



86ª EDIÇÃO

**SEQ  
UFRJ**

26 a 30 de agosto



# Introdução à programação para ciência e engenharia em *Python*

*Iuri Segtovich*

RESUMO

# Ferramentas

- python tutor
- Anaconda
  - Navigator
  - Conda
  - Qtconsole ipython
  - Spyder
  - Jupyter-lab
- Google colab
- Stack overflow
- Python
- Python científico
  - Ipython
  - Matplotlib
  - Numpy
  - Scipy
  - sympy
  - pandas

# Python tutor

The screenshot displays the Python Tutor web interface. At the top, a browser tab is labeled "Live Programming" with a Python logo. The address bar shows "pythontutor.com/live.html#mode=edit" with a "Not secure" warning. Below the browser, the code editor is set to "Python 3.6" and contains two lines of code: `x=1` and `l=[1,2,3]`. A green arrow points to the second line, indicating it has just executed. Below the code editor, a legend shows a green arrow for "line that has just executed" and a red arrow for "next line to execute". A progress bar is also present. At the bottom, navigation buttons include "<< First", "< Back", "Done running (2 steps)", "Forward >", and "Last >>".

Write code in **Python 3.6** (drag lower right corner to resize code editor)

```
1 x=1
2 l=[1,2,3]
```

→ line that has just executed  
→ next line to execute

Frames

Global frame	
x	1
l	→

Objects

list		
0	1	2
1	2	3

The diagram illustrates the memory state. The "Global frame" contains a variable `x` with value `1` and a variable `l` that points to a list object. The list object is shown as a table with indices `0`, `1`, and `2` containing values `1`, `2`, and `3` respectively. An arrow points from the `l` variable in the frame to the list object.

# python tutor

- Ferramenta online para treinar lógica de programação e sintaxe do python.
- Possibilita visualizar a tabela de nomes e objetos, e a execução do código passo-a-passo.
- Interrompe o código automaticamente ao se deparar com um bloco iterativo infinito
- Não permite usar as bibliotecas do python científico

<http://pythontutor.com/live.html>

# anaconda

Downloads - Anaconda x

← → ↻ 🏠  Secure | <https://www.anaconda.com/download/#linux>



[What is Anaconda?](#) [Products](#) [Support](#)

## Download Anaconda Distribution

Version 5.2 | Release Date: May 30, 2018

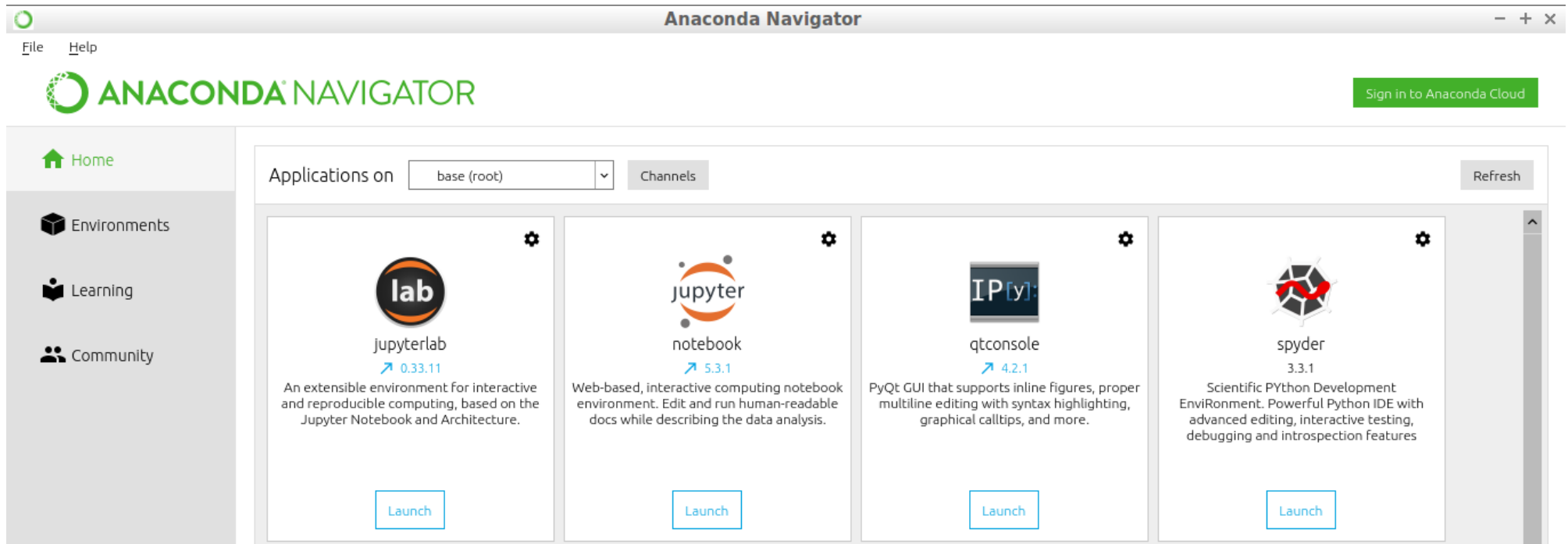
Download For:   

# anaconda

- Distribuição das ferramentas
  - Conda
  - Qtconsole ipython
  - Spyder
  - Jupyter-lab
- Gratuita
- Multiplataforma (windows, linux, mac)
- Rápida

**<https://www.anaconda.com/download/>**

# navigator



# navigator

- Menu de navegação dos programas que vem na distribuição anaconda
- Apresenta botões para criar ambientes virtuais do conda e atualizar pacotes sem precisar recorrer ao *anaconda prompt* (ver próximo slide)

[https://en.wikipedia.org/wiki/  
Anaconda\\_\(Python\\_distribution\)  
#Anaconda\\_Navigator](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#Anaconda_Navigator)



# conda

Anaconda Prompt

```
(base) C:\Users\Ashish>conda create --name MyEnv
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.4.10
  latest version: 4.4.11

Please update conda by running

  $ conda update -n base conda

## Package Plan ##

  environment location: C:\Users\Ashish\AppData\Local\conda\conda\envs\MyEnv

Proceed ([y]/n)? y

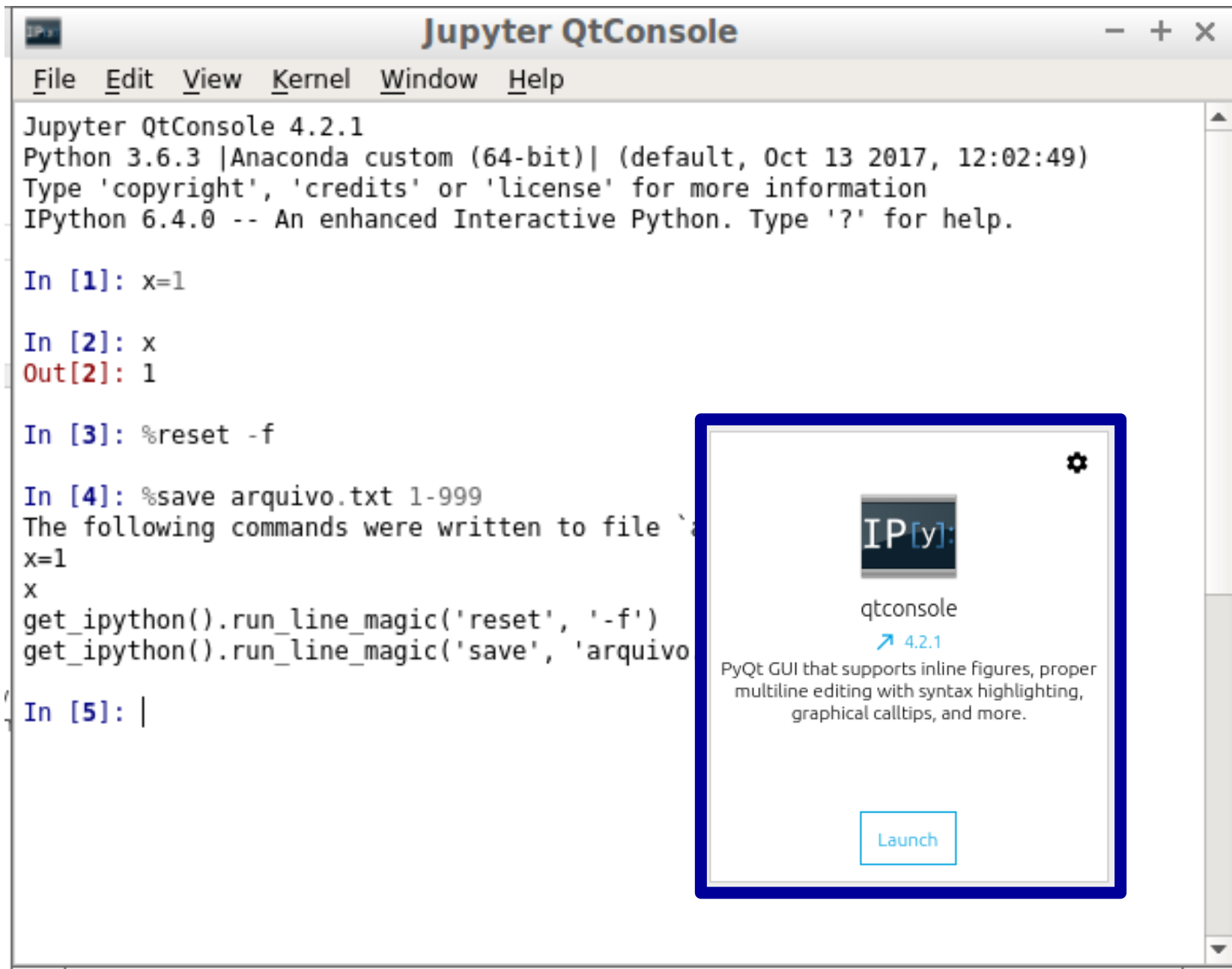
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate MyEnv
#
```

# conda

- Ferramenta para instalação de pacotes adicionais e obter atualizações.
  - (a ferramenta conda não foi apresentada no curso, fizemos uso apenas de pacotes que vêm na distribuição anaconda por padrão)
- Permite criar *ambientes virtuais*, que são uma forma de instalar duas ou mais versões do python e das bibliotecas independentemente.
  - Importante para desenvolver produtos que devam rodar em servidores de clientes que possuam configurações diferentes.
  - Importante para instalar pacotes para projetos pessoais que dependam de versões diferentes de outros pacotes (conflito de dependências)

<https://conda.io/docs/user-guide/getting-started.html>

# Qtconsole



# QtConsole

- Console leve para rodar cálculos usando a linguagem python e seus pacotes.
- Usa a linguagem python e os recursos extra do ipython
- Pode ser usado como uma calculadora simples
  - Ou científica (`import math` ou `from math import *`)
  - Ou gráfica (`import numpy as np; from matplotlib import pyplot as plt; import sympy as sym`)
- Para reiniciar a memória usar (`%reset -f`)
- Salva o histórico de cálculos com (`%save arquivo.txt 1-999`)

[https://ipython.org/  
ipython-doc/3/interactive/tutorial.html](https://ipython.org/ipython-doc/3/interactive/tutorial.html)

# spyder

The image shows the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.6)". The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The toolbar contains various icons for file operations, running, and debugging. The editor area shows a file named "temp.py" with the following content:

```
1 -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 Este é um arquivo de script temporário.
6 """
7
8
```

In the center of the editor, there is a large blue-bordered box containing the Spyder logo (a hexagon with a red squiggle), the text "spyder 3.3.1", and a description: "Scientific PYTHON Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features". At the bottom of this box is a blue "Launch" button.

On the right side, there is a "Usage" panel with the following text:

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

Below the "Usage" panel, there are tabs for "Variable explorer", "File explorer", and "Help". The "IPython console" tab is active, showing the following text:

```
Python 3.6.3 [Anaconda custom (64-bit)] (default, Oct 13 2017, 12:02:49)
Type "copyright", "credits" or "license" for more information.

IPython 6.4.0 -- An enhanced Interactive Python.

In [1]:
```

At the bottom of the window, there is a status bar with the following information: Permissions: RW, End-of-lines: LF, Encoding: UTF-8, Line: 1, Column: 1, Memory: 64%.

# Spyder

- Interface de desenvolvimento integrada IDE
- Permite gerenciar vários arquivos de código fonte (módulos e programa principal) e arquivos de texto de dados de entrada ou resultado
- Vem com um console de ipython para executar o seu código fonte e exibir os print
- Avisa quando há algum erro de sintaxe e sugere correções antes mesmo de tentar executar o código
- Permite executar códigos de projeto de forma passo a passo através do modo Debug

<https://pythonhosted.org/spyder/overview.html>

JupyterLab

localhost:8888/lab

FileEditViewRunKernelTabsSettingsHelp

Files

Untitled.ipynb9 minutes ago

Untitled.ipynb


Code

# Relatório de teste do jupyter-lab

```
In [1]: from matplotlib import pyplot as plt
%matplotlib inline
import numpy as np
```

- Desenho da função  $\sin(\frac{x}{2})$   
na faixa de 0 a  $\pi$

```
In [2]: x=np.linspace(0,np.pi,100)
plt.plot(x,np.sin(x));
```



jupyterlab

0.33.11

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch

# Jupyter-lab

- Organiza o fluxo de trabalho em forma de células que rodam código python no kernel de ipython, em arquivos jupyter (notebook.ipynb)
  - Permite dar import em módulos de projeto desenvolvidos no spyder
  - Permite fazer gráficos inline para testar interativamente resultados do projeto
  - Permite mesclar código fonte com texto formatado para apresentar resultados do projeto em um relatório intimamente ligado ao código ou apresentar um manual interativo de uso do seu pacote.

<https://jupyterlab.readthedocs.io/en/stable/>

<https://www.cheatography.com/>

[weidadeyue/cheat-sheets/jupyter-notebook/](https://www.cheatography.com/weidadeyue/cheat-sheets/jupyter-notebook/)



# Google colab

External data: Drive x

Secure | <https://colab.research.google.com/notebooks/io.ipynb#scrollTo=7Z2jcRKwUHqV>

External data: Drive, Sheets, and Cloud Storage

File Edit View Insert Runtime Tools Help

+ CODE + TEXT ↑ CELL ↓ CELL COPY TO DRIVE

Table of contents Code snippets Files X

Local file system

- Uploading files from your local file system
- Downloading files to your local file system

Google Drive

- PyDrive
- Drive REST API
- Creating a new Drive file with data from Python
- Downloading data from a Drive file into Python

Google Sheets

This notebook provides recipes for loading and saving data from external sources.

## Local file system

### Uploading files from your local file system

files.upload returns a dictionary of the files which were uploaded. The dictionary is keyed by the file name, the uploaded.

```
[ ] from google.colab import files
    uploaded = files.upload()
    for fn in uploaded.keys():
        print('User uploaded file "{name}" with length {length} bytes'.format(
            name=fn, length=len(uploaded[fn])))
```

# Google colab

- Permite rodar código python nos servidores do google
  - Os notebooks são salvos no seu google drive
  - A instalação deles vem com acesso às bibliotecas do python científico
  - São necessário alguns comandos especiais para levar módulos e arquivos de dados seus do seu google drive para o servidor de execução deles


<https://colab.research.google.com/>

<https://colab.research.google.com/notebooks/io.ipynb>

# Stack overflow

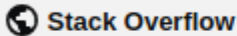
Posts containing 'p' x

Secure | https://stackoverflow.com/search?q=python+error

python error

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

Teams

Q&A for work

Learn More

## Search Results

Advanced Search Tips

Ask Question

results found containing python error

python error

Search

185,649 results

RelevanceNewestMore ▾

2109

votes

34

answers

### Q: Making a flat list out of list of lists in Python

I wonder whether there is a shortcut to make a simple list out of list of lists in **Python**. I can do that in a for loop, but maybe there is some cool "one-liner"? I tried it with reduce, but I get an ... **error**. Code I = [[1, 2, 3], [4, 5, 6], [7], [8, 9]] reduce(lambda x, y: x.extend(y), I) **Error** message Traceback (most recent call last): File "<stdin>", line 1, in <module> File "<stdin>", line 1, in <lambda> AttributeError: 'NoneType' object has no attribute 'extend' ...

pythonlistmultidimensional-arrayflatten

asked Jun 4 '09 by Emma

5792

votes

### A: What are metaclasses in Python?

Classes as objects Before understanding metaclasses, you need to master classes in **Python**. And **Python** has a very peculiar idea of what classes are, borrowed from the Smalltalk language. In most ... languages, classes are just pieces of code that describe how to produce an object. That's kinda true in **Python** too: >>> class ObjectCreator(object): ... pass ... >>> my\_object = ObjectCreator ...

answered Jul 5 '11 by e-satis

# Stack overflow

- Base de conhecimento com perguntas e respostas sobre lógica, sintaxe e uso de diversas ferramentas e linguagens de programação.
- Ferramenta de busca

<https://stackoverflow.com/search>

python



# python

- A linguagem que permite expressas, seguindo suas regras de sintaxe, a lógica do programa que desejamos criar para que o interpretador possa traduzir para o nosso hardware.
  - Bindings: nomes e objetos
  - Condicionais
  - Iterações
  - Funções
- Utilizamos a linguagem python em todas as ferramentas apresentadas
- Atualmente, utilizamos a versão 3.x da linguagem.
- Permite usar funções de módulos desenvolvidos em python mesmo ou em outras linguagens por meio de wrappers.

<https://www.python.org/>

# Scientific python



Install



Getting Started



Documentation

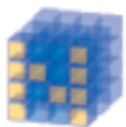


Report Bugs



Blogs

SciPy (pronounced "Sigh Pie") is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:



NumPy

Base N-dimensional  
array package



SciPy library

Fundamental library  
for scientific  
computing



Matplotlib

Comprehensive 2D  
Plotting

IP[y]:  
IPython

IPython

Enhanced Interactive  
Console



Sympy

Symbolic mathematics



pandas

Data structures &  
analysis

# Scientific python

- Ecosistema de bibliotecas
  - Ipython kernel
  - Numpy
  - Matplotlib
  - Pacote Scipy
  - Sympy
  - pandas

<https://www.scipy-lectures.org/>



ipython

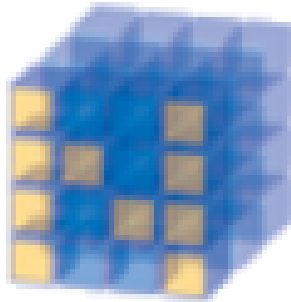
**IP**[y]: IPython  
Interactive Computing

- Kernel que executa o seu código python no modo interativo
  - Usado indiretamente ao abrir o qtconsole, spyder ou jupyterlab
  -

<https://en.wikipedia.org/wiki/IPython>

<https://ipython.readthedocs.io/en/stable/whatsnew/version6.html>

# Numpy



NumPy

Base N-dimensional  
array package

```
In [1]: import numpy as np
a=np.array([[11,12],
            [21,22]])
a
```

```
Out[1]: array([[11, 12],
               [21, 22]])
```

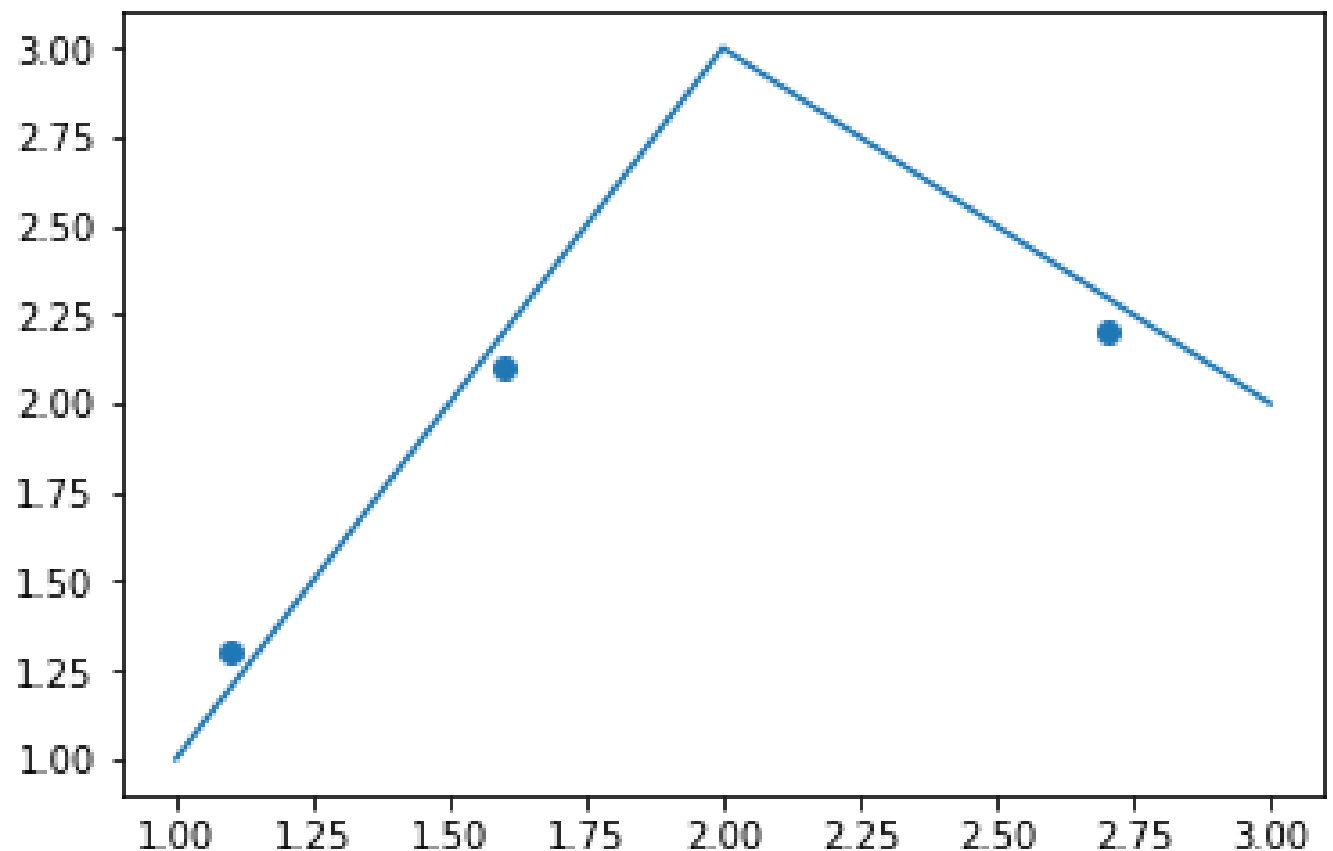
# numpy

- Pacote do python científico que dá suporte a arrays multidimensionais.
  - Permite operações elemento-a-elemento e calculos matriciais.
  - Cálculos com arrays numéricos no numpy são executados muito mais rápido que com listas nativas do python.

[https://docs.scipy.org/  
doc/numpy/user/quickstart.html](https://docs.scipy.org/doc/numpy/user/quickstart.html)

# Matplotlib / pyplot

```
In [1]: from matplotlib import pyplot as plt
%matplotlib inline
plt.scatter( [1.1,1.6,2.7], [1.3,2.1,2.2] )
plt.plot(    [1.0,2.0,3.0], [1.0,3.0,2.0] );
```



Matplotlib  
Comprehensive 2D  
Plotting

# Matplotlib / pyplot

- Matplotlib
  - Pacote para fazer gráficos
    - pontos, linhas, histogramas, etc...
- Pyplot
  - Modulo através do qual acessamos as funções do matplotlib
  - Usa comandos e argumentos similares aos do matlab

`%matplotlib inline`

- Comando para desenhar os gráficos na seção de output da própria célula do jupyter-lab.

[https://matplotlib.org/tutorials/introductory/sample\\_plots.html](https://matplotlib.org/tutorials/introductory/sample_plots.html)

# Scipy package



SciPy library

Fundamental library  
for scientific  
computing

```
In [1]: from scipy.optimize import fsolve
        from scipy.optimize import fmin
        from scipy.optimize import approx_fprime
        from scipy.integrate import quad
        from scipy.integrate import odeint
        from scipy.interpolate import interp1d
```

# scipy

- Pacote com métodos numéricos prontos para aplicar em problemas de ciência e engenharia
  - Raízes de polinômio
  - Solução de sistema de equações não linear
  - Otimização de função objetivo
  - Avaliação de integral definida
  - Solução de equação diferencial ordinária
  - Ajuste de curva
  - Interpolação de dados
  - estatísticas

[https://www.scipy-lectures.org/  
intro/scipy.html](https://www.scipy-lectures.org/intro/scipy.html)



# Sympy



Sympy

Symbolic mathematics

```
In [1]: import sympy as sym
sym.init_printing()
x = sym.Symbol('x')
f = sym.Function('f')
ODE=sym.Eq(sym.Derivative(f(x),x,x) + 9*f(x), 1)
ODE
```

Out[1]:

$$9f(x) + \frac{d^2}{dx^2}f(x) = 1$$

```
In [2]: sym.dsolve(ODE,f(x))
```

Out[2]:

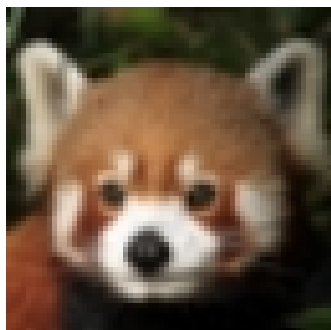
$$f(x) = C_1 \sin(3x) + C_2 \cos(3x) + \frac{1}{9}$$

# sympy

- Pacote para calculo simbolica (Computer algebra system)
  - Permite fazer derivada e integral simbolica
  - Permite encontrar solução analitica para equações
  - Permite gerar implementações numericas das expressões desenvolvidas

<https://github.com/sympy/sympy/wiki/Quick-examples>

# Pandas



pandas

Data structures &  
analysis

```
import pandas as pd
%matplotlib inline
s = pd.Series([1,3,5,6,8,9,11])
s
```

```
0    1
1    3
2    5
3    6
4    8
5    9
6   11
dtype: int64
```

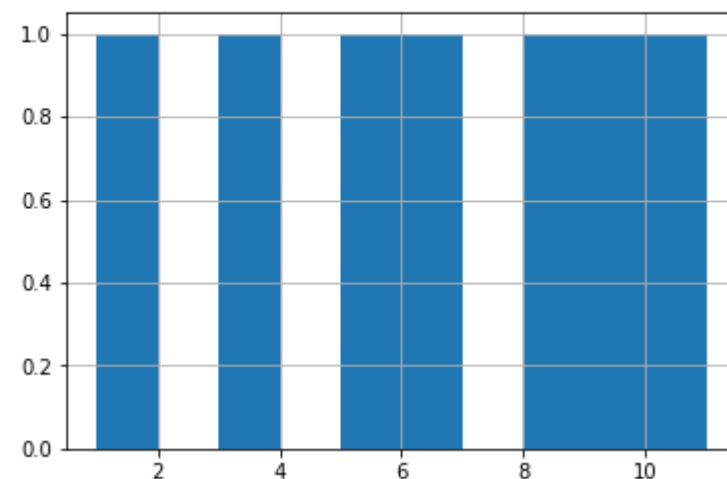
```
df=pd.DataFrame(s)
df
```

```
   0
0  1
1  3
2  5
3  6
4  8
5  9
6 11
```

```
df.describe()
```

	0
count	7.000000
mean	6.142857
std	3.484660
min	1.000000
25%	4.000000
50%	6.000000
75%	8.500000
max	11.000000

```
df[0].hist();
```



- Possui funções para tratar séries de dados
  - Leitura de arquivos de tabelas de dados
  - Organização em data frames
  - Descrição estatística
  - histogramas

[https://pandas.pydata.org/  
pandas-docs/stable/10min.html](https://pandas.pydata.org/pandas-docs/stable/10min.html)