

Instalação e Configuração do Ambiente de Desenvolvimento

Allan M. de Souza

Resumo: Neste tutorial apresentaremos o ambiente de desenvolvimento que será utilizado durante o curso. Utilizaremos Python como a linguagem de programação, além de um conjunto de bibliotecas auxiliares para nos ajudar no desenvolvimento das aplicações. Ao final desse tutorial, você será capaz de criar e configurar seu próprio ambiente de desenvolvimento e terá conhecimento sobre as principais ferramentas que serão utilizadas, sendo elas, Python, Miniconda e Jupyter.

Introdução

Instalar ou atualizar o Python em seu sistema é o primeiro passo para se tornar um desenvolvedor Python. Há muitas maneiras de começar a configurar e instalar: você pode baixar as distribuições oficiais do Python em [Python.org](https://python.org), instalar a partir de um gerenciador de pacotes (Anaconda) e até mesmo instalar distribuições especializadas para computação científica, Internet das Coisas e sistemas embarcados.

A linguagem Python tem se tornado muito popular nos últimos anos e vem sendo gradativamente utilizada em diversos campos devido à sua facilidade de uso e ricas bibliotecas de terceiros. No campo de aprendizado profundo, muitos frameworks baseados em Python são propostos. No primeiro capítulo, focamos nos recursos da linguagem Python e como usá-la.

Verifique se você já tem o Python instalado

Você pode perguntar: Como verificar sua versão do Python no Windows? Você pode verificar no aplicativo de linha de comando, como o PowerShell Dica: Veja como abrir o PowerShell:

- Pressione a tecla \boxtimes Win.
- Digite PowerShell ou cmd.
- Pressione Enter

Como alternativa, você pode clicar com o botão direito do mouse no botão Iniciar e selecionar Windows PowerShell ou Windows PowerShell (Admin). Se estiver interessado em saber onde a instalação está localizada, você pode usar o comando **where.exe** no prompt de comando ou no PowerShell.

Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\yoges> python --version
Python 3.7.9
PS C:\Users\yoges> python -V
Python 3.7.9
PS C:\Users\yoges> where.exe python
C:\Users\yoges\AppData\Local\Microsoft\WindowsApps\python.exe
PS C:\Users\yoges> █
```

Command Prompt

```
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\yoges>python --version
Python 3.7.9

C:\Users\yoges>python -V
Python 3.7.9

C:\Users\yoges>where.exe python
C:\Users\yoges\AppData\Local\Microsoft\WindowsApps\python.exe
```

Instalar o Python

Etapas 1: selecione a versão do Python para baixar o instalador completo e instalar

Os downloads mais estáveis do Windows estão disponíveis na página [Python para Windows](#). No Windows, você pode escolher entre as versões de 32 bits (rotulada como x86) e 64 bits (rotulada como x86-64) e vários tipos de instalador para cada uma. Abaixo das versões do Python para Windows, encontre a versão mais recente do Python 3 — Python 3.9.4

Python Releases for Windows

- [Latest Python 3 Release - Python 3.9.4](#)
- [Latest Python 2 Release - Python 2.7.18](#)

Files

| Version | Operating System | Description | MD5 Sum | File Size | GPG |
|---|------------------|--|----------------------------------|-----------|---------------------|
| Gzipped source tarball | Source release | | cc8507b3799ed4d8baa7534cd8d5b35f | 25411523 | SIG |
| XZ compressed source tarball | Source release | | 2a3dba5fc75b695c45cf1806156e1a97 | 18900304 | SIG |
| macOS 64-bit Intel installer | Mac OS X | for macOS 10.9 and later | 2b974bfd787f941fb8f80b5b8084e569 | 29866341 | SIG |
| macOS 64-bit universal2 installer | Mac OS X | for macOS 10.9 and later, including macOS 11 Big Sur on Apple Silicon (experimental) | 9aa68872b9582c6c71151d5dd4f5ebca | 37648771 | SIG |
| Windows embeddable package (32-bit) | Windows | | b4bd8ec0891891158000c6844222014d | 7580762 | SIG |
| Windows embeddable package (64-bit) | Windows | | 5c34eb7e79cfe8a92bf56b5168a459f4 | 8419530 | SIG |
| Windows help file | Windows | | aaacfe224768b5e4aa7583c12af68fb0 | 8859759 | SIG |
| Windows installer (32-bit) | Windows | | b790fdaff648f757bf0f233e4d05c053 | 27222976 | SIG |
| Windows installer (64-bit) | Windows | Recommended | ebc65aaa142b1d6de450ce241c50e61c | 28323440 | SIG |

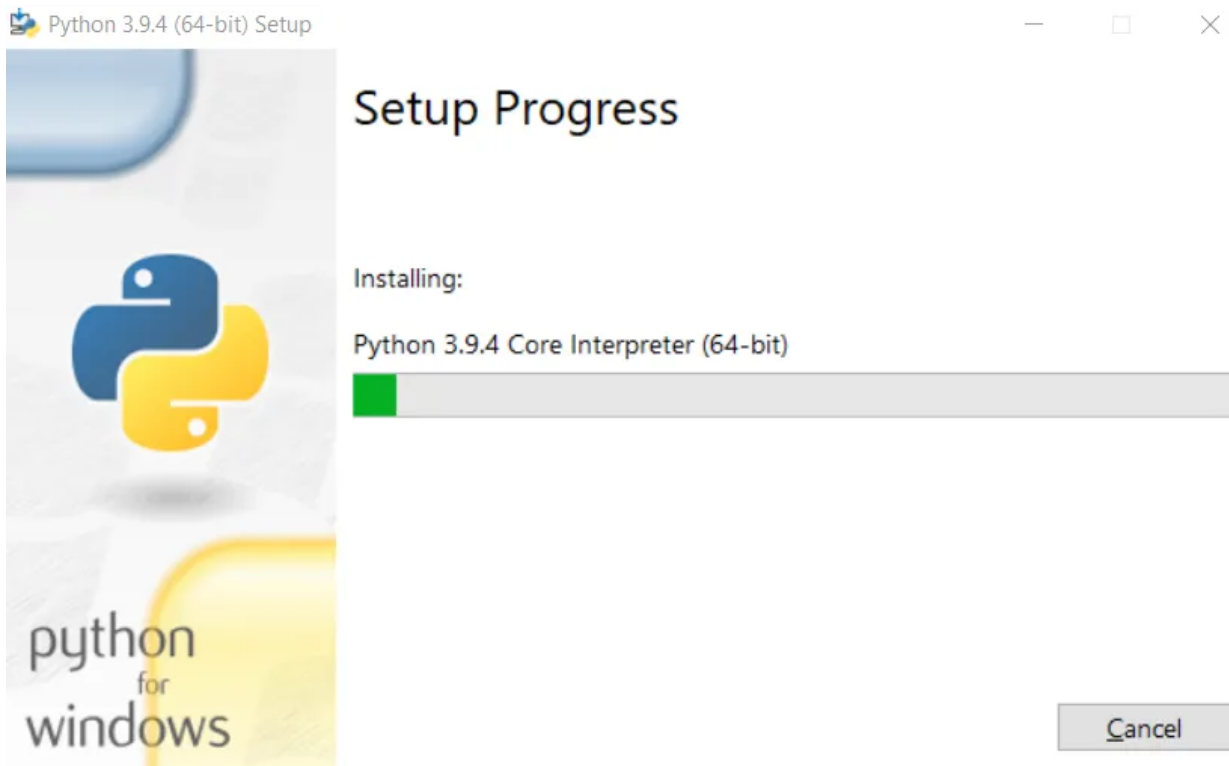
Etapa 2: Baixe o *Python Executable Installer* e instale-o

Clique duas vezes no arquivo executável, que é baixado; a seguinte janela será aberta. Selecione Personalizar instalação e prossiga. Clique na caixa de seleção Adicionar caminho, isso definirá o caminho do Python automaticamente.

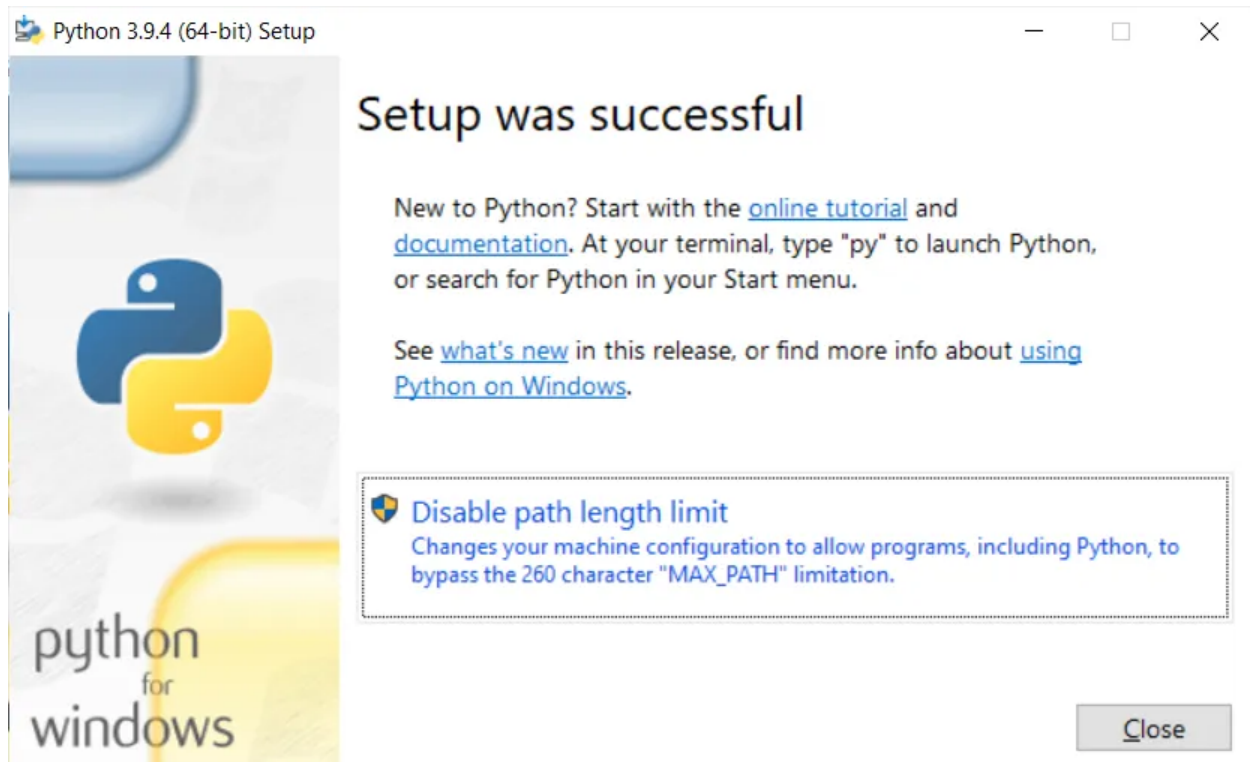
1. Execute o instalador do Python depois de baixado. (Neste exemplo, baixamos o Python 3.9.4)
2. Certifique-se de selecionar as caixas de seleção “Install launcher for all users” e “Add Python 3.9 to PATH”.
3. Selecione Instalar agora — as opções de instalação recomendadas.



Etapa 3: aguarde a conclusão do processo de instalação



A próxima caixa de diálogo solicitará que você selecione se deseja desativar o limite de comprimento do caminho. A escolha dessa opção permitirá que o Python ignore o limite de **MAX_PATH** de 260 caracteres. Efetivamente, isso permitirá que o Python use nomes de caminho longos.



A opção Desativar limite de comprimento do caminho não afetará nenhuma outra configuração do sistema. Ativá-lo resolverá possíveis problemas de tamanho de nome que podem surgir com projetos Python desenvolvidos no Linux.

Etapa 4: Verificação da instalação do python no Windows

Para verificar se o Python 3.9.4 foi instalado com sucesso em nosso sistema,

- Open Cmd prompt in your system
- Run "Python -V"

```
Command Prompt
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Win10>python --version
Python 3.9.4

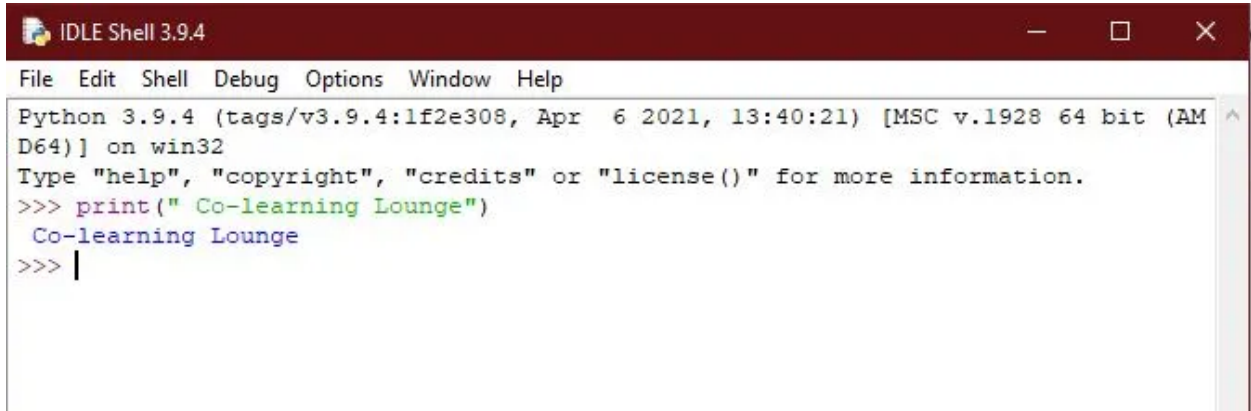
C:\Users\Win10>
```

Etapa 5: execute o Python

Python foi instalado em seu sistema, agora vá para

1. Pesquisa do Windows
2. Digite Idle
3. Abra.

Execute seu primeiro código Python

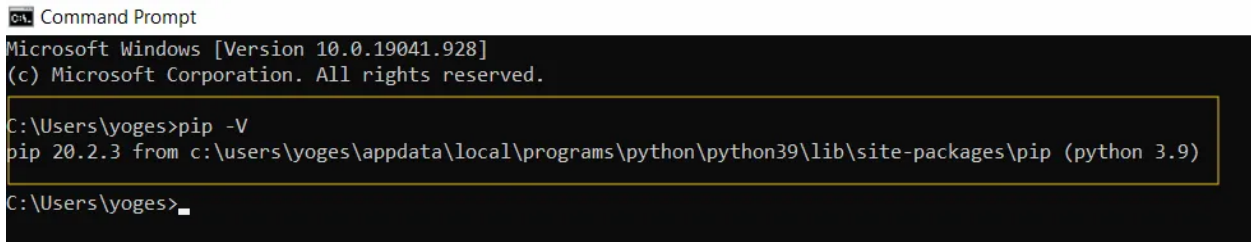


```
IDLE Shell 3.9.4
File Edit Shell Debug Options Window Help
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 6 2021, 13:40:21) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print(" Co-learning Lounge")
Co-learning Lounge
>>> |
```

Etapa 6: Verifique se o Pip foi instalado

Se você optou por instalar uma versão mais antiga do Python, é possível que ela não tenha vindo com o Pip pré-instalado. Pip é um poderoso sistema de gerenciamento de pacotes para pacotes de software Python. Portanto, certifique-se de tê-lo instalado.

Recomendamos o uso do Pip para a maioria dos pacotes Python, especialmente ao trabalhar em ambientes virtuais.



```
Command Prompt
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\yoges>pip -V
pip 20.2.3 from c:\users\yoges\appdata\local\programs\python\python39\lib\site-packages\pip (python 3.9)

C:\Users\yoges>
```

O Pip ainda não foi instalado se você obtiver a seguinte saída:

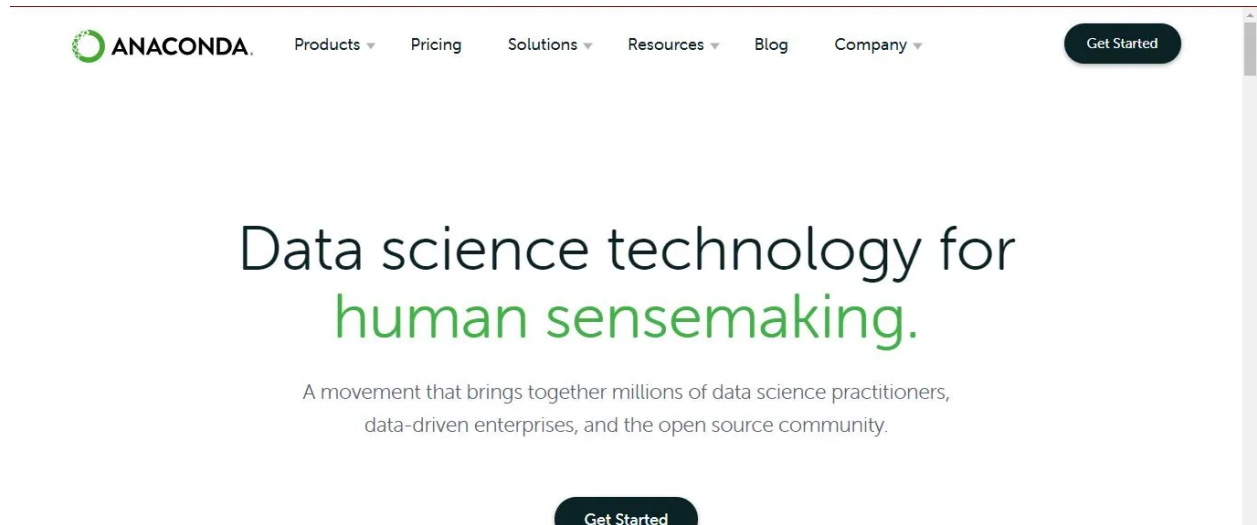
```
>'pip' is not recognized as an internal or external command,
Operable program or batch file.
```


Instale o Python usando o navegador Anaconda

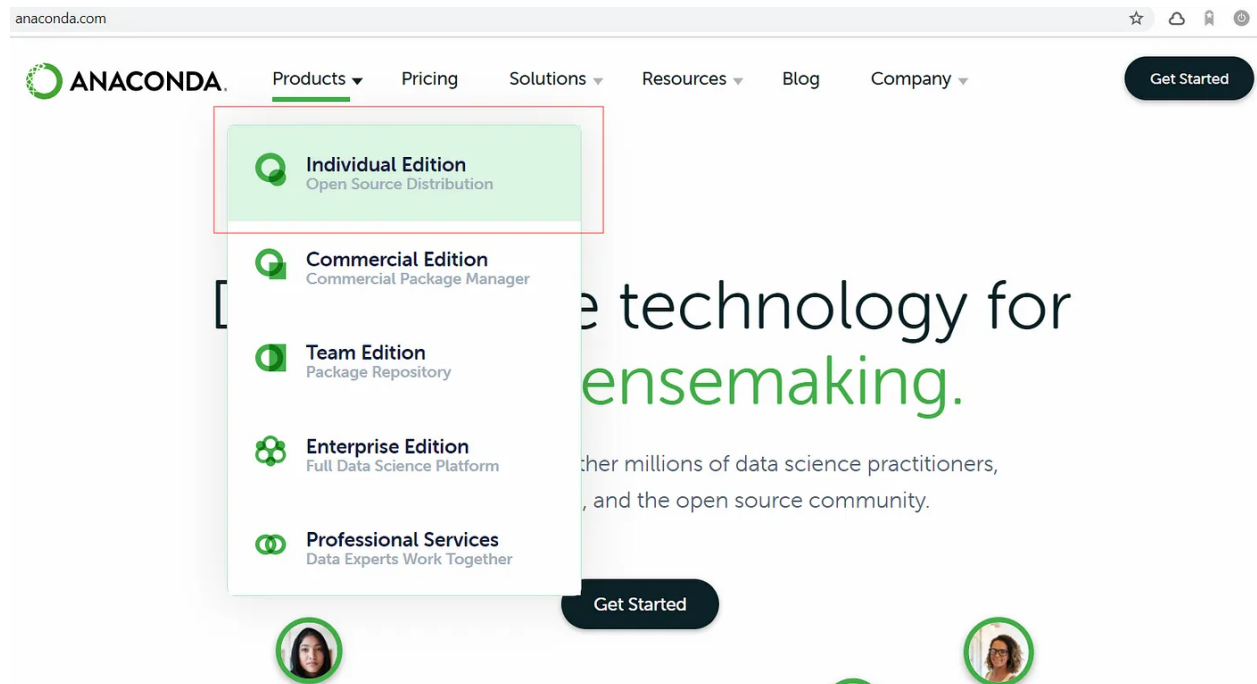
O Anaconda Navigator é uma interface GUI de desktop que vem com o Anaconda Individual Edition. Ele facilita o lançamento de aplicativos e o gerenciamento de pacotes e ambientes sem usar comandos de linha de comando (CLI).

Etapa 1: visite o site oficial do anaconda

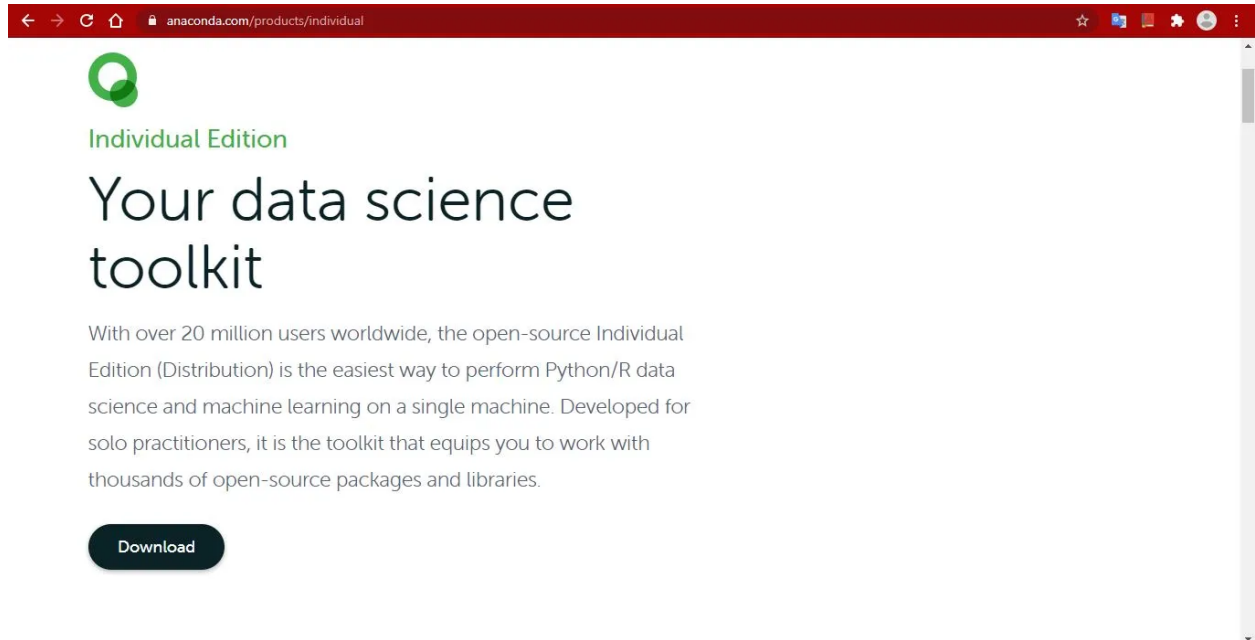
Download: [Anaconda](https://anaconda.com)



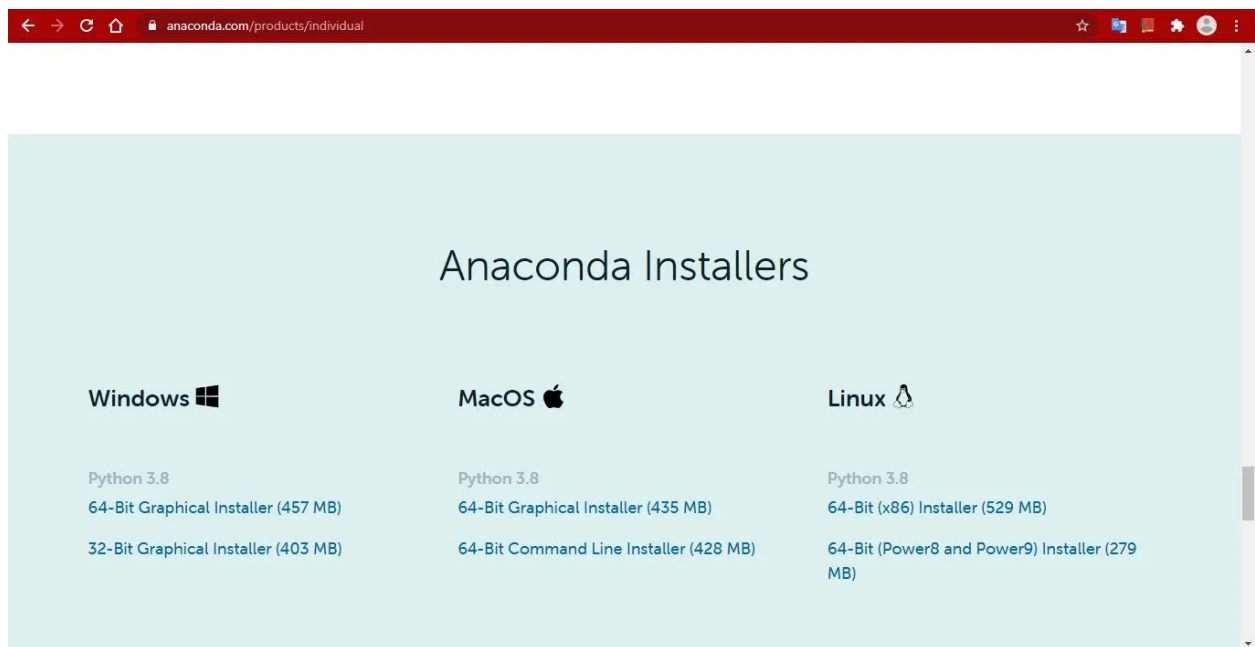
Etapa 2: selecione a distribuição de código aberto



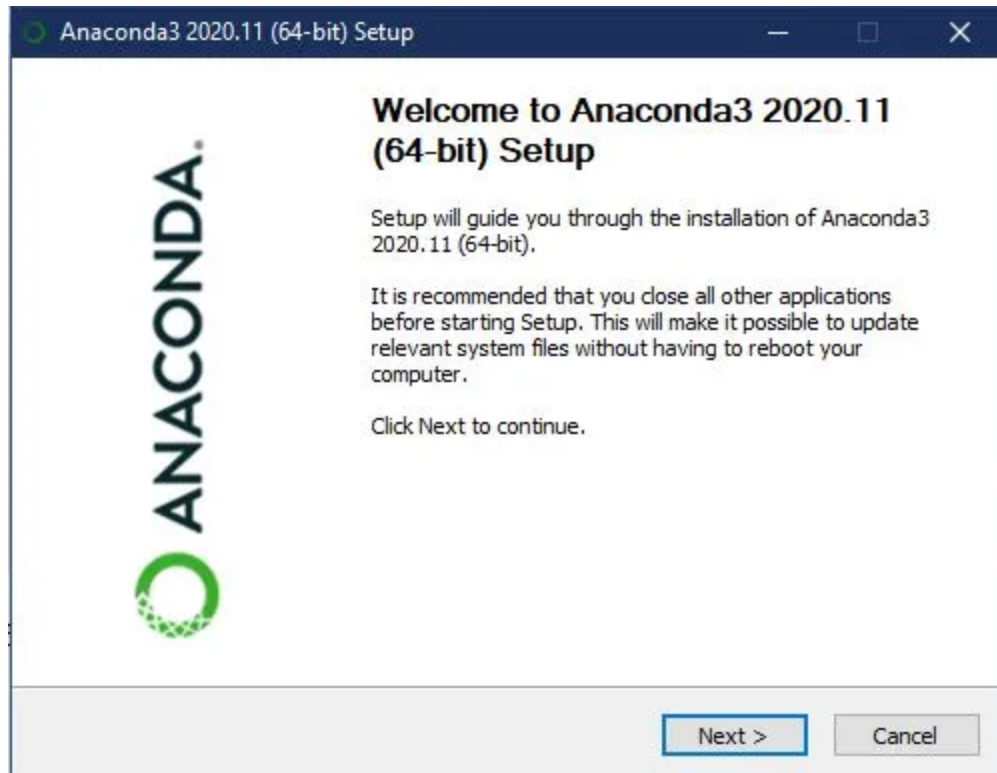
Passo 3: Clique no botão de download



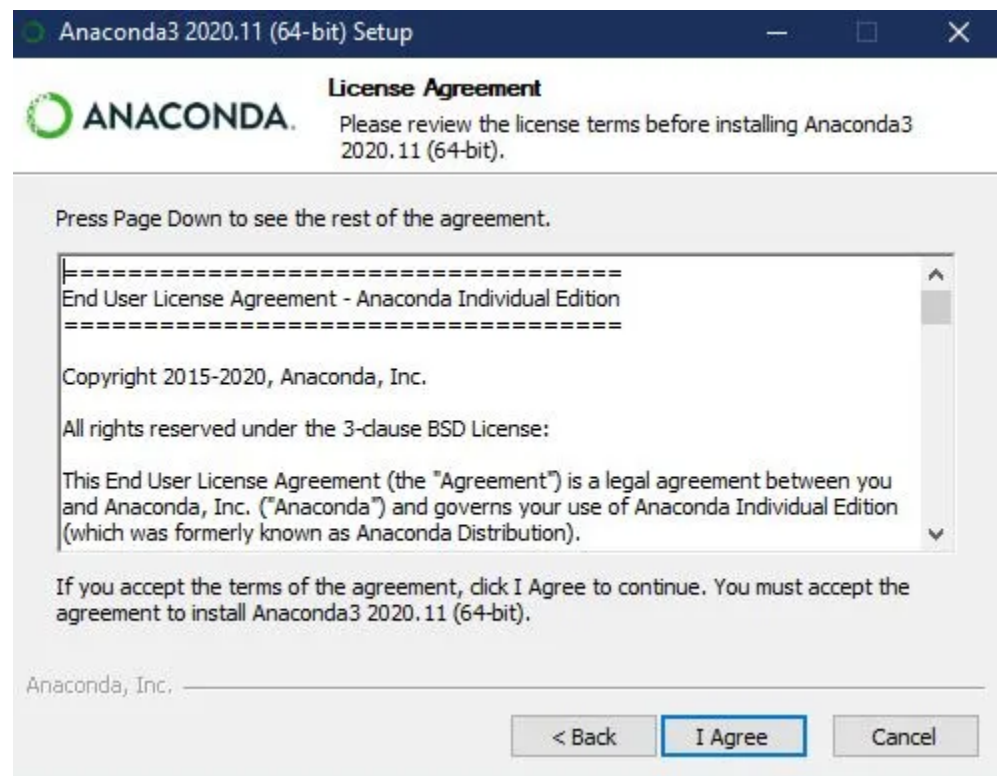
Etapas 4: selecione seu sistema operacional e faça o download de um pacote

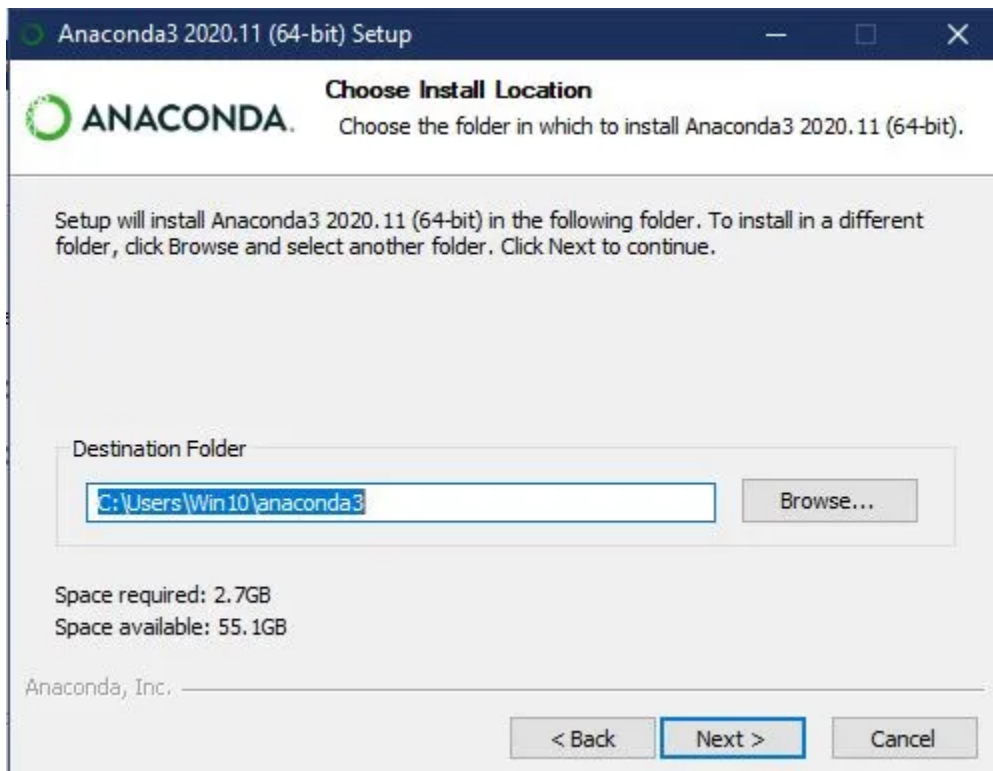
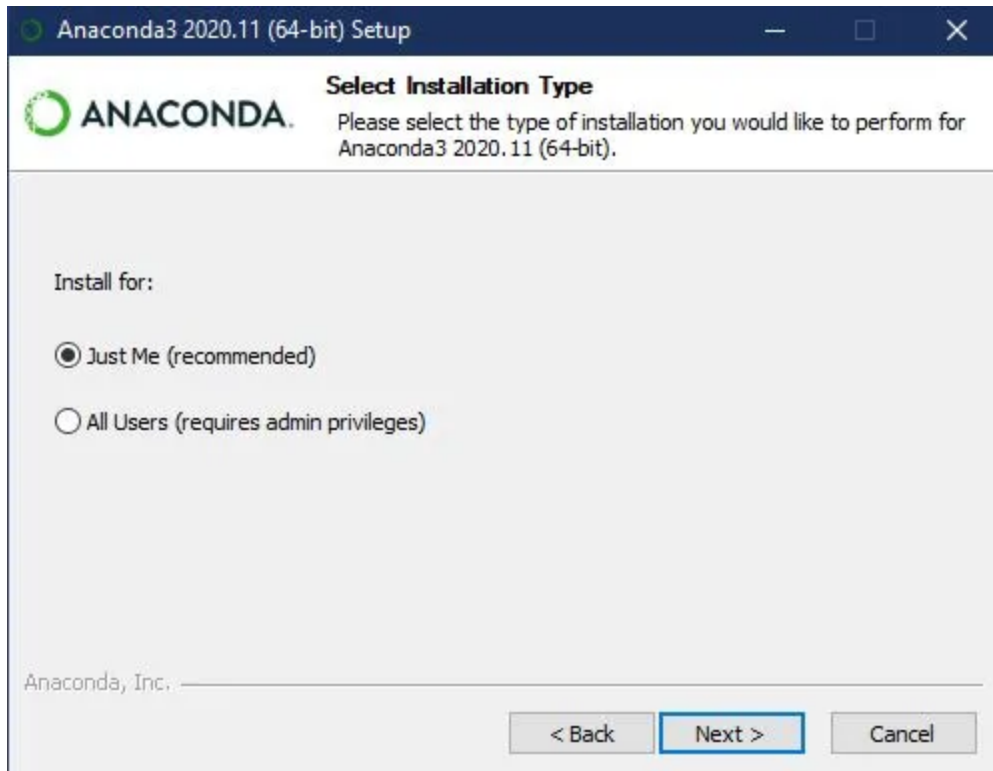


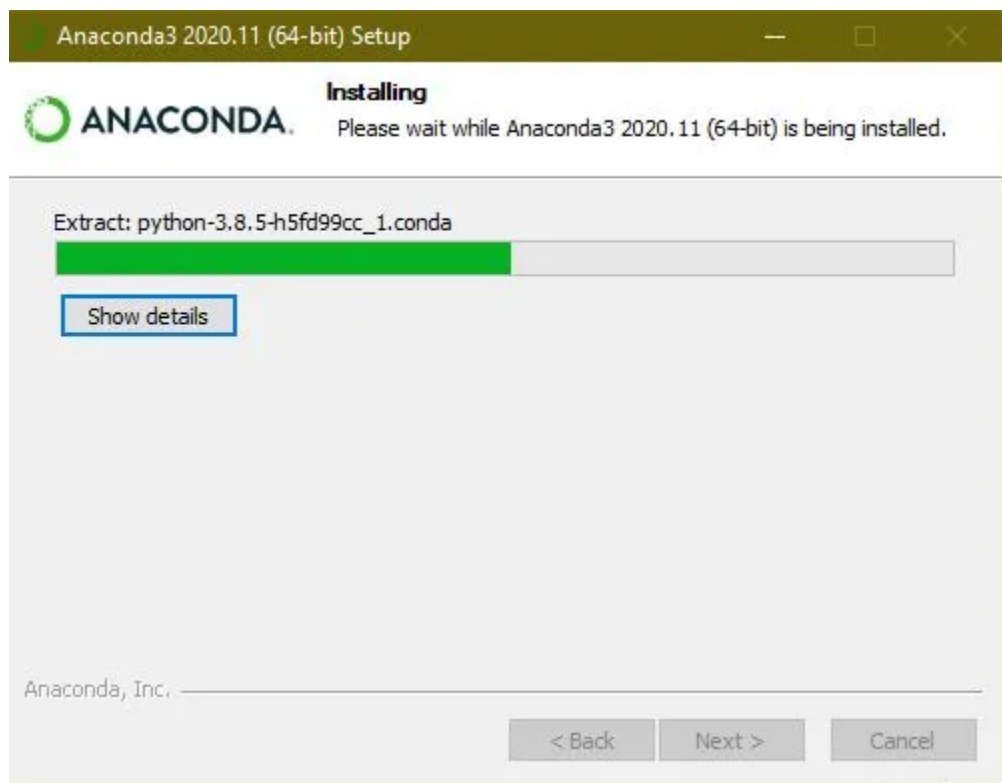
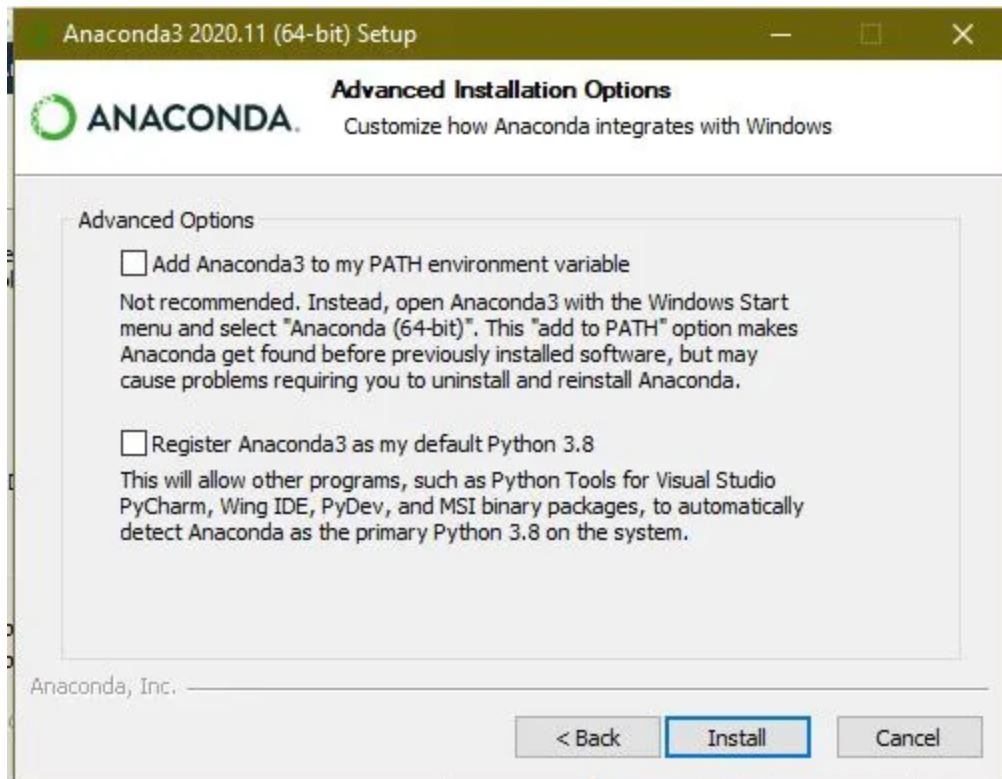
Etapas 5: após o download do instalador, abra-o

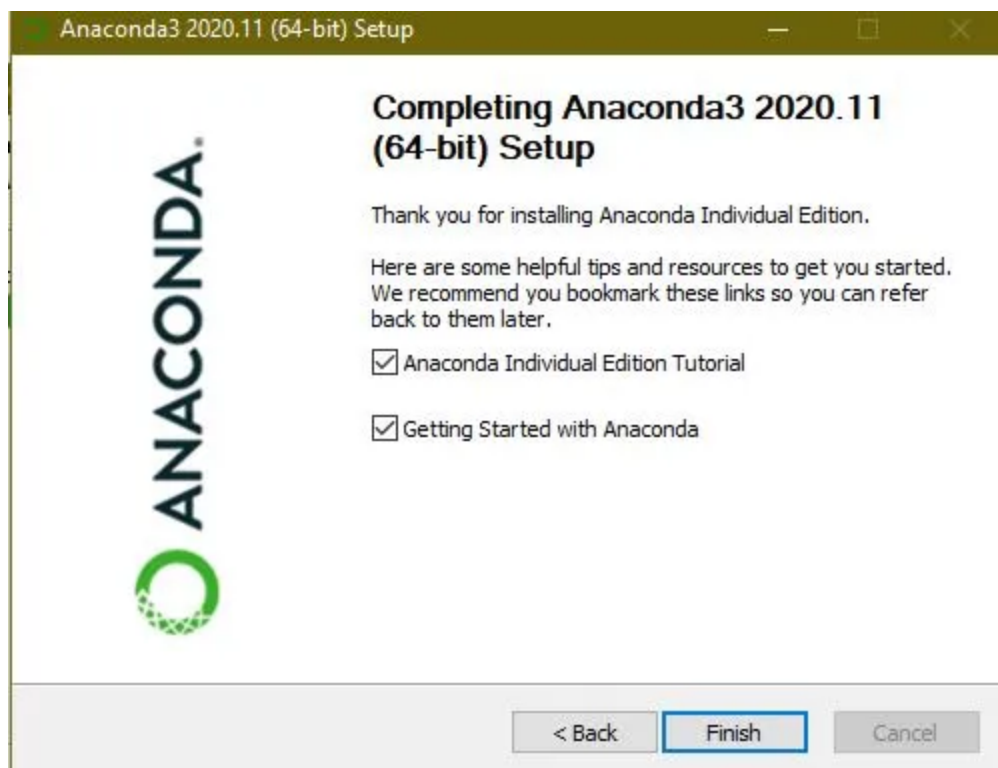


Passo 6: Inicie a instalação em seu sistema

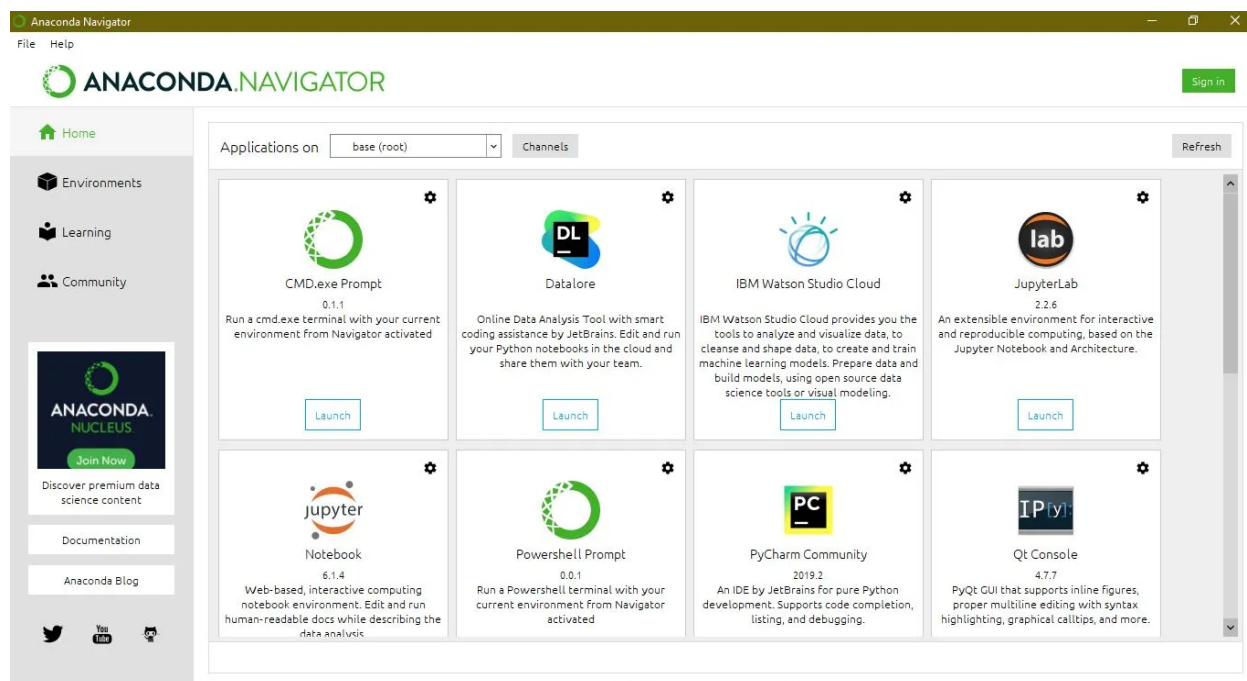








Passo 7: Parabéns! ANACONDA foi instalado com sucesso em seu sistema.



Interpretadores Python Online

Por algum ou outro motivo, se você não estiver configurando o Python em sua máquina, existem vários sites que oferecem um interpretador Python online:

- [Python.org Online Console](#)
- [Repl.it](#)
- [Trinket](#)
- [Python Anywhere](#)

Observação: esses interpretadores Python baseados em nuvem podem não ser capazes de executar alguns dos exemplos mais complexos, mas são adequados para executar a maior parte do código e talvez uma boa maneira de começar.

Instalando o Python 3 no Linux

Se você estiver usando o Ubuntu 16.10 ou mais recente, poderá instalar facilmente o Python 3.6 com os seguintes comandos:

```
$ sudo apt-get update
$ sudo apt-get install python3.6
```

Se você estiver usando outra versão do Ubuntu (por exemplo, a versão LTS mais recente) ou quiser usar um Python mais atual, recomendamos usar o [PPA deadsnakes](#) para instalar o Python 3.8:

```
$ sudo apt-get install software-properties-common
$ sudo add-apt-repository ppa:deadsnakes/ppa
$ sudo apt-get update
$ sudo apt-get install python3.8
```

Se você estiver usando outra distribuição Linux, é provável que já tenha o Python 3 pré-instalado também. Caso contrário, use o gerenciador de pacotes da sua distribuição. Por exemplo, no Fedora, você usaria dnf:

```
$ sudo dnf install python3
```

Setuptools & Pip

Os dois pacotes Python de terceiros mais cruciais são [setuptools](#) e [pip](#). Uma vez instalado, você pode baixar, instalar e desinstalar qualquer produto de software Python compatível com um único comando. Ele também permite adicionar esse recurso de instalação de rede ao seu próprio software Python com muito pouco trabalho. Para ver se o pip está instalado, abra um prompt de comando e execute:

```
$ command -v pip
```

Observe que em algumas distribuições Linux, incluindo Ubuntu e Fedora, o comando pip é destinado ao Python 2, enquanto o comando pip3 é destinado ao Python 3.

Jupyter Notebook

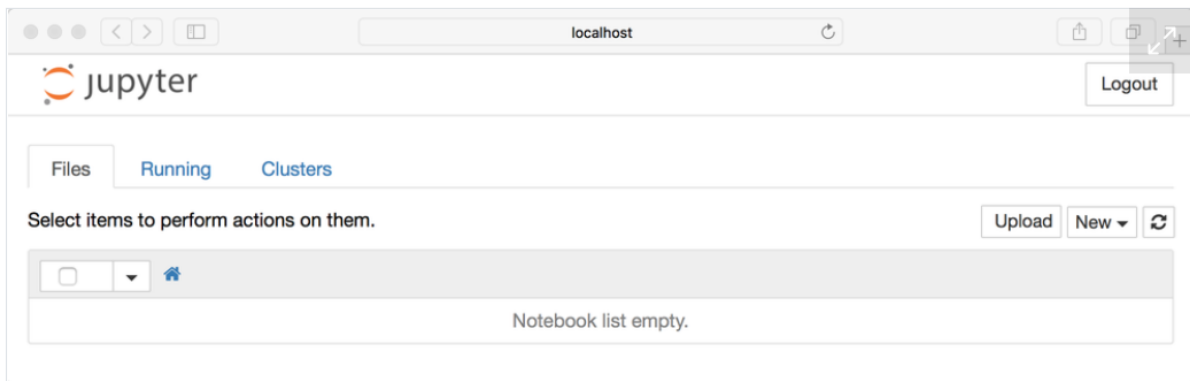
O Jupyter Notebook é um aplicativo da web de código aberto que você pode usar para criar e compartilhar documentos que contêm código ativo, equações, visualizações e texto. Para instalar o Jupyter, utilizamos o **conda** com o seguinte comando:

```
> conda install jupyter
```

Agora que você instalou o Jupyter, vamos aprender como usá-lo. Para começar, tudo que você precisa fazer é abrir seu terminal e ir para uma pasta de sua escolha. Eu recomendo usar uma pasta de documentos para o curso i.e., INF-1010. Em seguida, vá até esse local em seu terminal e execute o seguinte comando:

```
Shell
$ jupyter notebook
```

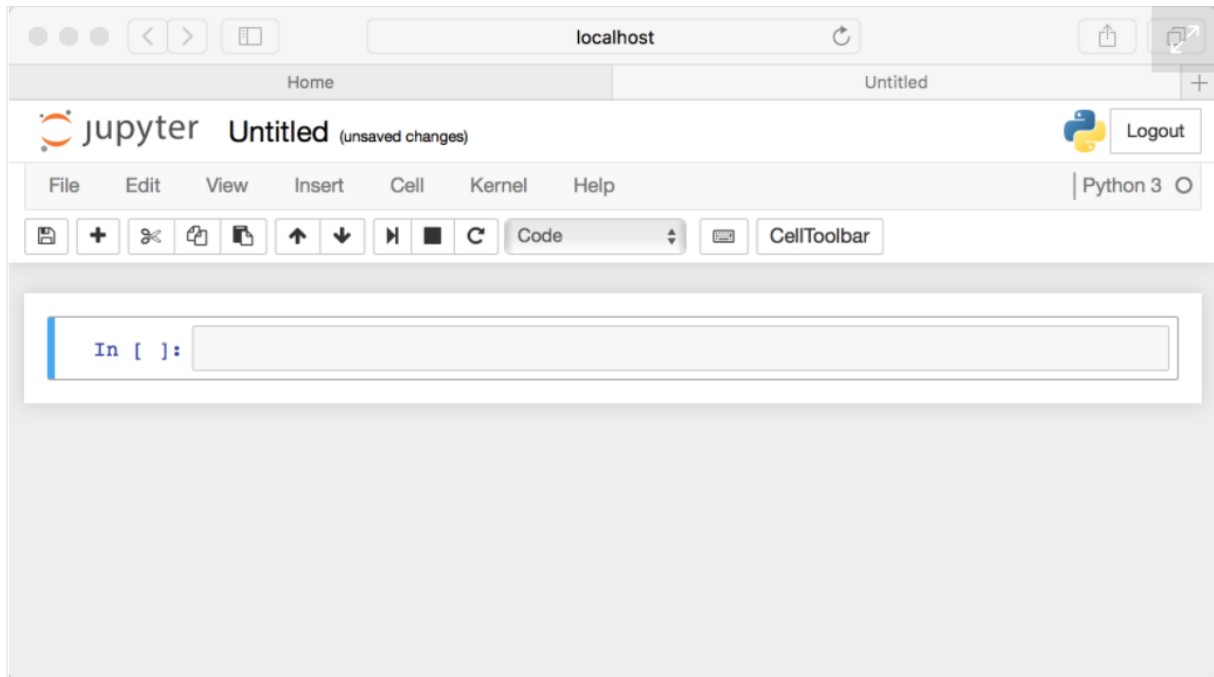
Isso iniciará o Jupyter e seu navegador padrão deve iniciar (ou abrir uma nova guia) no seguinte URL: <http://localhost:8888/tree>. Seu navegador agora deve ser parecido com isto:



Observe que agora você não está realmente executando um Notebook, mas apenas executando o servidor Notebook. Vamos criar um Notebook agora!

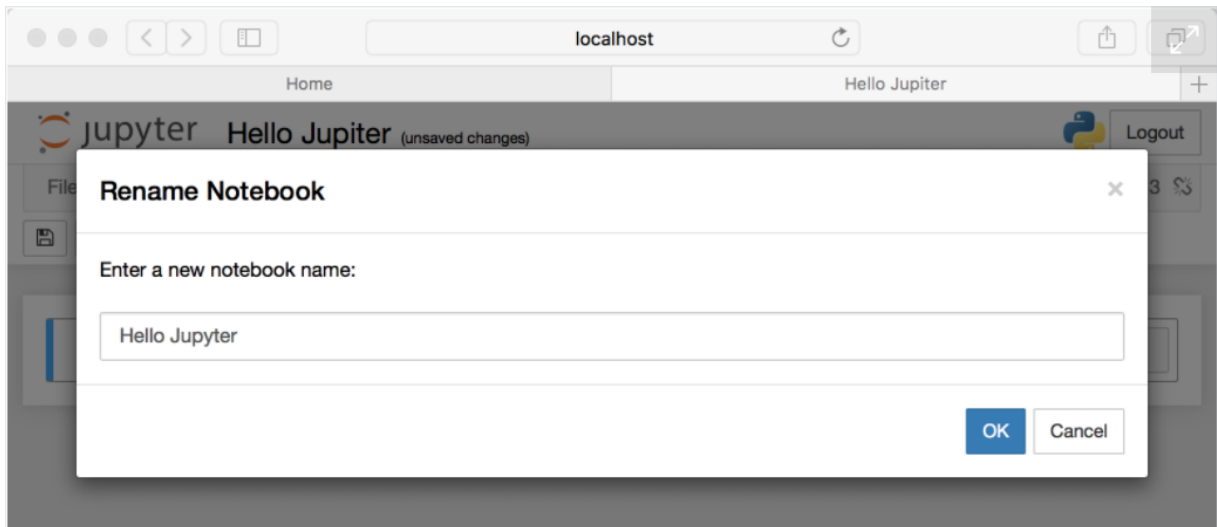
Criando um Notebook

Agora que você sabe como iniciar um servidor Notebook, provavelmente deve aprender como criar um documento real do Notebook. Tudo que você precisa fazer é clicar no botão Novo (canto superior direito), e uma lista de opções será aberta, vamos escolher Python 3. Assim, sua página da web agora deve ter a seguinte aparência:



Nomeando o Notebook

Você notará que no topo da página está a palavra Sem título. Este é o título da página e o nome do seu Notebook. Já que esse não é um nome muito descritivo, vamos mudá-lo! Basta passar o mouse sobre a palavra Sem título e clicar no texto. Agora você deve ver uma caixa de diálogo no navegador intitulada Renomear Bloco de anotações. Vamos renomear este para Hello Jupyter:



Executando as Células

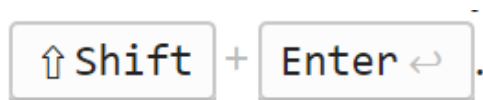
O padrão da célula de um Notebook é usar código sempre que você cria um pela primeira vez, e essa célula usa o kernel que você escolheu quando iniciou seu Notebook. Nesse caso, você iniciou o seu notebook com Python 3 como seu kernel, o que significa que você pode escrever código Python em suas células de código. Uma vez que seu Notebook inicial tem apenas uma célula vazia, o Notebook realmente não pode fazer nada.

Portanto, para verificar se tudo está funcionando como deveria, você pode adicionar algum código Python à célula e tentar executar seu conteúdo. Vamos tentar adicionar o seguinte código a essa célula:

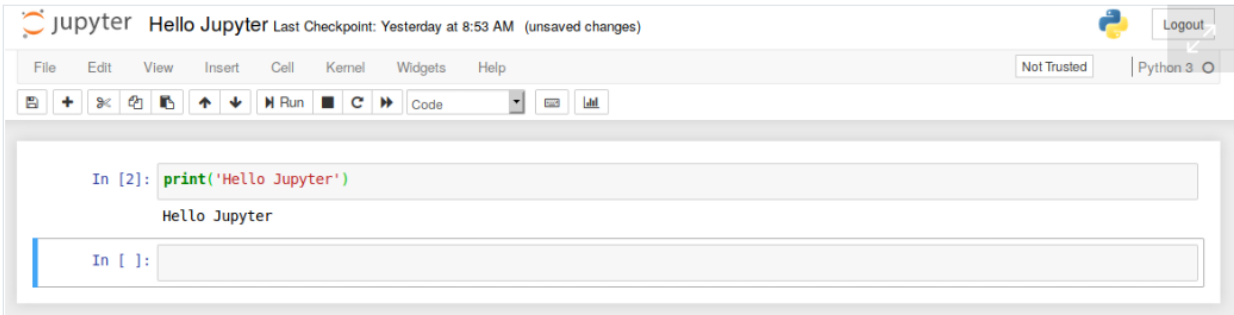
```
Python

print('Hello Jupyter!')
```

Executar uma célula significa que você executará o conteúdo da célula. Para executar uma célula, você pode apenas selecionar a célula e clicar no botão Executar que está na fileira de botões na parte superior. Se preferir usar o teclado, basta pressionar



Quando executar o código acima, a saída deve ficar assim:



Se você tiver várias células em seu Bloco de anotações e executar as células em ordem, poderá compartilhar suas variáveis e importações entre células. Isso torna mais fácil separar seu código em blocos lógicos sem a necessidade de re-importar bibliotecas ou re-criar variáveis ou funções em cada célula.

Ao executar uma célula, você notará que há alguns colchetes próximos à palavra **In []:** à esquerda da célula. Os colchetes serão preenchidos automaticamente com um número que indica a ordem em que você executou as células. Por exemplo, se você abrir um novo Notebook e executar a primeira célula na parte superior do Notebook, os colchetes serão preenchidos com o número 1.

Ver o que está em execução

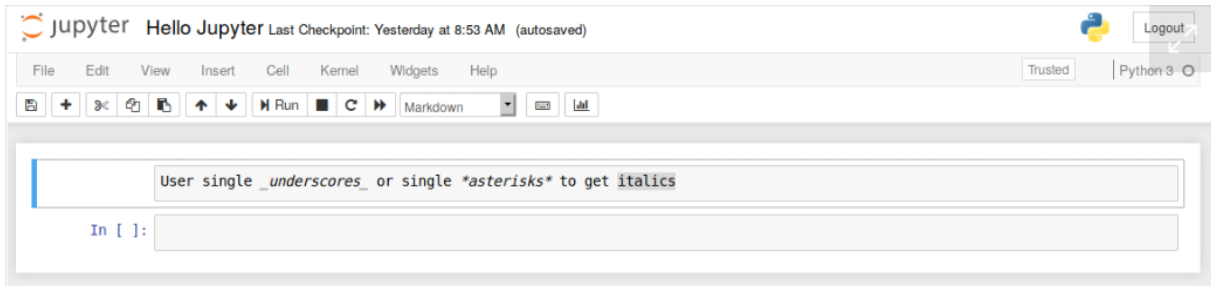
Na página inicial do seu servidor Jupyter (<http://localhost:8888/tree>) estão duas outras guias: **Running** e **Clusters**. A guia **Running** informará quais Notebooks e Terminais você está executando no momento. Isso é útil quando você deseja desligar o servidor, mas precisa ter certeza de que salvou todos os seus dados. Felizmente, os notebooks salvam automaticamente com bastante frequência, então você raramente perde dados. Mas é bom poder ver o que está funcionando quando você precisar. Outra coisa boa sobre esta guia é que você pode passar por seus aplicativos em execução e fechá-los lá.

Tipos de células

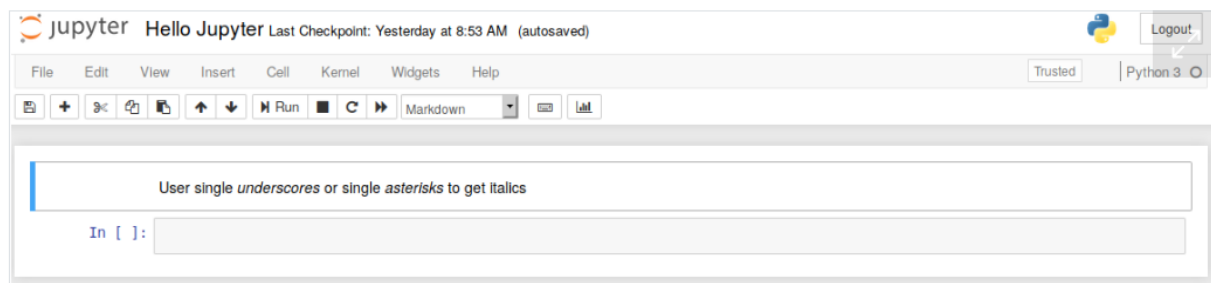
Os principais tipos de células que você usará são os tipos de células **Code** e **Markdown**. Você já aprendeu como as células de código funcionam, então vamos aprender como estilizar seu texto com **Markdown**.

O Jupyter Notebook suporta Markdown, que é uma linguagem de marcação que é um superconjunto do HTML. Este tutorial cobrirá alguns dos princípios básicos do que você pode fazer com o Markdown.

Defina uma nova célula para Markdown e, em seguida, adicione o seguinte texto à célula:

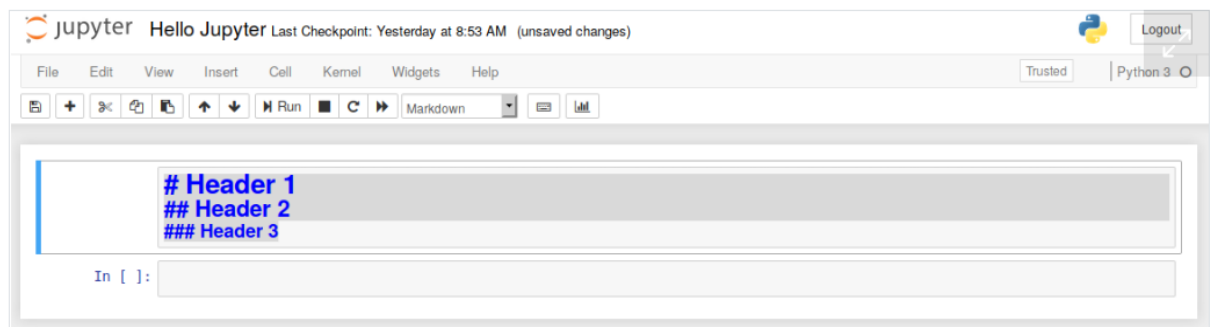


Quando você executar a célula, a saída deve ser semelhante a esta:

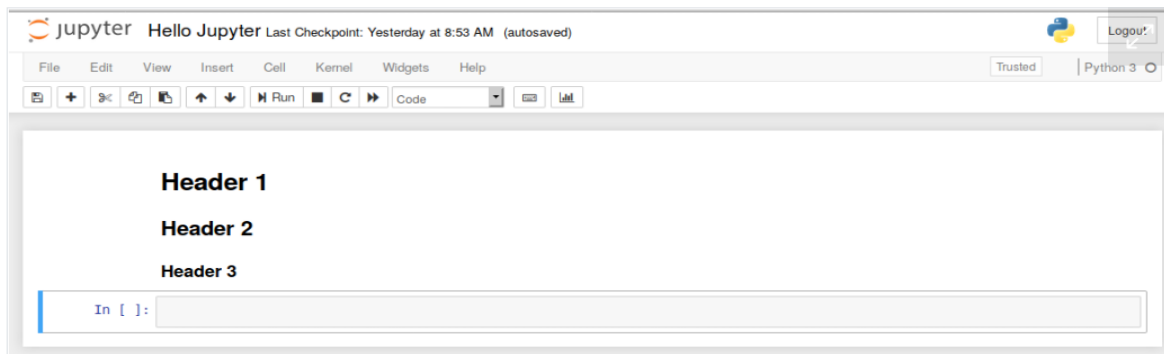


Cabeçalhos

Criar cabeçalhos no Markdown também é bastante simples. Você apenas tem que usar **# (i.e., cerquilha, hashtag)**. Quanto mais sinais de cerquilha você usar, menor será o cabeçalho. O Jupyter Notebook faz uma espécie de pré-visualização para você:



Então, ao executar a célula, você terá um cabeçalho bem formatado:



Exemplos de códigos Markdown e Comandos

Para mais informações sobre de Markdown para editar o texto de seu notebook utilize o link (<https://docs.pipz.com/central-de-ajuda/learning-center/guia-basico-de-markdown#open>). Além disso, o Jupyter Notebook possui um série de comandos que podem ser utilizados para navegar, editar, executar e gerenciar células, veja alguns desses comandos no seguinte link: <https://karloguidoni.com/til/cheatsheets/jupyter-notebook/>