

# LÓGICA DE PROGRAMAÇÃO PARA IA COM PYTHON

TECNOLOGIA DA INFORMAÇÃO E COMPUTAÇÃO

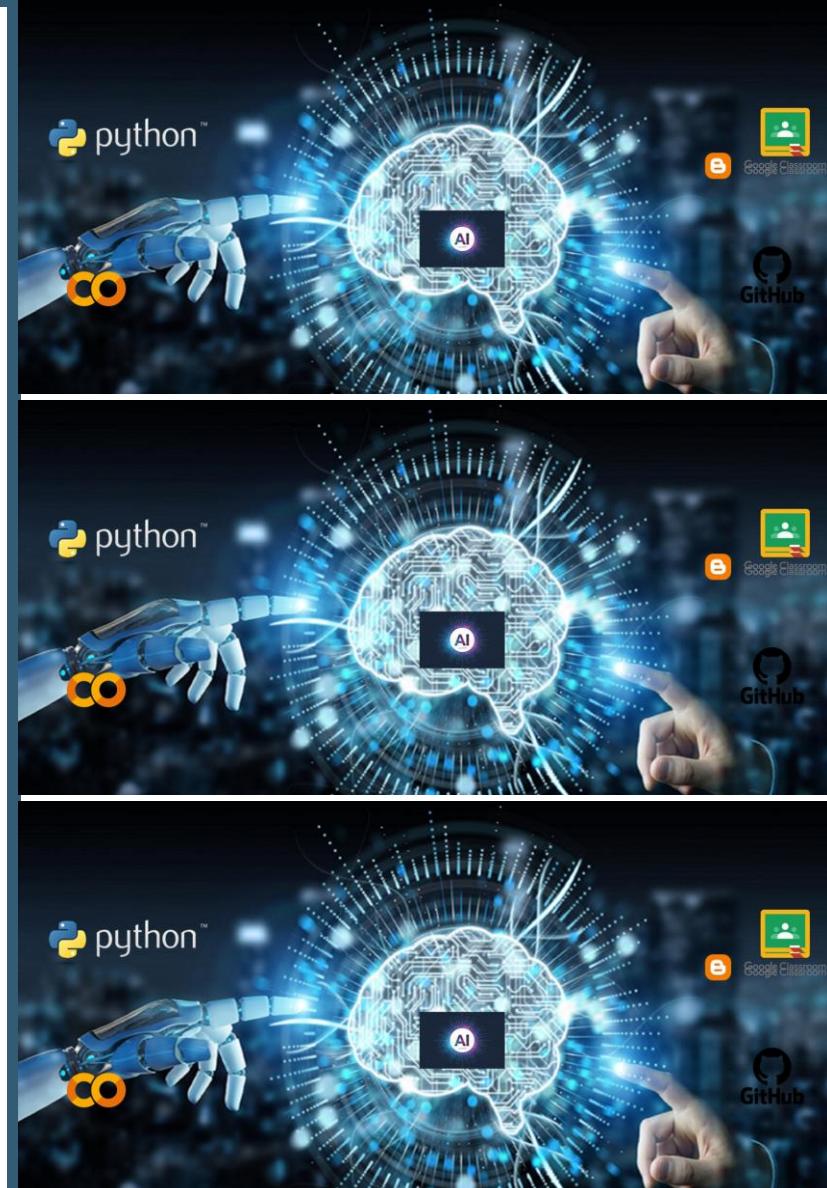


*Hiram Amaral*

**AULA 03**

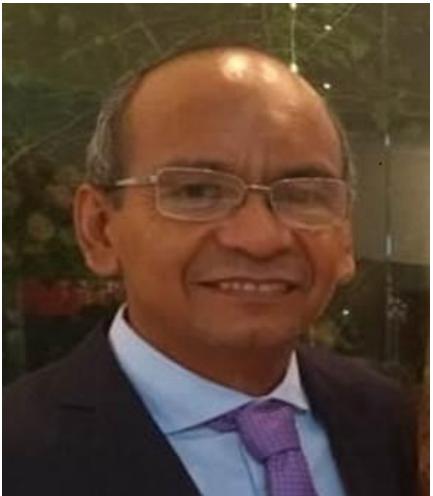
# SUMÁRIO DA APRESENTAÇÃO

1. Apresentação dos instrutores , monitores e coordenação
2. Apresentação do Módulo Lógica de Programação para IA com Python
3. Breve introdução à Linguagem de Programação
4. Calendário do módulo
5. Apresentação do Ambiente de Estudo
6. Apresentação da Estratégia de Ensino-Aprendizagem
7. Apresentação do Roteiro do Aluno Estudante
8. Comissionamento do ambiente (mãos à obra)





**Dhenifer Araújo**  
Coordenadora, MsC



**Hiram Amaral**  
Instrutor, MsC



**Diogo Rezende**  
Instrutor, Dr.



**Jhonatas**  
Monitor



**Nathalia**  
Monitora

# 1. APRESENTAÇÃO DA EQUIPE DO MÓDULO LÓGICA DE PROGRAMAÇÃO

→ C ⌂ https://classroom.google.com/u/0/c/NDgwMDk1NDk3NTE4

Curso de Inteligência Artificial - CETAM  
Módulo I - Lógica de Programação IA com Python

Mural Atividades Pessoas Notas

python™ CO Curso de Inteligência Artificial - CETAM Módulo I - Lógica de Programação IA com Python

Código da turma: jhnlpuy

Personalizar

1. Aulas expositivas
2. Interação com estudantes
3. Montagem do ambiente de estudo
4. Execução de exercício em Python
5. Utilização do Google Colab
6. Visita a sites de Machine Learning
7. Preparação desenvolvimento do miniprojeto
8. Elaboração do miniprojeto
9. Utilização do Github para armazenar código
10. Defesa do miniprojeto

## 2. APRESENTAÇÃO DO MÓDULO LP\_IA PYTHON LÓGICA DE PROGRAMAÇÃO

<https://logicadeprogramacaoparaiaacompython.blogspot.com/>



Março 2022  
webcid.com.br

Sem	Dom	Seg	Ter	Qua	Qui	Sex	Sáb
9			1	2	3	4	5
10	6	7	8	9	10	11	12
11	13	14	15	16	17	18	19
12	20	21	22	23	24	25	26
13	27	28	29	30	31		

1: Carnaval 2: Cinzas 8: Dia Internacional da Mulher 20: Início do outono  
2: Nova 10: Cresc. 18: Cheia 25: Ming.

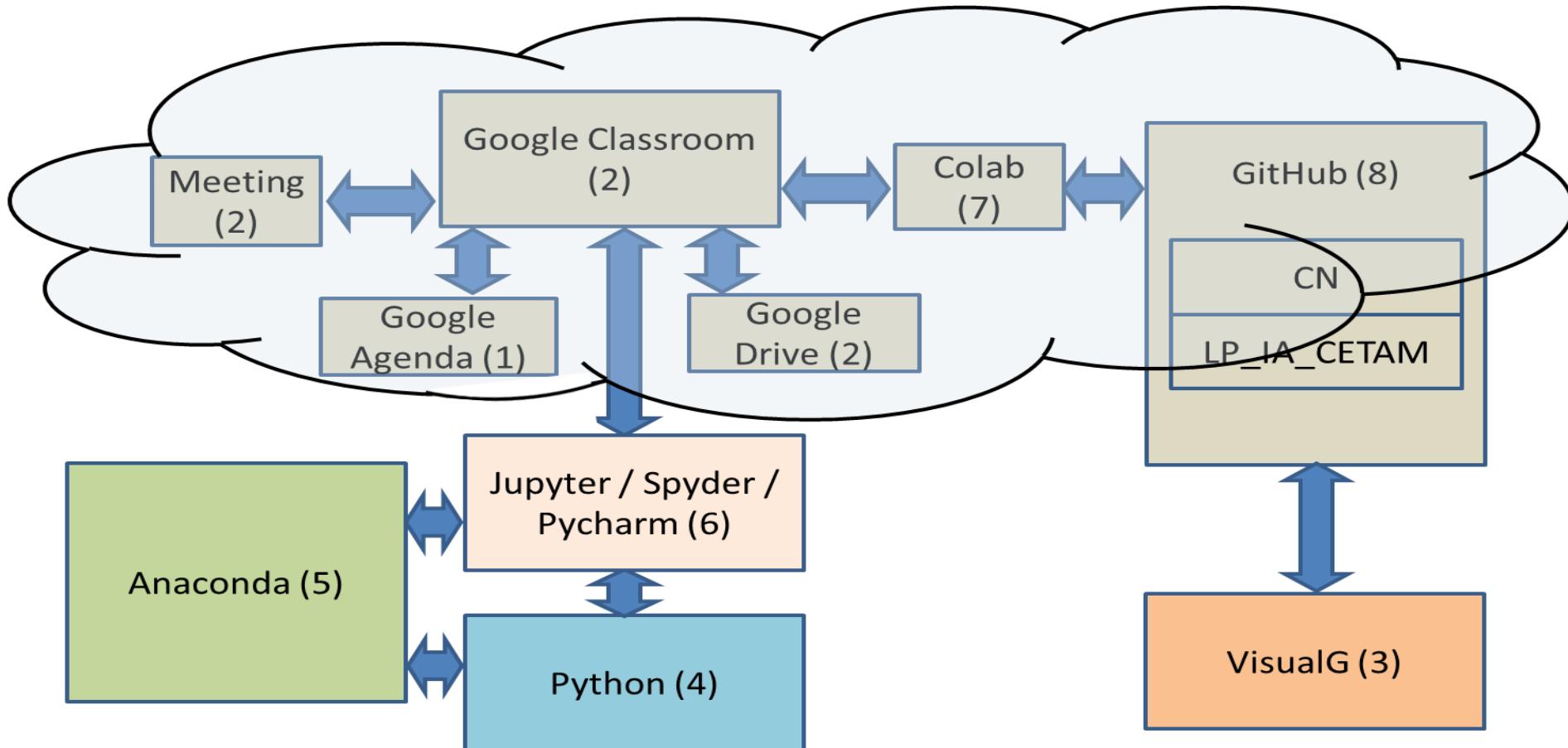


Abril 2022  
webcid.com.br

Sem	Dom	Seg	Ter	Qua	Qui	Sex	Sáb
13							1
14	3		4	5	6	7	8
15	10		11	12	13	14	15
16	17		18	19	20	21	22
17	24		25	26	27	28	29
							30

15: Paixão de Cristo 17: Páscoa 19: Dia do Índio 21: Tiradentes  
22: Descobrimento do Brasil  
1: Nova 9: Cresc. 18: Cheia 23: Ming. 30: Nova

### 3. CALENDÁRIO DO MÓDULO LP\_IA PYTHON LÓGICA DE PROGRAMAÇÃO



## 5. APRESENTAÇÃO DA ESTRATÉGIA DE ENSINO-APRENDIZAGEM

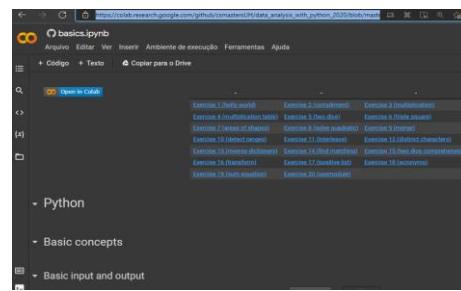
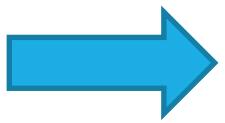
### VISÃO GERAL



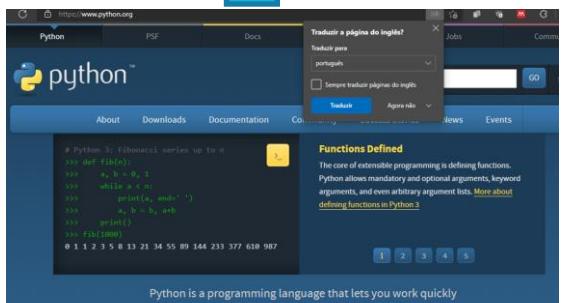
# 5. APRESENTAÇÃO DA ESTRATÉGIA DE ENSINO-APRENDIZAGEM **LÓGICA DE PROGRAMAÇÃO**



<https://classroom.google.com/u/0/c/NDM5MTQ4ODU2MzA4>



[basics.ipynb - Colaboratory \(google.com\)](#)



[Welcome to Python.org](#)

Miniprojeto



[Authorize application \(github.com\)](#)

# Montagem do ambiente de desenvolvimento e estudos

UNIDADE 1:	LÓGICA DE PROGRAMAÇÃO				
<b>VISÃO GERAL DO MÓDULO E DA ESTRATÉGIA DE ESTUDO</b> Aula 1 3 horas	1. Principais carreiras na área de IA; 2. Principais Aplicações de IA no mundo real; 3. Introdução a lógica de programação; 4. Introdução a algoritmos; 5. Introdução a linguagem Python; 6. Visão geral de <i>deep learning</i> ; 7. Algoritmos de <i>deep learning</i> ; 8. <i>Deep learning</i> para visão computacional; 9. Fundamentos da visão computacional; 10. Tipos de visão computacional; 11. Introdução a Inteligência Artificial; 12. Fundamentos teóricos e práticos de IA; 13. Fundamentos de ciência de dados; 14. Ferramentas de desenvolvimento em IA;	1- Quadro branco 2-Pinceis 3-Computadores 4-Acesso à Internet 5-Datashow	- Aula expositiva interativa - Acesso a sites de Programação, Algoritmos, Ciência de Dados, IA, Python, Projetos de IA, IDE's. -Exemplo de miniprojeto - Expectativa ou objetivos do estudante com o módulo de LP_IA_PY	- Aplicar questões de sensibilização e nivelamento por meio de ferramentas computacionais - Aplicação de exercício de assimilação.	

Aula 2 <b>AMBIENTE DE DESENVOLVIMENTO</b>	3 horas	<ul style="list-style-type: none"> <li>- Preparação de Ambiente Python;</li> <li>- Instalação de Bibliotecas;</li> <li>-Montagem do Ambiente de Desenvolvimento:</li> </ul>	<ul style="list-style-type: none"> <li>-instalação visualg (<a href="http://visualg3.com.br/baixar-o-visualg3-0/">http://visualg3.com.br/baixar-o-visualg3-0/</a>)</li> <li>-instalação python (<a href="https://www.python.org/downloads/">https://www.python.org/downloads/</a>)</li> <li>-Cadastro no colab (<a href="https://colab.research.google.com/">https://colab.research.google.com/</a>) <ul style="list-style-type: none"> <li>-integração com google drive</li> <li>-integração com o github</li> </ul> </li> <li>-instalação do github desktop (<a href="https://desktop.github.com/">https://desktop.github.com/</a>)</li> <li>-visualização do github</li> <li>-instalação pacote anaconda (<a href="https://docs.anaconda.com/anaconda/install/windows/">https://docs.anaconda.com/anaconda/install/windows/</a>)</li> </ul> <p>1- Quadro branco 2-Pinceis 3-Computadores 4-Acesso à Internet 5-Datashow</p>	<ul style="list-style-type: none"> <li>- Aula expositiva interativa</li> <li>- ReApresentação do Módulo Lógica de Programação para IA com Python</li> <li>- ReApresentação do Ambiente de Estudo</li> <li>ReApresentação da Estratégia de Ensino-Aprendizagem</li> <li>ReApresentação do Roteiro do Aluno Estudante</li> <li>Comissionamento do ambiente (mãos à obra)</li> <li>- Acompanhar instalações</li> <li>- Avaliar ambiente de desenvolvimento</li> </ul> <ul style="list-style-type: none"> <li>- Avaliar e pontuar p Ambiente de desenvolvimento</li> <li>- Realizar exercício de acesso e manipulação de arquivos no ambiente de desenvolvimento</li> </ul>

<p><b>Aula 3</b></p> <p><b>AMBIENTE DE DESENVOLVIMENTO</b></p>	<p>3 hora s</p>	<p><b>CONTINUAÇÃO DA AULA ANTERIOR</b></p>	<ul style="list-style-type: none"> <li>- Google Colab <a href="https://colab.research.google.com/#scrollTo=GJBs_fIRovLc">https://colab.research.google.com/#scrollTo=GJBs_fIRovLc</a></li> <li><a href="https://colab.research.google.com/github/csmastersUH/data_analysis_with_python_2020/blob/master/basics.ipynb#Exercise-1-(hello-world)">https://colab.research.google.com/github/csmastersUH/data_analysis_with_python_2020/blob/master/basics.ipynb#Exercise-1-(hello-world)</a></li> <li>- Python <a href="https://Welcome to Python.org">Welcome to Python.org</a></li> <li>- Visualg <a href="https://visualg3.com.br/baixe-o-visualg-3-0-7/">https://visualg3.com.br/baixe-o-visualg-3-0-7/</a></li> <li>- Classroom <a href="https://LOGICA DE PROGRAMACAO PARA IA COM PYTHON TECNOLOGIA DA INFORMACAO E COMPUTACAO (google.com)">LOGICA DE PROGRAMACAO PARA IA COM PYTHON TECNOLOGIA DA INFORMACAO E COMPUTACAO (google.com)</a></li> <li>- Github <a href="https://github.com/hiramaral/lp_ia_python">https://github.com/hiramaral/lp_ia_python</a></li> <li>- Googledrive <a href="https://classroom.google.com/u/0/c/NDM5MTQ4ODU2MzA4">https://classroom.google.com/u/0/c/NDM5MTQ4ODU2MzA4</a></li> <li>- Blog do módulo: <a href="https://logicadeprogramacaoparaiaacompython.blogspot.com/2021/12/lpiapthon-aula-13.html">https://logicadeprogramacaoparaiaacompython.blogspot.com/2021/12/lpiapthon-aula-13.html</a></li> </ul> <p>1- Quadro branco 2-Pinceis 3-Computadores 4-Acesso à Internet 5-Datashow</p>	<ul style="list-style-type: none"> <li>- Aula expositiva interativa</li> <li>- ReApresentação do Módulo Lógica de Programação para IA com Python</li> <li>- ReApresentação do Ambiente de Estudo</li> </ul> <p>ReApresentação da Estratégia de Ensino-Aprendizagem</p> <p>ReApresentação do Roteiro do Aluno Estudante</p> <p>Comissionamento do ambiente (mãos à obra)</p> <ul style="list-style-type: none"> <li>- Acompanhar instalações</li> <li>- Avaliar ambiente de desenvolvimento</li> </ul>	<ul style="list-style-type: none"> <li>- Avaliar e pontuar o Ambiente de desenvolvimento</li> <li>- Realizar exercício de acesso e manipulação de arquivos no ambiente de desenvolvimento</li> </ul>
--	-------------------------	--	---	---	--

Montando nosso ambiente de  
estudo e desenvolvimento

Continuação...

# Acesse o Google Colab com sua conta Google

[Olá, este é o Colaboratory - Colaboratory \(google.com\)](#)

# Exemplo

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

The screenshot shows the Google Colaboratory interface. At the top, there's a navigation bar with tabs: Exemplos, Recente (which is selected), Google Drive, GitHub, and Upload. To the right of the tabs are buttons for Compartilhar, settings, and a user profile icon. Below the navigation bar is a search bar labeled "Filtrar notebooks". Underneath is a table with columns: Título, Aberto pela última vez (sorted by ascending date), and Primeiro acesso (sorted by descending date). The table contains one row for a notebook titled "Olá, este é o Colaboratory", which was opened at 03:05 and first accessed at 03:05. There are icons for deleting and viewing the notebook next to the row. At the bottom of the modal window are buttons for "Novo notebook" and "Cancelar". On the left side of the main window, there's a sidebar with sections like Índice, Primeiros passos, Ciência de dados, Machine learning, Mais recursos, and Exemplos em destaque. The main workspace area is currently empty.

<https://colab.research.google.com/?authuser=3>

Olá, este é o Colaboratory

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda

Índice

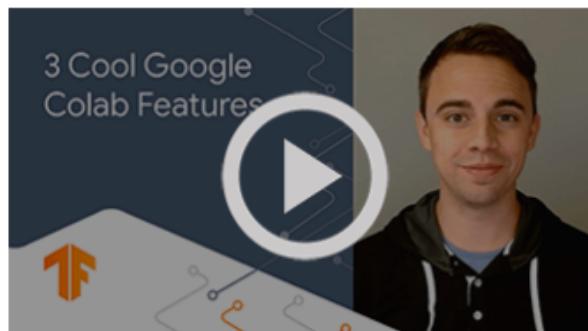
- Primeiros passos
- Ciência de dados
- Machine learning
- Mais recursos
  - Exemplos em destaque
- Seção

+ Código + Texto Copiar para o Drive

Conectar | Editar

## Conheça o Colab

Se você já conhece bem o Colab, confira este vídeo para saber mais sobre as tabelas interativas, a visualização do histórico de código executado e o Palette de comandos.



## O que é o Colab?

O Colab, ou "Colaboratory", permite escrever e executar Python no navegador e conta com o seguinte:

- Nenhuma configuração necessária
- Acesso gratuito a GPUs



Olá, este é o Colaboratory

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda

Compartilhar



Índice



Primeiros passos

Ciência de dados

Machine learning

Mais recursos

Exemplos em destaque



+ Seção

+ Código + Texto Copiar para o Drive

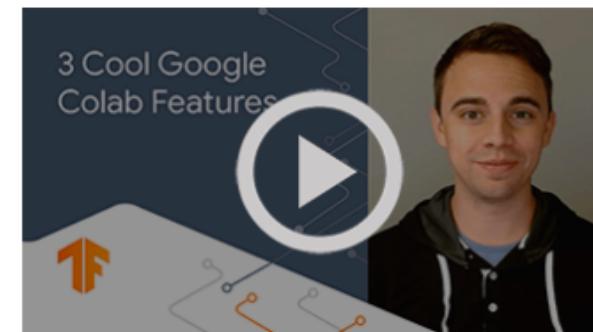
Conectar

Editar



## Conheça o Colab

Se você já conhece bem o Colab, confira este vídeo para saber mais sobre as tabelas interativas, a visualização do histórico de código executado e o Palette de comandos.



## O que é o Colab?

O Colab, ou "Colaboratory", permite escrever e executar Python no navegador e conta com o seguinte:

- Nenhuma configuração necessária
- Acesso gratuito a GPUs



<https://colab.research.google.com/drive/1e981Y4mtECLgx-Wajd1oKTqhojy37vvl?authuser=3>

 Untitled0.ipynb 

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda

Comentário Compartilhar  

+ Código + Texto Conectar |  Editar | ^

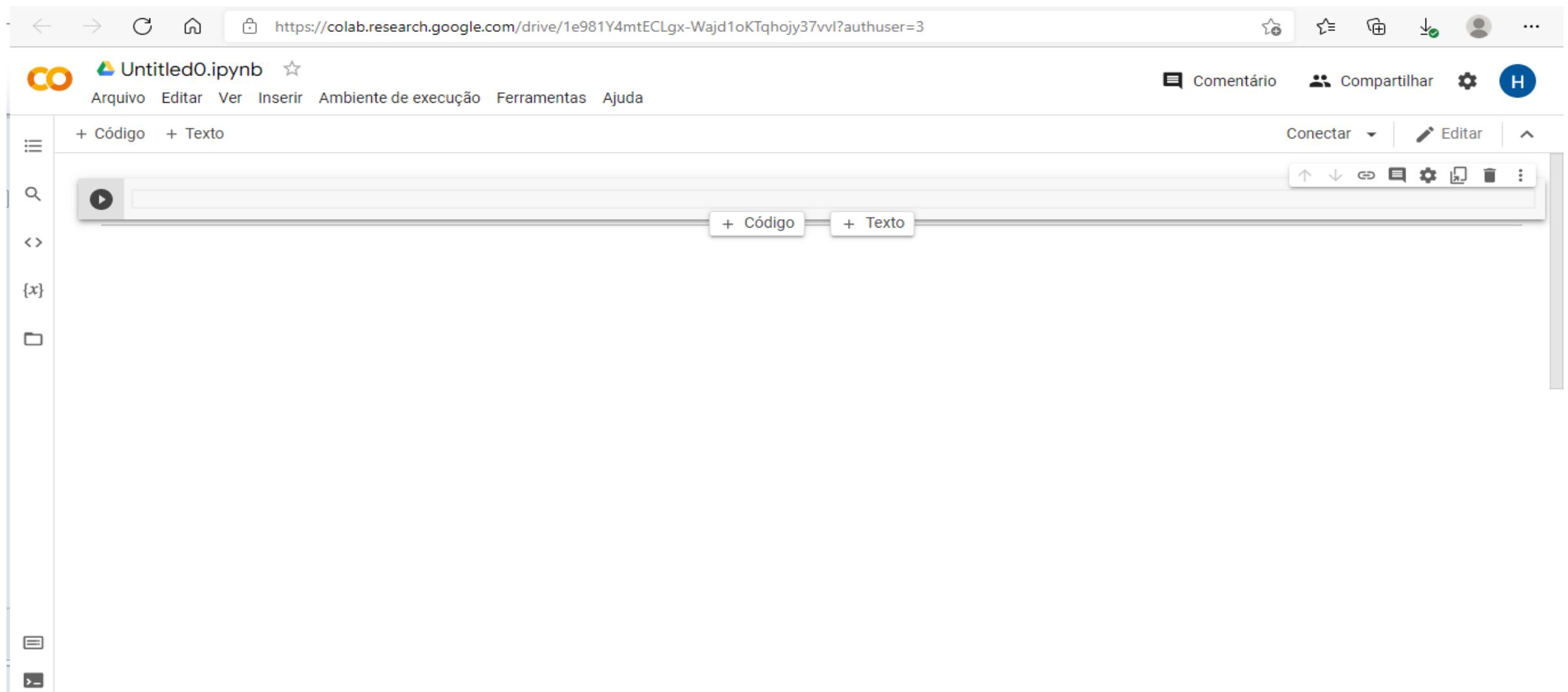
 + Código + Texto ↑ ↓ ↻ ⚙️ 📁 :

{x}

□

☰

☰





Untitled0.ipynb

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

Comentário

Compartilhar



H

+ Código + Texto

✓ RAM  
Disco

Editar



+ Código

+ Texto



{x}



 Untitled0.ipynb 

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

Comentário Compartilhar  

+ Código + Texto

RAM  | Disco  |  Editar 

0s  print("Olá estudantes de Python")

Olá estudantes de Python

{x}

□

☰

↳

# **PRIMEIROS PASSOS**

https://colab.research.google.com/?authuser=3#scrollTo=Primeiros\_passos

Olá, este é o Colaboratory

Arquivo Editar Ver Inserir

Índice

Primeiros passos

- Ciência de dados
- Machine learning
- Mais recursos
- Exemplos em destaque

+ Seção

Exemplos Recente Google Drive GitHub Upload

Filtrar notebooks

Título	Aberto pela última vez	Primeiro acesso	
Olá, este é o Colaboratory	03:15	03:05	
meuprimeironotebook1.ipynb	03:08	03:08	

Novo notebook Cancelar

U trabalho. Assista ao vídeo

chamado **notebook Colab** que

na-o em uma variável e imprime c



Olá, este é o Colaboratory

Arquivo Editar Ver Inserir

Compartilhar



## Índice

Primeiros passos

Ciência de dados

Machine learning

Mais recursos

Exemplos em destaque

## Seção

Exemplos	Recente	Google Drive	GithHub	Upload
Filtrar notebooks				
Título		Aberto pela última vez	Primeiro acesso	
 Olá, este é o Colaboratory		03:16	03:05	
 meuprimeironotebook1.ipynb		03:08	03:08	 

[Novo notebook](#)

[Cancelar](#)

Olá, este é o Colaboratory

Arquivo Editar Ver Inserir

Compartilhar  

Exemplos Recente Google Drive GitHub Upload

Conectar  Editar

Índice

- Primeiros passos
- Ciência de dados
- Machine learning
- Mais recursos
  - Exemplos em destaque
- Seção

Filtrar notebooks

Título

Thumbnail	Title	Actions
	Overview of Colaboratory Features	
	Markdown Guide	
	Charts in Colaboratory	
	External data: Drive, Sheets, and Cloud Storage	
	Getting started with BigQuery	

[Novo notebook](#) Cancelar

Olá, este é o Colaboratory

Arquivo Editar Ver Inserir

Compartilhar  

Exemplos Recente Google Drive GitHub Upload

Índice

Primeiros passos  
Ciência de dados  
Machine learning  
Mais recursos  
Exemplos em destaque

Filtrar notebooks

Título	Aberto pela última vez	Primeiro acesso	
Olá, este é o Colaboratory	03:42	03:05	
Overview of Colaboratory Features	03:42	03:27	
meuprimeironotebook1.ipynb	03:08	03:08	 

lização do histórico de código

+ Seção

[Novo notebook](#) Cancelar



Olá, este é o Colaboratory

Arquivo Editar Ver Inserir

Compartilhar



## Índice

Primeiros passos

Ciência de dados

Machine learning

Mais recursos

Exemplos em destaque

+ Seção

Exemplos

Recente

Google Drive

GitHub

Upload

Conectar

Editar

Filtrar notebooks

Título	Proprietário	Aberto pela última vez	Última modificação	
meuprimeironotebook1.ipynb	Hiram Amaral	03:11	03:11	

[Novo notebook](#) Cancelar



Olá, este é o Colaboratory

Arquivo Editar Ver Inserir

Compartilhar



Exemplos

Recente

Google Drive

Github

Upload

Conectar

Editar

## Índice

Primeiros passos

Ciência de dados

Machine learning

Mais recursos

Exemplos em destaque



+ Seção

Nova célula

Escolher Arquivo Nenhum arquivo escolhido

[Novo notebook](#) Cancelar



## Tabela de conteúdos

### Cells

- [Code cells](#)
  - [Text cells](#)
  - [Adding and moving cells](#)
  - [Working with python](#)
  - [System aliases](#)
  - [Magics](#)
  - [Automatic completions and exploring code](#)
  - [Exception Formatting](#)
  - [Rich, interactive outputs](#)
  - [Integration with Drive](#)
  - [Commenting on a cell](#)
- [+ Secção](#)

+ Código + Texto Copiar para dirigir

Connect Editing



## ▼ Células

Um caderno é uma lista de células. As células contêm texto explicativo ou código executável e sua saída. Clique em uma célula para selecioná-la.

## ▼ Células de código

Abaixo está uma **célula de código**. Uma vez que o botão da barra de ferramentas indique CONECTADO, clique na célula para selecioná-la e execute o conteúdo das seguintes maneiras:

- Clique no ícone **Reproduzir** na sarjeta esquerda da célula;
- **Digite Cmd/Ctrl+Enter** para executar a célula no lugar;
- **Digite Shift+Enter** para executar a célula e mover o foco para a próxima célula (adicionando uma se nenhuma existir); ou
- **Digite Alt+Enter** para executar a célula e insira uma nova célula de código imediatamente abaixo dela.

Há opções adicionais para executar algumas ou todas as células no menu **Runtime**.

```
[ ] 1 a = 10  
2 a
```



## Tabela de conteúdos



### Células

Células de código

Células de texto

Adicionando e movendo células

Trabalhando com python

Codinomes do sistema

Magia

Conclusões automáticas e código de exploração

Formatação de exceção

Saídas ricas e interativas

Integração com a unidade

Comentando sobre um celular

+ Secção



+ Código

+ Texto

Copiar para dirigir

Connect



Editing



## Células

Um caderno é uma lista de células. As células contêm texto explicativo ou código executável e sua saída. Clique em uma célula para selecioná-la.

## Células de código

Abaixo está uma **célula de código**. Uma vez que o botão da barra de ferramentas indique CONECTADO, clique na célula para selecioná-lo e execute o conteúdo das seguintes maneiras:

- Clique no ícone **Reproduzir** na sarjeta esquerda da célula;
- **Digite Cmd/Ctrl+Enter** para executar a célula no lugar;
- **Digite Shift+Enter** para executar a célula e mover o foco para a próxima célula (adicionando uma se nenhuma existir); ou
- **Digite Alt+Enter** para executar a célula e insira uma nova célula de código imediatamente abaixo dela.

Há opções adicionais para executar algumas ou todas as células no menu **Runtime**.

```
[ ] 1 a = 10  
2 um
```

# Células

Um caderno é uma lista de células. As células contêm texto explicativo ou código executável e sua saída. Clique em uma célula para selecioná-la.

## Células de código

Abaixo está uma célula de código. Uma vez que o botão da barra de ferramentas indique CONECTADO, clique na célula para selecioná-lo e execute o conteúdo das seguintes maneiras:

Clique no ícone Reproduzir na sarjeta esquerda da célula;

Digite **Cmd/Ctrl+Enter** para executar a célula no lugar;

Digite **Shift+Enter** para executar a célula e mover o foco para a próxima célula (adicionando uma se nenhuma existir); ou

Digite **Alt+Enter** para executar a célula e insira uma nova célula de código imediatamente abaixo dela.

Há opções adicionais para executar algumas ou todas as células no menu Runtime.



The screenshot shows a Jupyter Notebook cell. On the left, there is a play button icon inside a grey box. The cell contains the following code:

```
1 a = 10
2 a
```

The output below the cell shows the result:

```
10
```

To the right of the cell, there is a toolbar with various icons: up arrow, down arrow, copy, paste, settings, and more.

## Células de texto

Isto é uma célula de texto. Você pode clicar duas vezes para editar este celular. As células de texto usam sintaxe de marcação. Para saber mais, consulte nosso guia de marcação.

Você também pode adicionar matemática às células de texto usando LaTeX para ser renderizada pelo MathJax. Basta colocar a declaração dentro de um par de placas de \$. Por exemplo  $\sqrt{3x-1}+(1+x)^2$  torna-se  $\sqrt{3x-1}+(1+x)^2$ .

## Adicionando e movendo células

Você pode adicionar novas células usando os botões + CODE e + TEXT que mostram quando você pира entre as células. Esses botões também estão na barra de ferramentas acima do notebook, onde podem ser usados para adicionar uma célula abaixo da célula selecionada atualmente.

Você pode mover uma célula selecionando-a e clicando em **Cell Up** ou **Cell Down** na barra de ferramentas superior.

Células consecutivas podem ser selecionadas por "seleção de laços" arrastando-se de fora de uma célula e através do grupo. Células não adjacentes podem ser selecionadas simultaneamente clicando em uma e, em seguida, segurando Ctrl enquanto clicam em outra. Da mesma forma, o uso do Shift em vez de Ctrl selecionará todas as células intermediárias.

# Trabalhando com python

Colaboratory é construído em cima do Caderno Jupyter. Abaixo estão alguns exemplos de funções de conveniência fornecidas.

Processos python de longa duração podem ser interrompidos. Execute a seguinte célula e selecione Executar -> interromper a execução (tecla quente: Cmd/Ctrl-M I) para impedir a execução.

The screenshot shows the Google Colaboratory interface. On the left is a sidebar with navigation links like 'Índice', 'Cells' (selected), 'Code cells', 'Text cells', etc. The main area has a toolbar with icons for back, forward, search, and file operations, and a URL bar pointing to 'https://colab.research.google.com/notebooks/basic\_features\_overview.ipynb'. Below the toolbar is a menu bar with 'Arquivo', 'Editar', 'Ver', 'Inserir', 'Ambiente de execução', 'Ferramentas', and 'Ajuda'. To the right of the menu are sharing, settings, and profile icons. The main content area contains a code cell with the following Python code:

```
[ ] 1 import time
2 print("Sleeping")
3 time.sleep(30) # sleep for a while; interrupt me!
4 print("Done Sleeping")
```

When run, the cell prints "Sleeping" and then hangs. A red error message at the bottom indicates a 'KeyboardInterrupt' exception:

```
Sleeping
-----
KeyboardInterrupt                                     Traceback (most recent call last)
<ipython-input-3-626f81edbca4> in <module>()
      1 import time
      2 print "Sleeping"
----> 3 time.sleep(30) # sleep for a while; interrupt me!
      4 print "Done Sleeping"
```

Below the error message, another 'KeyboardInterrupt' message is shown:

```
KeyboardInterrupt:
```

# Trabalhando com python

Colaboratory é construído em cima do Caderno Jupyter. Abaixo estão alguns exemplos de funções de conveniência fornecidas.

Processos python de longa duração podem ser interrompidos. Execute a seguinte célula e selecione Executar -> interromper a execução (tecla quente: Cmd/Ctrl-M I) para impedir a execução.

The screenshot shows a Jupyter Notebook cell. The code in the cell is:

```
1 import time
2 print("Sleeping")
3 time.sleep(30) # sleep for a while; interrupt me!
4 print("Done Sleeping")
```

The output pane below the cell shows the results of the execution:

Sleeping  
Done Sleeping

The top right corner of the notebook interface has a toolbar with various icons for navigating and managing cells.

## Codinomes do sistema

Jupyter inclui atalhos para operações comuns, como ls:

Overview of Colaboratory Features

Arquivo Editar Vista Inserir Runtime Ferramentas Ajuda Não é possível salvar mudanças

Compartilhar | |

Tabela de conteúdos

Células

- Células de código
- Células de texto

Adicionando e movendo células

Trabalhando com python

**Codinomes do sistema**

- Magia
- Conclusões automáticas e código de exploração
- Formatação de exceção
- Saídas ricas e interativas
- Integração com a unidade
- Comentando sobre um celular

Secção

+ Código + Texto Copiar para dirigir

RAM Disk Editing

Jupyter inclui atalhos para operações comuns, como ls:

```
1 ! ls /bin
```

arch@	dmesg*	ls*	pwd*	true*
awk@	dnsdomainname*	lsmod*	readlink*	umount*
basename@	domainname*	mail*	red@	uname*
bash*	echo*	mkdir*	rm*	uncompress*
bunzip2@	ed@	mknod*	rmdir*	usleep*
busybox*	egrep*	mktemp*	run-parts*	ver*
bzip2@	false*	more*	sed*	which*
cat*	fgrep*	mount*	sh@	wrapper_checkpoints/
chgrp*	gawk@	mountpoint*	sleep*	zcat*
chmod*	grep*	mv*	sort@	zcmp*
chown*	gunzip*	nc*	stty*	zdiff*
cp*	gzexe*	netcat@	su*	zegrep*
cpio*	gzip*	netstat*	sync*	zfgrep*
csh@	hostname*	nice@	tailf*	zforce*
cut@	igawk@	pidof@	tar*	zgrep*
date*	kill*	ping*	tcsh@	zless*
dd*	ln*	ping6*	tempfile*	zmore*
df*	login*	ps*	touch*	znew*

Isso provavelmente gerou uma grande saída. Você pode selecionar a célula e limpar a saída por qualquer um: !ls

1. Clicando no botão de saída clara (x) na barra de ferramentas acima da célula; ou

Execute qualquer outro processo usando com interpolação de sequência de caracteres a partir de variáveis python, e observe que o resultado pode ser atribuído a uma variável!:!

The screenshot shows the 'Overview of Colaboratory Features' page. On the left, there's a sidebar with a tree view of content categories like 'Células', 'Trabalhando com python', 'Codinomes do sistema', and 'Magia'. The main area displays a text block with instructions and two code examples.

**Instructions:**

1. Clicando no botão de saída clara (x) na barra de ferramentas acima da célula; ou
2. Clique com a direita na sarjeta esquerda da área de saída e selecione "Saída clara" no menu de contexto.

**Text:**

Execute qualquer outro processo usando com interpolação de sequência de caracteres a partir de variáveis python, e observe que o resultado pode ser atribuído a uma variável!: !

**Code Example 1:**

```
1 mensagem = 'Colaboratory é ótimo!'
2 foo = !echo -e '$message\n$message\n$message'
3 foo
```

**Output:**

```
['Colaboratory is great!', 'Colaboratory is great!', 'Colaboratory is great!']
```

**Section: Magia**

Colaboratory compartilha a noção de magias de Jupyter. Há anotações taquigrafias que mudam a forma como o texto de uma célula é executado. Para saber mais, veja a [página de magias de Jupyter](#).

**Code Example 2:**

```
[ ] 1 %%html
2 <marquee style='largura: 30%; cor: azul;' ><b>Whee! </b></marquei>
```

Whe!

# Magia

Colaboratory compartilha a noção de magias de Jupyter. Há anotações taquigrafias que mudam a forma como o texto de uma célula é executado. Para saber mais, veja a [página de magias de Jupyter](#).

The screenshot shows the Google Colaboratory interface at the URL [https://colab.research.google.com/notebooks/basic\\_features\\_overview.ipynb#scrollTo=5OCYEvK5QSHf](https://colab.research.google.com/notebooks/basic_features_overview.ipynb#scrollTo=5OCYEvK5QSHf). The page title is "Overview of Colaboratory Features". The sidebar on the left contains links such as "Índice", "Cells", "Working with python", "System aliases", "Integration with Drive", and "Commenting on a cell". The main content area displays two code cells. The first cell contains:[ ] 1 %%html  
2 <marquee style='width: 30%; color: blue;'><b>Whee!</b></marquee>The second cell contains:[ ] 1 %%html  
2 <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 450 400" width="200" height="200">  
3 <rect x="80" y="60" width="250" height="250" rx="20" style="fill:red; stroke:black; fill-  
4 <rect x="180" y="110" width="250" height="250" rx="40" style="fill:blue; stroke:black; fi  
5 </svg>Below the code cells, there is a preview of the SVG output, which consists of two overlapping rectangles: a red one on top and a blue one underneath.

## Conclusões automáticas e código de exploração

A Colab fornece conclusões automáticas para explorar atributos de objetos Python, bem como para visualizar rapidamente as sequências de documentação. Como exemplo, execute primeiro a seguinte célula para importar o módulo numpy

```
[ ] 1 import numpy as np
```

Se agora você inserir o cursor após e pressionar o Período(), você verá a lista de conclusões disponíveis dentro do módulo. As conclusões podem ser abertas novamente usando o Ctrl+Space.np.`np`

```
[ ] 1 np
```

Se você digitar um parêntese aberto após qualquer função ou classe no módulo, você verá um pop-up de sua sequência de documentação:

```
[ ] 1 np.ndarray
```

A documentação pode ser aberta novamente usando Ctrl+Shift+Space ou você pode visualizar a documentação para o método por mouse pairando sobre o nome do método.

Ao pairar sobre o nome do método, o link abrirá a documentação em um painel persistente. O link navegará até o código-fonte para o método.[Open in tab](#)[View source](#)

## Formatação de exceção

As exceções são bem formatadas nas saídas colab:

As exceções são bem formatadas nas saídas colab:

```
1 x = 1
2 y = 4
3 z = y/(1-x)

ZeroDivisionError                               Traceback (most recent call last)
<ipython-input-14-dc39888fd1d2> in <module>()
      1 x = 1
      2 y = 4
----> 3 z = y/(1-x)

ZeroDivisionError: integer division or modulo by zero
```

SEARCH STACK OVERFLOW

## Saídas ricas e interativas

Até agora todas as saídas geradas foram texto, mas podem ser mais interessantes, como o gráfico abaixo.

```
1 importação numpy como np
2 de matplotlib importação pyplot como plt
3
4 ys = 200 + np.random.randn(100)
5 x = [x para x na faixa (len(ys))]
6
7 plt.plot(x, ys, '-')
8 plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alfa=0,6)
9
10 plt.title ("Fills and Alpha Example")
11 plt.show()
```



# **Integração com a unidade**

Colaboratory é integrado ao Google Drive. Ele permite que você compartilhe, comente e colabore no mesmo documento com várias pessoas:

O botão SHARE (canto superior direito da barra de ferramentas) permite compartilhar o notebook e controlar as permissões definidas nele.

**File->Make a Copy** cria uma cópia do notebook em Drive.

**File->Save** salva o arquivo para o Drive.

**File->Save and checkpoint** pins a versão para que ela não seja excluída do histórico de revisão.

**File->Revision history** mostra arquivo do histórico de revisão do notebook..

## Comentando sobre um celular

Você pode comentar em um notebook colaborativo como faria em um Documento do Google. Os comentários são anexados às células e são exibidos ao lado da célula a que se referem. Se você tiver permissões somente para comentários, você verá um botão de comentário no canto superior direito da célula quando você pairar sobre ele.

Se você tiver permissões de edição ou comentário, você pode comentar em um celular de uma das três maneiras:

Selecione um celular e clique no botão de comentário na barra de ferramentas acima do canto superior direito da célula.

Clique com o botão direito de clicar em uma célula de texto e selecione Adicionar um comentário no menu de contexto.

Use o atalho Ctrl+Shift+M para adicionar um comentário à célula selecionada no momento.

Você pode resolver e responder a comentários, e você pode direcionar comentários para colaboradores específicos digitando +[endereço de e-mail] (por exemplo, ). Colaboradores endereçados serão enviados por e-mail. +user@domain.com

O botão Comment no canto superior direito da página mostra todos os comentários anexados ao notebook.

# O que é Markdown?

Colab tem dois tipos de células: texto e código. As células de texto são formatadas usando uma linguagem de marcação simples chamada Markdown.

Para ver a fonte Markdown, clique duas vezes em uma célula de texto, mostrando tanto a fonte Markdown quanto a versão renderizada. Acima da fonte Markdown há uma barra de ferramentas para auxiliar a edição.

The screenshot shows the Colab Markdown Guide interface. At the top, there's a navigation bar with icons for file, edit, view, insert, runtime, tools, and help, along with sharing and settings options. Below the bar, there are two tabs: '+ Código' and '+ Texto'. A 'Copiar para dirigir' button is also present. The main area contains two text cells:

- O que é Markdown?**

Colab tem dois tipos de células: texto e código. As células de texto são formatadas usando uma linguagem de marcação simples chamada Markdown.
- Para ver a fonte Markdown, clique duas vezes em uma célula de texto, mostrando tanto a fonte Markdown quanto a versão renderizada. Acima da fonte Markdown há uma barra de ferramentas para auxiliar a edição.

On the left side, there are sidebar icons for search, refresh, and file operations. On the right, there's a toolbar with icons for back, forward, search, and other editing functions. Below the text cells, there's a 'Referência' section containing a table of Markdown syntax:

Redução de preço	Visualizar
**bold text**	texto ousado
*italicized text* ou _italicized text_	texto itálico
`Monospace`	Monospace
~~strikethrough~~	strikethrough
[A link](https://www.google.com)	Um link
![An image](https://www.google.com/images/rss.png)	

At the bottom, there's a note about titles and a footer with '# Section 1'.



+ Código + Texto

A link (<https://www.google.com>)

[um link](#)

✓ RAM Disk

 Editing

H

Títulos são renderizados como títulos.

```
# Section 1  
# Section 2  
## Sub-section under Section 2  
### Sub-section under the sub-section under Section 2  
# Section 3
```

Seção 1

Seção 2

Subseção sob a Seção 2

## Subseção sob a subseção sob a Seção 2

Seção 3

A tabela de conteúdo, disponível no lado esquerdo do Colab, é preenchida usando no máximo um título de seção de cada célula de texto.



## Markdown Guide

Arquivo Editar Vista Inserir Runtime Ferramentas Ajuda

Compartilhar



+ Código

+ Texto



Copiar para dirigir



RAM



▼

Disk



▼

Editing



A tabela de conteúdo, disponível no lado esquerdo do Colab, é preenchida usando no máximo um título de seção de cada célula de texto.



>One level of indentation



Um nível de recuo



>>Two levels of indentation



Dois níveis de recuo



Blocos de código

```
```python
print("a")
```

```

```
print("a")
```



Listas encomendadas:

[+ Code](#)[+ Text](#) [Copy to Drive](#)RAM  
Disk

Editing



### Ordered lists:

1. One

1. Two

1. Three



1. One

2. Two

3. Three

### Unordered lists:

\* One

\* Two

\* Three

• One

• Two

• Three

### Equations:



## Markdown Guide

File Edit View Insert Runtime Tools Help

Share

+ Code + Text

RAM Disk Editing

• Three

Equations:

$y=x^2$

$e^{i\pi} + 1 = 0$

$e^x=\sum_{i=0}^{\infty} \frac{1}{i!}x^i$

$\frac{n!}{k!(n-k)!} = {n \choose k}$

$A_{m,n} =$   
$$\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

$y = x^2$

$e^{i\pi} + 1 = 0$



## Markdown Guide

File Edit View Insert Runtime Tools Help

Share

+ Code + Text

Copy to Drive

RAM Disk Editing

Equations:

```
$y=x^2$  
  
$e^{i\pi} + 1 = 0$  
  
$e^x=\sum_{i=0}^{\infty} \frac{1}{i!}x^i$  
  
$\frac{n!}{k!(n-k)!} = {n \choose k}$  
  
$A_{m,n} =  
\begin{pmatrix}  
a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\\cdots & a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\\vdots & \vdots & \vdots & \ddots & \vdots \\a_{m,1} & a_{m,2} & \cdots & a_{m,n}\end{pmatrix}$  
  

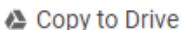
$$y = x^2$$
  
  

$$e^{i\pi} + 1 = 0$$
  
  

$$e^x=\sum_{i=0}^{\infty} \frac{1}{i!}x^i$$
  

$$e^x=\sum_{i=0}^{\infty} \frac{1}{i!}x^i$$

```

[+ Code](#)[+ Text](#)

✓ RAM Disk

Editing

 $y = x^2$  $e^{i\pi} + 1 = 0$  $\$e^x = \sum_{i=0}^{\infty} \frac{1}{i!} x^i \$e^x = \sum_{i=0}^{\infty} \frac{1}{i!} x^i$  $\$\\frac{n!}{k!(n-k)!} = \\{n \\choose k} \$\\frac{n!}{k!(n-k)!} = \\{n \\choose k}$ 
$$\$A_{m,n} = \\begin{pmatrix} a_{1,1} & a_{1,2} & \\cdots & a_{1,n} \\\\ a_{2,1} & a_{2,2} & \\cdots & a_{2,n} \\\\ \\vdots & \\vdots & \\ddots & \\vdots \\\\ a_{m,1} & a_{m,2} & \\cdots & a_{m,n} \\end{pmatrix}$$
$$A_{m,n} = \\begin{pmatrix} a_{1,1} & a_{1,2} & \\cdots & a_{1,n} \\\\ a_{2,1} & a_{2,2} & \\cdots & a_{2,n} \\\\ \\vdots & \\vdots & \\ddots & \\vdots \\\\ a_{m,1} & a_{m,2} & \\cdots & a_{m,n} \\end{pmatrix}$$

Tables:

| First column name | Second column name |
|-------------------|--------------------|
| Row 1, Col 1      | Row 1, Col 2       |
| Row 2, Col 1      | Row 2, Col 2       |

| First column name | Second column name |
|-------------------|--------------------|
| Row 1, Col 1      | Row 1, Col 2       |
| Row 2, Col 1      | Row 2, Col 2       |

Diferenças entre Colab Markdown e outros dialetos de Markdown

Colab usa marcado.js e por isso é semelhante, mas não completamente idêntico ao Markdown usado por Jupyter e Github.

Colab suporta (MathJax)  $\backslash\text{LaTeX}\backslash\text{LaTeX}$  equações como Jupyter, mas não permite tags HTML no Markdown.  
Colab não suporta algumas adições do GitHub, como emojis e caixas de seleção.

Se HTML deve ser incluído em um notebook Colab, consulte a magia `%%html`.

## Referências úteis

[Github Markdown basics](#)

[Github flavored Markdown](#)

[Original Markdown spec: Syntax](#)

[Original Markdown spec: Basics](#)

[marked.js library used by Colab](#)

[LaTex mathematics for equations](#)

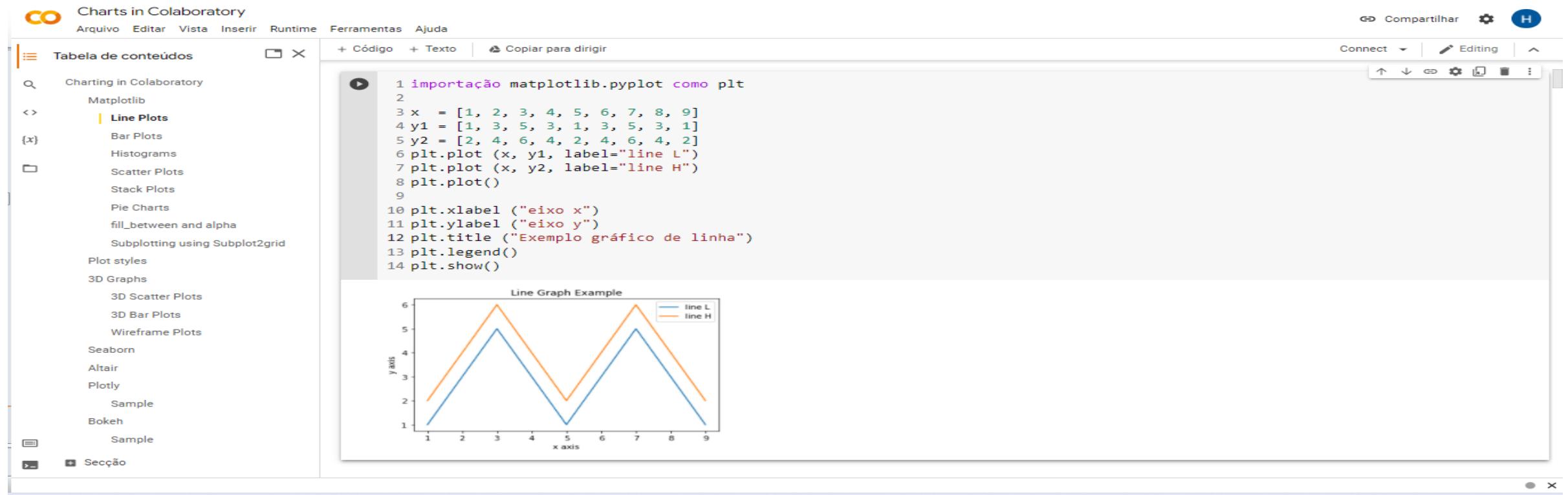
## Gráficos em Colaboratório

Um uso comum para notebooks é a visualização de dados usando gráficos. Colaboratory facilita com várias ferramentas de gráficos disponíveis como importações python.

### Matplotlib

Matplotlib é o pacote de gráficos mais comum, veja sua documentação para detalhes e seus exemplos de inspiração.

### Gráficos de Linha



The screenshot shows a Jupyter Notebook interface titled "Charts in Colaboratory". The left sidebar contains a "Tabela de conteúdos" (Table of contents) with categories like "Charting in Colaboratory", "Matplotlib", "Seaborn", "Altair", "Plotly", and "Bokeh". Under "Matplotlib", "Line Plots" is selected. The main area has tabs for "Código" (Code), "Texto" (Text), and "Copiar para dirigir" (Copy to drive). The code cell contains the following Python code:

```
1 importação matplotlib.pyplot como plt
2
3 x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
4 y1 = [1, 3, 5, 3, 1, 3, 5, 3, 1]
5 y2 = [2, 4, 6, 4, 2, 4, 6, 4, 2]
6 plt.plot(x, y1, label="line L")
7 plt.plot(x, y2, label="line H")
8 plt.plot()
9
10 plt.xlabel ("eixo x")
11 plt.ylabel ("eixo y")
12 plt.title ("Exemplo gráfico de linha")
13 plt.legend()
14 plt.show()
```

The output cell displays a line graph titled "Line Graph Example". The x-axis is labeled "x axis" and ranges from 1 to 9. The y-axis is labeled "y axis" and ranges from 1 to 6. Two lines are plotted: "line L" (blue line) and "line H" (orange line). Both lines follow a similar triangular pattern, peaking at x=3 and x=7, and dipping at x=5.



## Índice

## Charting in Colaboratory

- Matplotlib
  - Line Plots
- {x} Bar Plots
- Histograms

- Scatter Plots
- Stack Plots

- Pie Charts

- fill\_between and alpha

- Subplotting using Subplot2grid

- Plot styles

- 3D Graphs

- 3D Scatter Plots

- 3D Bar Plots

- Wireframe Plots

- Seaborn

- Altair

- Plotly

- Sample

- Bokeh

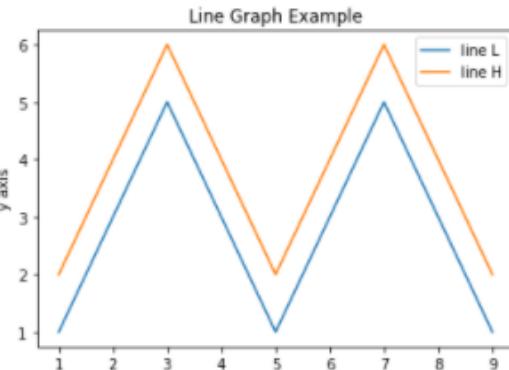
- Sample

- Seção

[+ Código](#) [+ Texto](#) [Copiar para o Drive](#)[Conectar](#)  [Editar](#)

## Line Plots

```
1 import matplotlib.pyplot as plt
2
3 x  = [1, 2, 3, 4, 5, 6, 7, 8, 9]
4 y1 = [1, 3, 5, 3, 1, 3, 5, 3, 1]
5 y2 = [2, 4, 6, 4, 2, 4, 6, 4, 2]
6 plt.plot(x, y1, label="line L")
7 plt.plot(x, y2, label="line H")
8 plt.plot()
9
10 plt.xlabel("x axis")
11 plt.ylabel("y axis")
12 plt.title("Line Graph Example")
13 plt.legend()
14 plt.show()
```





## Índice

Charting in Colaboratory

Matplotlib

Line Plots

Bar Plots

Histograms

Scatter Plots

Stack Plots

Pie Charts

fill\_between and alpha

Subplotting using Subplot2grid

Plot styles

3D Graphs

3D Scatter Plots

3D Bar Plots

Wireframe Plots

Seaborn

Altair

Plotly

Sample

Bokeh

Sample

## + Seção



+ Código

+ Texto

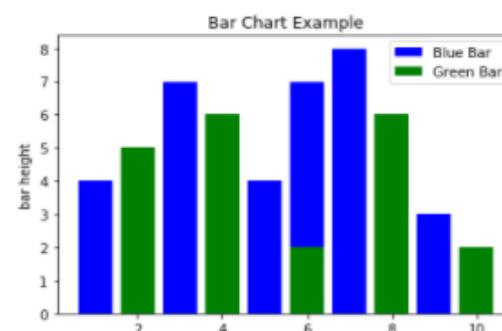
Copiar para o Drive

Conectar

Editar



```
1 import matplotlib.pyplot as plt
2
3 # Look at index 4 and 6, which demonstrate overlapping cases.
4 x1 = [1, 3, 4, 5, 6, 7, 9]
5 y1 = [4, 7, 2, 4, 7, 8, 3]
6
7 x2 = [2, 4, 6, 8, 10]
8 y2 = [5, 6, 2, 6, 2]
9
10 # Colors: https://matplotlib.org/api/colors\_api.html
11
12 plt.bar(x1, y1, label="Blue Bar", color='b')
13 plt.bar(x2, y2, label="Green Bar", color='g')
14 plt.plot()
15
16 plt.xlabel("bar number")
17 plt.ylabel("bar height")
18 plt.title("Bar Chart Example")
19 plt.legend()
20 plt.show()
```





## Índice

Charting in Colaboratory

Matplotlib

&lt;&gt;

Line Plots

{x}

Bar Plots

☰

Histograms

Scatter Plots

Stack Plots

Pie Charts

fill\_between and alpha  
Subplotting using Subplot2grid

Plot styles

3D Graphs

3D Scatter Plots

3D Bar Plots

Wireframe Plots

Seaborn

Altair

Plotly

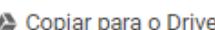
Sample

...



+ Código

+ Texto



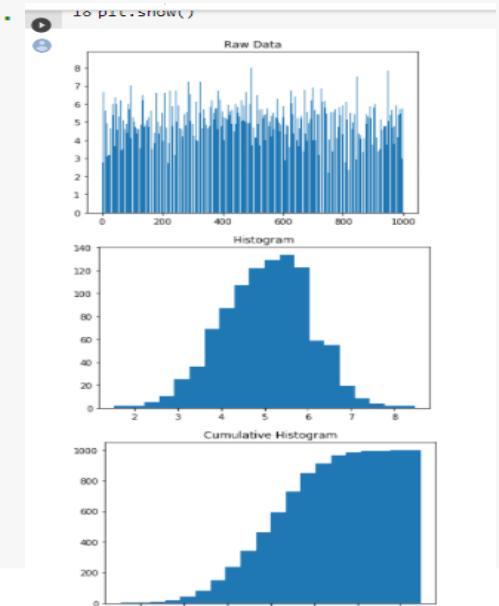
Conectar

Editar

...

## ▼ Histograms

```
[ ] 1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 # Use numpy to generate a bunch of random data in a bell curve around 5.
5 n = 5 + np.random.randn(1000)
6
7 m = [m for m in range(len(n))]
8 plt.bar(m, n)
9 plt.title("Raw Data")
10 plt.show()
11
12 plt.hist(n, bins=20)
13 plt.title("Histogram")
14 plt.show()
15
16 plt.hist(n, cumulative=True, bins=20)
17 plt.title("Cumulative Histogram")
18 plt.show()
```



Raw Data

R



## Índice



Charting in Colaboratory

Matplotlib

Line Plots

Bar Plots

Histograms

Scatter Plots

Stack Plots

Pie Charts

fill\_between and alpha

Subplotting using Subplot2grid

Plot styles

3D Graphs

3D Scatter Plots

3D Bar Plots

Wireframe Plots

Seaborn

Altair

Plotly

Sample

Bokeh

Sample

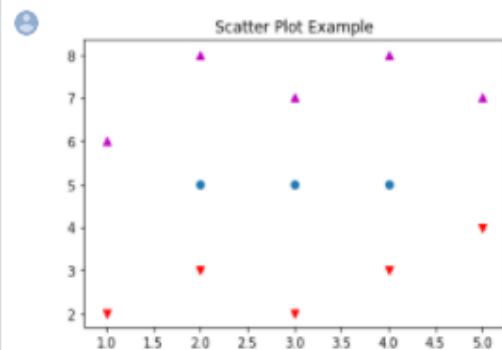
Seção

+ Código + Texto Copiar para o Drive

Conectar

## ▼ Scatter Plots

```
1 import matplotlib.pyplot as plt
2
3 x1 = [2, 3, 4]
4 y1 = [5, 5, 5]
5
6 x2 = [1, 2, 3, 4, 5]
7 y2 = [2, 3, 2, 3, 4]
8 y3 = [6, 8, 7, 8, 7]
9 # Markers: https://matplotlib.org/api/markers\_api.html
10
11 plt.scatter(x1, y1)
12 plt.scatter(x2, y2, marker='v', color='r')
13 plt.scatter(x2, y3, marker='^', color='m')
14 plt.title('Scatter Plot Example')
15 plt.show()
```



## Índice

## Charting in Colaboratory

## Matplotlib

Line Plots

Bar Plots

Histograms

## Scatter Plots

Stack Plots

Pie Charts

fill\_between and alpha

Subplotting using Subplot2grid

## Plot styles

## 3D Graphs

3D Scatter Plots

3D Bar Plots

Wireframe Plots

## Seaborn

## Altair

## Plotly

Sample

## Bokeh

Sample

## Seção

+ Código + Texto Copiar para o Drive

Conectar | Editar



## ▼ Stack Plots

```
[ ] 1 import matplotlib.pyplot as plt
2
3 idxes = [ 1, 2, 3, 4, 5, 6, 7, 8, 9]
4 arr1 = [23, 40, 28, 43, 8, 44, 43, 18, 17]
5 arr2 = [17, 30, 22, 14, 17, 17, 29, 22, 30]
6 arr3 = [15, 31, 18, 22, 18, 19, 13, 32, 39]
7
8 # Adding legend for stack plots is tricky.
9 plt.plot([], [], color='r', label = 'D 1')
10 plt.plot([], [], color='g', label = 'D 2')
11 plt.plot([], [], color='b', label = 'D 3')
12
13 plt.stackplot(idxes, arr1, arr2, arr3, colors= ['r', 'g', 'b'])
14 plt.title('Stack Plot Example')
15 plt.legend()
16 plt.show()
```



## ▼ Pie Charts

# CO Charts in Colaboratory

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Não é possível salvar as alterações

Compartilhar

## Índice

Charting in Colaboratory

Matplotlib

Line Plots

Bar Plots

Histograms

Scatter Plots

Stack Plots

Pie Charts

fill\_between and alpha

Subplotting using Subplot2grid

Plot styles

3D Graphs

3D Scatter Plots

3D Bar Plots

Wireframe Plots

Seaborn

Altair

Plotly

Sample

Bokeh

Sample

Seção



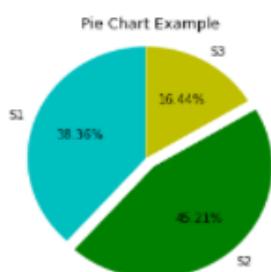
Conectar

Editar



## Pie Charts

```
1 import matplotlib.pyplot as plt
2
3 labels = 'S1', 'S2', 'S3'
4 sections = [56, 66, 24]
5 colors = ['c', 'g', 'y']
6
7 plt.pie(sections, labels=labels, colors=colors,
8         startangle=90,
9         explode = (0, 0.1, 0),
10        autopct = '%1.2f%%')
11
12 plt.axis('equal') # Try commenting this out.
13 plt.title('Pie Chart Example')
14 plt.show()
```



## fill between and alpha



Índice

- Charting in Colaboratory
- Matplotlib
  - Line Plots
  - Bar Plots
  - Histograms
  - Scatter Plots
  - Stack Plots**
  - Pie Charts
  - fill\_between and alpha
  - Subplotting using Subplot2grid
- Plot styles
- 3D Graphs
  - 3D Scatter Plots
  - 3D Bar Plots
  - Wireframe Plots
- Seaborn
- Altair
- Plotly
  - Sample
- Bokeh
  - Sample
- Seção

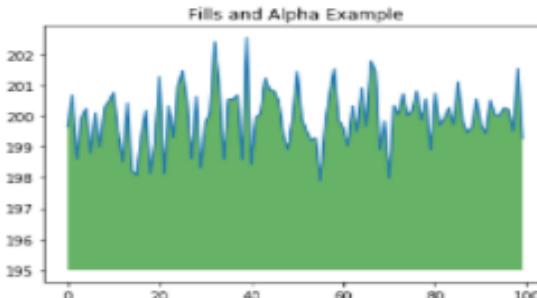
+ Código + Texto | Copiar para o Drive

[ ]

52

▼ fill\_between and alpha

```
[ ] 1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 ys = 200 + np.random.randn(100)
5 x = [x for x in range(len(ys))]
6
7 plt.plot(x, ys, '-')
8 plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)
9
10 plt.title("Fills and Alpha Example")
11 plt.show()
```

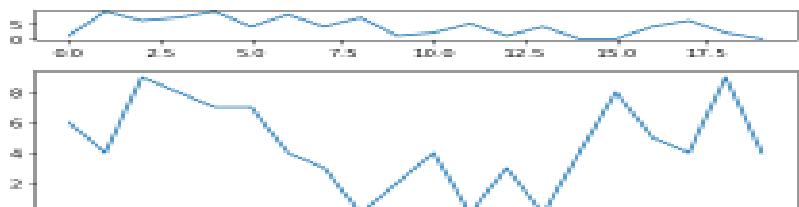


▼ Subplotting using Subplot2grid

```
[ ] 1 import matplotlib.pyplot as plt
2 import numpy as np
3
```

## Subplotting using Subplot2grid

```
 1 import matplotlib.pyplot as plt
 2 import numpy as np
 3
 4 def random_plots():
 5     xs = []
 6     ys = []
 7
 8     for i in range(20):
 9         x = i
10         y = np.random.randint(10)
11
12         xs.append(x)
13         ys.append(y)
14
15     return xs, ys
16
17 fig = plt.figure()
18 ax1 = plt.subplot2grid((5, 2), (0, 0), rowspan=1, colspan=2)
19 ax2 = plt.subplot2grid((5, 2), (1, 0), rowspan=3, colspan=2)
20 ax3 = plt.subplot2grid((5, 2), (4, 0), rowspan=1, colspan=1)
21 ax4 = plt.subplot2grid((5, 2), (4, 1), rowspan=1, colspan=1)
22
23 x, y = random_plots()
24 ax1.plot(x, y)
25
26 x, y = random_plots()
27 ax2.plot(x, y)
28
29 x, y = random_plots()
30 ax3.plot(x, y)
31
32 x, y = random_plots()
33 ax4.plot(x, y)
34
35 plt.tight_layout()
36 plt.show()
```



## Charts in Colaboratory

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Não é possível salvar as alterações

### Índice

#### Charting In Colaboratory

##### Matplotlib

###### Line Plots

###### Bar Plots

###### Histograms

###### Scatter Plots

###### Stack Plots

###### Pie Charts

###### fill\_between and alpha

#### Subplotting using Subplot2grid

##### Plot styles

##### 3D Graphs

###### 3D Scatter Plots

###### 3D Bar Plots

###### Wireframe Plots

##### Seaborn

##### Altair

##### Plotly

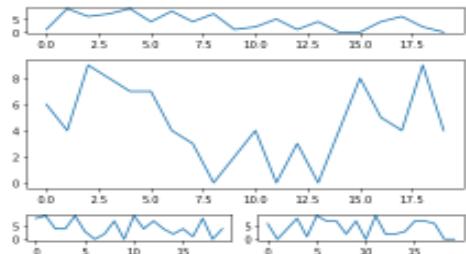
###### Sample

##### Bokeh

###### Sample

#### Sepção

```
14
15     return xs, ys
16
17 fig = plt.figure()
18 ax1 = plt.subplot2grid((5, 2), (0, 0), rowspan=1, colspan=2)
19 ax2 = plt.subplot2grid((5, 2), (1, 0), rowspan=3, colspan=2)
20 ax3 = plt.subplot2grid((5, 2), (4, 0), rowspan=1, colspan=1)
21 ax4 = plt.subplot2grid((5, 2), (4, 1), rowspan=1, colspan=1)
22
23 x, y = random_plots()
24 ax1.plot(x, y)
25
26 x, y = random_plots()
27 ax2.plot(x, y)
28
29 x, y = random_plots()
30 ax3.plot(x, y)
31
32 x, y = random_plots()
33 ax4.plot(x, y)
34
35 plt.tight_layout()
36 plt.show()
```



### Plot styles

Colaboratory charts use [Seaborn's](#) custom styling by default. To customize styling further please see the [matplotlib docs](#).

#### ▼ 3D Graphs

#### ▼ 3D Scatter Plots

```
[ ] 1 import matplotlib.pyplot as plt
2 import numpy as np
3 from mpl_toolkits.mplot3d import Axes3D
```



Índice

- Charting in Colaboratory
- Matplotlib
  - Line Plots
  - Bar Plots
  - Histograms
- Scatter Plots
  - Stack Plots
  - Pie Charts
  - fill\_between and alpha
  - Subplotting using Subplot2grid
- Plot styles
- 3D Graphs
  - 3D Scatter Plots
  - 3D Bar Plots
  - Wireframe Plots
- Seaborn
- Altair
- Plotly
  - Sample
- Bokeh
  - Sample
- Seção

+ Código + Texto Copiar para o Drive

Conectar Editar

▼ 3D Graphs

▼ 3D Scatter Plots

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 from mpl_toolkits.mplot3d import axes3d
4
5 fig = plt.figure()
6 ax = fig.add_subplot(111, projection = '3d')
7
8 x1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
9 y1 = np.random.randint(10, size=10)
10 z1 = np.random.randint(10, size=10)
11
12 x2 = [-1, -2, -3, -4, -5, -6, -7, -8, -9, -10]
13 y2 = np.random.randint(-10, 0, size=10)
14 z2 = np.random.randint(10, size=10)
15
16 ax.scatter(x1, y1, z1, c='b', marker='o', label='blue')
17 ax.scatter(x2, y2, z2, c='g', marker='D', label='green')
18
19 ax.set_xlabel('x axis')
20 ax.set_ylabel('y axis')
21 ax.set_zlabel('z axis')
22 plt.title("3D Scatter Plot Example")
23 plt.legend()
24 plt.tight_layout()
25 plt.show()
```

## Índice

Charting in Colaboratory

Matplotlib

Line Plots

Bar Plots

Histograms

Scatter Plots

Stack Plots

Pie Charts

fill\_between and alpha

Subplotting using Subplot2grid

Plot styles

3D Graphs

### 3D Scatter Plots

3D Bar Plots

Wireframe Plots

Seaborn

Altair

Plotly

Sample

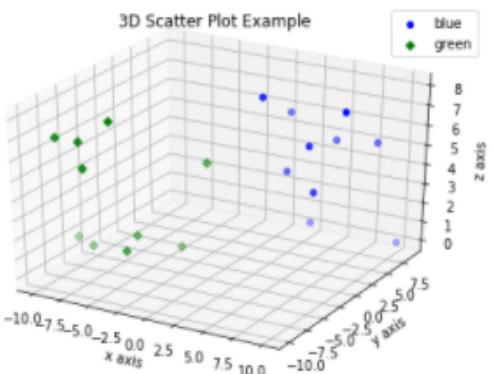
Bokeh

Sample

+ Seção

+ Código + Texto Copiar para o Drive

```
17 ax.scatter(x2, y2, z2, c='g', marker='D', label='green')
18
19 ax.set_xlabel('x axis')
20 ax.set_ylabel('y axis')
21 ax.set_zlabel('z axis')
22 plt.title("3D Scatter Plot Example")
23 plt.legend()
24 plt.tight_layout()
25 plt.show()
```



### 3D Bar Plots

```
[ ] 1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure()
5 ax = fig.add_subplot(111, projection = '3d')
```



## Índice

Charting in Colaboratory

Matplotlib

Line Plots

Bar Plots

Histograms

Scatter Plots

Stack Plots

Pie Charts

fill\_between and alpha

Subplotting using Subplot2grid

Plot styles

3D Graphs

3D Scatter Plots

3D Bar Plots

Wireframe Plots

Seaborn

Altair

Plotly

Sample

Bokeh

Sample

+ Código + Texto Copiar para o Drive

Conectar Editar

## 3D Bar Plots

```
[ ] 1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure()
5 ax = fig.add_subplot(111, projection = '3d')
6
7 x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
8 y = np.random.randint(10, size=10)
9 z = np.zeros(10)
10
11 dx = np.ones(10)
12 dy = np.ones(10)
13 dz = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
14
15 ax.bar3d(x, y, z, dx, dy, dz, color='g')
16
17 ax.set_xlabel('x axis')
18 ax.set_ylabel('y axis')
19 ax.set_zlabel('z axis')
20 plt.title("3D Bar Chart Example")
21 plt.tight_layout()
22 plt.show()
```

## Índice



Charting in Colaboratory

Matplotlib

Line Plots

Bar Plots

Histograms

Scatter Plots

Stack Plots

Pie Charts

fill\_between and alpha

Subplotting using Subplot2grid

Plot styles

3D Graphs

## 3D Scatter Plots

3D Bar Plots

Wireframe Plots

Seaborn

Altair

Plotly

Sample

Bokeh

Sample

## Seção

+ Código

+ Texto

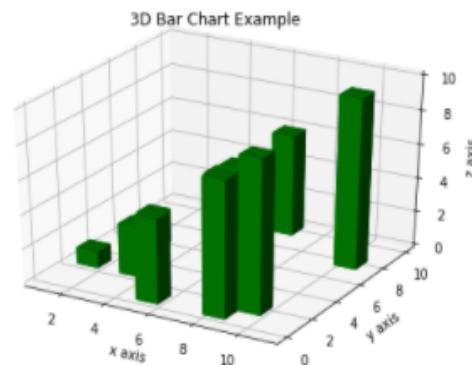
Copiar para o Drive

```
19 ax.set_zlabel( z axis )
[ ] 20 plt.title("3D Bar Chart Example")
21 plt.tight_layout()
22 plt.show()
```

Conectar

Editar

^



```
[ ] 1 import matplotlib.pyplot as plt
2
3 fig = plt.figure()
4 ax = fig.add_subplot(111, projection = '3d')
5
6 x, y, z = axes3d.get_test_data()
7
8 ax.plot_wireframe(x, y, z, rstride = 2, cstride = 2)
9
10 plt.title("Wireframe Plot Example")
11 plt.tight_layout()
```

## Índice

Charting in Colaboratory

Matplotlib

Line Plots

Bar Plots

Histograms

Scatter Plots

Stack Plots

Pie Charts

fill\_between and alpha

Subplotting using Subplot2grid

Plot styles

3D Graphs

3D Scatter Plots

3D Bar Plots

Wireframe Plots

Seaborn

Altair

Plotly

Sample

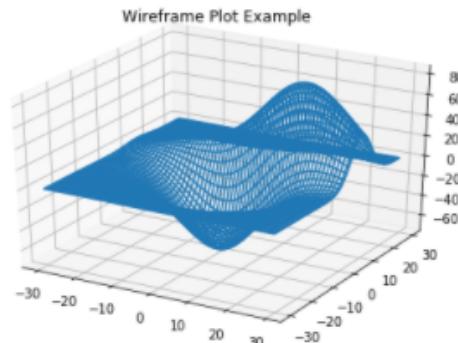
Bokeh

Sample

Seção

## Wireframe Plots

```
[ ] 1 import matplotlib.pyplot as plt
2
3 fig = plt.figure()
4 ax = fig.add_subplot(111, projection = '3d')
5
6 x, y, z = axes3d.get_test_data()
7
8 ax.plot_wireframe(x, y, z, rstride = 2, cstride = 2)
9
10 plt.title("Wireframe Plot Example")
11 plt.tight_layout()
12 plt.show()
```



## Seaborn



## Índice



## Charting in Colaboratory

Matplotlib

Line Plots

Bar Plots

Histograms

Scatter Plots

Stack Plots

Pie Charts

fill\_between and alpha

Subplotting using Subplot2grid

Plot styles

3D Graphs

## 3D Scatter Plots

3D Bar Plots

Wireframe Plots

Seaborn

Altair

Plotly

Sample

Bokeh

Sample

+ Seção

+ Código + Texto Copiar para o Drive

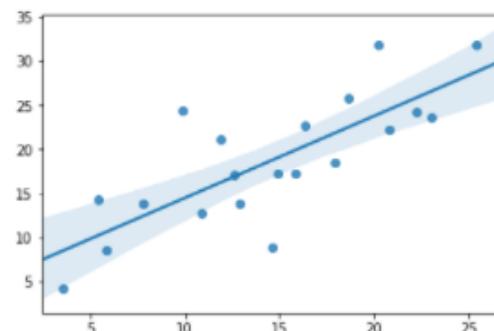
20 30 -30

Conectar | Editar |

## ▼ Seaborn

There are several libraries layered on top of Matplotlib that you can use in Colab. One that is worth highlighting is [Seaborn](#):

```
[ ] 1 import matplotlib.pyplot as plt
2 import numpy as np
3 import seaborn as sns
4
5 # Generate some random data
6 num_points = 20
7 # x will be 5, 6, 7... but also twiddled randomly
8 x = 5 + np.arange(num_points) + np.random.randn(num_points)
9 # y will be 10, 11, 12... but twiddled even more randomly
10 y = 10 + np.arange(num_points) + 5 * np.random.randn(num_points)
11 sns.regplot(x, y)
12 plt.show()
```



## Índice

Charting in Colaboratory

Matplotlib

Line Plots

Bar Plots

Histograms

Scatter Plots

Stack Plots

Pie Charts

fill\_between and alpha

Subplotting using Subplot2grid

Plot styles

3D Graphs

**3D Scatter Plots**

3D Bar Plots

Wireframe Plots

Seaborn

Altair

Plotly

Sample

Bokeh

Sample

+ Seção

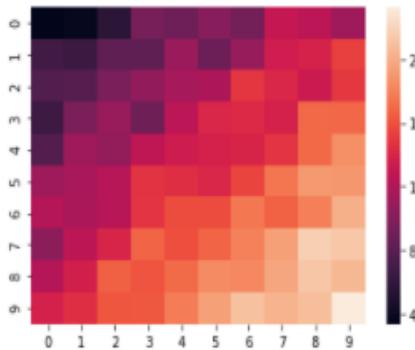
+ Código + Texto Copiar para o Drive

Conectar | Editar

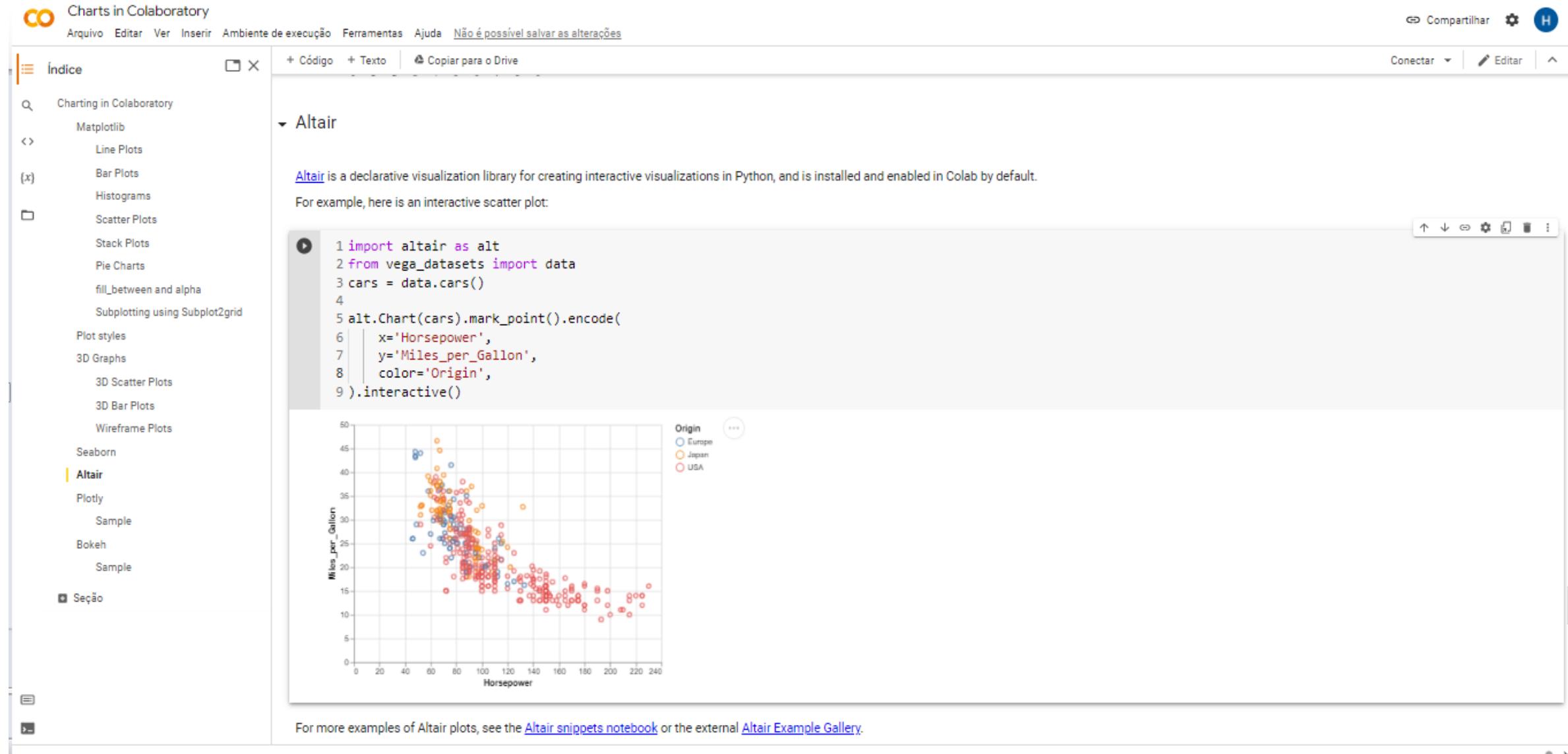
That's a simple scatterplot with a nice regression line fit to it, all with just one call to Seaborn's [regplot](#).

Here's a Seaborn [heatmap](#):

```
[ ] 1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 # Make a 10 x 10 heatmap of some random data
5 side_length = 10
6 # Start with a 10 x 10 matrix with values randomized around 5
7 data = 5 + np.random.randn(side_length, side_length)
8 # The next two lines make the values larger as we get closer to (9, 9)
9 data += np.arange(side_length)
10 data += np.reshape(np.arange(side_length), (side_length, 1))
11 # Generate the heatmap
12 sns.heatmap(data)
13 plt.show()
```



Altair



## Índice

Charting in Colaboratory

Matplotlib

Line Plots

Bar Plots

Histograms

Scatter Plots

Stack Plots

Pie Charts

fill\_between and alpha

Subplotting using Subplot2grid

Plot styles

3D Graphs

3D Scatter Plots

3D Bar Plots

Wireframe Plots

Seaborn

Altair

Plotly

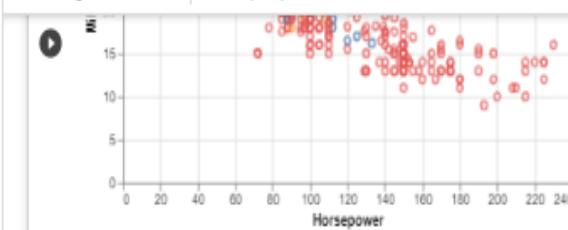
Sample

Bokeh

Sample

Seção

+ Código + Texto Copiar para o Drive



Conectar | Editar

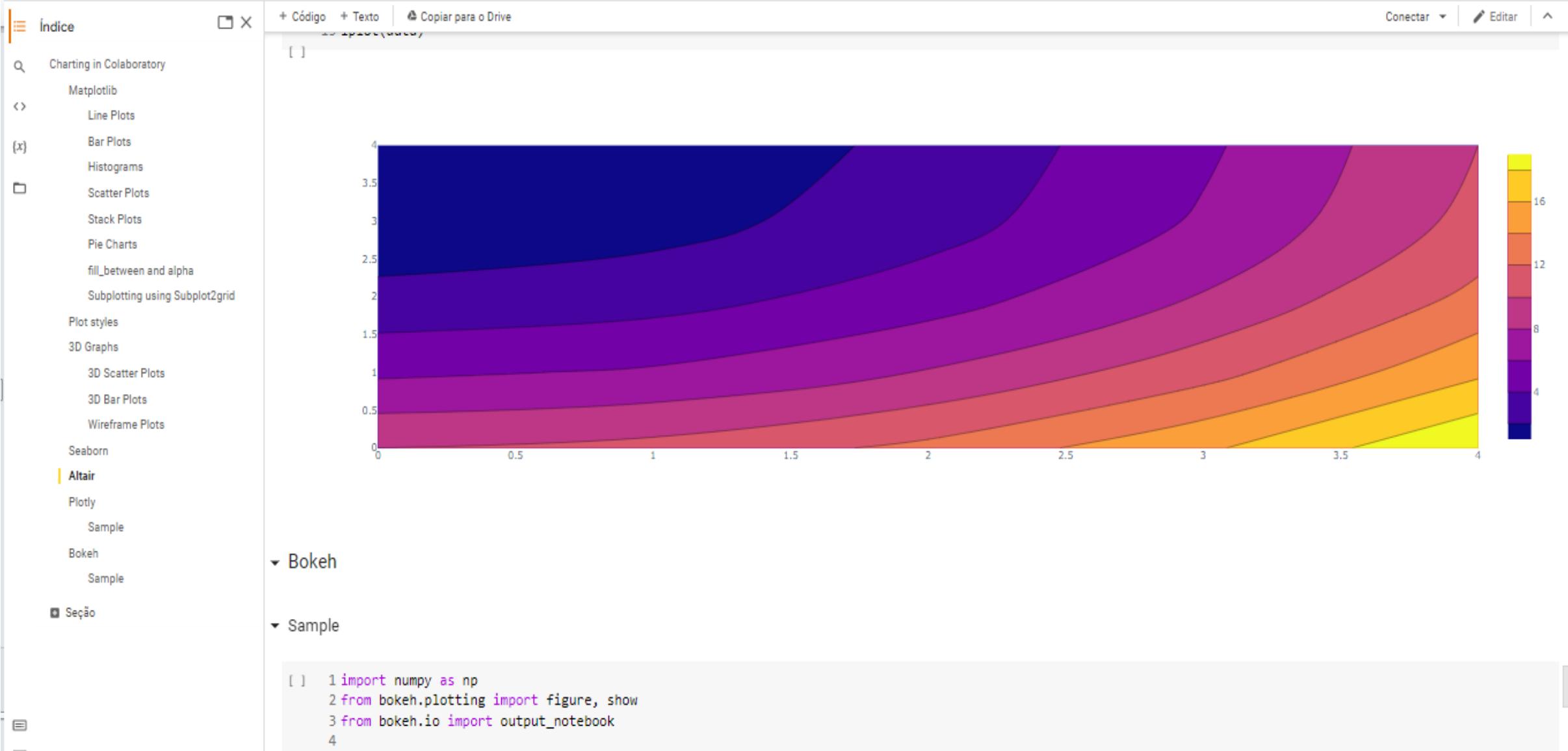


For more examples of Altair plots, see the [Altair snippets notebook](#) or the external [Altair Example Gallery](#).

### Plotly

### Sample

```
[ ] 1 from plotly.offline import iplot
2 import plotly.graph_objs as go
3
4 data = [
5     go.Contour(
6         z=[[10, 10.625, 12.5, 15.625, 20],
7          [5.625, 6.25, 8.125, 11.25, 15.625],
8          [2.5, 3.125, 5., 8.125, 12.5],
9          [0.625, 1.25, 3.125, 6.25, 10.625],
10         [0, 0.625, 2.5, 5.625, 10]]
11     )
12 ]
13 iplot(data)
```





Charts in Colaboratory

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Não é possível salvar as alterações Compartilhar Configurações

Índice + Código + Texto Copiar para o Drive Conectar Editar

Charting in Colaboratory

- Matplotlib
- Line Plots
- Bar Plots
- Histograms
- Scatter Plots
- Stack Plots
- Pie Charts
- fill\_between and alpha
- Subplotting using Subplot2grid
- Plot styles
- 3D Graphs
- 3D Scatter Plots
- 3D Bar Plots
- Wireframe Plots
- Seaborn
- Altair
- Plotly
- Sample
- Bokeh
- Sample

Seção

```
[1]: 4 radii = np.random.random(size=N) * 1.5
[2]: 5 colors = ["#%02x%02x%02x" % (r, g, 150) for r, g in zip(np.floor(50+2*x).astype(int), np.floor(30+2*y).astype(int))]
[3]:
[4]: p = figure()
[5]: p.circle(x, y, radius=radii, fill_color=colors, fill_alpha=0.6, line_color=None)
[6]: show(p)
```

A scatter plot showing a 100x100 grid of colored circles. The color of each circle is determined by its coordinates (x, y) using a hex color code. The color gradient transitions from purple at the bottom left to yellow at the top right, with a higher density of points in the center.

## **Sistema de arquivos local**



## Índice

Sistema de arquivos local

**Fazer upload de arquivos a partir de sistema de arquivos local**

Fazer download de arquivos para o sistema de arquivos local

Google Drive

Montar o Google Drive localmente

PyDrive

API REST do Drive

Criar um novo arquivo do Drive com dados do Python

Fazer download de dados de um arquivo do Drive para Python

Planilhas Google

Criar uma nova planilha com dados do Python

Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

Google Cloud Storage (GCS)

+ Código + Texto Copiar para o Drive

Conectar

Editar

Este notebook contém instruções para carregar e salvar dados de fontes externas.

## Sistema de arquivos local

### Fazer upload de arquivos a partir de sistema de arquivos local

`files.upload` retorna um dicionário com os arquivos incluídos no upload. O dicionário é indexado com o nome do arquivo e os valores são os dados enviados por upload.

```
[ ] 1 from google.colab import files  
2  
3 uploaded = files.upload()  
4  
5 for fn in uploaded.keys():  
6   print('User uploaded file "{name}" with length {length} bytes'.format(  
7     name=fn, length=len(uploaded[fn])))
```

### Fazer download de arquivos para o sistema de arquivos local



## Índice

### Sistema de arquivos local

Fazer upload de arquivos a partir de sistema de arquivos local

Fazer download de arquivos para o sistema de arquivos local

Google Drive

Montar o Google Drive localmente

PyDrive

API REST do Drive

Criar um novo arquivo do Drive com dados do Python

Fazer download de dados de um arquivo do Drive para Python

Planilhas Google

Criar uma nova planilha com dados do Python

Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

Google Cloud Storage (GCS)

+ Código + Texto Copiar para o Drive

Conectar

Editar

### Fazer download de arquivos para o sistema de arquivos local

`files.download` invocará o download do arquivo pelo navegador para seu computador local.

```
[ ] 1 from google.colab import files  
2  
3 with open('example.txt', 'w') as f:  
4     f.write('some content')  
5  
6 files.download('example.txt')
```

### Google Drive

É possível acessar os arquivos no Drive de várias maneiras. Por exemplo:

- Montar o Google Drive na máquina virtual do ambiente de execução
- Usar um wrapper ao redor da API, [como o PyDrive](#)
- Usar a [API REST nativa](#)

Veja abaixo exemplos de cada um eles.



## Índice



## Sistema de arquivos local

Fazer upload de arquivos a partir de sistema de arquivos local

Fazer download de arquivos para o sistema de arquivos local

## Google Drive

Montar o Google Drive localmente

## PyDrive

## API REST do Drive

Criar um novo arquivo do Drive com dados do Python

Fazer download de dados de um arquivo do Drive para Python

## Planilhas Google

Criar uma nova planilha com dados do Python

Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

## Google Cloud Storage (GCS)

+ Código

+ Texto

Copiar para o Drive

Conectar

Editar



## ▼ Montar o Google Drive localmente

O exemplo abaixo mostra como montar o Google Drive no seu ambiente de execução usando um código de autorização, além de como gravar e ler arquivos nele. Depois de executado, você verá o novo arquivo (`foo.txt`) no <https://drive.google.com/>.

Isto permite somente ler, gravar e mover arquivos. Para modificar de maneira programática as configurações de compartilhamento ou outros metadados, use uma das opções abaixo.

**Observação:** ao usar o botão "Montar Drive" no navegador de arquivos, não é necessário usar nenhum código de autenticação para notebooks que tenham sido editados somente pelo usuário atual.

1 `from google.colab import drive`  
2 `drive.mount('/content/drive')`

Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0t](https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0t)

Enter your authorization code:

.....

Mounted at /content/drive

```
[ ] 1 with open('/content/drive/My Drive/foo.txt', 'w') as f:  
2 |   f.write('Hello Google Drive!')  
3 |   f.close()
```



## Índice



### Sistema de arquivos local

Fazer upload de arquivos a partir de sistema de arquivos local

Fazer download de arquivos para o sistema de arquivos local

### Google Drive

Montar o Google Drive localmente

#### PyDrive

#### API REST do Drive

Criar um novo arquivo do Drive com dados do Python

Fazer download de dados de um arquivo do Drive para Python

### Planilhas Google

Criar uma nova planilha com dados do Python

Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

### Google Cloud Storage (GCS)

+ Código + Texto Copiar para o Drive

Conectar

Editar

Mounted at /content/drive

```
[ ] 1 with open('/content/drive/My Drive/foo.txt', 'w') as f:  
2 | f.write('Hello Google Drive!')  
3 | !cat /content/drive/My\ Drive/foo.txt
```

Hello Google Drive!

```
[ ] 1 drive.flush_and_unmount()  
2 print('All changes made in this colab session should now be visible in Drive.')
```

All changes made in this colab session should now be visible in Drive.

## PyDrive

Os exemplos abaixo demonstram a autenticação e o upload/download de arquivos usando o PyDrive. Existem mais exemplos disponíveis na [documentação do PyDrive](#).

```
[ ] 1 from pydrive.auth import GoogleAuth  
2 from pydrive.drive import GoogleDrive  
3 from google.colab import auth
```



| Índice  |
|---|
| Sistema de arquivos local   |
| Fazer upload de arquivos a partir de sistema de arquivos local                  |
| Fazer download de arquivos para o sistema de arquivos local                     |
| Google Drive  |
| Montar o Google Drive localmente  |
| PyDrive   |
| API REST do Drive   |
| Criar um novo arquivo do Drive com dados do Python                              |
| Fazer download de dados de um arquivo do Drive para Python                      |
| Planilhas Google  |
| Criar uma nova planilha com dados do Python                                     |
| Fazer o download de dados de uma planilha em Python como um DataFrame da pandas |
| Google Cloud Storage (GCS)  |

[+ Código](#) [+ Texto](#) [Copiar para o Drive](#)[Conectar](#) [Editar](#)

## ▼ PyDrive

Os exemplos abaixo demonstram a autenticação e o upload/download de arquivos usando o PyDrive. Existem mais exemplos disponíveis na [documentação do PyDrive](#).

```
[ ] 1 from pydrive.auth import GoogleAuth  
2 from pydrive.drive import GoogleDrive  
3 from google.colab import auth  
4 from oauth2client.client import GoogleCredentials
```

Autentique e crie o cliente do PyDrive.

```
[ ] 1 auth.authenticate_user()  
2 gauth = GoogleAuth()  
3 gauth.credentials = GoogleCredentials.get_application_default()  
4 drive = GoogleDrive(gauth)
```

Crie e faça upload de um arquivo de texto.

```
[ ] 1 uploaded = drive.CreateFile({'title': 'Sample upload.txt'})
```



## Índice



+ Código + Texto Copiar para o Drive

Conectar

Editar



[ ] 4 drive = GoogleDrive(gauth)

Crie e faça upload de um arquivo de texto.

[ ] 1 uploaded = drive.CreateFile({'title': 'Sample upload.txt'})  
2 uploaded.SetContentString('Sample upload file content')  
3 uploaded.Upload()  
4 print('Uploaded file with ID {}'.format(uploaded.get('id')))

Uploaded file with ID 14vDAdqp7BSCQnoougmgy1BexIr2AQx2T

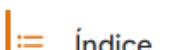
Use o ID do arquivo para carregá-lo e imprima seu conteúdo.

[ ] 1 downloaded = drive.CreateFile({'id': uploaded.get('id')})  
2 print('Downloaded content "{}".format(downloaded.GetContentString())')

Downloaded content "Sample upload file content"

## ▼ API REST do Drive

Para usar a API REST do Drive, primeiro é necessário autenticar e construir um cliente da API.



## Sistema de arquivos local

Fazer upload de arquivos a partir de sistema de arquivos local

Fazer download de arquivos para o sistema de arquivos local

## Google Drive

Montar o Google Drive localmente

## PyDrive

## API REST do Drive

Criar um novo arquivo do Drive com dados do Python

Fazer download de dados de um arquivo do Drive para Python

## Planilhas Google

Criar uma nova planilha com dados do Python

Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

## Google Cloud Storage (GCS)



## Índice

### Sistema de arquivos local

Fazer upload de arquivos a partir de sistema de arquivos local

Fazer download de arquivos para o sistema de arquivos local

### Google Drive

Montar o Google Drive localmente

PyDrive

### API REST do Drive

Criar um novo arquivo do Drive com dados do Python

Fazer download de dados de um arquivo do Drive para Python

### Planilhas Google

Criar uma nova planilha com dados do Python

Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

### Google Cloud Storage (GCS)

+ Código

+ Texto

Copiar para o Drive

Conectar

Editar



Downloaded content "Sample upload file content"

### API REST do Drive

Para usar a API REST do Drive, primeiro é necessário autenticar e construir um cliente da API.

```
[ ] 1 from google.colab import auth  
2 auth.authenticate_user()  
3 from googleapiclient.discovery import build  
4 drive_service = build('drive', 'v3')
```

Com esse cliente, é possível usar todas as funções da [Referência da API do Google Drive](#). Veja os exemplos a seguir.

### Criar um novo arquivo do Drive com dados do Python

Primeiro, crie um arquivo local para fazer upload.

```
[ ] 1 with open('/tmp/to_upload.txt', 'w') as f:  
2     f.write('my sample file')  
3
```



## Índice

- [Sistema de arquivos local](#)
  - Fazer upload de arquivos a partir de sistema de arquivos local
  - Fazer download de arquivos para o sistema de arquivos local
- [Google Drive](#)
  - Montar o Google Drive localmente
  - PyDrive
  - API REST do Drive
    - Criar um novo arquivo do Drive com dados do Python
    - Fazer download de dados de um arquivo do Drive para Python
- [Planilhas Google](#)
  - Criar uma nova planilha com dados do Python
  - Fazer o download de dados de uma planilha em Python como um DataFrame da pandas
- [Google Cloud Storage \(GCS\)](#)

+ Código + Texto Copiar para o Drive

Conectar

Editar



### Criar um novo arquivo do Drive com dados do Python

Primeiro, crie um arquivo local para fazer upload.

```
[ ] 1 with open('/tmp/to_upload.txt', 'w') as f:  
2     f.write('my sample file')  
3  
4 print('/tmp/to_upload.txt contains:')  
5 !cat /tmp/to_upload.txt
```

```
/tmp/to_upload.txt contains:  
my sample file
```

Faça upload usando o método [files.create](#). Mais detalhes sobre como fazer upload de arquivos estão disponíveis na [documentação do desenvolvedor](#).

```
[ ] 1 from googleapiclient.http import MediaFileUpload  
2  
3 file_metadata = {  
4     'name': 'Sample file',  
5     'mimeType': 'text/plain'
```



## Índice

## Sistema de arquivos local

- Fazer upload de arquivos a partir de sistema de arquivos local
- Fazer download de arquivos para o sistema de arquivos local

## Google Drive

- Montar o Google Drive localmente
- PyDrive
- API REST do Drive
- Criar um novo arquivo do Drive com dados do Python
- Fazer download de dados de um arquivo do Drive para Python

## Planilhas Google

- Criar uma nova planilha com dados do Python
- Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

## Google Cloud Storage (GCS)

[+ Código](#) [+ Texto](#)  [Copiar para o Drive](#)[Conectar](#)

Editar

```
[ ] /tmp/to_upload.txt contains:  
my sample file
```

Faça upload usando o método [files.create](#). Mais detalhes sobre como fazer upload de arquivos estão disponíveis na [documentação do desenvolvedor](#).

```
[ ] 1 from googleapiclient.http import MediaFileUpload  
2  
3 file_metadata = {  
4     'name': 'Sample file',  
5     'mimeType': 'text/plain'  
6 }  
7 media = MediaFileUpload('/tmp/to_upload.txt',  
8                         mimetype='text/plain',  
9                         resumable=True)  
10 created = drive_service.files().create(body=file_metadata,  
11                                         media_body=media,  
12                                         fields='id').execute()  
13 print('File ID: {}'.format(created.get('id')))
```

File ID: 1Cw9CqiyU6zbXFD9ViPZu\_3yX-sYF4W17



## Índice

### Sistema de arquivos local

Fazer upload de arquivos a partir de sistema de arquivos local

Fazer download de arquivos para o sistema de arquivos local

### Google Drive

Montar o Google Drive localmente

PyDrive

API REST do Drive

Criar um novo arquivo do Drive com dados do Python

Fazer download de dados de um arquivo do Drive para Python

### Planilhas Google

Criar uma nova planilha com dados do Python

Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

### Google Cloud Storage (GCS)

+ Código + Texto Copiar para o Drive

Conectar

Editar

^

Depois de executar a célula acima, você verá um novo arquivo chamado "Sample file" no <https://drive.google.com/>.

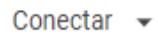
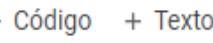
### ▼ Fazer download de dados de um arquivo do Drive para Python

Faça o download do arquivo enviado por upload anteriormente.

```
[ ] 1 file_id = created.get('id')
2
3 import io
4 from googleapiclient.http import MediaIoBaseDownload
5
6 request = drive_service.files().get_media(fileId=file_id)
7 downloaded = io.BytesIO()
8 downloader = MediaIoBaseDownload(downloaded, request)
9 done = False
10 while done is False:
11     # _ is a placeholder for a progress object that we ignore.
12     # (Our file is small, so we skip reporting progress.)
13     _, done = downloader.next_chunk()
14
15 downloaded.seek(0)
```



## Índice



### Sistema de arquivos local

Fazer upload de arquivos a partir de sistema de arquivos local

Fazer download de arquivos para o sistema de arquivos local

Google Drive

Montar o Google Drive localmente

PyDrive

API REST do Drive

Criar um novo arquivo do Drive com dados do Python

Fazer download de dados de um arquivo do Drive para Python

Planilhas Google

Criar uma nova planilha com dados do Python

Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

Google Cloud Storage (GCS)

+ Código + Texto Copiar para o Drive

Fazer download de dados de um arquivo do Drive para Python

```
[ ] 1 file_id = created.get('id')
2
3 import io
4 from googleapiclient.http import MediaIoBaseDownload
5
6 request = drive_service.files().get_media(fileId=file_id)
7 downloaded = io.BytesIO()
8 downloader = MediaIoBaseDownload(downloaded, request)
9 done = False
10 while done is False:
11     # _ is a placeholder for a progress object that we ignore.
12     # (Our file is small, so we skip reporting progress.)
13     _, done = downloader.next_chunk()
14
15 downloaded.seek(0)
16 print('Downloaded file contents are: {}'.format(downloaded.read()))
```

Downloaded file contents are: b'my sample file'



Índice

Sistema de arquivos local

- Fazer upload de arquivos a partir de sistema de arquivos local
- Fazer download de arquivos para o sistema de arquivos local

Google Drive

- Montar o Google Drive localmente
- PyDrive
- API REST do Drive
- Criar um novo arquivo do Drive com dados do Python
- Fazer download de dados de um arquivo do Drive para Python

Planilhas Google

- Criar uma nova planilha com dados do Python
- Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

Google Cloud Storage (GCS)

+ Código + Texto Copiar para o Drive

Fazer download de dados de um arquivo do Drive para Python

Conectar

Editar

^

Faça o download do arquivo enviado por upload anteriormente.

```
[ ] 1 file_id = created.get('id')
2
3 import io
4 from googleapiclient.http import MediaIoBaseDownload
5
6 request = drive_service.files().get_media(fileId=file_id)
7 downloaded = io.BytesIO()
8 downloader = MediaIoBaseDownload(downloaded, request)
9 done = False
10 while done is False:
11     # _ is a placeholder for a progress object that we ignore.
12     # (Our file is small, so we skip reporting progress.)
13     _, done = downloader.next_chunk()
14
15 downloaded.seek(0)
16 print('Downloaded file contents are: {}'.format(downloaded.read()))
```

Downloaded file contents are: b'my sample file'



## Índice



### Sistema de arquivos local

Fazer upload de arquivos a partir de sistema de arquivos local

Fazer download de arquivos para o sistema de arquivos local

Google Drive

Montar o Google Drive localmente

PyDrive

API REST do Drive

Criar um novo arquivo do Drive com dados do Python

Fazer download de dados de um arquivo do Drive para Python

Planilhas Google

Criar uma nova planilha com dados do Python

Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

Google Cloud Storage (GCS)

+ Código + Texto Copiar para o Drive

Conectar

Editar



Para fazer o download de um arquivo diferente, configure o `file_id` acima com o ID desse arquivo, que será similar a este: "1uBtlaggVyWshwcyP6kEI-y\_W3P8D26sz".

## Planilhas Google

Os exemplos abaixo usam a biblioteca [gspread](#) de código aberto para interagir com o Planilhas Google.

Importe a biblioteca, autentique e crie a interface com o Planilhas.

```
[ ] 1 from google.colab import auth  
2 auth.authenticate_user()  
3  
4 import gspread  
5 from google.auth import default  
6 creds, _ = default()  
7  
8 gc = gspread.authorize(creds)
```

Veja abaixo alguns exemplos de `gspread`. Há outros exemplos disponíveis na [página sobre gspread do GitHub](#).



| Índice  |
|---|
| Sistema de arquivos local   |
| Fazer upload de arquivos a partir de sistema de arquivos local                  |
| Fazer download de arquivos para o sistema de arquivos local                     |
| Google Drive  |
| Montar o Google Drive localmente  |
| PyDrive   |
| API REST do Drive   |
| Criar um novo arquivo do Drive com dados do Python                              |
| Fazer download de dados de um arquivo do Drive para Python                      |
| Planilhas Google  |
| Criar uma nova planilha com dados do Python                                     |
| Fazer o download de dados de uma planilha em Python como um DataFrame da pandas |
| Google Cloud Storage (GCS)  |

+ Código + Texto Copiar para o Drive

Conectar

Editar

Veja abaixo alguns exemplos de `gspread`. Há outros exemplos disponíveis na [página sobre `gspread` do GitHub](#).

## ▼ Criar uma nova planilha com dados do Python

```
[ ] 1 sh = gc.create('My cool spreadsheet')
```

Depois de executar a célula acima, você verá uma nova planilha chamada "My cool spreadsheet" no <https://drive.google.com/>.

Abra sua nova planilha e adicione alguns dados aleatórios.

```
[ ] 1 worksheet = gc.open('My cool spreadsheet').sheet1
2
3 cell_list = worksheet.range('A1:C2')
4
5 import random
6 for cell in cell_list:
7     cell.value = random.randint(1, 10)
8
9 worksheet.update_cells(cell_list)
```



## Índice

## Sistema de arquivos local

Fazer upload de arquivos a partir de sistema de arquivos local

Fazer download de arquivos para o sistema de arquivos local

## Google Drive

Montar o Google Drive localmente

## PyDrive

## API REST do Drive

Criar um novo arquivo do Drive com dados do Python

Fazer download de dados de um arquivo do Drive para Python

## Planilhas Google

Criar uma nova planilha com dados do Python

Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

## Google Cloud Storage (GCS)

+ Código + Texto Copiar para o Drive

Conectar

Editar



```
[ ] 7 cell.value = random.randint(1, 10)
8
9 worksheet.update_cells(cell_list)

{'spreadsheetId': '1dsQeN0YzXuM3871_CuyEbsYzL2ew9TJFzR-E-RQnwxs',
 'updatedCells': 6,
 'updatedColumns': 3,
 'updatedRange': 'Sheet1!A1:C2',
 'updatedRows': 2}
```

## ▼ Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

Leia novamente os dados aleatórios inseridos anteriormente e converta o resultado para um [Pandas DataFrame](#).

```
[ ] 1 worksheet = gc.open('My cool spreadsheet').sheet1
2
3 # get_all_values gives a list of rows.
4 rows = worksheet.get_all_values()
5 print(rows)
6
7 import pandas as pd
8 pd.DataFrame.from_records(rows)
```



|     |   |                     |         |
|-----|---|---------------------|---------|
|     | Índice  |                     |         |
|     |   | + Código            | + Texto |
|     |   | Copiar para o Drive |         |
|     | <b>Sistema de arquivos local</b>  |                     |         |
| <>  | Fazer upload de arquivos a partir de sistema de arquivos local                  |                     |         |
| {x} | Fazer download de arquivos para o sistema de arquivos local                     |                     |         |
|     | Google Drive  |                     |         |
|     | Montar o Google Drive localmente  |                     |         |
|     | PyDrive   |                     |         |
|     | API REST do Drive   |                     |         |
|     | Criar um novo arquivo do Drive com dados do Python                              |                     |         |
|     | Fazer download de dados de um arquivo do Drive para Python                      |                     |         |
|     | Planilhas Google  |                     |         |
|     | Criar uma nova planilha com dados do Python                                     |                     |         |
|     | Fazer o download de dados de uma planilha em Python como um DataFrame da pandas |                     |         |
|     | Google Cloud Storage (GCS)  |                     |         |

[ ] 7 `import pandas as pd`

8 `pd.DataFrame.from_records(rows)`

`[['6', '3', '4'], ['7', '2', '1']]`

1 to 2 of 2 entries [Filter](#)

| index | 0 | 1 | 2 |
|-------|---|---|---|
| 0     | 6 | 3 | 4 |
| 1     | 7 | 2 | 1 |

Show [25](#) per page

## ▼ Google Cloud Storage (GCS)

Para usar o Colaboratory com GCS, será preciso criar um [projeto do Google Cloud](#) ou usar um preexistente.

Especifique abaixo o ID do projeto:

[ ] 1 `project_id = 'Your_project_ID_here'`

Os arquivos no GCS são contidos em [buckets](#).

Os buckets precisam ter nomes exclusivos mundialmente. Por isso, vamos gerar um aqui.



| Índice  | X |
|---|---|
| Sistema de arquivos local   |   |
| <> Fazer upload de arquivos a partir de sistema de arquivos local               |   |
| {x} Fazer download de arquivos para o sistema de arquivos local                 |   |
| Google Drive  |   |
| Montar o Google Drive localmente  |   |
| PyDrive   |   |
| API REST do Drive   |   |
| Criar um novo arquivo do Drive com dados do Python                              |   |
| Fazer download de dados de um arquivo do Drive para Python                      |   |
| Planilhas Google  |   |
| Criar uma nova planilha com dados do Python                                     |   |
| Fazer o download de dados de uma planilha em Python como um DataFrame da pandas |   |
| - Google Cloud Storage (GCS)  |   |

+ Código + Texto Copiar para o Drive

[Conectar](#)[Editar](#)

## Google Cloud Storage (GCS)

Para usar o Colaboratory com GCS, será preciso criar um [projeto do Google Cloud](#) ou usar um preexistente.

Especifique abaixo o ID do projeto:

```
[ ] 1 project_id = 'Your_project_ID_here'
```

Os arquivos no GCS são contidos em [buckets](#).

Os buckets precisam ter nomes exclusivos mundialmente. Por isso, vamos gerar um aqui.

```
[ ] 1 import uuid  
2 bucket_name = 'colab-sample-bucket-' + str(uuid.uuid1())
```

Para acessar o GCS, é preciso autenticar.

```
[ ] 1 from google.colab import auth  
2 auth.authenticate_user()
```

O GCS pode ser acessado pelo utilitário `gsutil` da linha de comando ou através da API Python nativa.



| Índice  |
|---|
| Sistema de arquivos local   |
| Fazer upload de arquivos a partir de sistema de arquivos local                  |
| Fazer download de arquivos para o sistema de arquivos local                     |
| Google Drive  |
| Montar o Google Drive localmente  |
| PyDrive   |
| API REST do Drive   |
| Criar um novo arquivo do Drive com dados do Python                              |
| Fazer download de dados de um arquivo do Drive para Python                      |
| Planilhas Google  |
| Criar uma nova planilha com dados do Python                                     |
| Fazer o download de dados de uma planilha em Python como um DataFrame da pandas |
| Google Cloud Storage (GCS)  |

+ Código + Texto Copiar para o Drive

Conectar

Editar

## gsutil

Primeiro, é preciso configurar gsutil para usar o projeto especificado anteriormente usando gcloud.

```
[ ] 1 !gcloud config set project {project_id}
```

Updated property [core/project].

Crie um arquivo local para fazer upload.

```
[ ] 1 with open('/tmp/to_upload.txt', 'w') as f:  
2 | f.write('my sample file')  
3  
4 print('/tmp/to_upload.txt contains:')  
5 !cat /tmp/to_upload.txt
```

/tmp/to\_upload.txt contains:  
my sample file

Faça um bucket para onde você quer fazer o upload ([acesse a documentação aqui](#)).





**Índice**

- Fazer download de arquivos para o sistema de arquivos local
- Google Drive
- Montar o Google Drive localmente
- PyDrive
- API REST do Drive
  - Criar um novo arquivo do Drive com dados do Python
  - Fazer download de dados de um arquivo do Drive para Python
- Planilhas Google
  - Criar uma nova planilha com dados do Python
  - Fazer o download de dados de uma planilha em Python como um DataFrame da pandas
- Google Cloud Storage (GCS)
- gsutil**
  - API do Python
  - Seção

+ Código + Texto Copiar para o Drive Conectar | Editar | ^

Quando o upload for concluído, os dados aparecerão no navegador do armazenamento do Cloud Console do seu projeto:

[Mostrar código](https://console.cloud.google.com/storage/browser?project=Your_project_ID_here)

[https://console.cloud.google.com/storage/browser?project=Your\\_project\\_ID\\_here](https://console.cloud.google.com/storage/browser?project=Your_project_ID_here)

Por último, faça download do arquivo que acaba de ser enviado no exemplo acima. É muito simples: basta inverter a ordem do comando `gsutil cp`.

```
[ ] 1 !gsutil cp gs://{{bucket_name}}/to_upload.txt /tmp/gsutil_download.txt
2
3 # Imprima o resultado para conferir se a transferência funcionou.
4 !cat /tmp/gsutil_download.txt
```

Copying gs://colab-sample-bucket483f20dc-baaf-11e7-ae30-0242ac110002/to\_upload.txt...
/ [1 files][ 14.0 B/ 14.0 B]
Operation completed over 1 objects/14.0 B.
my sample file

▼ API do Python

Estes snippets se baseiam em [um exemplo maior](#) que mostra outros usos da API.



| Índice  |
|---|
| <a href="#">Fazer download de arquivos para o sistema de arquivos local</a>                     |
| <a href="#">Google Drive</a>  |
| <a href="#">Montar o Google Drive localmente</a>  |
| <a href="#">PyDrive</a>   |
| <a href="#">API REST do Drive</a>   |
| <a href="#">Criar um novo arquivo do Drive com dados do Python</a>                              |
| <a href="#">Fazer download de dados de um arquivo do Drive para Python</a>                      |
| <a href="#">Planilhas Google</a>  |
| <a href="#">Criar uma nova planilha com dados do Python</a>                                     |
| <a href="#">Fazer o download de dados de uma planilha em Python como um DataFrame da pandas</a> |
| <a href="#">Google Cloud Storage (GCS)</a>  |
| <a href="#">gsutil</a>  |
| <a href="#">API do Python</a>   |
| <a href="#">+ Seção</a>   |

[+ Código](#) [+ Texto](#) [Copiar para o Drive](#)[Conectar](#)[Editar](#)[^](#)

## ▼ API do Python

Estes snippets se baseiam em [um exemplo maior](#) que mostra outros usos da API.

Primeiro, crie o cliente do serviço.

```
[ ] 1 from googleapiclient.discovery import build  
2 gcs_service = build('storage', 'v1')
```

Crie um arquivo local para fazer upload.

```
[ ] 1 with open('/tmp/to_upload.txt', 'w') as f:  
2 | f.write('my sample file')  
3 |  
4 print('/tmp/to_upload.txt contains:')  
5 !cat /tmp/to_upload.txt
```

```
/tmp/to_upload.txt contains:  
my sample file
```

Crie um bucket no projeto especificado acima.



Índice + Código + Texto Copiar para o Drive Conectar ^

Fazer download de arquivos para o sistema de arquivos local

Google Drive

Montar o Google Drive localmente

PyDrive

API REST do Drive

Criar um novo arquivo do Drive com dados do Python

Fazer download de dados de um arquivo do Drive para Python

Planilhas Google

Criar uma nova planilha com dados do Python

Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

Google Cloud Storage (GCS)

**gsutil**

API do Python

+ Seção

my sample file

Crie um bucket no projeto especificado acima.

```
[ ] 1 # Use um nome de bucket exclusivo mundialmente do exemplo gsutil acima.  
2 import uuid  
3 bucket_name = 'colab-sample-bucket-' + str(uuid.uuid1())  
4  
5 body = {  
6   'name': bucket_name,  
7   # For a full list of locations, see:  
8   # https://cloud.google.com/storage/docs/bucket-locations  
9   'location': 'us',  
10 }  
11 gcs_service.buckets().insert(project=project_id, body=body).execute()  
12 print('Done')
```

Done

Faça upload do arquivo para o bucket recém criado.

```
[ ] 1 from googleapiclient.http import MediaFileUpload
```



|   |
|---|
| Índice  |
| Fazer download de arquivos para o sistema de arquivos local                     |
| Google Drive  |
| Montar o Google Drive localmente  |
| PyDrive   |
| API REST do Drive   |
| Criar um novo arquivo do Drive com dados do Python                              |
| Fazer download de dados de um arquivo do Drive para Python                      |
| Planilhas Google  |
| Criar uma nova planilha com dados do Python                                     |
| Fazer o download de dados de uma planilha em Python como um DataFrame da pandas |
| Google Cloud Storage (GCS)  |
| <b>gsutil</b>   |
| API do Python   |
| + Seção   |

+ Código

+ Texto

Copiar para o Drive

Conectar

Editar

Faça upload do arquivo para o bucket recém criado.

```
[ ] 1 from googleapiclient.http import MediaFileUpload
2
3 media = MediaFileUpload('/tmp/to_upload.txt',
4                         mimetype='text/plain',
5                         resumable=True)
6
7 request = gcs_service.objects().insert(bucket=bucket_name,
8                                         name='to_upload.txt',
9                                         media_body=media)
10
11 response = None
12 while response is None:
13     # _ is a placeholder for a progress object that we ignore.
14     # (Our file is small, so we skip reporting progress.)
15     _, response = request.next_chunk()
16
17 print('Upload complete')
```

Upload complete



|     |   |   |
|-----|---|---|
|     | Índice  | X |
|     | Fazer download de arquivos para o sistema de arquivos local                     |   |
| Q   | Google Drive  |   |
| <>  | Montar o Google Drive localmente  |   |
| {x} | PyDrive   |   |
|     | API REST do Drive   |   |
| 📁   | Criar um novo arquivo do Drive com dados do Python                              |   |
|     | Fazer download de dados de um arquivo do Drive para Python                      |   |
|     | Planilhas Google  |   |
|     | Criar uma nova planilha com dados do Python                                     |   |
|     | Fazer o download de dados de uma planilha em Python como um DataFrame da pandas |   |
|     | Google Cloud Storage (GCS)  |   |
|     | gsutil  |   |
| ☰   | API do Python   |   |
| [-] | + Seção   |   |

+ Código

+ Texto

Copiar para o Drive

Conectar

Editar

^

Faça upload do arquivo para o bucket recém criado.

```
[ ] 1 from googleapiclient.http import MediaFileUpload
2
3 media = MediaFileUpload('/tmp/to_upload.txt',
4                         mimetype='text/plain',
5                         resumable=True)
6
7 request = gcs_service.objects().insert(bucket=bucket_name,
8                                         name='to_upload.txt',
9                                         media_body=media)
10
11 response = None
12 while response is None:
13     # _ is a placeholder for a progress object that we ignore.
14     # (Our file is small, so we skip reporting progress.)
15     _, response = request.next_chunk()
16
17 print('Upload complete')
```

Upload complete



## Índice



Fazer download de arquivos para o sistema de arquivos local



Google Drive



Montar o Google Drive localmente



PyDrive



API REST do Drive



Criar um novo arquivo do Drive com dados do Python



Fazer download de dados de um arquivo do Drive para Python



Planilhas Google



Criar uma nova planilha com dados do Python



Fazer o download de dados de uma planilha em Python como um DataFrame da pandas



Google Cloud Storage (GCS)



gsutil



API do Python



+ Seção

+ Código + Texto Copiar para o Drive

Conectar



Quando o upload for concluído, os dados aparecerão no navegador do armazenamento do Cloud Console do seu projeto:

[Mostrar código](#)[https://console.cloud.google.com/storage/browser?project=Your\\_project\\_ID\\_here](https://console.cloud.google.com/storage/browser?project=Your_project_ID_here)

Faça o download do arquivo que acaba de ser enviado.

```
[ ] 1 from apiclient.http import MediaIoBaseDownload
2
3 with open('/tmp/downloaded_from_gcs.txt', 'wb') as f:
4     request = gcs_service.objects().get_media(bucket=bucket_name,
5                                                 object='to_upload.txt')
6     media = MediaIoBaseDownload(f, request)
7
8     done = False
9     while not done:
10        # _ is a placeholder for a progress object that we ignore.
11        # (Our file is small, so we skip reporting progress.)
12        _, done = media.next_chunk()
13
14 print('Download complete')
```



## Índice



Fazer download de arquivos para o sistema de arquivos local



Google Drive



Montar o Google Drive localmente



PyDrive

API REST do Drive



Criar um novo arquivo do Drive com dados do Python

Fazer download de dados de um arquivo do Drive para Python

Planilhas Google

Criar uma nova planilha com dados do Python

Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

Google Cloud Storage (GCS)

**gsutil**



API do Python



+ Seção

+ Código + Texto Copiar para o Drive

Conectar

Editar



```
[ ] 11 # (Our file is small, so we skip reporting progress.)  
12 _, done = media.next_chunk()  
13  
14 print('Download complete')
```

Download complete

Inspecione o arquivo salvo.

```
[ ] 1 !cat /tmp/downloaded_from_gcs.txt
```

my sample file

## Antes de começar

Use o Gerenciador de Recursos na Nuvem para criar um projeto de plataforma em nuvem se você ainda não tiver um.  
Habilite o faturamento do projeto.

Habilitar o BigQuery APIs para o projeto.

Forneça suas credenciais para o tempo de execução

```
[ ] 1 from google.colab import auth  
2 auth.authenticate_user()  
3 print('Authenticated')
```



Opcional: Habilitar a exibição da tabela de dados

Colab inclui o pacote que pode ser usado para exibir grandes dataframes pandas como uma tabela de dados interativa. Pode ser habilitado com:`google.colab.data_table`

```
[ ] 1 %load_ext google.colab.data_table
```

```
[ ] 1 %load_ext google.colab.data_table
```

## Use BigQuery através de magias

A biblioteca também inclui um comando mágico que executa uma consulta e exibe o resultado ou salva-o para uma variável como um `.google.cloud.bigqueryDataFrame`

The screenshot shows a Jupyter Notebook interface with the title "Getting started with BigQuery". The code cell contains the following BigQuery magic command:

```
1 # Display query output immediately
2
3 %%bigquery --project yourprojectid
4 SELECT
5   COUNT(*) as total_rows
6 FROM `bigquery-public-data.samples.gsod`
```

The resulting DataFrame is displayed below:

| index | total_rows |
|-------|------------|
| 0     | 114420316  |

Below the DataFrame, another code cell shows the same query with annotations:

```
[ ] 1 # Save output in a variable `df`
2
3 %%bigquery --project yourprojectid df
4 SELECT
5   COUNT(*) as total_rows
6 FROM `bigquery-public-data.samples.gsod`
```

Finally, the variable `df` is displayed in the next cell:

```
[ ] 1 df
```



## Getting started with BigQuery

File Edit View Insert Runtime Tools Help

Share



+ Code + Text Copy to Drive

RAM Disk Editing

[ ]

index total\_rows

| index | total_rows |
|-------|------------|
| 0     | 114420316  |

Show 25 per page

### Use BigQuery through google-cloud-bigquery

See [BigQuery documentation](#) and [library reference documentation](#).

The [GSOD sample table](#) contains weather information collected by NOAA, such as precipitation amounts and wind speeds from late 1929 to early 2010.

#### Declare the Cloud project ID which will be used throughout this notebook

```
[ ] 1 project_id = '[your project ID]'
```

#### Sample approximately 2000 random rows

Use o BigQuery através do google-cloud-bigquery

Consulte a documentação do BigQuery e a documentação de referência da biblioteca.

A tabela de amostras GSOD contém informações meteorológicas coletadas pela NOAA, como quantidades de precipitação e velocidades de vento entre o final de 1929 e o início de 2010.

Declare o ID do projeto Cloud que será usado em todo este notebook

```
The GSOD sample table contains weather information collected by NOAA, such as precipitation amounts and wind speeds from late 1929 to early 2010.
```

Declare the Cloud project ID which will be used throughout this notebook

```
1 project_id = '[your project ID]'
```

Sample approximately 2000 random rows

```
[ ] 1 from google.cloud import bigquery
2
3 client = bigquery.Client(project=project_id)
4
5 sample_count = 2000
6 row_count = client.query('''
7     SELECT
8         COUNT(*) as total
9     FROM `bigquery-public-data.samples.gsod` ''').to_dataframe().total[0]
10
```

[+ Code](#) [+ Text](#) [Copy to Drive](#)[▼ Sample approximately 2000 random rows](#)

```
1 from google.cloud import bigquery
2
3 client = bigquery.Client(project=project_id)
4
5 sample_count = 2000
6 row_count = client.query('''
7   SELECT
8     COUNT(*) as total
9   FROM `bigquery-public-data.samples.gsod`''').to_dataframe().total[0]
10
11 df = client.query('''
12   SELECT
13     *
14   FROM
15     `bigquery-public-data.samples.gsod`
16   WHERE RAND() < %d/%d
17 ''' % (sample_count, row_count)).to_dataframe()
18
19 print('Full dataset has %d rows' % row_count)
```

Full dataset has 114420316 rows

[▼ Describe the sampled data](#)



## Getting started with BigQuery

Share

File Edit View Insert Runtime Tools Help

RAM Disk Editing

+ Code

+ Text

Copy to Drive

Full dataset has 114420316 rows



## ▼ Describe the sampled data



{x} [ ] 1 df.describe()

1 to 8 of 8 entries

| index | station_number    | wban_number        | year               | month             | day                | mean_temp         | num_mean_temp_samples | mean_dew_point     | num_mean_dew_point_samples |
|-------|-------------------|--------------------|--------------------|-------------------|--------------------|-------------------|-----------------------|--------------------|----------------------------|
| count | 1979.0            | 1979.0             | 1979.0             | 1979.0            | 1979.0             | 1979.0            | 1979.0                | 1883.0             | 1883.0                     |
| mean  | 505585.599292572  | 89647.44113188479  | 1987.1814047498738 | 6.525517938352704 | 15.715512885295604 | 52.39186460128738 | 13.018696311268318    | 42.01869358592016  | 12.98247477429633          |
| std   | 302491.1873178059 | 27088.238467141528 | 15.993487610677022 | 3.419259475533833 | 8.661562572969782  | 23.32984218910621 | 7.886976768461429     | 21.691901647557273 | 7.91400451168853           |
| min   | 10100.0           | 13.0               | 1933.0             | 1.0               | 1.0                | -91.4000015258789 |                       | 4.0                | -63.79999923706055         |
| 25%   | 238255.0          | 99999.0            | 1978.0             | 4.0               | 8.0                | 39.29999923706055 |                       | 7.0                | 30.34999942779541          |
| 50%   | 538980.0          | 99999.0            | 1990.0             | 7.0               | 16.0               | 55.0              |                       | 8.0                | 44.0                       |
| 75%   | 725273.5          | 99999.0            | 2000.0             | 10.0              | 23.0               | 69.80000305175781 |                       | 23.0               | 56.70000076293945          |
| max   | 999999.0          | 99999.0            | 2010.0             | 12.0              | 31.0               | 105.0999984741211 |                       | 24.0               | 80.5                       |

...  
Show 25 per page

## ▼ View the first 10 rows



{x} [ ] 1 df.head(10)

[+ Code](#) [+ Text](#) [Copy to Drive](#)

RAM Disk Editing

Full dataset has 114420316 rows

[↑](#) [↓](#) [↶](#) [↷](#) [⚙️](#) [🖨️](#) [✖️](#) [⋮](#)

## Describe the sampled data



[ ] 1 df.describe()

1 to 8 of 8 entries [Filter](#) [?](#)

| index | station_number    | wban_number        | year               | month             | day                | mean_temp         | num_mean_temp_samples | mean_dew_point     | num_mean_dew_point_sample |
|-------|-------------------|--------------------|--------------------|-------------------|--------------------|-------------------|-----------------------|--------------------|---------------------------|
| count | 1979.0            | 1979.0             | 1979.0             | 1979.0            | 1979.0             | 1979.0            | 1979.0                | 1883.0             | 1883.0                    |
| mean  | 505585.599292572  | 89647.44113188479  | 1987.1814047498738 | 6.525517938352704 | 15.715512885295604 | 52.39186460128738 | 13.018696311268318    | 42.01869358592016  | 12.98247477429633         |
| std   | 302491.1873178059 | 27088.238467141528 | 15.993487610677022 | 3.419259475533833 | 8.661562572969782  | 23.32984218910621 | 7.886976768461429     | 21.691901647557273 | 7.91400451168853          |
| min   | 10100.0           | 13.0               | 1933.0             | 1.0               | 1.0                | -91.4000015258789 | 4.0                   | -63.79999923706055 | 4.0                       |
| 25%   | 238255.0          | 99999.0            | 1978.0             | 4.0               | 8.0                | 39.29999923706055 | 7.0                   | 30.34999942779541  | 7.0                       |
| 50%   | 538980.0          | 99999.0            | 1990.0             | 7.0               | 16.0               | 55.0              | 8.0                   | 44.0               | 8.0                       |
| 75%   | 725273.5          | 99999.0            | 2000.0             | 10.0              | 23.0               | 69.80000305175781 | 23.0                  | 56.70000076293945  | 23.0                      |
| max   | 999999.0          | 99999.0            | 2010.0             | 12.0              | 31.0               | 105.0999984741211 | 24.0                  | 80.5               | 24.0                      |

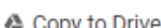
[>Show 25](#) per page

## View the first 10 rows



[ ] 1 df.head(10)

1 to 10 of 10 entries [Filter](#) [?](#)

[+ Code](#)[+ Text](#)

Show 25 per page



[ ] 1 # 10 highest total\_precipitation samples

2 df.sort\_values('total\_precipitation', ascending=False).head(10)[['station\_number', 'year', 'month', 'day', 'total\_precipitation']]



1 to 10 of 10 entries

[Filter](#)

| index | station_number | year | month | day | total_precipitation |
|-------|----------------|------|-------|-----|---------------------|
| 644   | 230220         | 1964 | 7     | 15  | 5.909999847412109   |
| 1155  | 985430         | 2008 | 12    | 8   | 3.4600000381469727  |
| 1196  | 248260         | 1961 | 11    | 1   | 2.950000047683716   |
| 1588  | 257670         | 1959 | 8     | 9   | 2.950000047683716   |
| 980   | 299150         | 1962 | 3     | 1   | 2.950000047683716   |
| 1325  | 470250         | 1965 | 11    | 25  | 2.950000047683716   |
| 1917  | 288380         | 1994 | 8     | 6   | 2.319999933242798   |
| 1211  | 585190         | 1995 | 4     | 14  | 2.319999933242798   |
| 250   | 647000         | 2005 | 8     | 19  | 2.200000047683716   |
| 1418  | 964710         | 1975 | 9     | 8   | 1.9700000286102295  |

Show 25 per page

## Use BigQuery through pandas-gbq

The library is a community led project by the pandas community. It covers basic functionality, such as writing a DataFrame to BigQuery and running a query, but as a third-party library it may not handle all BigQuery features or use cases. [pandas-gbq](#)

[Pandas GBO Documentation](#)

## Use BigQuery através de pandas-gbq

A biblioteca é um projeto liderado pela comunidade pandas. Ele cobre funcionalidades básicas, como escrever um DataFrame para o BigQuery e executar uma consulta, mas como uma biblioteca de terceiros, ele pode não lidar com todos os recursos do BigQuery ou usar casos.pandas-gbq

## Documentação pandas GBQ

The screenshot shows a Jupyter Notebook interface with the title "Getting started with BigQuery". The code cell contains the following Python code:

```
1 import pandas as pd
2
3 sample_count = 2000
4 df = pd.io.gbq.read_gbq('''
5   SELECT name, SUM(number) as count
6   FROM `bigquery-public-data.usa_names.usa_1910_2013`
7   WHERE state = 'TX'
8   GROUP BY name
9   ORDER BY count DESC
10  LIMIT 100
11 ''', project_id=project_id, dialect='standard')
12
13 df.head()
```

The resulting output is a pandas DataFrame:

| index | name    | count  |
|-------|---------|--------|
| 0     | James   | 272793 |
| 1     | John    | 235139 |
| 2     | Michael | 225320 |
| 3     | Robert  | 220399 |
| 4     | David   | 219028 |

At the bottom, there is a note about "Syntax highlighting".

## Destaque da sintaxe

google.colab.syntax pode ser usado para adicionar destaque de sintaxe a quaisquer literais de string Python que são usados em uma consulta mais tarde.

The screenshot shows a Google Colab notebook titled "Getting started with BigQuery". The sidebar on the left includes icons for code, text, search, and syntax highlighting. The main area displays a query in a code cell:

```
1 from google.colab import syntax
2 query = syntax.sql("""
3 SELECT
4   COUNT(*) as total_rows
5 FROM
6   `bigquery-public-data.samples.gsod`
7 """)
8
9 pd.io.gbq.read_gbq(query, project_id=project_id, dialect='standard')
```

The code uses syntax highlighting where Python keywords like `from`, `SELECT`, and `pd` are colored blue, and SQL keywords like `COUNT` and `FROM` are colored red. The resulting output is a table:

| index | total_rows |
|-------|------------|
| 0     | 114420316  |

At the bottom of the code cell, there are buttons for play, stop, and other execution controls. The status bar at the bottom right shows "220559" and "219028".



Olá, este é o Colaboratory

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda

Compartilhar



Índice

Primeiros passos

Ciência de dados

Machine learning

Mais recursos

Exemplos em destaque

Seção

Exemplos

Recente

Google Drive

GitHub

Upload

Conectar

Editar



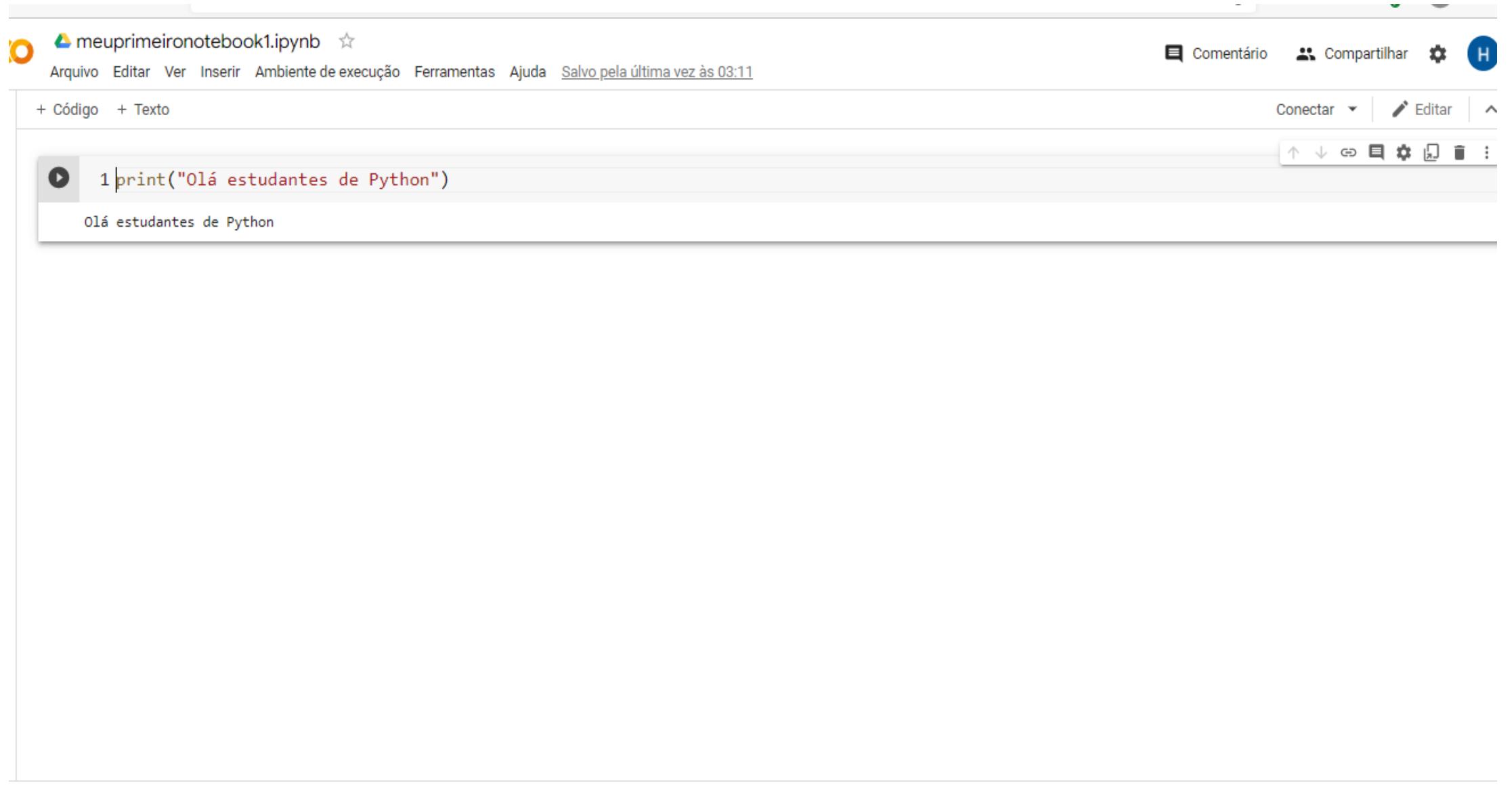
Filtrar notebooks

| Título                            | Aberto pela<br>última vez | Primeiro acesso |  |
|-----------------------------------|---------------------------|-----------------|--|
| Overview of Colaboratory Features | 03:42                     | 03:27           |  |
| Olá, este é o Colaboratory        | 03:41                     | 03:05           |  |
| meuprimeironotebook1.ipynb        | 03:08                     | 03:08           |  |

histórico de código

[Novo notebook](#) [Cancelar](#)

Você pode ser um [estudante](#), um [cientista de dados](#) ou um [pesquisador de IA](#), o Colab pode facilitar seu trabalho. Assista ao vídeo [Introdução ao Colab](#) para saber mais ou simplesmente comece a usá-lo abaixo!



# Instalando o Anaconda

→ ⌂ ⌄ ⌅ https://www.anaconda.com/products/individual

ANACONDA. Products ▾ Pricing Solutions ▾ Resources ▾ Partners ▾ Blog Company ▾ Get Started

Q Individual Edition

# Your data science toolkit

With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with

Anaconda Individual Edition

Download 

For Windows  
Python 3.9 • 64-Bit Graphical Installer • 510 MB

[Get Additional Installers](#)

Nova gu... Atividad... AULA01... Caixa de... python... Download... anacond... Anacond... tutorial... Python... + - □ X

https://www.anaconda.com/products/individual

# Thank you for downloading

## Get Started with Anaconda Individual Edition

Getting Started with Anaconda Individual Edition

Available only in Anaconda Nucleus, this free, hands-on Jupyter notebook provides an introduction to help you get started using this powerful tool.

### Using Anaconda for Commercial Purposes?

Experience guaranteed uptime and additional security features with access to Anaconda's premium repository. Customers also get dedicated support hours and compliance with our Terms of Service. [Learn more.](#)

Downloads

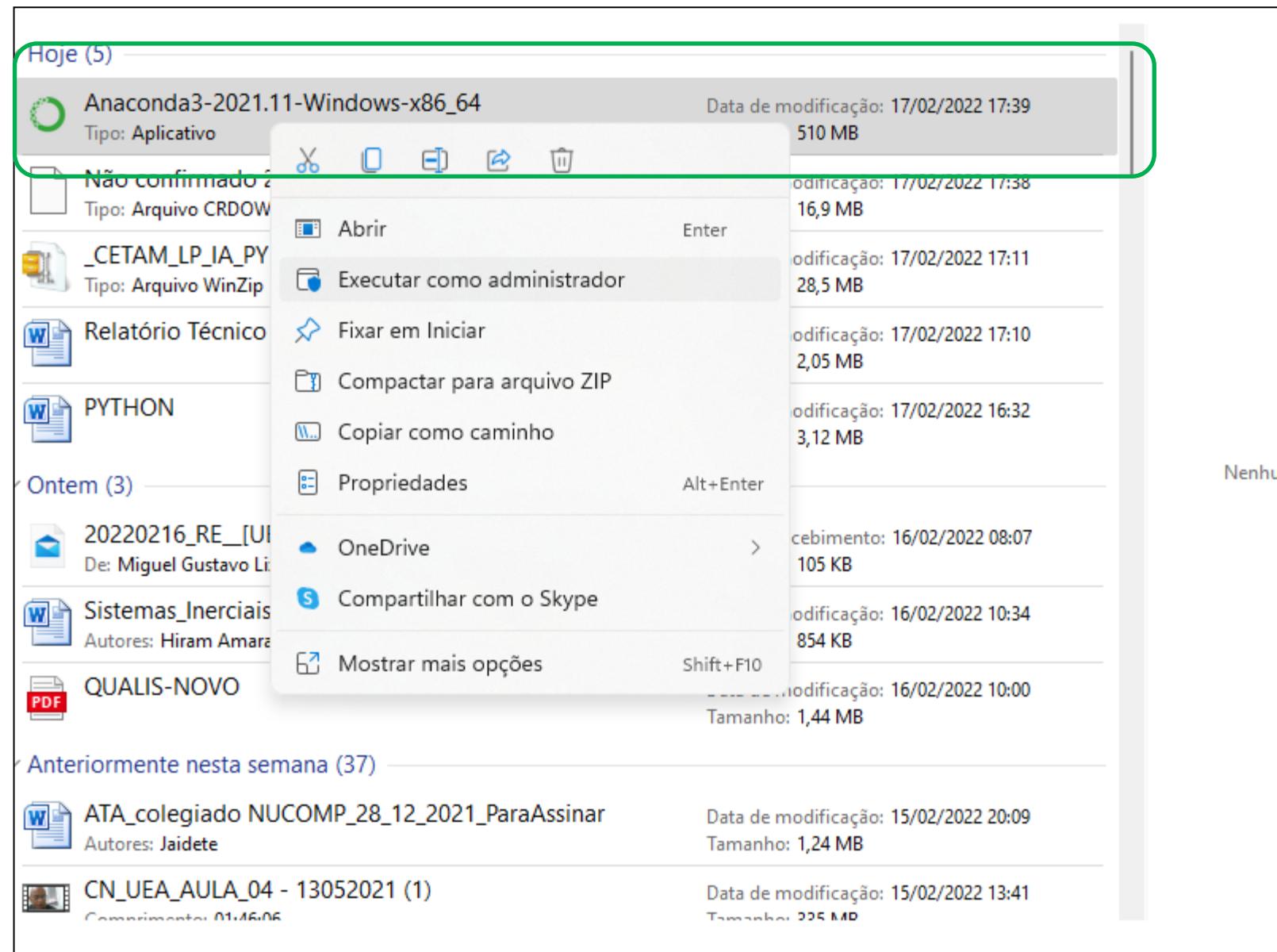
- Anaconda3-2021.11-Windows-x86\_64.exe  
243 KB/s — 1.214 KB de 510 MB, 36 min restantes
- python-3.10.2-amd64.exe  
50,9 KB/s — 10,9 MB de 26,9 MB, 5 min restantes
- \_CETAM\_LP\_IA\_PY\_APONTAMENTOS-20220217T20393...  
Removido
- PYTHON.docx  
[Abrir arquivo](#)

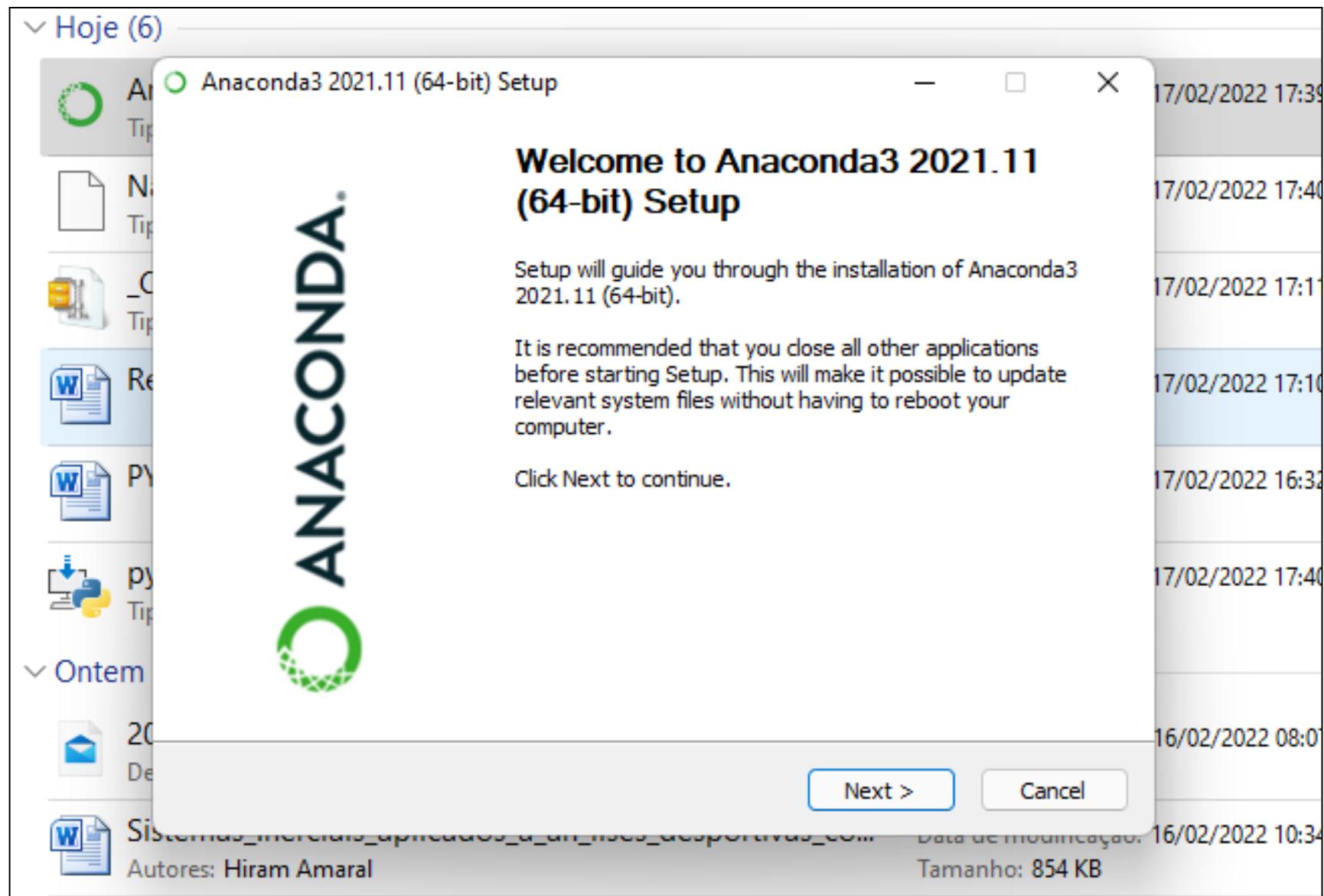
I have read and agree to Anaconda's [Terms of Service](#).\*

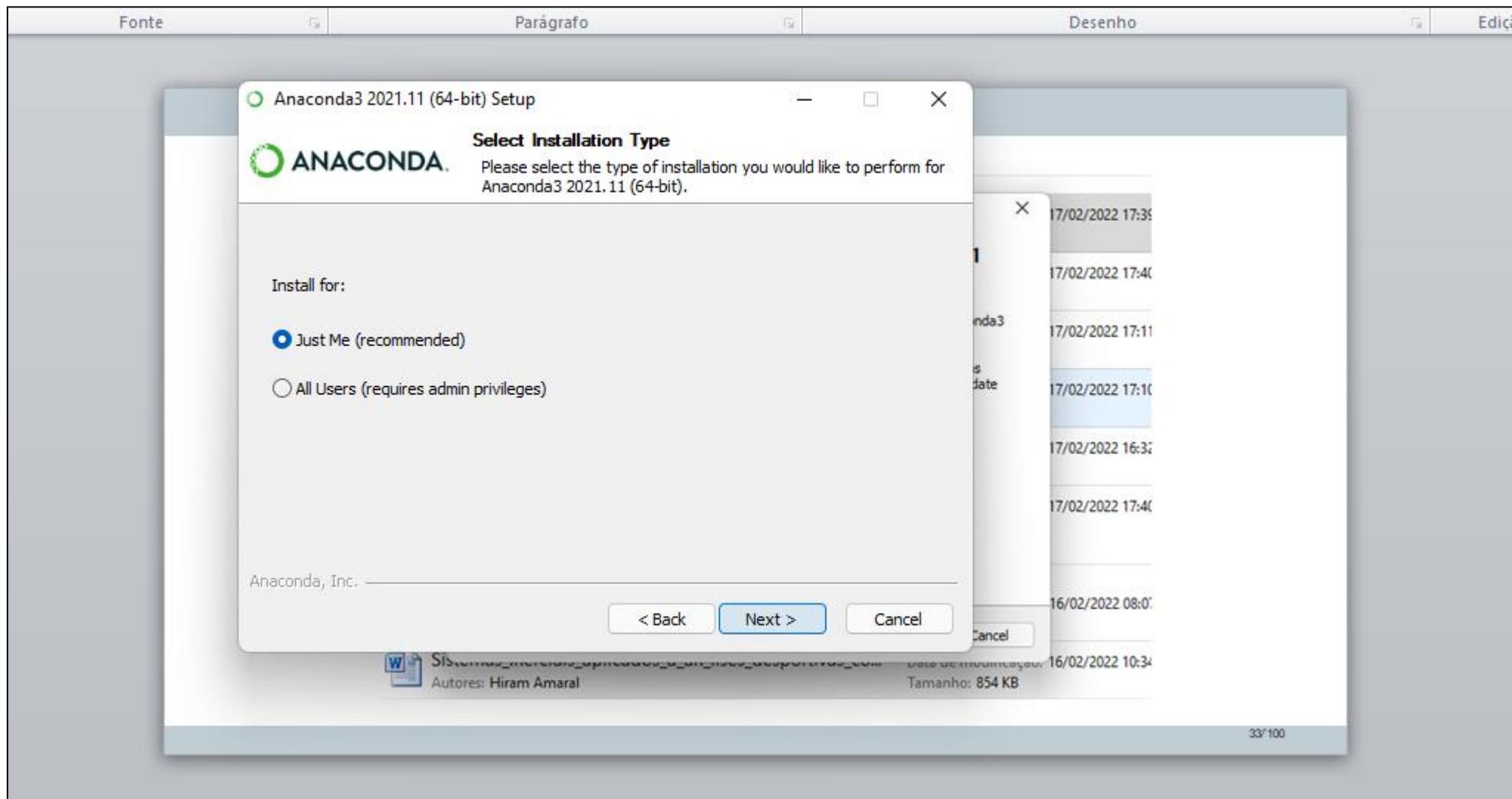
Não sou um robô

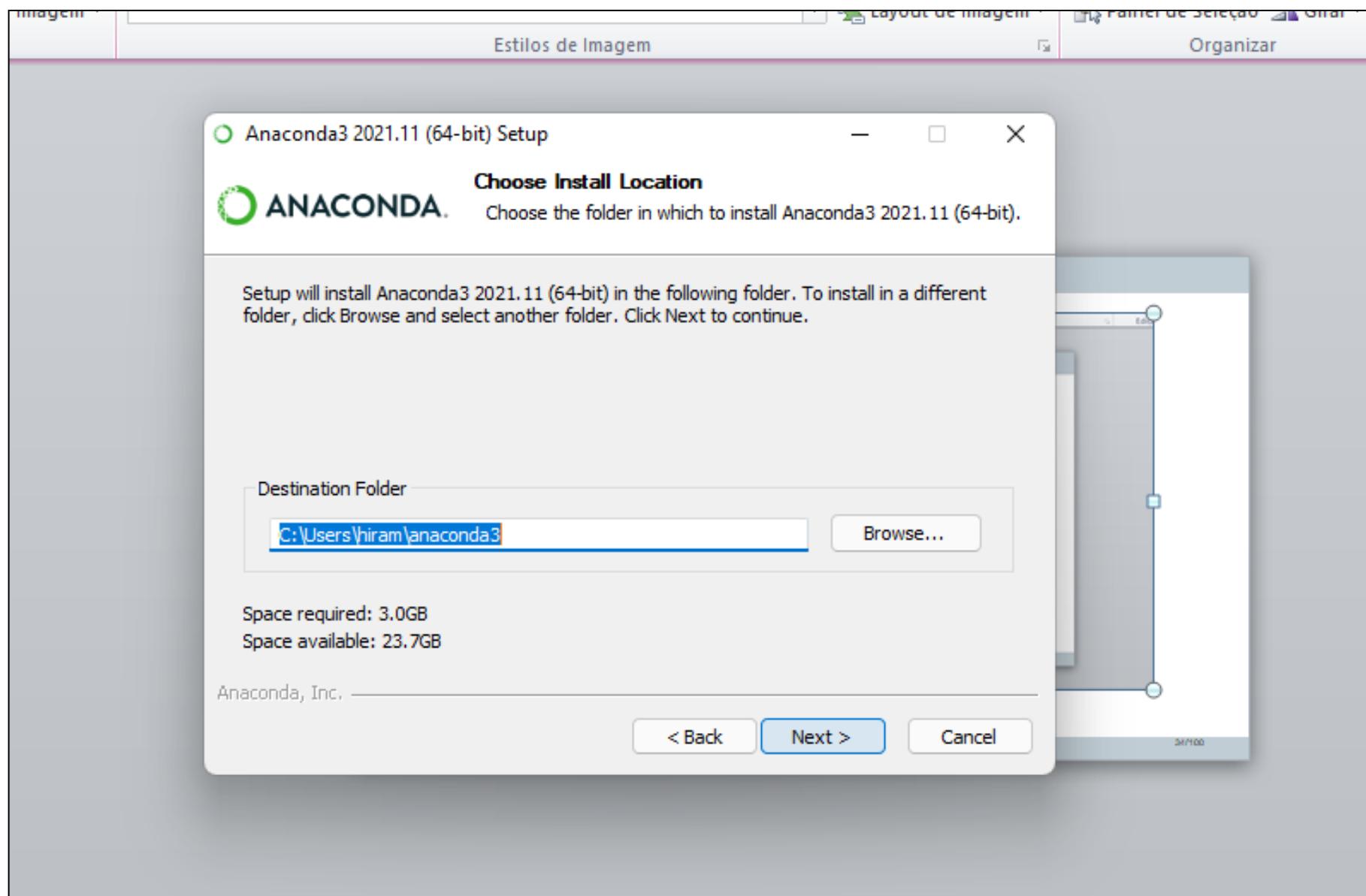
reCAPTCHA  
Privacidade - Termos

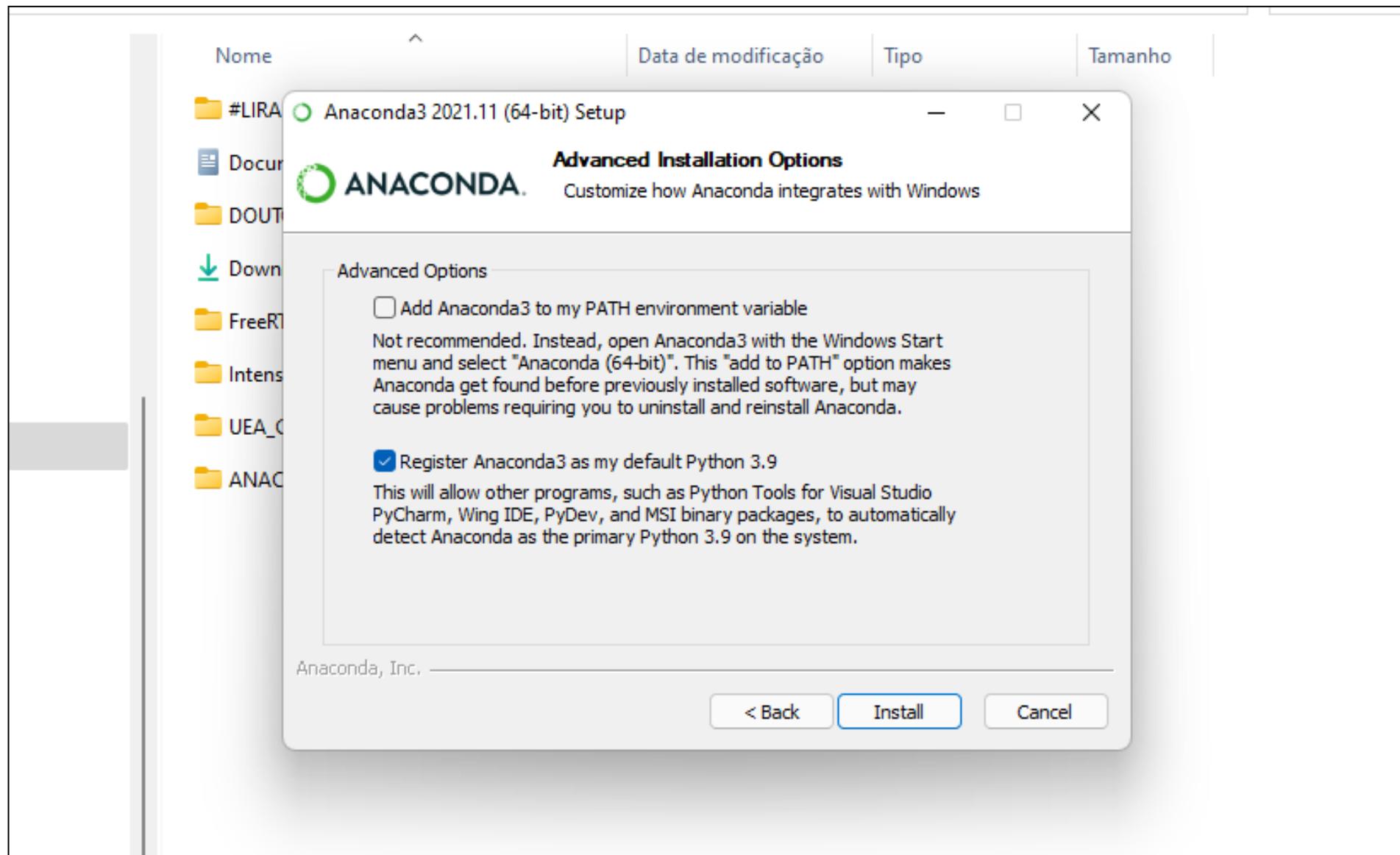
Get Started

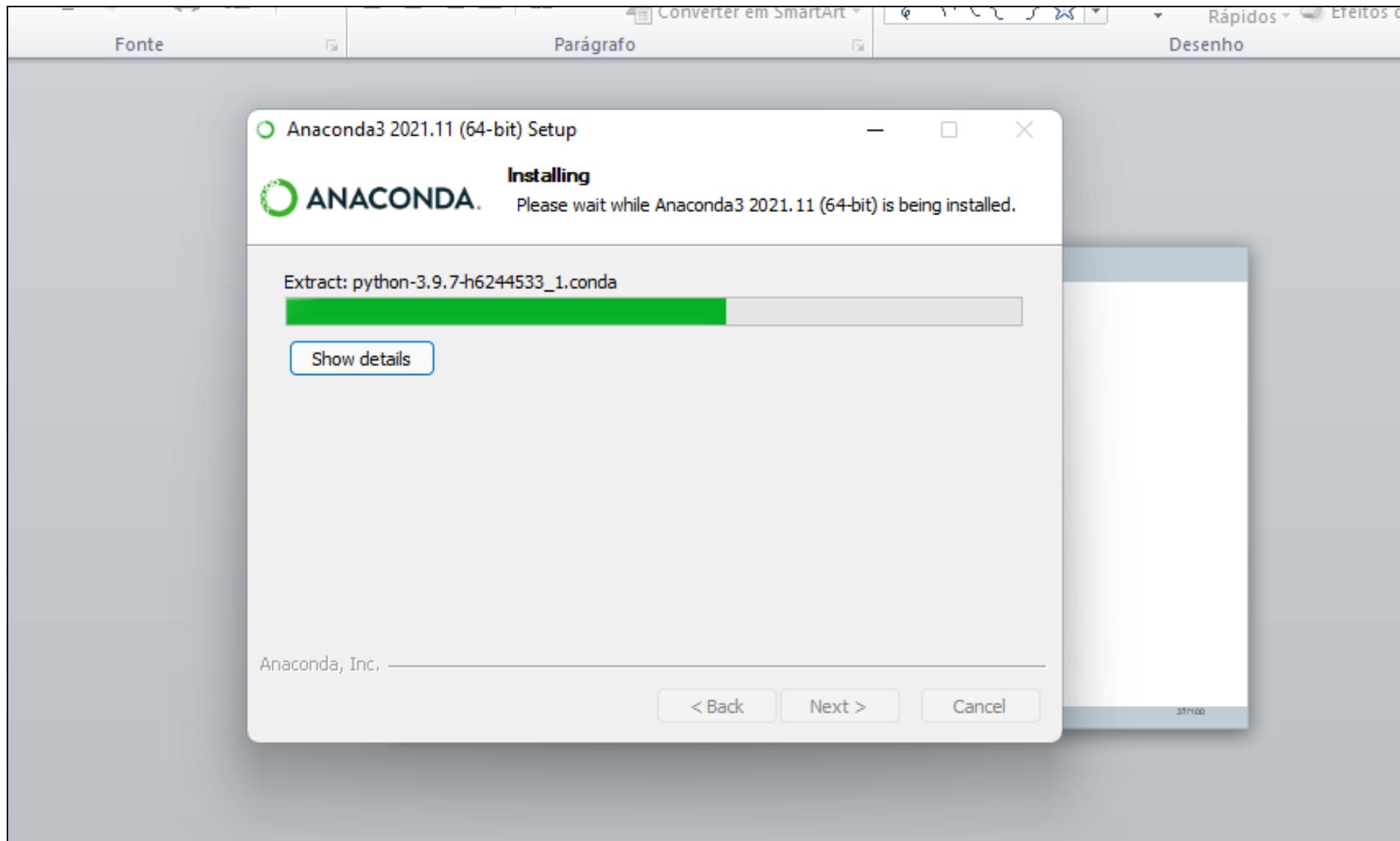














Anaconda3 2020.02 (64-bit) Setup



**Anaconda3 2020.02 (64-bit)**

Anaconda + JetBrains

Anaconda and JetBrains are working together to bring you Anaconda-powered environments tightly integrated in the PyCharm IDE.

PyCharm for Anaconda is available at:

<https://www.anaconda.com/pycharm>



**ANACONDA.**



Anaconda, Inc.

< Back

Next >

Cancel



## Completing Anaconda3 2020.02 (64-bit) Setup

Thank you for installing Anaconda Individual Edition.

Here are some helpful tips and resources to get you started.  
We recommend you bookmark these links so you can refer  
back to them later.

[Anaconda Individual Edition Tutorial](#)

[Learn More About Anaconda](#)

< Back

Finish

Cancel

# ANACONDA.NAVIGATOR

Connect ▾

Home

Environments

Learning

Community



Premium packages and dedicated support.

Documentation

Anaconda Blog



Applications on

base (root)

Channels



CMD.exe Prompt

0.1.1

Run a cmd.exe terminal with your current environment from Navigator activated

Launch



Datalore

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

Launch



IBM Watson Studio Cloud

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

Launch



JupyterLab

3.2.1

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

Launch



Notebook

6.4.5

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the



Powershell Prompt

0.0.1

Run a Powershell terminal with your current environment from Navigator activated



PyCharm Community

2021.3

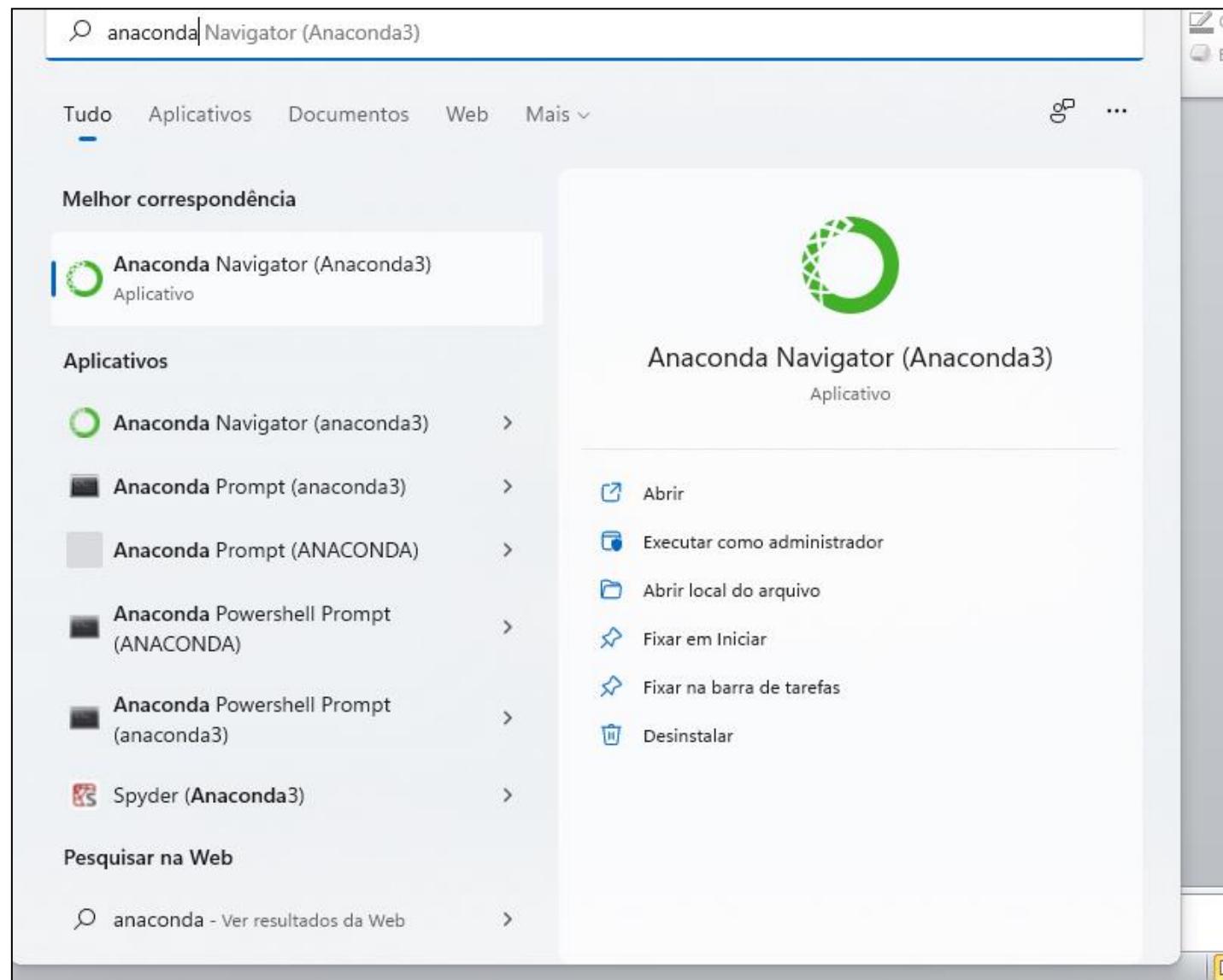
An IDE by JetBrains for pure Python development. Supports code completion, testing and debugging.



Qt Console

5.1.1

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calling, and more



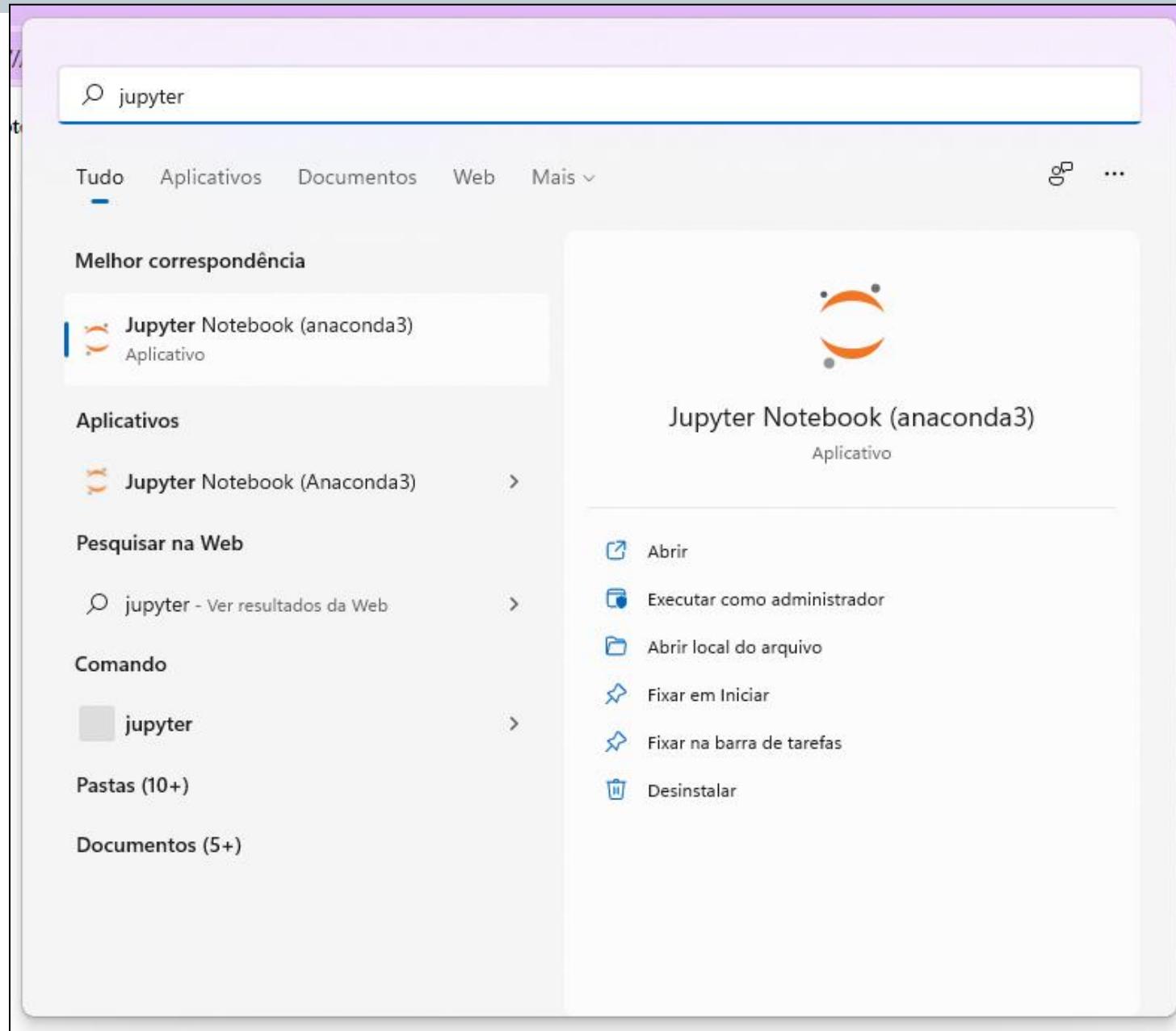
```
Anaconda Prompt (anaconda3)
(base) C:\Users\hiram> conda -v
usage: conda-script.py [-h] [-V] command ...
conda-script.py: error: the following arguments are required: command

(base) C:\Users\hiram> conda -V
conda 4.10.3

(base) C:\Users\hiram> conda --v
conda 4.10.3

(base) C:\Users\hiram>
```

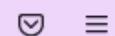
# Testando o Jupyter



```
Jupyter Notebook (anaconda3)
```

```
[W 2022-02-17 18:33:16.476 LabApp] 'notebook_dir' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2022-02-17 18:33:16.476 LabApp] 'notebook_dir' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[I 2022-02-17 18:33:16.495 LabApp] JupyterLab extension loaded from C:\Users\hiram\anaconda3\lib\site-packages\jupyterlab
[I 2022-02-17 18:33:16.496 LabApp] JupyterLab application directory is C:\Users\hiram\anaconda3\share\jupyter\lab
[I 18:33:16.500 NotebookApp] The port 8888 is already in use, trying another port.
[I 18:33:16.575 NotebookApp] Serving notebooks from local directory: C:\Users\hiram
[I 18:33:16.576 NotebookApp] Jupyter Notebook 6.4.5 is running at:
[I 18:33:16.576 NotebookApp] http://localhost:8889/?token=5f74c77fd76ca95670196c28ab5ba8da2971c89b37b70dc
[I 18:33:16.576 NotebookApp] or http://127.0.0.1:8889/?token=5f74c77fd76ca95670196c28ab5ba8da2971c89b37b70dc
[I 18:33:16.576 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 18:33:16.702 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/hiram/AppData/Roaming/jupyter/runtime/nbserver-12484-open.html
Or copy and paste one of these URLs:
http://localhost:8889/?token=5f74c77fd76ca95670196c28ab5ba8da2971c89b37b70dc
or http://127.0.0.1:8889/?token=5f74c77fd76ca95670196c28ab5ba8da2971c89b37b70dc
```



Quit

Logout

Files    Running    Clusters

Select items to perform actions on them.

Upload    New ▾

| <input type="checkbox"/> | 0 |                          | Documents                        | Name | Last Modified         | File size |
|--------------------------|---|--------------------------|----------------------------------|------|-----------------------|-----------|
|                          |   |                          | ..                               |      | poucos segundos atrás |           |
|                          |   | <input type="checkbox"/> | Apowersoft                       |      | 13 dias atrás         |           |
|                          |   | <input type="checkbox"/> | Arduino                          |      | 4 dias atrás          |           |
|                          |   | <input type="checkbox"/> | Colab Notebooks_hcca             |      | 9 horas atrás         |           |
|                          |   | <input type="checkbox"/> | DOUTORADO 2022                   |      | 3 dias atrás          |           |
|                          |   | <input type="checkbox"/> | GAIT_2022                        |      | 3 dias atrás          |           |
|                          |   | <input type="checkbox"/> | GATEWAY GAIT 2022                |      | um dia atrás          |           |
|                          |   | <input type="checkbox"/> | GitHub                           |      | 10 dias atrás         |           |
|                          |   | <input type="checkbox"/> | Jupyter                          |      | 9 dias atrás          |           |
|                          |   | <input type="checkbox"/> | lixo                             |      | 24 dias atrás         |           |
|                          |   | <input type="checkbox"/> | LTspiceXVII                      |      | 2 meses atrás         |           |
|                          |   | <input type="checkbox"/> | MATLAB                           |      | 8 horas atrás         |           |
|                          |   | <input type="checkbox"/> | Modelos Personalizados do Office |      | 2 meses atrás         |           |
|                          |   | <input type="checkbox"/> | Python Scripts                   |      | 2 meses atrás         |           |
|                          |   | <input type="checkbox"/> | texstudio                        |      | um dia atrás          |           |

The screenshot shows a Jupyter Notebook interface running at `localhost:8890/tree/Documents/Jupyter`. The top navigation bar includes a shield icon, a document icon, a star icon, a user icon, and a three-dot menu icon. The title bar says "jupyter". On the right, there are "Quit" and "Logout" buttons.

The main area has tabs for "Files", "Running", and "Clusters". Below the tabs, a message says "Select items to perform actions on them." A sidebar shows a file tree with a root folder containing two files: "Untitled.ipynb" and "Untitled1.ipynb".

A context menu is open on the right side of the screen, triggered by a click on the "New" button. The menu includes:

- Upload
- New ▾ (selected)
- Reset

Under "Notebook:":

- Name: Python 3 (ipykernel) (selected)
- Create a new notebook with Python 3 (ipykernel)

Under "Text File":

- Text File (1 KB)

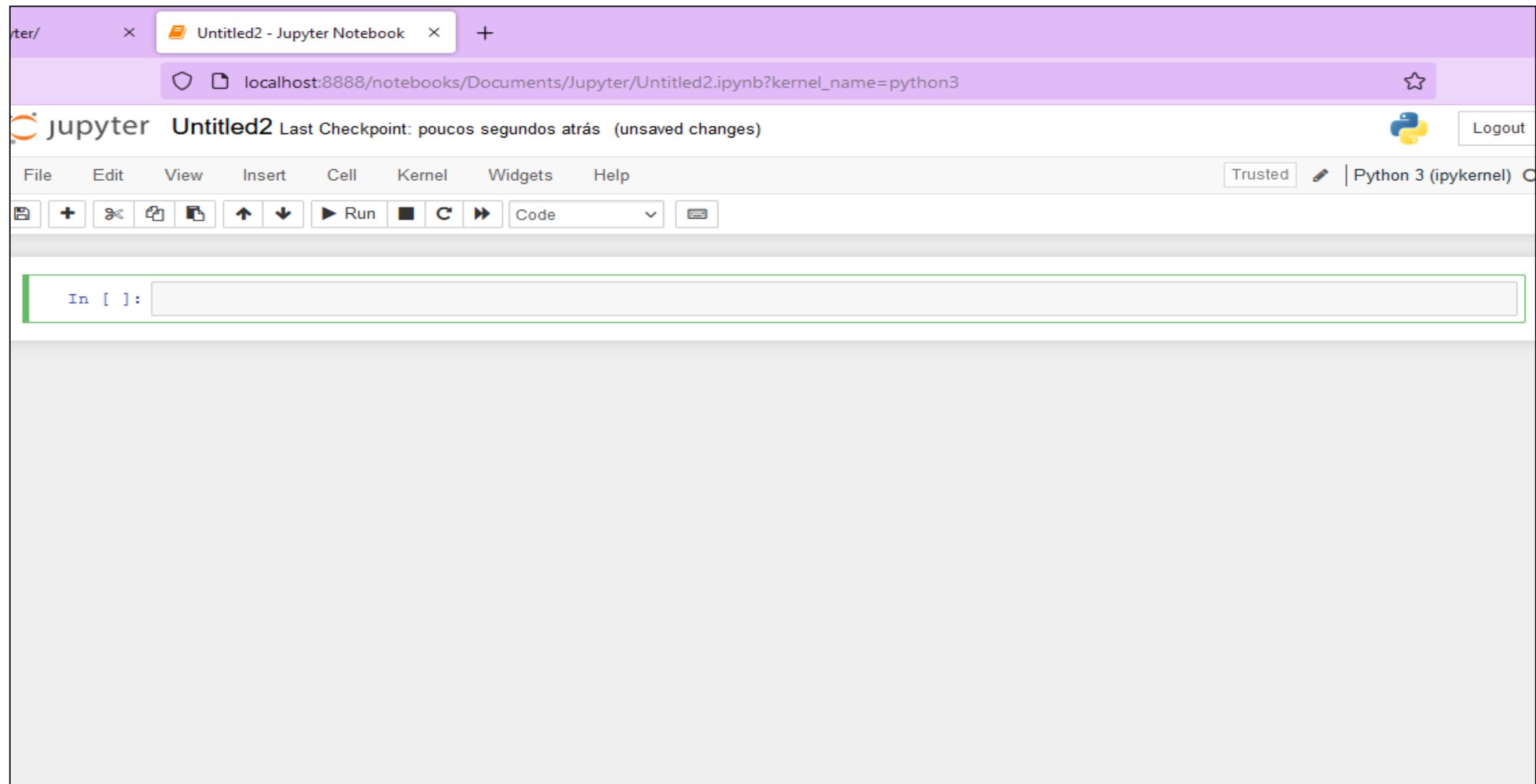
Under "Folder":

- Folder (6 B)

Under "Terminal":

- Terminal

At the bottom left, there is a footer bar with the text "Documents/Jupyter#".



A screenshot of a Jupyter Notebook interface. The title bar shows "Untitled2 - Jupyter Notebook". The URL in the address bar is "localhost:8888/notebooks/Documents/Jupyter/Untitled2.ipynb?kernel\_name=python3". The notebook header includes the Jupyter logo, the title "Untitled2", a message about last checkpoint, and icons for Python 3 (ipykernel), Logout, Trusted status, and a code editor icon. The menu bar has options: File, Edit, View, Insert, Cell, Kernel, Widgets, Help. Below the menu is a toolbar with icons for file operations like new, open, save, and run cells. The main content area displays a code cell with the following content:

```
In [1]: print("Olá Cetam!")
```

The output of the cell is:

```
Olá Cetam!
```

Below the cell, there is an empty input field labeled "In [ ]:".

# Instalando o Matplotlib

<https://matplotlib.org/>

## Matplotlib: Visualização com Python

Matplotlib é uma biblioteca abrangente para criar visualizações estáticas, animadas e interativas em Python. O Matplotlib torna as coisas fáceis fáceis e as difíceis possíveis.

- Crie gráficos de qualidade de publicação .
  - Faça figuras interativas que podem ampliar, deslocar, atualizar.
  - Personalize o estilo visual e o layout .
  - Exporte para vários formatos de arquivo .
  - Incorporar no JupyterLab e nas interfaces gráficas do usuário .
  - Use uma rica variedade de pacotes de terceiros criados no Matplotlib.

Experimente o Matplotlib (no Binder)



## Começando



## Exemplos



## Referência



## Folhas de dicas



Documentação

 Pesquise nos documentos...

```
plot(x,y)
dispersão (x, y)
bar(x, altura) / barh(y, largura)
haste (x, y)
passo(x, y)
fill_between(x, y1, y2)
imshow(Z)
pcolormesh(X, Y, Z)
contorno(X, Y, Z)
contornof(X, Y, Z)
farpas (X, Y, U, V)
aljava(X, Y, U, V)
streamplot(X, Y, U, V)
hist(x)
boxplot(X)
```

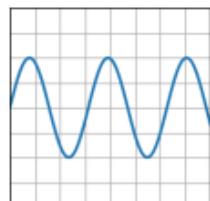
## Tipos de plotagem

Visão geral de muitos comandos de plotagem comuns no Matplotlib.

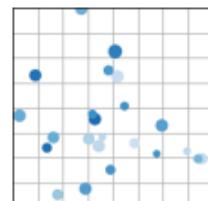
Observe que removemos todos os rótulos, mas eles estão presentes por padrão. Veja a galeria para muitos outros exemplos e a página de tutoriais para exemplos mais longos.

Básico¶

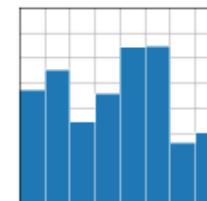
## Tipos básicos de plotagem, geralmente y versus x.



`plot(x,y)`



dispersão ( $x, y$ )



bar(x, altura) / barh(y,  
larura)

Nesta página

- Básico
- Plotagens de arrays e campo:
- Gráficos de estatísticas
- Coordenadas não estruturadas

 Pesquise nos documentos...

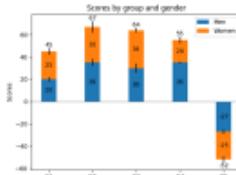
- Demonstração da etiqueta da barra
  - Gráfico de barras empilhadas
  - Gráfico de barras agrupado com rótulos
  - Gráfico de barras horizontais
  - Barra quebrada
  - CapStyle
  - Plotando variáveis categóricas
  - Traçar a coerência de dois sinais
  - Demonstração de refrigerante
  - Curva com banda de erro
  - Seleção de limite da barra de erros
  - Subamostragem da barra de erros
  - Demonstração da coleção de eventos
  - Demonstração do gráfico de eventos
  - Polígono preenchido
  - Preencher Entre e Alfa

## Exemplos

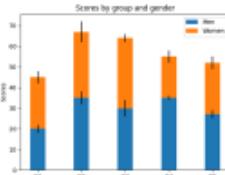
Esta página contém gráficos de exemplo. Clique em qualquer imagem para ver a imagem completa e o código-fonte.

Para tutoriais mais longos, consulte nossa página de tutoriais . Você também pode encontrar recursos externos e perguntas frequentes em nosso guia do usuário .

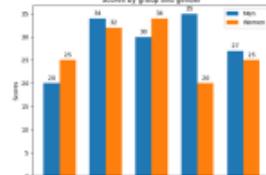
Linhos, barras e marcadores ¶



## Demonstração da etiqueta da barra



## Gráfico de barras empilhadas



## Gráfico de barras agrupado com rótulos

 Nesta página

## Linhos, barras e marcadores

Imagens, contornos e  
campos

## Subtramas, eixos e figuras

## Gráficos de pizza e polares Texto, rótulos e anotações matplotlib

Cor

Formas e coleções

## Folhas de estilo

axes\_grid1

artista do eixo

Mostruário

## Animação

### Manipulação de eventos

Primeira p

Diversos

 Pesquise nos documentos...

- Uso básico
  - Pyplot tutorial
  - Tutorial de imagem
  - O ciclo de vida de um enredo
  - Personalizando Matplotlib com folhas de estilo e rcParams
  - Tutorial do artista
  - Guia de legendas
  - Modelagem com cicladora
  - Guia de layout restrito
  - Guia de layout rígido
  - Organizando vários eixos em uma figura
  - Escalonamento automático
  - origem e extensão em imshow*
  - Renderização mais rápida usando blitting
  - Tutorial de caminho

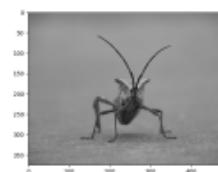
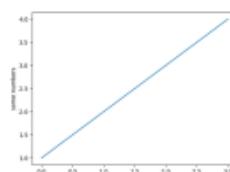
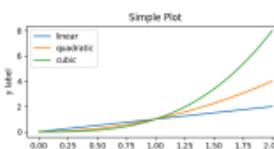
## Tutoriais

Esta página contém guias mais detalhados para usar o Matplotlib. Ele é dividido em seções para iniciantes, intermediários e avançados, bem como seções que cobrem tópicos específicos.

Para exemplos mais curtos, consulte nossa página de exemplos . Você também pode encontrar recursos externos e perguntas frequentes em nosso quia do usuário .

Introdutório

Esses tutoriais cobrem os fundamentos da criação de visualizações com o Matplotlib, bem como algumas práticas recomendadas para usar o pacote de forma eficaz.



## Uso básico

Pyplot tutorial

Tutorial de imagem

 Nesta página

Introdutório

Intermediário

Avances

6000

200

PROVIS

lexto

 Pesquise nos documentos...

- matplotlib
- matplotlib.afm
- matplotlib.animation
- matplotlib.artist
- matplotlib.axes
- matplotlib.axis
- matplotlib.backend\_bases
- matplotlib.backend\_managers
- matplotlib.backend\_tools
- matplotlib.backends
- matplotlib.bezier
- matplotlib.blocking\_input
- matplotlib.category
- matplotlib.cbook
- matplotlib.cm
- matplotlib.collections
- matplotlib.colorbar
- matplotlib.colors
- matplotlib.container
- matplotlib.contour
- matplotlib.dates
- matplotlib.docstring
- matplotlib.dviread
- ... ...

## Referência da

Ao usar a biblioteca, você normalmente criará objetos Figura e Eixos e chamará seus métodos para adicionar conteúdo e modificar a aparência.

- [matplotlib.figure](#): criação de eixos, conteúdo em nível de figura
- [matplotlib.axes](#): a maioria dos métodos de plotagem, rótulos de eixos, acesso ao estilo de eixo, etc.

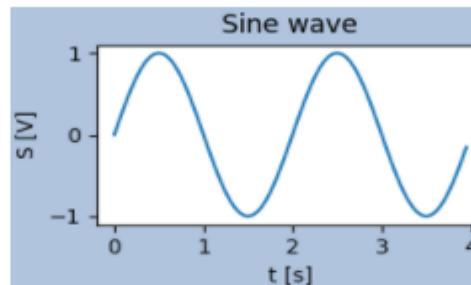
Exemplo: Criamos uma Figura `fig` Eixos `ax`. Em seguida, chamamos métodos neles para plotar dados, adicionar rótulos de eixo e um título de figura.

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

x = np.arange(0, 4, 0.05)
y = np.sin(x*np.pi)

fig, ax = plt.subplots(figsize=(3,2), constrained_layout=True)
ax.plot(x, y)
ax.set_xlabel('t [s]')
ax.set_ylabel('s [V]')
ax.set_title('Sine wave')
fig.set_facecolor('lightsteelblue')
```

( [Código fonte](#) , [png](#) , [pdf](#) )



Nesta página

Módulos

Padrões de uso

 Pesquise nos documentos...

## Guia do usuário

Nesta página

- Em geral
- Tutoriais e exemplos
- Referência

Geral

- Começando
    - Instalação rápida de início
    - Desenhe um primeiro gráfico
    - Para onde ir a seguir
  - Instalação
    - Instalando uma versão oficial
    - Distribuições de terceiros
    - Instalando da fonte
    - Instalando para desenvolvimento
    - Perguntas frequentes
  - Explicações
    - Back-ends
    - Figuras interativas
    - Fontes no mecanismo de texto Matplotlib
    - Tratamento e picking de eventos
    - Desempenho



Pesquise nos documentos...

Contribuindo

Triagem de bugs e curadoria de problemas

Configurando o Matplotlib para desenvolvimento

Teste

Escrevendo documentação

Guia de estilo de documentação

Trabalhando com o código-fonte do *Matplotlib* ▾

Diretrizes de solicitação de pull

Guia de lançamento

Dependências

Versão mínima da política de dependências

Propostas de aprimoramento do Matplotlib ▾

Licenças

Alterações de cor padrão

## Contribuindo

Obrigado pelo seu interesse em ajudar a melhorar o Matplotlib! Existem várias maneiras de contribuir para o Matplotlib. Todos eles são super valiosos, mas não exigem necessariamente escrever código. Por exemplo:

- contribuindo com a documentação
- abrindo novos problemas para bugs
- solicitando novos recursos
- pedindo esclarecimentos sobre coisas que você acha pouco claras
- corrigindo bugs

Se você tiver alguma dúvida sobre o processo ou como consertar algo, sinta-se à vontade para perguntar no [gitter](#) para perguntas curtas e no [discurso](#) para perguntas mais longas.

[Reportar um erro](#)

[Solicite um recurso](#)

[Contribuir com código](#)

[Escrever documentação](#)

- Contribuindo
  - Incubadora de colaboradores
  - Envio de um relatório de bug



Pesquise nos documentos...

O que há de novo no Matplotlib 3.5.0 (15 de novembro de 2021)

Alterações de API para 3.5.0

Estatísticas do GitHub (11 de dezembro de 2021)

Estatísticas do GitHub para 3.5.0 (15 de novembro de 2021)

O que há de novo no Matplotlib 3.4.0 (26 de março de 2021)

Mudanças de API para 3.4.2

Alterações de API para 3.4.0

Estatísticas do GitHub para 3.4.1 (31 de março de 2021)

Estatísticas do GitHub para 3.4.0 (26 de março de 2021)

O que há de novo no Matplotlib 3.3.0 (16 de julho de 2020)

Mudanças de API para 3.3.0

# Notas de lançamento

☰ Nesta página

Versão 3.5

Versão 3.4

Versões anteriores

## Versão

- [O que há de novo no Matplotlib 3.5.0 \(15 de novembro de 2021\)](#)
- [Alterações de API para 3.5.0](#)
- [Estatísticas do GitHub \(11 de dezembro de 2021\)](#)
- [Estatísticas do GitHub para 3.5.0 \(15 de novembro de 2021\)](#)

## Versão

- [O que há de novo no Matplotlib 3.4.0 \(26 de março de 2021\)](#)
- [Mudanças de API para 3.4.2](#)
- [Alterações de API para 3.4.0](#)
- [Estatísticas do GitHub para 3.4.1 \(31 de março de 2021\)](#)
- [Estatísticas do GitHub para 3.4.0 \(26 de março de 2021\)](#)

## Versões anteriores

# Instalando o Scikit-learn

<https://scikit-learn.org/stable/index.html>

[←](#) [→](#) [C](#) [Home](#) [scikit-learn.org/stable/index.html](#)

 [Instalar](#) [Guia de usuário](#) [API](#) [Exemplos](#) [Mais](#) [Vai](#)

# scikit-learn

## Aprendizado de máquina em Python

[Começando](#) [Destaque da versão 1.0](#) [GitHubGenericName](#)

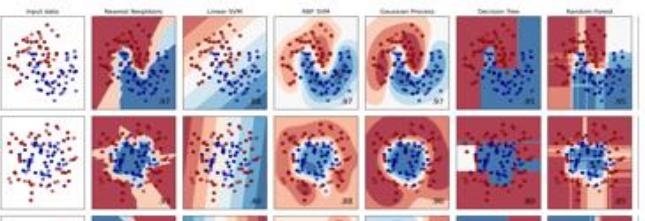
- Ferramentas simples e eficientes para análise preditiva de dados
- Acessível a todos e reutilizável em vários contextos
- Construído em NumPy, SciPy e matplotlib
- Código aberto, comercialmente utilizável - licença BSD

## Classificação

Identificar a qual categoria um objeto pertence.

**Aplicações:** Detecção de spam, reconhecimento de imagem.

**Algoritmos:** SVM , vizinhos mais próximos , floresta aleatória e mais...

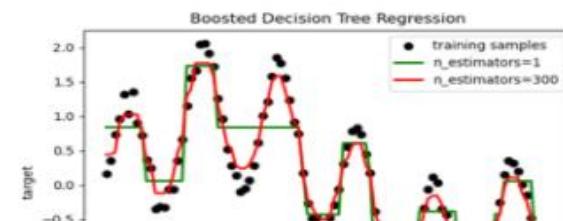


## Regressão

Prevendo um atributo de valor contínuo associado a um objeto.

**Aplicações:** Resposta a medicamentos, Preços de ações.

**Algoritmos:** SVR , vizinhos mais próximos , floresta aleatória e mais...

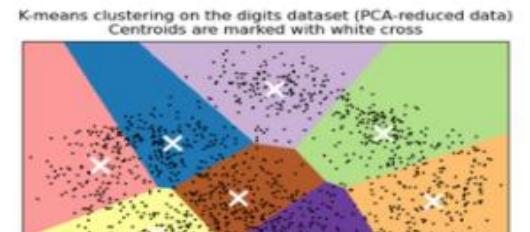


## Agrupamento

Agrupamento automático de objetos semelhantes em conjuntos.

**Aplicações:** Segmentação de clientes, Resultados de experimentos de agrupamento

**Algoritmos:** k-Means , clustering espectral , mean-shift , e muito mais...



- → C H scikit-learn.org/stable/install.html



Instalar Guia de usuario API Exemplos Mais ▾

Anterior Acima Próximo

Você pode favoritarnos se você usar o software.

Instalando o scikit-learn

- Instalando a versão mais recente
- Instalando no hardware Apple com M1
- Distribuições de terceiros do scikit-learn
- Solução de problemas

## Instalando o scikit-learn

Existem diferentes maneiras de instalar o scikit-learn:

- Instale a versão oficial mais recente. Esta é a melhor abordagem para a maioria dos usuários. Ele fornecerá uma versão estável e os pacotes pré-criados estão disponíveis para a maioria das plataformas.
- Instale a versão do scikit-learn fornecida pelo seu sistema operacional ou distribuição do Python. Esta é uma opção rápida para quem tem sistemas operacionais ou distribuições Python que distribuem scikit-learn. Pode não fornecer a versão de lançamento mais recente.
- Construindo o pacote a partir do código-fonte. Isso é melhor para usuários que desejam os melhores e mais recentes recursos e não têm medo de executar um código totalmente novo. Isso também é necessário para usuários que desejam contribuir com o projeto.

### Instalando a versão mais recente

Sistema operacional

janelas Mac OS Linux

Empacotador

pip conda

Use pip virtualenv

Instale a versão de 64 bits do Python 3, por exemplo, em <https://www.python.org>.  
Então corra:

Internar menu

← → C Home scikit-learn.org/stable/user\_guide.html

 scikit-learn Instalar Guia de usuário API Exemplos Mais ▾

Anterior Acima Próximo

**scikit-learn 1.0.2**  
Outras versões

Por favor cite-nos se você usar o software.

**Guia de usuário**

- 1. Aprendizado supervisionado
- 2. Aprendizado não supervisionado
- 3. Seleção e avaliação do modelo
- 4. Inspeção
- 5. Visualizações
- 6. Transformações do conjunto de dados
- 7. Utilitários de carregamento do conjunto de dados
- 8. Computação com scikit-learn
- 9. Persistência do modelo
- 10. Armadilhas comuns e práticas recomendadas

Alternar menu

# Guia do usuário

## 1. Aprendizado supervisionado

- 1.1. Modelos lineares
- 1.2. Análise Discriminante Linear e Quadrática
- 1.3. Regressão da crista do kernel
- 1.4. Máquinas de vetor de suporte
- 1.5. Descida do Gradiente Estocástico
- 1.6. Vizinhos mais próximos
- 1.7. Processos Gaussianos
- 1.8. Decomposição cruzada
- 1.9. Baías ingênuas
- 1.10. Árvores de decisão
- 1.11. Métodos de conjunto
- 1.12. Algoritmos multiclasse e multisaída

[Anterior](#) [Acima](#) [Próximo](#)**scikit-learn 1.0.2**[Outras versões](#)

Por favor [cite-nos](#) se você usar o software.

**Referência da API**[sklearn.base: classes base e funções utilitárias](#)[sklearn.calibration: Calibração de probabilidade](#)[sklearn.cluster: Agrupamento](#)[sklearn.compose: Estimadores compostos](#)[sklearn.covariance: Estimadores de covariância](#)[sklearn.cross\\_decomposition: Decomposição cruzada](#)[sklearn.datasets: conjuntos de dados](#)[sklearn.decomposition: Decomposição da Matriz](#)[sklearn.discriminant\\_analysis: Análise discriminante](#)[Alternar menu](#)

## Referência da API

Esta é a referência de classe e função do scikit-learn. Consulte o [guia do usuário completo](#) para obter mais detalhes, pois as especificações brutas de classe e função podem não ser suficientes para fornecer orientações completas sobre seus usos. Para referência sobre conceitos repetidos na API, consulte [Glossário de termos comuns e elementos da API](#).

### sklearn.base: Classes base e funções utilitárias

Classes base para todos os estimadores.

#### Classes

[base.BaseEstimator](#) Classe base para todos os estimadores no scikit-learn.

[base.BiclusterMixin](#) Classe Mixin para todos os estimadores bicluster no scikit-learn.

[base.ClassifierMixin](#) Classe Mixin para todos os classificadores no scikit-learn.

[base.ClusterMixin](#) Classe Mixin para todos os estimadores de cluster no scikit-learn.

[base.DensityMixin](#) Classe Mixin para todos os estimadores de densidade no scikit-learn.

[base.RegressorMixin](#) Classe Mixin para todos os estimadores de regressão no scikit-learn.

[base.TransformerMixin](#) Classe Mixin para todos os transformadores no scikit-learn.

[feature\\_selection.SelectorMixin](#) Mixin do transformador que executa a seleção de recursos com uma máscara de suporte

#### Funções

[base.clone\(estimador, \\*\[  
copy=True\]\)](#) Construa um novo estimador não ajustado com os mesmos parâmetros.





Por favor **cite-nos** se você usar o software.

Bem-vindo ao scikit-learn  
Tutoriais scikit-learn

Uma introdução ao aprendizado de máquina com scikit-learn  
Um tutorial sobre aprendizado estatístico para processamento de dados científicos

Trabalhando com dados de texto  
Escolhendo o estimador certo  
Recursos externos, vídeos e  
palestras

### **Comercando**

Guía de usuario

Glossário de Termos Comuns e

## Elementos da API

[Exercises](#)

## Alternar menu

## Tutoriais scikit-learn

Uma introdução ao aprendizado de máquina com scikit-learn

- Aprendizado de máquina: a configuração do problema
  - Carregando um conjunto de dados de exemplo
  - Aprendizagem e previsão
  - Convocações

# Um tutorial sobre aprendizado estatístico para processamento de dados científicos

- Aprendizagem estatística: o cenário e o objeto estimador no scikit-learn
  - Aprendizado supervisionado: prevendo uma variável de saída a partir de observações de alta dimensão

[github.com/scikit-learn/scikit-learn](https://github.com/scikit-learn/scikit-learn)

Por que GitHub? Equipe Empreendimento Explorar Mercado Preços

Procurar Entrar Inscrever-se

scikit-learn / scikit-learn Público

Patrocinador Notificações Garfo 22,7 mil Estrela 49,1 mil

Código Problemas 1,7 mil Solicitações de pull 763 Discussões Ações Projetos 18 Wiki Segurança Intuições

a Principal 23 filiais 121 etiquetas Ir para o arquivo Código

|   |  |  |                     |
|---|--|--|---------------------|
|  thomasjpfan | DOC Mova a persistência do modelo para o guia do usuário (#225...) | 593495e há 3 horas   | 27.915 confirmações |
|              | .encadernador  | URL de compilação noturna de atualização MNT e configuração de teste...      | há 2 anos           |
|              | .circleci  | CI Remove o diretório de compilação do cache (#22192 )                       | mês passado         |
|              | .github  | CI corrige compilações de roda do Windows 64 (#22532 )                       | há 5 horas          |
|             | asv_benchmarks   | MNT Torna o algoritmo='auto' padrão para usar 'full' em vez de 'elkan' (...) | há 3 meses          |
|            | referências  | Atualização MNT de preto para versão estável (#22474 )                       | há 3 dias           |
|            | ferramentas_construção   | MAINT Ajustar testes para numpydoc 1.2 (#22287 )                             | há 7 dias           |
|            | doc  | DOC Mover a persistência do modelo para o guia do usuário (#22511 )          | há 3 horas          |
|            | exemplos   | O tipo ENH aceita uma lista de str em plot_partial_dependence (#19438 )      | ontem               |
|            | ferramentas_manutenção   | MNT Corrigir bot para falha de coleta de teste (#22512 )                     | ontem               |

Sobre

scikit-learn: aprendizado de máquina em Python

scikit-learn.org

Pitão ciência de dados  
aprendizado de máquina Estatísticas  
análise de dados

Leia-me Licença BSD-3-Cláusula Código de Conduta 49,1 mil estrelas 2,2k assistindo garfos de 22,7k

[Anterior](#) [Acima](#) [Próximo](#)

**scikit-learn 1.1.dev0**

[Outras versões](#)

Por favor [cite-nos](#)se você usar o software.

**Documentação disponível para Scikit-learn**

[Alternar menu](#)

## Documentação disponível para Scikit-learn

A documentação baseada na Web está disponível para as versões listadas abaixo:

- Documentação do Scikit-learn 1.1.dev0 (dev) ( ZIP 60,0 MB )
- Documentação do Scikit-learn 1.0.2 (estável) ( ZIP 59,4 MB )
- Documentação do Scikit-learn 0.24.2 ( ZIP 67,6 MB )
- Documentação do Scikit-learn 0.23.2 ( PDF 51,0 MB )
- Documentação do Scikit-learn 0.22.2 ( PDF 48,5 MB )
- Documentação do Scikit-learn 0.21.3 ( PDF 46,7 MB )
- Documentação do Scikit-learn 0.20.4 ( PDF 45,2 MB )
- Documentação do Scikit-learn 0.19.2 ( PDF 42,2 MB )
- Documentação do Scikit-learn 0.18.2 ( PDF 46,5 MB )
- Documentação do Scikit-learn 0.17.1 ( PDF 46,0 MB )
- Documentação do Scikit-learn 0.16.1 ( PDF 56,8 MB )
- Documentação do Scikit-learn 0.15-git

© 2007 - 2022, desenvolvedores scikit-learn (licença BSD). [Mostrar esta fonte de página](#)

```
C:\ Prompt de Comando - python
Microsoft Windows [versão 10.0.22000.493]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\hiram>python
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

```
C:\ Prompt de Comando

C:\Users\hiram>pip install -U scikit-learn
Requirement already satisfied: scikit-learn in c:\users\hiram\appdata\local\programs\python\python310\lib\site-packages
(1.0.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\hiram\appdata\local\programs\python\python310\lib\site-packages
(from scikit-learn) (3.0.0)
Requirement already satisfied: numpy>=1.14.6 in c:\users\hiram\appdata\local\programs\python\python310\lib\site-packages
(from scikit-learn) (1.21.5)
Requirement already satisfied: joblib>=0.11 in c:\users\hiram\appdata\local\programs\python\python310\lib\site-packages
(from scikit-learn) (1.1.0)
Requirement already satisfied: scipy>=1.1.0 in c:\users\hiram\appdata\local\programs\python\python310\lib\site-packages
(from scikit-learn) (1.7.3)

C:\Users\hiram>█
```

```
D  
le C:\Users\hiram>python -m pip show scikit-learn  
Sc  
Name: scikit-learn  
Version: 1.0.2  
Summary: A set of python modules for machine learning and data mining  
Home-page: http://scikit-learn.org  
Author:  
Author-email:  
License: new BSD  
Location: c:\users\hiram\appdata\local\programs\python\python310\lib\site-packages  
Requires: joblib, numpy, scipy, threadpoolctl  
Required-by:  
  
C:\Users\hiram>
```

C:\ Prompt de Comando

```
C:\Users\hiram>python -m pip freeze
anyio==3.5.0
argon2-cffi==21.3.0
argon2-cffi-bindings==21.2.0
asttokens==2.0.5
attrs==21.4.0
Babel==2.9.1
backcall==0.2.0
black==22.1.0
bleach==4.1.0
certifi==2021.10.8
cffi==1.15.0
charset-normalizer==2.0.12
click==8.0.3
colorama==0.4.4
cycler==0.11.0
debugpy==1.5.1
decorator==5.1.1
defusedxml==0.7.1
entrypoints==0.4
executing==0.8.2
fonttools==4.28.5
idna==3.3
imageio==2.13.5
ipykernel==6.9.1
ipython==8.0.1
ipython-genutils==0.2.0
```

```
C:\Users\hiram>python -c "import sklearn; sklearn.show_versions()"  
C:\Users\hiram\AppData\Local\Programs\Python\Python310\lib\site-packages\_distutils_hack\__init__.py:36: UserWarning: Se  
tuptools is replacing distutils.  
  warnings.warn("Setuptools is replacing distutils.")  
  
System:  
  python: 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)]  
  executable: C:\Users\hiram\AppData\Local\Programs\Python\Python310\python.exe  
  machine: Windows-10-10.0.22000-SP0  
  
Python dependencies:  
  pip: 22.0.3  
  setuptools: 60.1.1  
  sklearn: 1.0.2  
    numpy: 1.21.5  
    scipy: 1.7.3  
    Cython: None  
    pandas: None  
    matplotlib: 3.5.1  
    joblib: 1.1.0  
  threadpoolctl: 3.0.0  
  
  Built with OpenMP: True  
  
C:\Users\hiram>
```

# Integração Colab com o GitHub

# Realizando os primeiros códigos

# Obrigada(o) pela Atenção !!!

Somos hoje o que penamos ontem, seremos amanhã,  
o que pensarmos hoje! Carl Roger



Dhenifer Araújo  
Coordenadora, MsC



Hiram Amaral  
Instrutor, MsC



Diogo Rezende  
Instrutor, Dr.



Jhonatas  
Monitor



Nathalia  
Monitora

## LINGUAGEM DE PROGRAMAÇÃO PARA IA COM PYTHON