

Para Começar a Programar: Curso Básico de Python

Semana Universitária UnB 2022

Professor: Adriano Possebon Rosa

Departamento de Engenharia Mecânica
Faculdade de Tecnologia
Universidade de Brasília

- 1 Introdução
- 2 O que é Programar?
- 3 Linguagens de Programação
- 4 Python?
- 5 O que vamos ver nesse curso?
- 6 Pra quem é esse curso?
- 7 GitHub
- 8 Grupo no WhatsApp
- 9 Instalando o Python



- 1 Introdução
- 2 O que é Programar?
- 3 Linguagens de Programação
- 4 Python?
- 5 O que vamos ver nesse curso?
- 6 Pra quem é esse curso?
- 7 GitHub
- 8 Grupo no WhatsApp
- 9 Instalando o Python





SU22

Semana Universitária UnB

29 ago - 2 set

100 anos de Darcy Ribeiro



ft_extensao [Edit Profile](#) 

42 posts 399 followers 56 following

FT Extensão
Divulgação de ações de extensão realizadas por alunos, técnicos e professores da Faculdade de Tecnologia da UnB
sig.unb.br/sigaa/public/extendensao/paginaListaPeriodosInscricoesAtividadesPublico...



O professor, quem é?

Formação:

- **Graduação** em Engenharia Mecânica (UnB, 2012)
- **Mestrado** em Ciências Mecânicas (UnB, 2014)
- **Doutorado** em Ciências Mecânicas (UnB, 2018)

Professor na **UnB** desde 2016.

Departamento de **Engenharia Mecânica**.

Atualmente ministro as **disciplinas:**

- Transporte de Calor e Massa
- Métodos Numéricos em Termofluidos



Trabalho com **programação numérica** desde 2010 (+-).

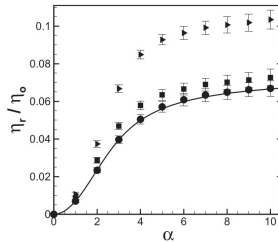
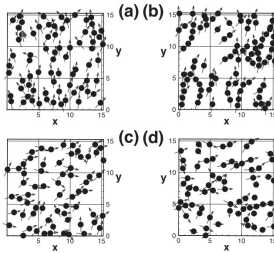
Programação Numérica: uso do computador para resolver equações complicadas.

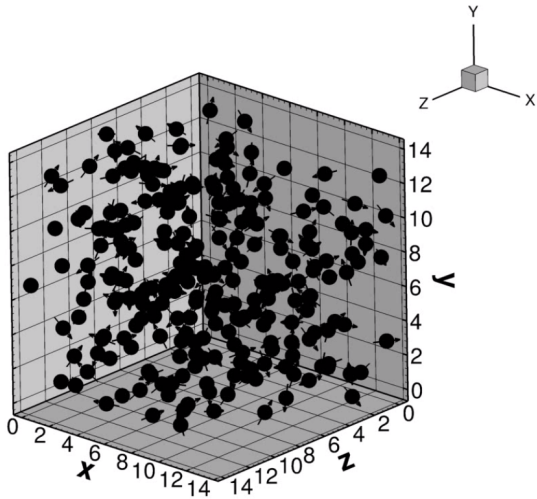
Minha área de **pesquisa**: **Mecânica dos Fluidos**.

O **computador** é usado para resolver equações que governam o movimento do fluido.

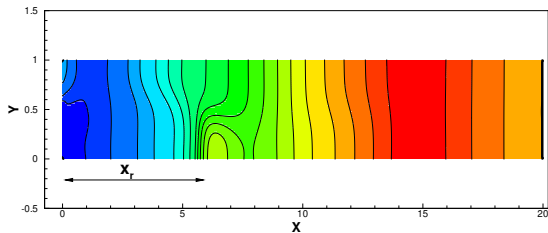
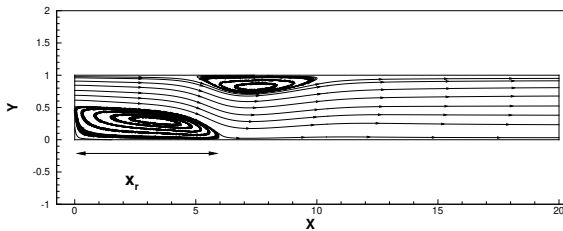


Simulação de partículas em um fluido magnético.





Simulação do comportamento de um fluido.



Comecei a programar com o **MATLAB**.

Depois passei muitos anos com o bom e velho **Fortran**.
(Ainda convivo muito com ele, e acho que vou conviver pra sempre.)

Tive uma rápida temporada com o **C++** em 2016.

Até que conheci o **Python**, uns 4 anos atrás.
(E foi amor à primeira vista.)

Python é uma linguagem extraordinária.
Para mim, para o que eu preciso fazer, é a melhor linguagem.
(E ainda tenho muito para aprender.)

Espero que você goste do **Python** também.



Os **alunos**, quem são?

Quero saber:

- **curso/instituição**
- **time de futebol do coração**
- **experiência com programação** (zero, pouca, média, avançada)
- e o **motivo de você estar nesse curso**



- 1 Introdução
- 2 O que é Programar?
- 3 Linguagens de Programação
- 4 Python?
- 5 O que vamos ver nesse curso?
- 6 Pra quem é esse curso?
- 7 GitHub
- 8 Grupo no WhatsApp
- 9 Instalando o Python



Textão do **Wikipédia**:

Programação é o processo de escrita, teste e manutenção de um programa de computador. O programa é escrito em uma linguagem de programação, embora seja possível, com alguma dificuldade, o escrever diretamente em linguagem de máquina. Diferentes partes de um programa podem ser escritas em **diferentes linguagens**.

Diferentes **linguagens** de programação funcionam de diferentes modos. Por esse motivo, os **programadores** podem criar programas muito diferentes para diferentes linguagens; muito embora, teoricamente, a maioria das linguagens possa ser usada para criar **qualquer programa**.

Há várias décadas se debate se a programação é mais semelhante a uma **arte**, a uma **ciência**, à **matemática**, à **engenharia**, ou se é um campo completamente novo.



Programa: sequência de instruções que dizem ao computador o que fazer.

Linguagem: traduz o que está escrito no programa em linguagem de máquina (zeros e uns).

```
37 double delta_x = 1.0/N;  
38 double delta_y = delta_x;  
39 double delta_t = 0.25*Re*delta_x*delta_x;  
40  
41 //Restriction For delta_t  
42 if (delta_t > 0.5*delta_x)  
43 { delta_t = 0.25*delta_x;}  
44  
45 //X-velocity Array  
46 double u[N+2][N+2], u_s[N+2][N+2];  
47  
48 for (int i=0; i<N+2; i++)  
49 {  
50     for (int j=0; j<N+2; j++)  
51     {  
52         if (j==N+1)  
53         {  
54             u[i][j] = 2.0;  
55             u_s[i][j] = 0.0;  
56         }  
57         else
```

```
207 def calculate_v_star(v_star, u, v, p, Re, N_x, N_y, dx, dy, dt):  
208     """  
209     Calcula v_star seguindo o algoritmo do método de  
210     projeção de primeira ordem. O cálculo de v_star é  
211     explícito.  
212     """  
213  
214     # Pontos do interior do domínio.  
215     # a1, a2, a3 e a4 são variáveis auxiliares.  
216     for i in range(0, N_x):  
217         for j in range(1, N_y):  
218             a1 = 0.25*(u[i+1, j] + u[i, j] + u[i+1, j-1] + u[i, j-1])  
219             a2 = (a1*(v[i+1, j] - v[i-1, j]))/(2.0*dx) +  
220                 v[i, j]*(v[i, j+1] - v[i, j-1])/(2.0*dy)  
221             a3 = (v[i+1, j] - 2.0*v[i, j] + v[i-1, j])/(Re*dx*dx)  
222             a4 = (v[i, j+1] - 2.0*v[i, j] + v[i, j-1])/(Re*dy*dy)  
223             v_star[i, j] = v[i, j] + dt*(-a2 + a3 + a4)
```



- 1 Introdução
- 2 O que é Programar?
- 3 Linguagens de Programação**
- 4 Python?
- 5 O que vamos ver nesse curso?
- 6 Pra quem é esse curso?
- 7 GitHub
- 8 Grupo no WhatsApp
- 9 Instalando o Python



Linguagens mais famosas.

Front-end:

- HTML, CSS, JavaScript

Back-end:

- PHP, C, C++, C#, Ruby, Java, Perl, Python

Desenvolvimento de Apps:

- Swift, Java, Objective-C, C++, C#, JavaScript, Python

Programação Científica:

- C, C++, Fortran, Matlab, Julia, Octave, R, Python

Temos mais de **1000** linguagens de programação!

No começo, foco em **uma linguagem!!**



- 1 Introdução
- 2 O que é Programar?
- 3 Linguagens de Programação
- 4 Python?**
- 5 O que vamos ver nesse curso?
- 6 Pra quem é esse curso?
- 7 GitHub
- 8 Grupo no WhatsApp
- 9 Instalando o Python





- Criada por **Guido van Rossum** em 1991
- Nome vem de *Monty Python*, grupo de comédia britânico
- Gratuita
- Open source
- Interpretada
- GPL: *General Purpose Language*



- Fácil de aprender (boa para **iniciantes**)
- Códigos mais curtos
- Muitas **bibliotecas** e **frameworks** disponíveis
- Utilizada por grandes empresas (Google, Instagram, Amazon, Spotify, Youtube)
- Utilizada no meio científico/acadêmico
- Conversa muito bem com outras linguagens



O que dá pra fazer com Python?

- Programas (scripts, códigos, roteiros) para automatizar tarefas do dia a dia (mover arquivos, criar arquivos, ler e-mails, ...)
- Desenvolvimento para desktop; aplicativos de computador; programas com interface para o usuário (usando PyQt, Tkinter, PySimpleGUI)
- Análise de Dados
- Inteligência Artificial (algoritmo que aprende com os dados de entrada)
- Desenvolvimento de jogos
- Desenvolvimento web; desenvolvimento de sites
- Resolução de problemas científicos



Algumas desvantagens:

- Lenta (mas dá pra otimizar em muitos casos)
- Consome mais memória do que algumas outras linguagens



Escrevendo "Hello World!" na tela, com C++, Fortran e Python.

```
1#include <iostream>
2
3using namespace std;
4
5int main()
6{
7    cout << "Hello World!" << endl;
8
9    return 0;
10}
```

```
1program hello_world
2
3write(*,*) "Hello World!"
4
5end program
```

```
1print("Hello World!")
```



- 1 Introdução
- 2 O que é Programar?
- 3 Linguagens de Programação
- 4 Python?
- 5 O que vamos ver nesse curso?**
- 6 Pra quem é esse curso?
- 7 GitHub
- 8 Grupo no WhatsApp
- 9 Instalando o Python



Aqui vamos ver o básico de **Python**, o que é comum a todos os programadores.

Teremos **10 horas de curso**, em **5 dias**.

Vamos tentar entender como a linguagem funciona.

No fim deste curso você será capaz de usar o Python para desenvolver um **projeto**, para resolver um **problema** e/ou para criar uma **aplicação**.

Vamos com calma, um passo de cada vez.

No final de cada aula serão apresentados alguns **exercícios**.

É **muito importante** que você os faça.

Só se aprende a programar programando.



As 5 aulas estão divididas da seguinte forma:

- **Aula 1.** Introdução. O que é programação? Por que programar? Linguagens de Programação. Python? Onde Python é utilizado? Instalando o Python. IDEs. Input e Print. Tarefa de Casa.
- **Aula 2.** Variáveis e tipos de dados no Python. Strings. Inteiros e Floats. Listas, Tuplas, Conjuntos e Dicionários. Conversão. Booleanos. Condicionais. Loops. Erros no código. Tarefa de Casa.
- **Aula 3.** Funções. Por que usar funções? Parâmetros de entrada e de saída. Parâmetros predefinidos, args e kwargs. Tarefa de Casa.



- **Aula 4.** Classes e Objetos. Métodos. Classes definidas pelo Python. Classes definidas pelo usuário. Herança. Programação orientada a objetos. Tarefa de Casa.
- **Aula 5.** Usando bibliotecas externas. Numpy. Plotando gráficos com o Matplotlib. Bibliotecas e Frameworks. Próximos passos. Projetos, Github! Tarefa de Casa.

E é claro que eu vou puxar o curso um pouco mais para a minha área. Veremos vários exemplos com equações.



Aula 1.

```
nome = input('Qual é o seu nome? ')

idade = input('Qual é a sua idade hoje? ')

ano_de_nascimento = 2022 - int(idade)

print('\n0 seu nome é', nome, '. Você tem', idade, 'anos e nasceu em',
      ano_de_nascimento, 'ou em', ano_de_nascimento - 1, '.')
```



Aula 2.

```
In [3]: my_number = 100

if my_number > 100:
    print("Meu número é maior do que 100.")
elif my_number == 100:
    print("Meu número é igual a 100.")
else:
    print("Meu número é menor do que 100.")
```

Meu número é igual a 100.



Aula 3.

```
In [171]: def calcular_media_final(notas1, notas2, notas3):  
  
    media_final = (notas1 + notas2 + notas3)/3  
  
    return media_final  
  
def calcular_mencao(nome, sobrenome, notas1 = 10, notas2 = 10, notas3 = 10):  
  
    media_final = calcular_media_final(notas1, notas2, notas3)  
  
    if media_final >= 9.0:  
        mencao = 'SS'  
    elif media_final >= 7.0:  
        mencao = 'MS'  
    elif media_final >= 5.0:  
        mencao = 'MM'  
    elif media_final >= 3.0:  
        mencao = 'MI'  
    else:  
        mencao = 'II'  
  
    return print(f'A média final de {nome} {sobrenome} foi {media_final:.2f}.',  
                f' A menção final é {mencao}.')  
  
calcular_mencao('Joseph', 'Klimber', 8.0, 8.0, 1.0)  
  
calcular_mencao('Douglas', 'Adams')  
  
calcular_mencao('Ford', 'Prefect', notas2 = 5.0, notas3 = 3.0)
```



Aula 4.

```
In [251]: class ContaCorrente:

    conta_tipo = 'Corrente'

    def __init__(self, nome, numero, saldo = 0):
        self.nome = nome
        self.numero = numero
        self.saldo = saldo
        self.printar_saldo()

    def deposito(self, valor_deposito):
        self.saldo += valor_deposito
        self.printar_saldo()

    def printar_saldo(self):
        print(f'\n0 saldo da conta {self.numero} é de R$ {self.saldo:.2f}.')

conta1 = ContaCorrente('José', 1111)
conta1.deposito(50)
conta1.deposito(150)

conta2 = ContaCorrente('João', 2224, 35)
conta2.deposito(35)
```



Aula 5.

```
In [3]: plt.rcParams['text.usetex'] = True # Esse comando deixa as fontes no padrão LaTeX.

fig, ax = plt.subplots()

x = np.linspace(0.0, 2.0, 201)

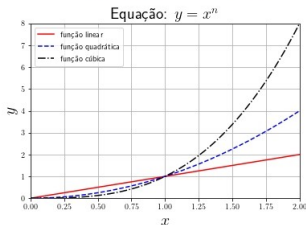
ax.plot(x, x, 'r-', label='função linear')
ax.plot(x, x**2.0, 'b--', label='função quadrática')
ax.plot(x, x**3.0, 'k-.', label='função cúbica')

ax.set_title(r"Equação:  $y = x^n$ ", fontsize=20)
ax.set_xlabel("$x$", fontsize=20)
ax.set_ylabel("$y$", fontsize=20)
ax.set_xlim(0.0, 2.0)
ax.set_ylim(0.0, 8.0)

ax.grid("on")

ax.legend()

fig.savefig("figural.pdf", format="pdf", bbox_inches='tight', pad_inches=0.2)
fig.savefig("figural.png", format="png", bbox_inches='tight', pad_inches=0.2)
```



E depois? Aí é com você! Vai depender do que você quer fazer com **Python**.

- **Computação Científica, Análise de Dados e Inteligência Artificial:** Numpy, Matplotlib, Pandas, TensorFlow, Scikit-learn, Keras, PyTorch, OpenCV.
- **Desenvolvimento Web:** Django, Flask, CherryPy, Pyramid, Dash, Falcon, web2py, Streamlit, Bottle.
- **Desenvolvimento App:** Kivy, BeeWare, python-for-android
- **Desenvolvimento de Jogos:** pygame, pykya, Arcade, PyOpenGL, Panda3D.
- **Automação Web e de Tarefas:** OS, Urllib, Requests, PyAutoGUI, Webbot, Scrapy, Pyinput, BeautifulSoup.



- 1 Introdução
- 2 O que é Programar?
- 3 Linguagens de Programação
- 4 Python?
- 5 O que vamos ver nesse curso?
- 6 Pra quem é esse curso?**
- 7 GitHub
- 8 Grupo no WhatsApp
- 9 Instalando o Python



Este curso é para quem

- nunca programou e quer ter uma primeira experiência com programação;
- já conhece um pouco de programação e quer saber um pouco mais;
- já programa usando outra linguagem e quer conhecer o famoso Python;
- usa/quer usar alguma biblioteca do Python mas não está entendendo muito bem os detalhes, e por isso precisa saber um pouco mais sobre como a linguagem funciona.



- 1 Introdução
- 2 O que é Programar?
- 3 Linguagens de Programação
- 4 Python?
- 5 O que vamos ver nesse curso?
- 6 Pra quem é esse curso?
- 7 GitHub**
- 8 Grupo no WhatsApp
- 9 Instalando o Python



GitHub é uma rede social para programadores.

Faça a sua conta hoje mesmo!!! E visite as contas de outras pessoas.

O material deste curso pode ser encontrado em

github.com/adrianopossebon





Adriano Possebon

adrianopossebon

Follow

Professor do Departamento de Engenharia Mecânica da Universidade de Brasília.

[Overview](#) [Repositories 1](#) [Projects](#) [Packages](#) [Stars 4](#)

Popular repositories

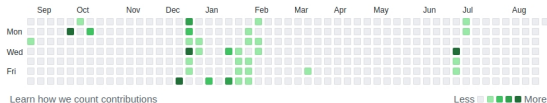
[Semuni_2021_Curso_Python](#)

Public

Material do Curso de Python para Iniciantes ministrado na Semana Universitária de 2021 da Universidade de Brasília.

 Jupyter Notebook  14  2

88 contributions in the last year



- 1 Introdução
- 2 O que é Programar?
- 3 Linguagens de Programação
- 4 Python?
- 5 O que vamos ver nesse curso?
- 6 Pra quem é esse curso?
- 7 GitHub
- 8 Grupo no WhatsApp**
- 9 Instalando o Python



Criei um **grupo no WhatsApp** para podermos interagir.

O link é [este](#).

Coloquem suas dúvidas com relação às aulas e aos exercícios.

Esse grupo é pra vocês. Eu vou responder raramente.

Tente ajudar na dúvida do colega. Essa é uma ótima forma de aprender.



- 1 Introdução
- 2 O que é Programar?
- 3 Linguagens de Programação
- 4 Python?
- 5 O que vamos ver nesse curso?
- 6 Pra quem é esse curso?
- 7 GitHub
- 8 Grupo no WhatsApp
- 9 Instalando o Python



Temos várias maneiras de instalar o Python no computador.

A mais simples delas é por meio do **Pacote Anaconda**. Esse pacote possui muitas ferramentas do Python.

Entre em www.anaconda.com

Para um passo a passo da instalação no Windows, veja esse [vídeo no youtube](#).

Eu não uso Windows, por isso não sei explicar como fazer a instalação.

A **boa notícia** é que não é necessário instalar.

É bom, mas não é necessário.

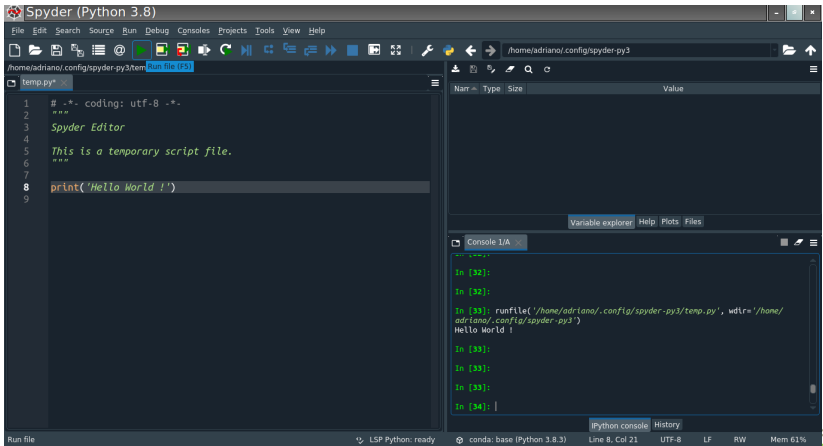


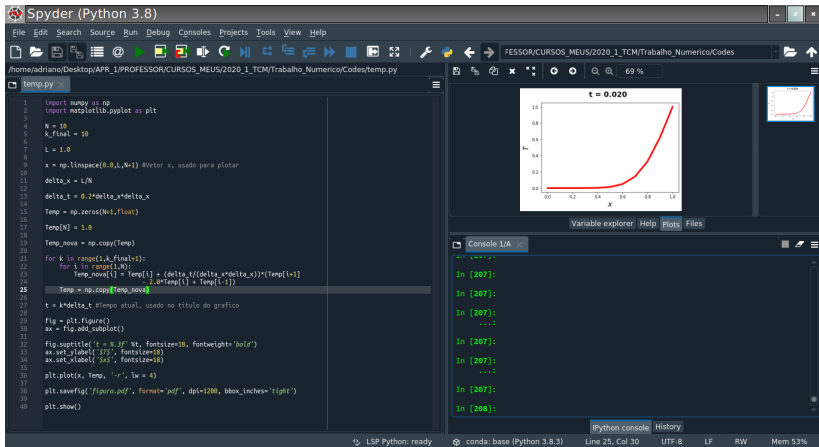
Para ajudar na hora de criar os códigos, temos as **IDEs** (**Am-
bientes de Desenvolvimento Integrado**).

Temos várias IDEs para Python. As mais famosas são:

- PyCharm
- Visual Studio Code
- Sublime Text
- Atom
- IDLE
- Spyder







Um ambiente muito utilizado para o desenvolvimento em **Python** é o **Jupyter Notebook**.

O Jupyter permite incluir texto, imagens, equações e códigos em um mesmo arquivo.

Algoritmo: Método de Newton

Entradas: função f e sua derivada f' ; aproximação inicial x_0 ; tolerância $atol$.

Saída: valor aproximado da solução

1. Comece com uma aproximação inicial x_0
2. $k \leftarrow 0$
3. $dif \leftarrow 2 \cdot atol$
4. enquanto $dif \geq atol$, faça:
 - A. $x_{k+1} \leftarrow x_k - \frac{f(x_k)}{f'(x_k)}$
 - B. $dif \leftarrow |x_{k+1} - x_k|$
 - C. $k \leftarrow k + 1$

Fim do Algoritmo

Exemplo: vamos encontrar os zeros da função $f(x) = 2 \cosh(x/4) - x$ utilizando uma implementação do algoritmo acima.

```
In [48]: def f(y):
          return 2.0*np.cosh(y/4.0) - y

          def f_linha(y):
              return 0.5*np.sinh(x/4.0) - 1.0

          x = 10.0      # 2.0, 10.0
          atol = 1.e-8
          k = 0
          dif = 2.0*atol
          while dif >= atol:
              x_new = x - f(x)/f_linha(x)
              dif = np.abs(x_new - x)
              x = x_new
              k += 1
          print('Na iteração k = %2.21, o valor de x é %20.15f'%(k,x))
```



O **Jupyter** vem instalado no **Anaconda**.

Mas é possível usar o **Jupyter** online, na nuvem, por meio do **Google Colab**.

O **Google Colab** é uma ferramenta poderosíssima que permite rodar códigos Python na nuvem (em um servidor Google), sem ter nada instalado no computador.

O **Google Colab** é gratuito (versão básica, muito boa): você só precisa ter uma conta do Google (gmail).

Os arquivos ficam salvos no seu Google Drive.

Para começar, basta entrar no Google, digitar “Google Colab”, e entrar na primeira página.





Curso_Python_Semuni_v1.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment

Share



+ Code + Text

✓ RAM
Disk

Editing

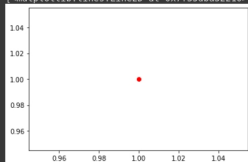
- Alternatively, if your data is already a 2d array, you can pass it directly to *x*, *y*. A separate data set will be drawn for every column.

Vamos começar com um gráfico simples.

Primeiro nós vamos criar dois objetos: uma figura (*fig*), que representa a figura como um todo, e um objeto plot (*ax*), representado pelos eixos, que é onde serão acrescentados os dados.

```
fig, ax = plt.subplots() # Criando os objetos.  
ax.plot(1.0, 1.0, 'ro') # Plotando o ponto 1.0, 1.0 como um círculo vermelho.
```

[<matplotlib.lines.Line2D at 0x7f35dba32210>]



No código acima estamos criando os objetos *fig* e *ax* e estamos adicionando os pontos que queremos plotar no objeto *ax*.

✓ 0s completed at 5:04 PM



Vou usar o **Google Colab** aqui no nosso curso.

Vou passar pra lá agora.

Alguma dúvida??? Comentário???

