



Live de
Python

Introdução ao matplotlib

Obrigado!

apoia.se/livedepython

```
dunossauro at bouman in ~/git/apoiase on master*  
$ python apoiadores.py  
Alysson Oliveira      Amaziles Carvalho    Andre Machado        Bruno Guizi  
Carlos Augusto        Cleber Santos        David Reis           Dayham Soares  
Diego Ubirajara       Edimar Fardim        Eliabe Silva         Elias Soares  
Emerson Lara          Fabiano Silos        Fabiano Teichmann    Fabiano Gomes  
Fernando Furtado      Fábio Serrão         Gleison Oliveira     Humberto Rocha  
JONATHAN DOMINGOS     Jean Vetorello       Johnny Tardin        Jonatas Oliveira  
Jonatas Simões        João Lugão           Jucélio Silva        Júlia Kastrup  
Leon Teixeira         Magno Malkut         Maria Boladona       Matheus Francisco  
Nilo Pereira          Pablo Henrique       Paulo Tadei          Pedro Alves  
Rafael Galleani       Regis Santos         Renan Moura          Renato Santos  
Rennan Almeida        Rodrigo Vaccari       Sérgio Passos        Thiago Araujo  
Tiago Cordeiro        Vergil Valverde      Vicente Marcal       Wander Silva  
Wellington Carlos     Wellington Camargo    Welton Souza         William Oliveira  
Willian Gl            Yros Aguiar          Falta você           Falta você
```

Python Pro

Python para Profissionais

Roteiro

- Entendo o básico sobre o matplotlib
- Alguns conceitos básicos primordiais
 - Títulos
 - Labels
 - Legendas
- Explorando alguns tipos de gráficos
 - `plt.lines`
 - `plt.bar`
 - `plt.barh`
 - `plt.pie`
 - `plt.hist`
- Trabalhando com dados

BAIXE A BASE
(<https://bit.ly/2HFkCtb>)

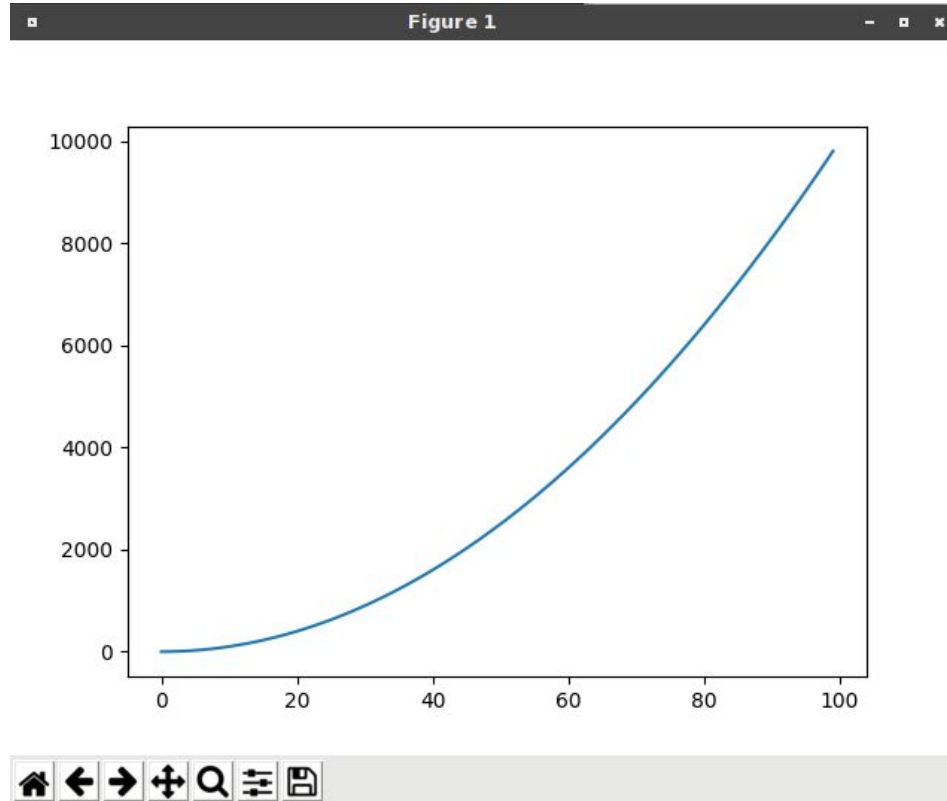
Que raios é matplotlib?

Matplotlib é um módulo Python para plotting, e é um componente da suíte de módulos 'scientificPython'. Matplotlib permite que você prepare facilmente figuras de nível profissional com uma API abrangente para personalizar todos os aspectos das figuras.

5 linhas sem perder a amizade [0]

```
import matplotlib.pyplot as plt
X = range(100)
Y = [value ** 2 for value in X]
plt.plot(X, Y)
plt.show()
```

5 linhas sem perder a amizade [1]

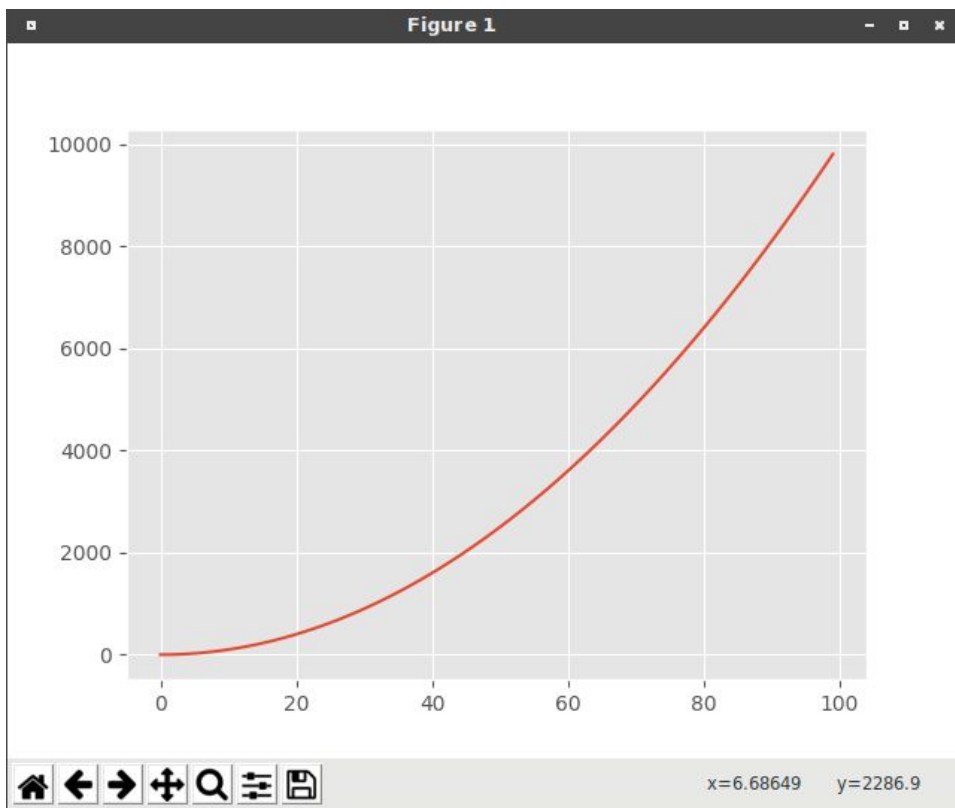


O raio embelezador

Agora com 7 e mais bonito [0]

```
import matplotlib.pyplot as plt
from matplotlib import style
style.use('ggplot')
X = range(100)
Y = [value ** 2 for value in X]
plt.plot(X, Y)
plt.show()
```

Agora com 7 e mais bonito [1]



Uma consideração anterior

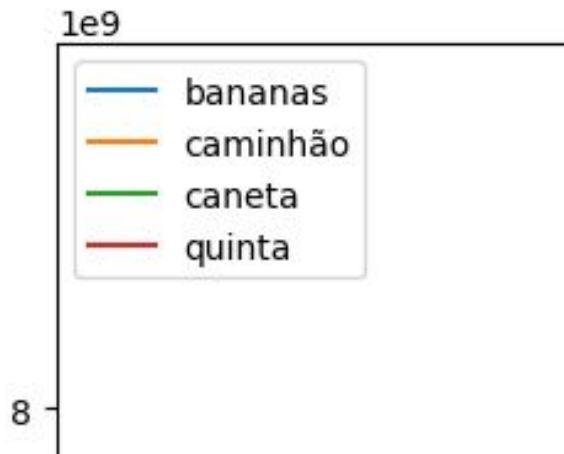
```
from matplotlib.pyplot as plt
```

(Tudo que vamos ver daqui pra frente vai omitir esses passos)

```
plt.show()
```

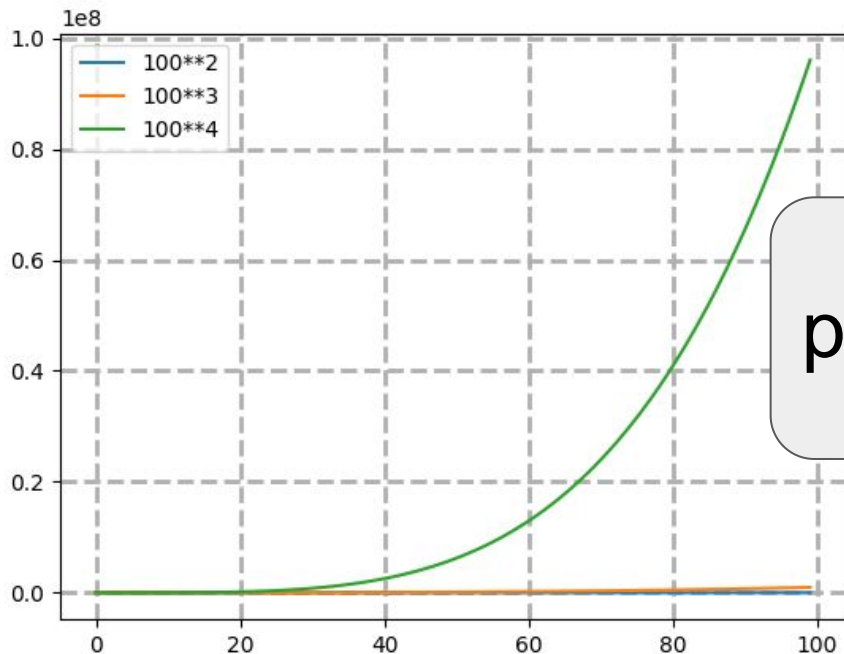
Labels

Uma maneira simples de explicar o que cada coisa do gráfico significa



```
plt.plot(x, y, label='bananas')
```

Grids

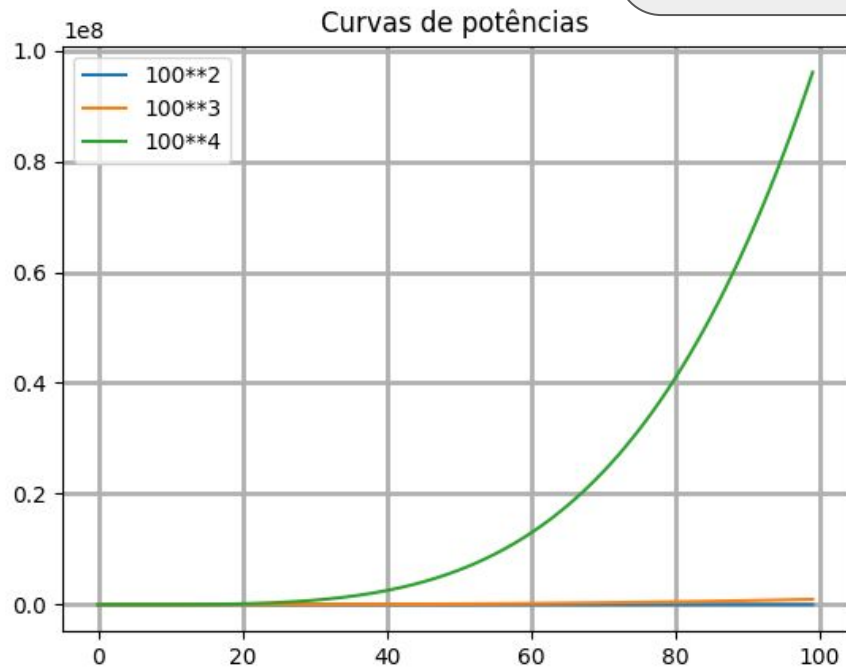


Grids (grades) ajudam a visualizar as grandezas do gráfico

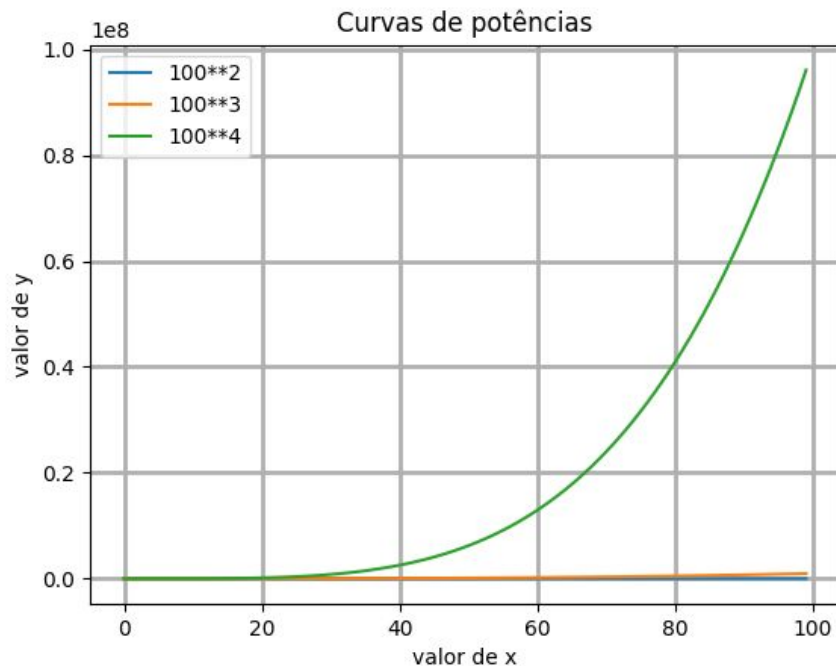
```
plt.grid(True, lw=2, ls='--')
```

Título

```
plt.title('Curvas de potência')
```



xlabel, ylabel



São as explicações do que o gráfico está exibindo em x e y (como medidas):

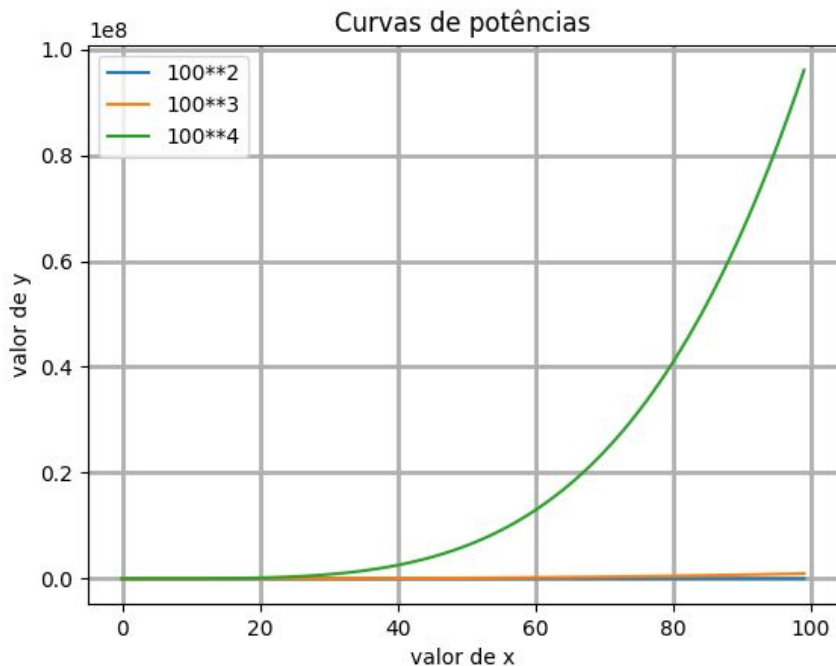
Tempo x Tamanho

Altura x Largura

Custo x Tempo

...

xlabel, ylabel



São as explicações do que o gráfico está exibindo em x e y (como medidas):

Tempo x Tamanho

```
plt.xlabel('valor de x')  
plt.ylabel('valor de y')
```

...

Gráficos de linha

São necessários dois iteráveis (listas, tuplas, conjuntos, ...) um para que seja feita a escala do eixo x, outra do eixo y.

```
x = (1, 2, 3, 4)  
y = (1, 2, 3, 4)  
plt.plot(x, y)
```

**Mas, vamos
falar só dessa
linha agora**

Gráficos de linha

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

Formato	O que faz?
-	Linha contínua
--	Linha tracejada
-.	Linha tracejada com pontos
:	Linha pontilhada
.	Exibe somente os pontos
o	círculos
^	Triangulos
*	Estrela
...	COLA

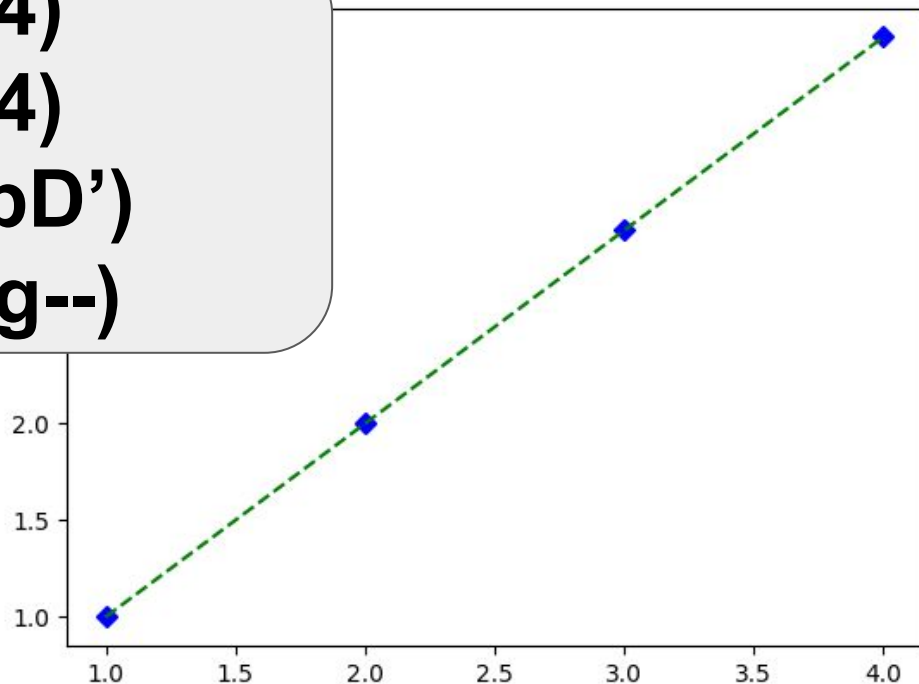
Gráficos de linha

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

Formato	O que faz?
b	Azul
g	verde
r	vermelho
c	ciano
m	magenta
y	amarelo
k	preto
w	branco
...	COLA

Gráficos de linha (linha_02.py)

```
x = (1, 2, 3, 4)  
y = (1, 2, 3, 4)  
plt.plot(x, y, 'bD')  
plt.plot(x, y, 'g--')
```



Gráficos de barra

São necessários dois iteráveis (listas, tuplas, conjuntos, ...) um para que seja feita a escala do eixo x, outra do eixo y.

```
x = (1, 2, 3, 4)  
y = ('um', 'dois', 'três', 'quatro')  
plt.plot(x, y)
```

Gráficos de pizza

```
data = [5, 25, 50, 20]
```

```
plt.pie(data)
```

Gráficos de histograma

```
from random import randint
```

```
x = [randint(1, 5000) for x in range(10)]  
plt.hist(x)
```

Lendo um arquivo e entendendo os dados

```
texto = ''  
with open(<file>) as variável:  
    texto = variável.read()  
#texto é uma string
```


Exercício #1

Monte um gráfico de linhas com as temperaturas médias do ano de 1999.

Exercício #2

Faça um histograma das temperaturas máximas de todos os anos.

Exercício #3

Exiba um gráfico de linhas de todas as temperaturas, utilizando cores e símbolos diferentes para cada temperatura.

Exercício #4

Faça um gráfico de barras com nomes de animais e dizendo o quanto você gosta deles de 0 a 100

Roteiro [2]

- Subplots
- Manipulando figuras
- Gráficos de 3 dimensões

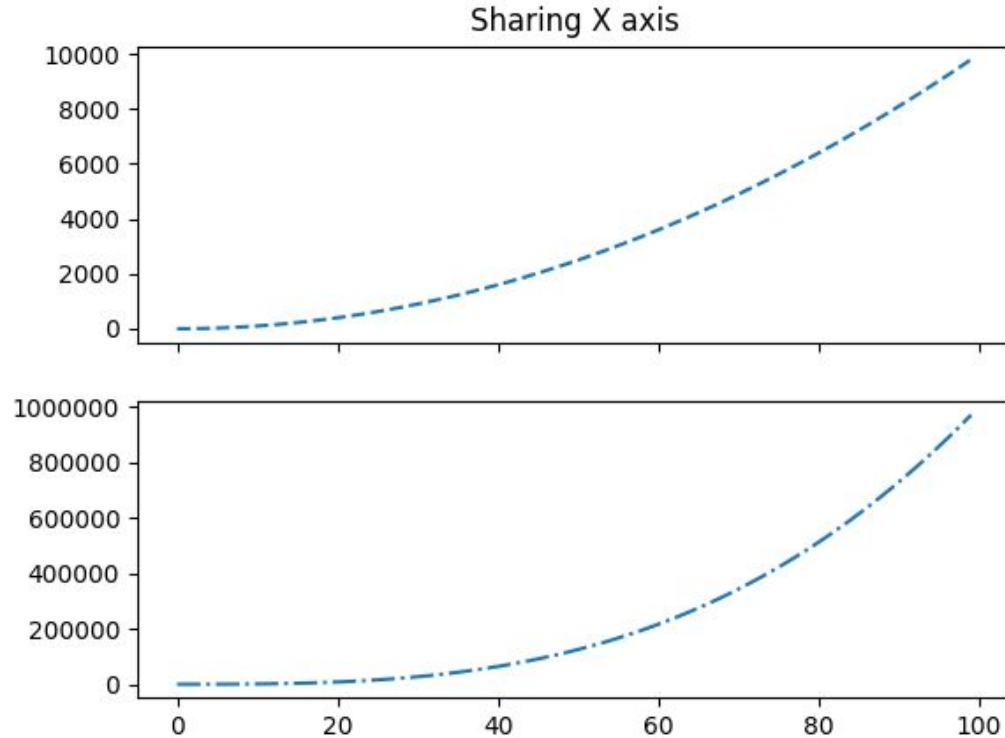
subplots

```
plt.subplot(<n de linhas>,  
            <n de colunas>,  
            share<x ou y>=<bool>)
```

subplots (compartilhando eixo x)

```
f, axarr = plt.subplots(2, sharex=True)
axarr[0].plot(x, y0, '--')
axarr[0].set_title('Sharing X axis')
axarr[1].plot(x, y1, '-.')
```

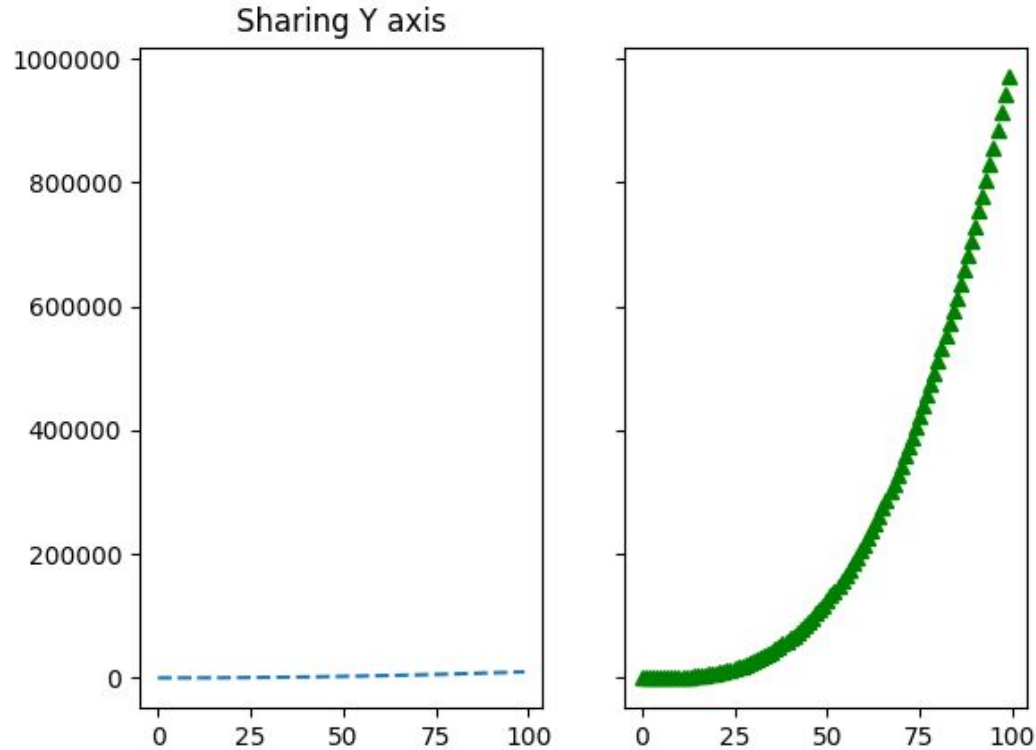
subplots (compartilhando x) [subplot_00.py]



subplots (compartilhando eixo y) [subplot_01.py]

```
f, (ax1, ax2) = plt.subplots(1, 2, sharey=True)
ax1.plot(x, y0, '--')
ax1.set_title('Sharing Y axis')
ax2.plot(x, y1, 'g^')
```

subplots (compartilhando eixo y) [subplot_01.py]



Exercício #5

Monte um gráfico que contenha 4 subplots com quatro tipos de gráficos de linha com cores diferentes e 'pontos lúdicos' diferentes e com cores diferentes

plt.figure

Responsável por criar uma nova figura, lembre-se, nós não criamos uma ainda, só usamos a pronta que o matplotlib nos deixa de brinde

```
figure(num=None, figsize=None,  
       dpi=None, facecolor=None,  
       edgecolor=None, frameon=True,  
       FigureClass=<sua_figura>)
```

plt.figure

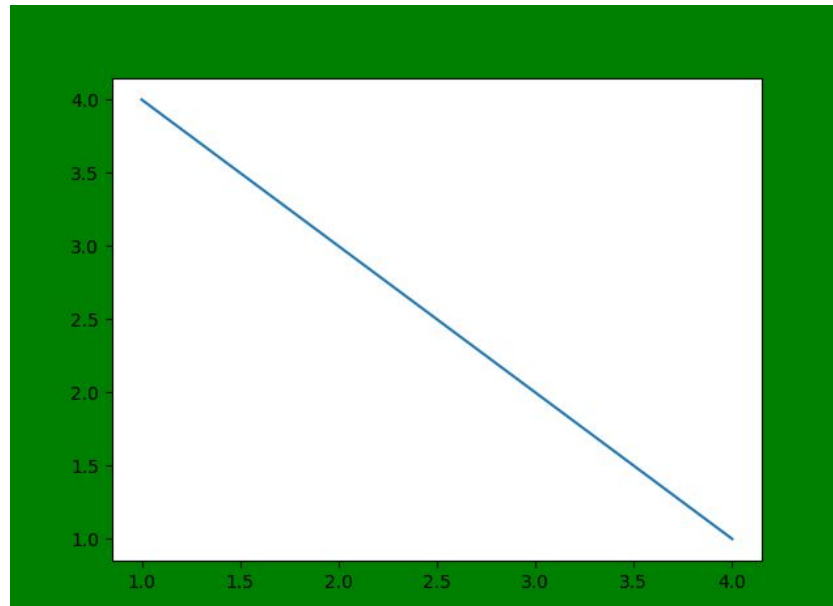
```
x = [1,2,3,4]
y = [4,3,2,1]

fig = plt.figure()

style = fig.patch

style.set_facecolor('green')

plt.plot(x, y)
```



Não deu muito certo, não é mesmo?

plt.plot -> criou uma nova figura usando as propriedades de figure

plt.figure

```
import matplotlib.pyplot as plt

fig = plt.figure()

style = fig.patch

style.set_facecolor('green')

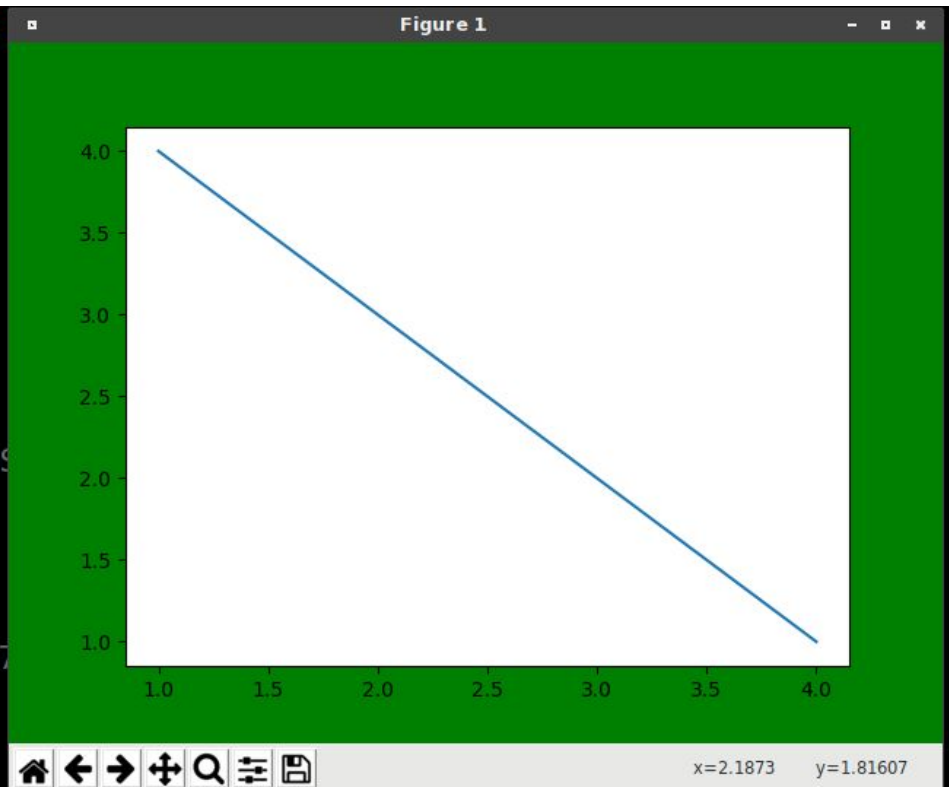
x = [1,2,3,4]; y = [4,3,2,1]

fig.add_subplot(1, 1, 1)
<matplotlib.axes._subplots.AxesS

ax1 = fig.add_subplot(1, 1, 1)

ax1.plot(x, y)
[<matplotlib.lines.Line2D at 0x7

plt.show()
```



Então, qual é a ideia do figure com subplots?

subplot(2, 1, 1)

subplot(2, 1, 2)

canvas

Então, qual é a ideia do `fig` (linhas, colunas, gráfico)

`subplot(2, 1, 1)`

`subplot(2, 1, 2)`

`canvas`

Então, qual é a ideia do figure com subplots?

subplot(2, 2, 1)

subplot(2, 2, 2)

subplot(2, 2, 3)

subplot(2, 2, 4)

canvas

Então, qual é a ideia do figure com subplots?

```
import matplotlib.pyplot as plt

fig = plt.figure()

style = fig.patch
style.set_facecolor('grey')

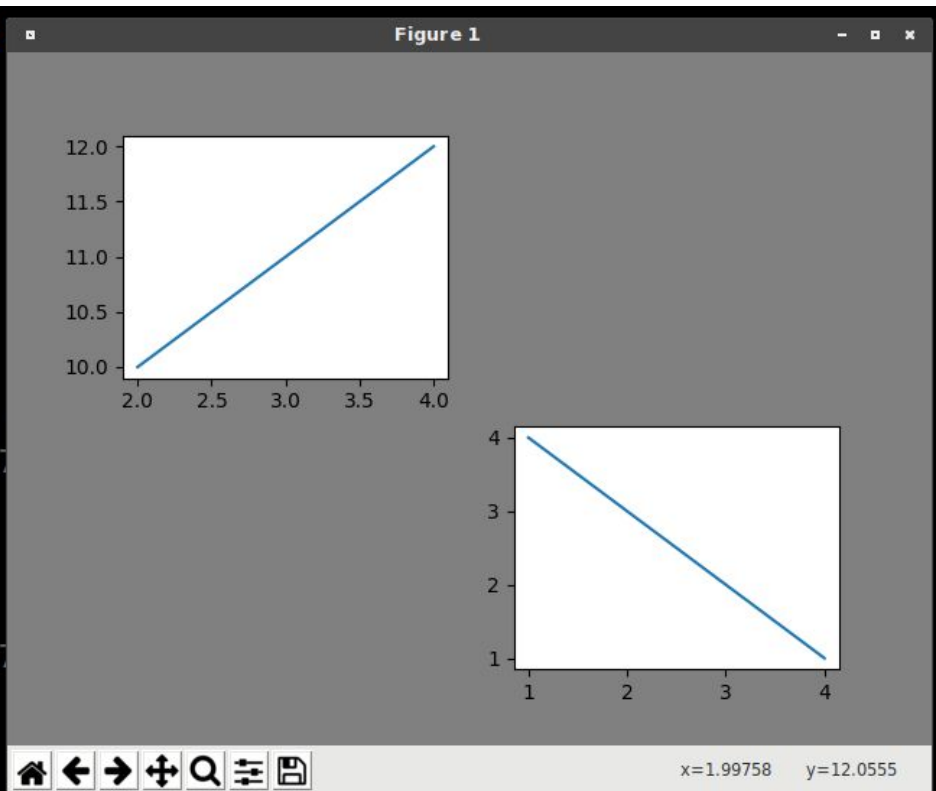
ax1 = fig.add_subplot(2, 2, 1)

ax1.plot([2,3,4], [10, 11, 12])
[<matplotlib.lines.Line2D at 0x...]

ax2 = fig.add_subplot(2, 2, 4)

ax2.plot([4,3,2,1], [1,2,3,4])
[<matplotlib.lines.Line2D at 0x...]

plt.show()
```



Exercício #6

Monte um gráfico que contenha 4 subplots com quatro tipos de gráficos de linha com cores diferentes e 'pontos lúdicos' diferentes e com cores diferentes (usando `plt.figure`) e com um fundo colorido

Gráficos 3D

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

fig = plt.figure()

fig.add_subplot(111, projection='3d')
<matplotlib.axes._subplots.Axes3DSubplot at

ax = fig.add_subplot(111, projection='3d')

x = [1, 2, 3, 4]

y = [4, 3, 2, 1]

z = [50, 60, 70, 80]

ax.plot_wireframe(x, y, z)
```