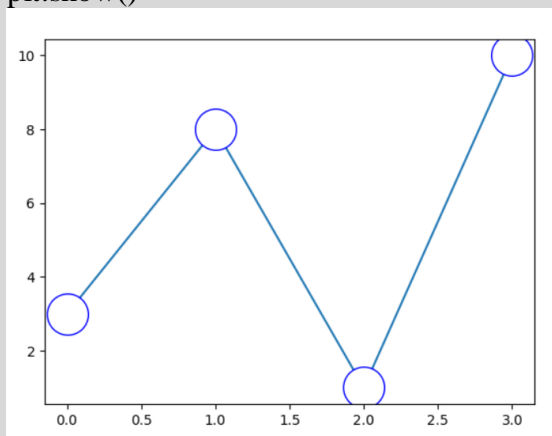
Bạn có thể sử dụng đối số từ khóa `markerfacecolor` hoặc đối số ngắn hơn `mfc` để đặt màu bên trong cạnh của điểm đánh dấu:

```
plt.plot(ypoints, marker = 'o', ms = 20, mfc = 'r')
```



=> Sử dụng cả các `mec` và `mfc` đối số cho màu sắc của toàn bộ marker

```
from __future__ import division, print_function,  
unicode_literals  
import numpy as np  
import matplotlib.pyplot as plt  
ypoints = np.array([3, 8, 1, 10])  
plt.plot(ypoints, marker = 'o', ms =  
30, mec='b', mfc='w')  
plt.show()
```

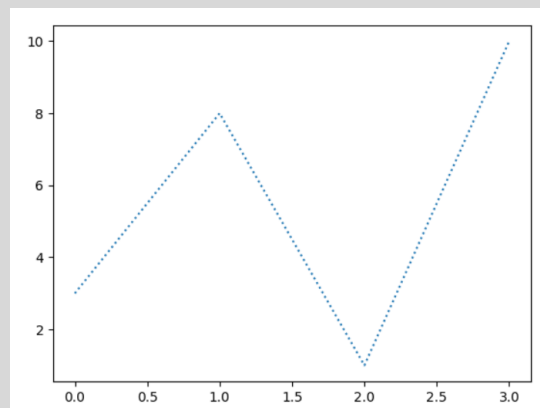


Matplotlib Line

Line Styles

You can use the keyword argument `linestyle`, or shorter `ls`, to change the style of the plotted line:

```
from __future__ import division, print_function,  
unicode_literals  
import numpy as np  
import matplotlib.pyplot as plt  
ypoints = np.array([3, 8, 1, 10])  
plt.plot(ypoints, ls = ':')  
plt.show()
```



You can choose any of these styles: '-':'---
'. or ''

Line Color

You can use the keyword argument `color` or the shorter `c` to set the color of the line:

```
plt.plot(ypoints, color = 'r')
```

```
plt.plot(ypoints, c = 'r')
```

Line Width

You can use the keyword argument `linewidth` or the shorter `lw` to change the width of the line.

```
plt.plot(ypoints, linewidth = '20.5')
```

Matplotlib Labels and Title

Create Labels for a Plot

With Pyplot, you can use

the `xlabel()` and `ylabel()` functions to set a label for the x- and y-axis.

Create a Title for a Plot

With Pyplot, you can use the `title()` function to set a title for the plot.

```
from __future__ import division, print_function,
unicode_literals
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
```

```
y = np.array([240, 250, 260, 270, 280, 290, 300, 310,
320, 330])
```

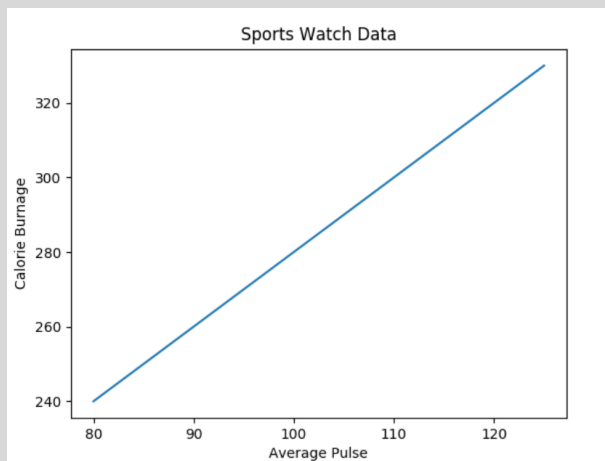
```
plt.plot(x, y)
```

```
plt.title("Sports Watch Data")
```

```
plt.xlabel("Average Pulse")
```

```
plt.ylabel("Calorie Burnage")
```

```
plt.show()
```



Matplotlib Adding Grid Lines (lưới)

```
from __future__ import division, print_function,
unicode_literals
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
```

```
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])
```

```
plt.title("Sports Watch Data")
```

```
plt.xlabel("Average Pulse")
```

```
plt.ylabel("Calorie Burnage")
```

```
plt.plot(x, y)
```

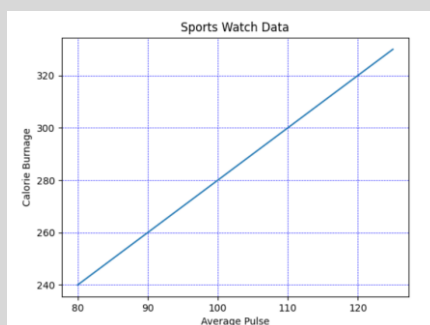
```
plt.grid(c= 'b', ls = '--', lw = 0.5)
```

plt.show()

```
plt.grid() #vẽ lưới theo cả 2 trục
```

```
plt.grid(axis = 'x') #vẽ lưới dọc theo trục x
```

```
plt.grid(axis = 'y') ##vẽ lưới ngang theo trục y
```



Matplotlib Subplots

With the `subplots()` function you can draw multiple plots in one figure:

```
from __future__ import division, print_function,
unicode_literals
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])
plt.subplot(1, 2, 1)
plt.plot(x,y)
plt.title("SALES")
```

```
#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])
plt.subplot(1, 2, 2)
plt.plot(x,y)
plt.title("INCOMES")
#Super Title : siêu tiêu đề
plt.suptitle("MY SHOP")
plt.show()
```



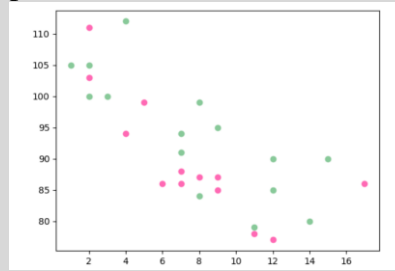
```
plt.subplot(1, 2, 1)
#the figure has 1 row, 2 columns, and this plot is
the first plot.
plt.subplot(1, 2, 2)
#the figure has 1 row, 2 columns, and this plot is
the second plot.
```

Matplotlib Scatter :biểu đồ phân tán

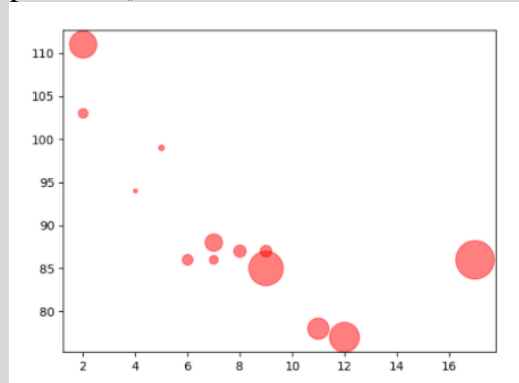
With Pyplot, you can use the `scatter()` function to draw a scatter plot.

```
from __future__ import division, print_function,
unicode_literals
import numpy as np
import matplotlib.pyplot as plt
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y =
np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
plt.scatter(x, y, color = 'hotpink')
x = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])
y =
np.array([100,105,84,105,90,99,90,95,94,100,79,112,
91,80,85])
```

```
plt.scatter(x, y, color = '#88c999')
plt.show()
```



```
Size::
from __future__ import division, print_function,
unicode_literals
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import numpy as np
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y =
np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
sizes =
np.array([20,50,100,200,500,1000,60,90,10,300,600,8
00,75])
#kích thước s=size // s=10...
#Bạn có thể điều chỉnh độ trong suốt của các dấu
chấm bằng alpha đối số.
plt.scatter(x, y, s=sizes, alpha=0.5,c='r')
plt.show()
```

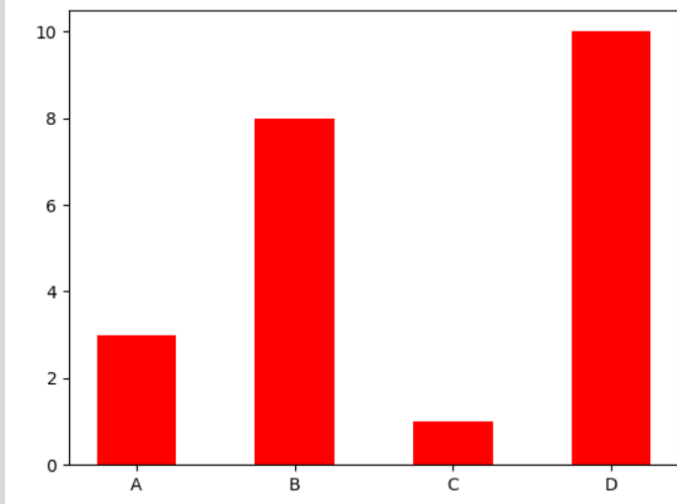


Matplotlib Bars

```
from __future__ import division, print_function,
unicode_literals
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import numpy as np
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])
#Với Pyplot, bạn có thể sử dụng bar() chức năng để vẽ
biểu đồ thanh:
plt.bar(x,y,color='r',width=0.5)
```

#Lưu ý: Đối với thanh ngang, sử dụng height thay thế width.

plt.show()

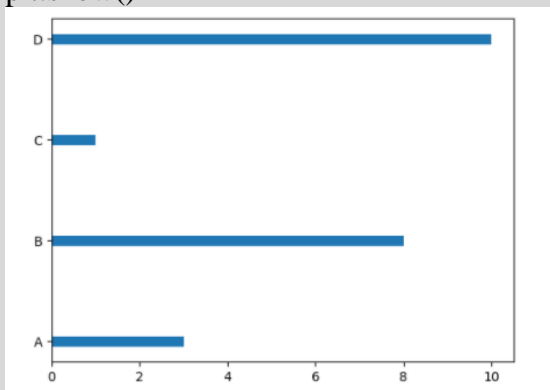


Thanh ngang

Nếu bạn muốn các thanh được hiển thị theo chiều ngang thay vì theo chiều dọc, hãy sử dụng `barh()` chức năng:

```
from __future__ import division, print_function,
unicode_literals
import numpy as np
import matplotlib.pyplot as plt
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])
```

```
plt.barh(x, y, height = 0.1)
plt.show()
```



Matplotlib Histograms Biểu đồ Matplotlib

Biểu đồ là một biểu đồ hiển thị phân bố *tần số*.

Đó là một biểu đồ thể hiện số lượng quan sát trong mỗi khoảng thời gian nhất định.

```
from __future__ import division, print_function,
unicode_literals
import numpy as np
import matplotlib.pyplot as plt
```

x =

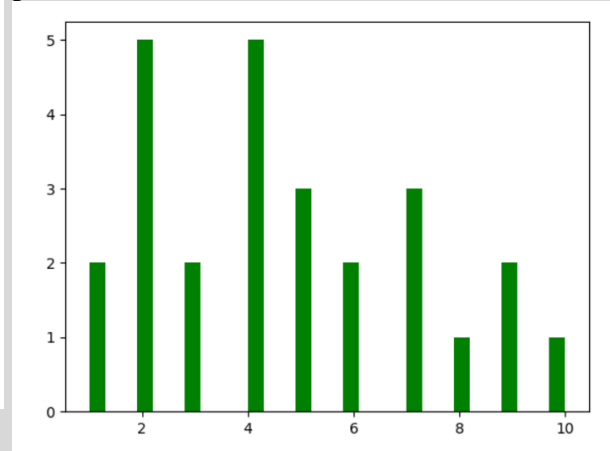
[2,1,6,4,2,4,8,9,4,2,4,10,6,4,5,7,7,3,2,7,5,3,5,9,2,1]

#plot for a histogram

#bín : khoảng cách giữa hai cột

plt.hist(x, bins = 8, color='green', alpha=1)

plt.show()



Matplotlib Pie Charts: biểu đồ bánh

```
from __future__ import division, print_function,
unicode_literals
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
y = np.array([35, 25, 25, 15])
```

```
mylabels = ["Apples", "Bananas", "Cherries",
"Dates"]
```

#labels: gán nhãn, startangle trực (90-0-270-180) theo chiều kim đồng hồ

```
plt.pie(y, labels = mylabels, startangle = 90)
```

```
plt.show()
```

