

Fundamentos Matemáticos e Computacionais de Machine Learning

Especialização em Machine Learning e Big Data



Profa. Dra. Juliana Felix

jufelix16@uel.br



Python

Breve revisão ...

- Python é uma linguagem de programação de alto nível, interpretada/*script*, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte.
- Foi lançada por Guido van Rossum em 1991.

Breve revisão ...

- Família 3 - Versão atual: 3.11.3 (5 de abril de 2023).
 - O uso da família 3 é altamente recomendada.
- Família 2 - Versão atual: 2.7.18 (Python 2.7.18).
 - O uso da família 2 não é recomendado, existe apenas para suporte a sistemas legados.

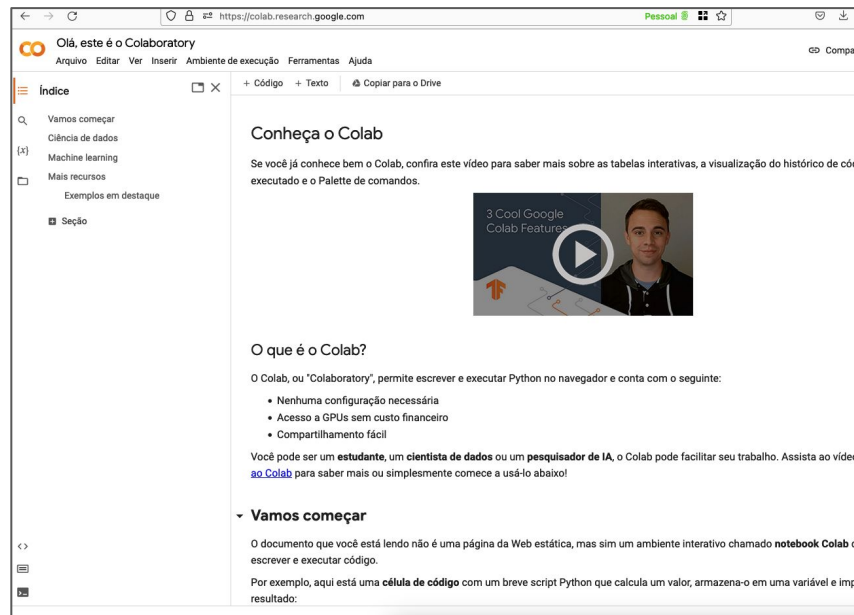
O que eu preciso para usar o Python?

- Atualmente, o Python é uma linguagem muito popular e, portanto, há diversas opções para usá-lo.
- Podemos classificar seu uso de duas formas:
 - Instalação local
 - Instalação remota

Uso Remoto

Para uso remoto, há diversas opções disponíveis.

A mais popular delas é utilizando o [Google Colab](https://colab.research.google.com)



Uso Remoto

Há ainda opções menos populares, como:

- OnlineGDB - https://www.onlinegdb.com/online_python_compiler
- OnlinePython - <https://www.online-python.com/>
- Replit - <https://replit.com/languages/python3>
- WayScript - <https://www.wayscript.com/>
- Pytutor - <https://pythontutor.com/>
- PythonAnyware - <https://www.pythonanywhere.com/>
- E muitos outros

Uso Remoto

Muitas das ferramentas de Python online disponíveis são gratuitas, porém cada uma tem suas limitações

- Ex: bibliotecas precisam ser instaladas a cada inicialização do sistema, não oferece terminal ou é limitado, etc

Muitas das ferramentas de Python online disponíveis são pagas, e o custo pode iniciar em zero, e escalar à medida que suas necessidades escalam

- Ex: instalar bibliotecas, visualizar gráficos, execução de longo prazo, etc.

Portanto, a escolha da ferramenta deve considerar todas as variáveis em questão antes de começar a usar e depender de alguma delas.

Uso Local

Se você usa ambientes baseados em Windows

- Você provavelmente irá precisar instalar o Python.

Se você estiver utilizando sistemas baseados em Linux, ou MacOS

- Provavelmente, você tem alguma versão de Python no seu computador.

De qualquer forma, para se trabalhar com Aprendizado de Máquina será necessário instalar diversas **bibliotecas** Python.

Uso Local

Instalar bibliotecas no Python que já vem em seu ambiente pode não ser muito interessante, e pode até criar inconveniências, pois pode "quebrar" coisas que dependam da versão disponível.

Além disso, há bibliotecas que podem depender de versões específicas do Python, ou de versões diferentes das bibliotecas pré-instaladas.

Assim, para uso local, a instalação do Python é altamente recomendada.

O que preciso para instalar o Python?

Considere usar alguma distribuição de Python disponível!

- Atualmente o Python é oficialmente distribuído na página: <https://www.python.org/downloads/> , onde é possível baixar versões para Windows e MacOS
- Instalações para Linux são comumente feitas com gerenciadores de pacotes da distribuição (APT, YUM, etc)
- Nesta página é possível baixar todas (ou quase todas) as versões de Python existentes.
 - Considere que você irá partir da versão 3.<alguma>.<coisa>
 - Versões anteriores à 3 não são mantidas e ou recomendadas

O que preciso para instalar o Python

- Todavia, para trabalhar com Aprendizado de Máquina, há bibliotecas do Python que dependem de Frameworks não Python (C, C++, Java, etc)
- Assim, usar a distribuição padrão pode trazer alguma dificuldade extra (ou muita)
- Para isto, uma alternativa bastante popular é o Anaconda!



O que preciso para instalar o Python

- O Anaconda é uma das distribuições do tipo CONDA, que foca em instalar o Python e bibliotecas desejadas.
- O Anaconda traz dentro de seu repositório bibliotecas não Python (OpenCV, OpenCL, etc) e auxilia na instalação destas de forma simplificada.



O que preciso para instalar o Python

- A instalação de bibliotecas em Python é geralmente feita usando algum gerenciador de pacotes.
- O mais famoso é o PIP (Preferred Installer Program - Instalador de Programas Preferido)
- O PIP instala os pacotes da fonte Python Package Index ou PYPI
 - <https://pypi.org/>

O que preciso para instalar o Python

- O PYPI contém uma infinidade de pacotes, bibliotecas, etc, em diversas versões.
- Portanto, instalar a partir deste repositório parece uma excelente opção.
- Porém ...

O que preciso para instalar o Python

- Algumas bibliotecas podem "brigar" entre si, devido à versão de Python em que foi desenvolvida, ou outras bibliotecas de que a biblioteca dependa.
- Portanto, instalar "cegamente" a partir do PYPI com o PIP pode se tornar uma grande dor de cabeça em um prazo muito curto.
 - Principalmente se você for iniciante.

O que preciso para instalar o Python

- Além disso, nem todas as bibliotecas de Aprendizado de Máquina precisam estar juntas, pois depende do estudo e ou ferramenta que se estiver produzindo.
 - Todavia, você não precisa ficar desinstalando e instalando bibliotecas sempre que iniciar um novo projeto.
 - Para isso, existem diversos gerenciadores de ambientes Python, que permitem montar um conjunto adequado de versão do Python e bibliotecas.
- São os chamados Ambientes Virtuais!

O que preciso para instalar o Python

Ambiente Virtual é um ambiente Python isolado, onde as dependências de um projeto são instaladas em diretórios diferentes entre ambientes e entre o Python padrão instalado no sistema.

O que preciso para instalar o Python

Para gerenciar ambientes em Python existem diversas ferramentas.

- Venv
- Virtualenv
- Pipenv
- **Conda**
- Poetry

Observe que o Conda é uma das opções, logo o Anaconda traz mais outra vantagem!

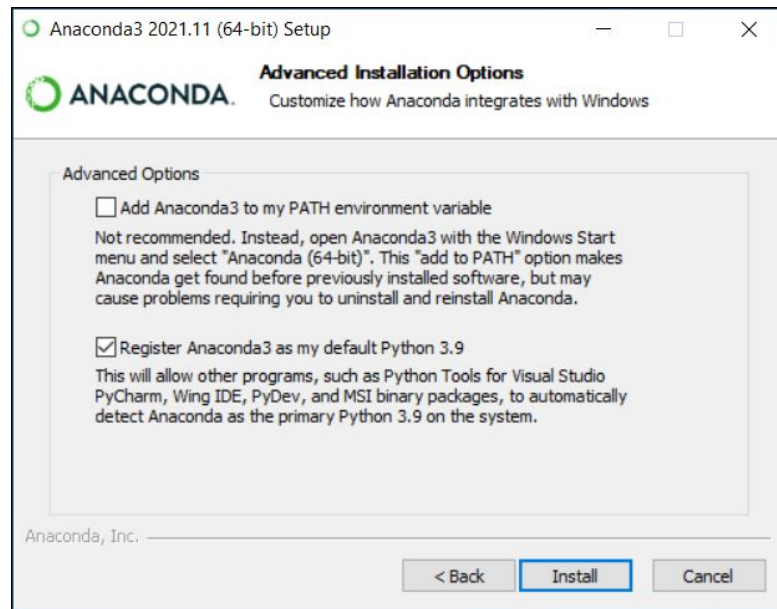




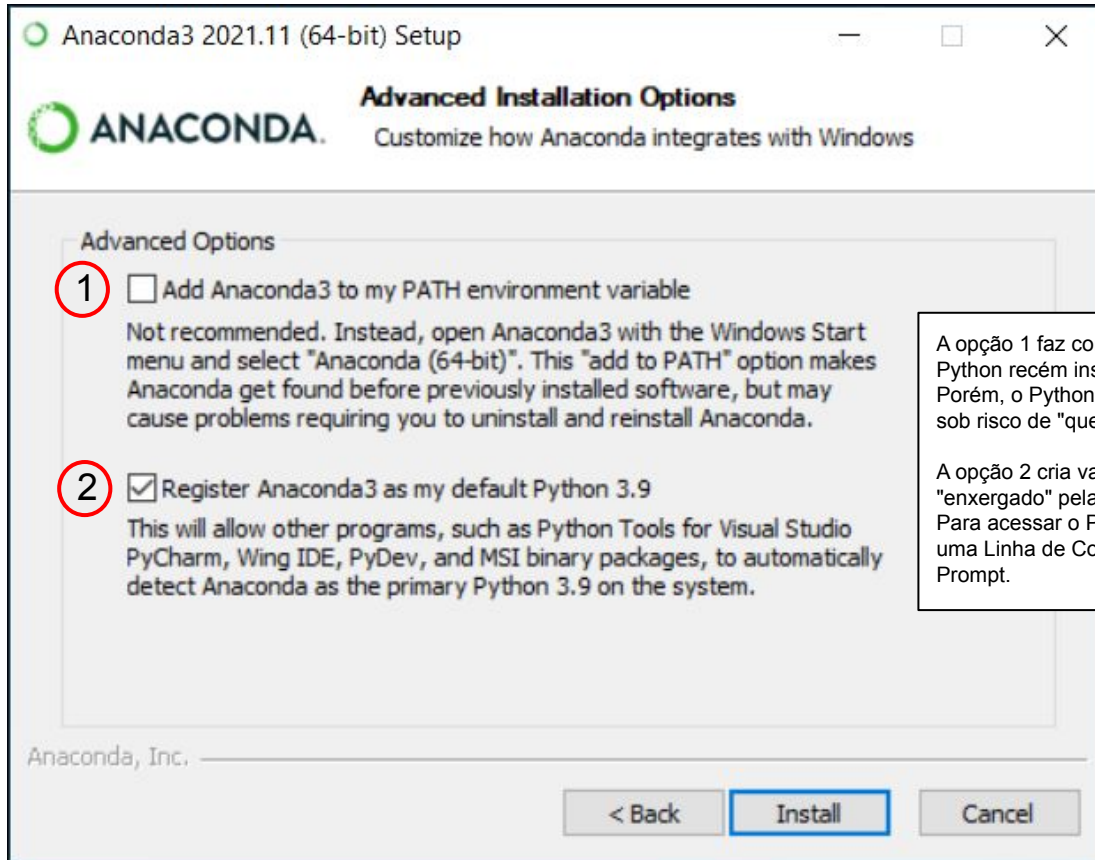
Instalando o Anaconda

Instalação do Anaconda no Windows

- Basta "baixar" o .exe instalador na página do Anaconda
- A instalação no Windows segue o famoso modelo:
 - Executar, próximo, próximo, ..., Finalizar
- Ainda, é importante ficar atento a uma tela similar a esta ao lado.
- Maiores detalhes podem ser obtidos aqui:
 - <https://docs.anaconda.com/free/anaconda/install/windows/>



Instalação do Anaconda no Windows



A opção 1 faz com que, ao abrir um terminal, os comandos do Python recém instalados estejam disponíveis. Porém, o Python padrão que já esteja no Windows será sobreposto, sob risco de "quebrar" algo.

A opção 2 cria variáveis especiais para acesso ao Python novo ser "enxergado" pelas suas ferramentas, mas não na linha de comando. Para acessar o Python na linha de comando será necessário abrir uma Linha de Comando especial para o Python chamado Anaconda Prompt.

Instalação do Anaconda no MacOS

Basta "baixar" o .pkg na página do Anaconda (escolha Intel ou M1/M2). A instalação do Anaconda no MacOS também segue o famoso modelo:

- Executar, próximo, próximo, ..., Finalizar

Sem maiores perguntas, exceto onde instalar

- Para você
- Para todos

Também é possível instalar via linha de comando.

- Detalhes podem ser obtidos aqui:
 - <https://docs.anaconda.com/free/anaconda/install/mac-os/>

Instalação do Anaconda no Linux

- Basta "baixar" o .sh na página do Anaconda A instalação do
- A partir do terminal, execute o "script" instalador.
- Detalhes podem ser obtidos aqui:
 - <https://docs.anaconda.com/free/anaconda/install/mac-os/>

Instalação do Anaconda em outros ambientes

Tutorial de instalação:

<https://docs.anaconda.com/free/anaconda/install/>

Vídeo de tutorial.

Legenda em português (tradução automática) disponível no vídeo.

Linux: <https://www.youtube.com/watch?v=MUZtVEDKXsk&t=7s>

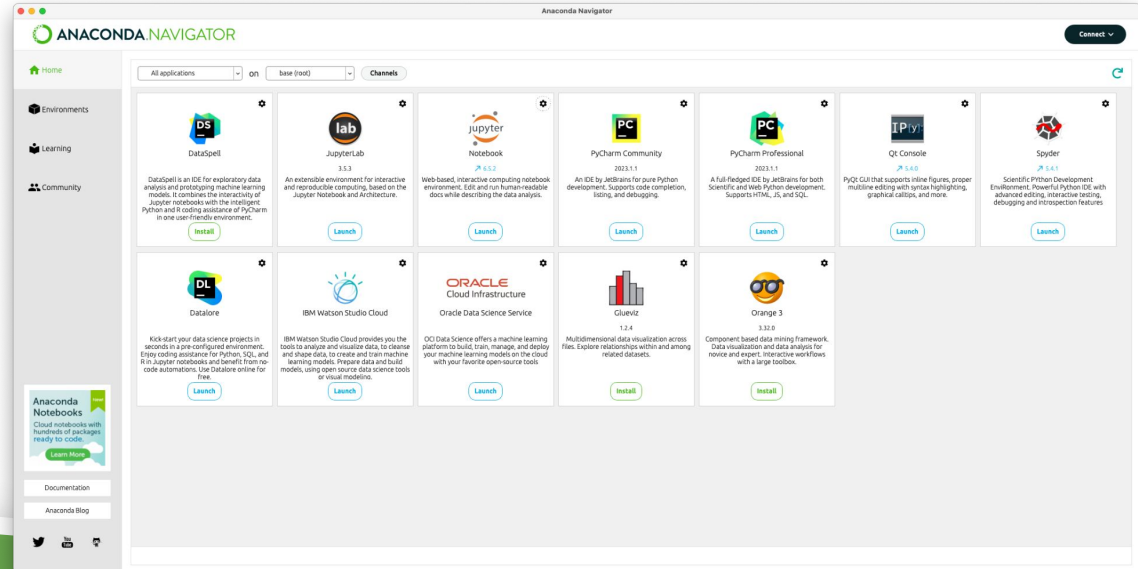
Windows:

<https://www.youtube.com/watch?v=MUZtVEDKXsk&t=695s>

Gerenciando os ambientes no Anaconda

Quando instalado, o Anaconda traz uma ferramenta visual para auxílio de instalação de pacotes e gerenciamento de ambientes

- Anaconda Navigator.



Gerenciando os ambientes no Anaconda

Uma vez instalado, o anaconda cria o ambiente *base (root)*.

- Todavia, para fins de criar seus projetos, recomenda-se criar outros ambientes com versões específicas do Python e instalar dentro deles suas bibliotecas.

Gerenciando os ambientes no Anaconda

Quando você abrir seu terminal no MacOS ou Linux, ou a Linha de Comando especialmente criada para o Anaconda, você poderá fazer tudo usando os comandos do Conda.

- Para saber mais sobre os comandos do Conda, você pode usar o Conda Cheat Sheet
 - https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf

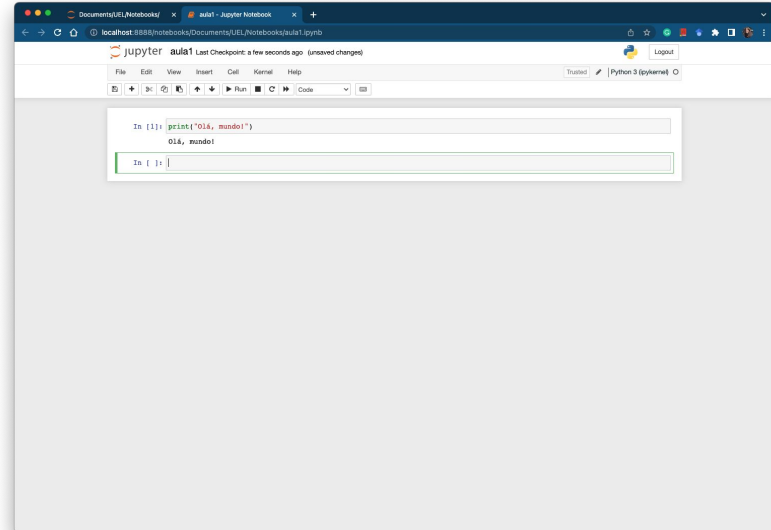
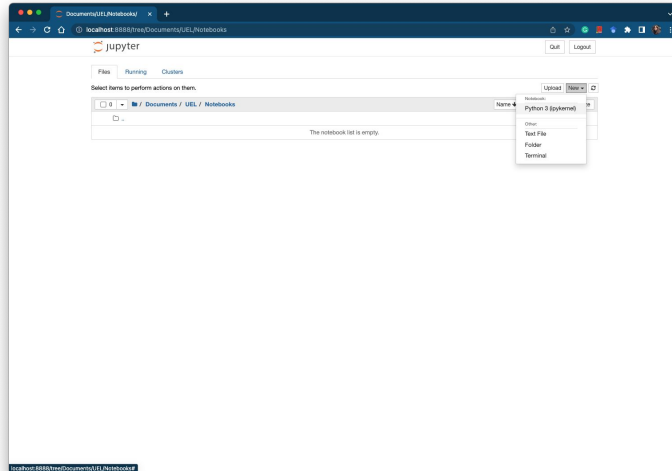
Ambientes de programação com Python

Após instalar o Python você irá querer instalar alguma IDE ou ambiente para auxiliá-lo a programar. Há diversas opções, e segue abaixo a lista de algumas mais populares:

- Spyder
- PyCharm
- Jupyter Notebook
- Jupyter Lab
- Visual Studio Code

Jupyter Notebook

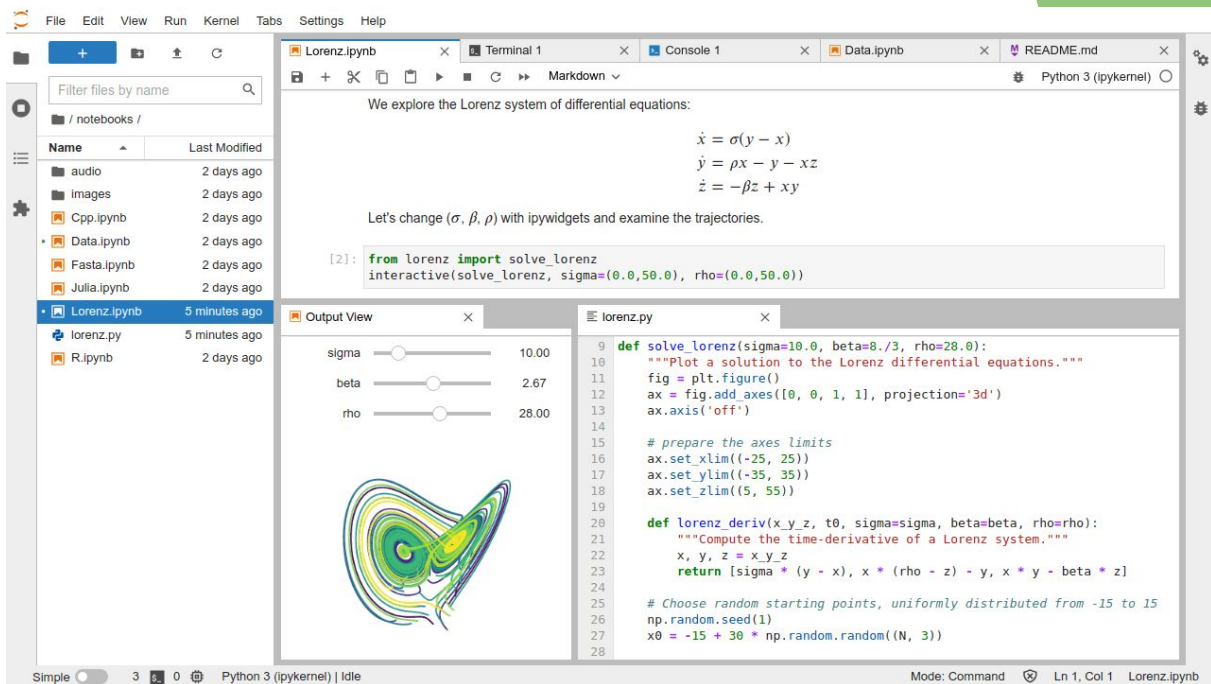
- O Jupyter Notebook é um ambiente simples que roda em um mini *webserver* local, e é aberto no navegador.
- Ele permite visualizar arquivos do seu computador, e criar "notebooks" python para a realização de experimentos.



Jupyter Lab

O Jupyter Lab é uma versão do Jupyter Notebook mais sofisticada.

- Traz um ambiente mais parecido com o Google Colab.
- Na verdade, o Google Colab é o Jupyter Notebook "vestido" e "perfumado" de Google.



The screenshot displays the Jupyter Lab environment. On the left is a file browser showing a list of notebooks and files. The main area is divided into several panes:

- Code Editor:** Contains the Lorenz system of differential equations:
$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$
Below the equations, it says: "Let's change (σ, β, ρ) with ipywidgets and examine the trajectories." A code cell [2]: `from lorenz import solve_lorenz`
`interactive(solve_lorenz, sigma=(0.0,50.0), rho=(0.0,50.0))`
- Output View:** Shows three sliders for parameters: `sigma` (set to 10.00), `beta` (set to 2.67), and `rho` (set to 28.00). Below the sliders is a 3D plot of the Lorenz attractor, showing its characteristic butterfly shape with trajectories in various colors.
- Code Editor (lorenz.py):** Contains the implementation of the Lorenz system solver and the interactive plot.

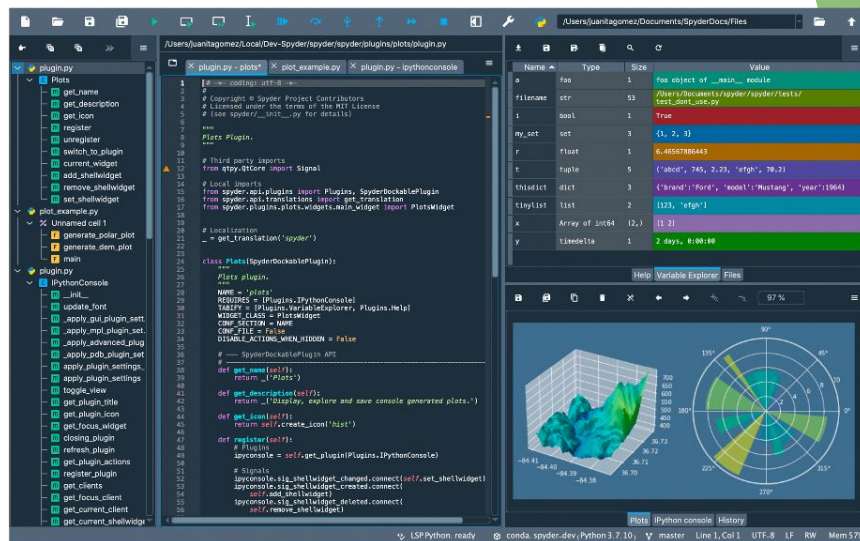
```
9 def solve_lorenz(sigma=10.0, beta=8./3, rho=28.0):
10     """Plot a solution to the Lorenz differential equations."""
11     fig = plt.figure()
12     ax = fig.add_axes([0, 0, 1, 1], projection='3d')
13     ax.axis('off')
14
15     # prepare the axes limits
16     ax.set_xlim((-25, 25))
17     ax.set_ylim((-35, 35))
18     ax.set_zlim((5, 55))
19
20     def lorenz_deriv(x,y,z, t0, sigma=sigma, beta=beta, rho=rho):
21         """Compute the time-derivative of a Lorenz system."""
22         x, y, z = x,y,z
23         return [sigma * (y - x), x * (rho - z) - y, y * y - beta * z]
24
25     # Choose random starting points, uniformly distributed from -15 to 15
26     np.random.seed(1)
27     x0 = -15 + 30 * np.random.random((N, 3))
28
```

The bottom status bar indicates the environment is "Python 3 (ipykernel) | Idle" and the mode is "Command".

Spyder



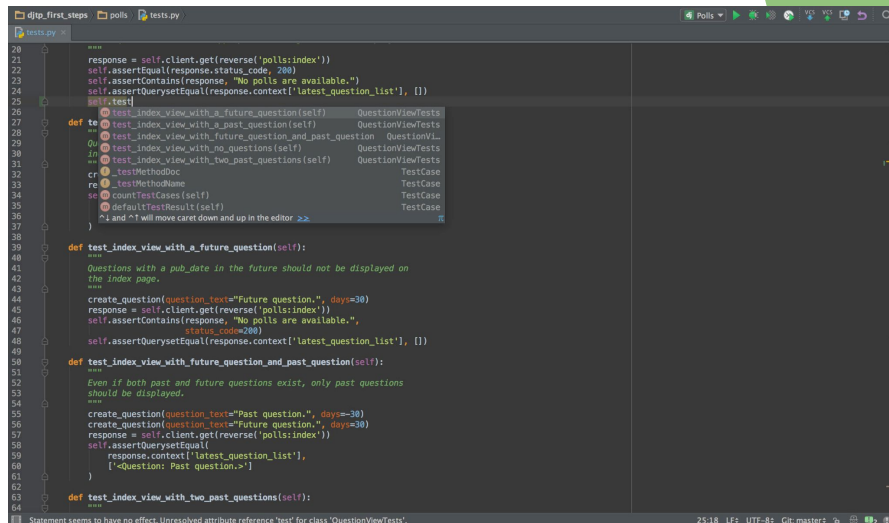
- O Spyder é um ambiente que se auto-intitula o ambiente de desenvolvimento Python.
- Traz visualizador de variáveis, gráficos, dentre outros.
- Possui ferramentas de depuração, e gerenciamento de projetos.



PyCharm

O Pycharm está disponível em duas versões:

- Community Edition - Gratuita, mas com limitações.
- Professional Edition - Paga, mas gratuita por 1 ano (renovável) se possuir email de instituição de ensino.
- Possui ferramentas de depuração, e gerenciamento de projetos.



```
20 response = self.client.get(reverse('polls:index'))
21
22 self.assertEqual(response.status_code, 200)
23 self.assertContains(response, "No polls are available.")
24 self.assertQuerysetEqual(response.context['latest_question_list'], [])
25
26
27 def test_index_view_with_a_future_question(self):
28     """
29     test_index_view_with_a_future_question_and_past_question
30     test_index_view_with_no_questions
31     test_index_view_with_two_past_questions
32 """
33     @unittest.skipUnless(settings.DEBUG, 'Test with debug enabled')
34     @unittest.skipUnless(settings.DEBUG, 'Test with debug enabled')
35     @unittest.skipUnless(settings.DEBUG, 'Test with debug enabled')
36     @unittest.skipUnless(settings.DEBUG, 'Test with debug enabled')
37     @unittest.skipUnless(settings.DEBUG, 'Test with debug enabled')
38
39     def test_index_view_with_a_future_question(self):
40         """
41         Questions with a pub_date in the future should not be displayed on
42         the index page.
43         """
44         create_question(question_text="Future question.", days=30)
45         response = self.client.get(reverse('polls:index'))
46         self.assertContains(response, "No polls are available.")
47         self.assertEqual(response.status_code, 200)
48         self.assertQuerysetEqual(response.context['latest_question_list'], [])
49
50     def test_index_view_with_future_question_and_past_question(self):
51         """
52         Even if both past and future questions exist, only past questions
53         should be displayed.
54         """
55         create_question(question_text="Past question.", days=-30)
56         create_question(question_text="Future question.", days=30)
57         response = self.client.get(reverse('polls:index'))
58         self.assertQuerysetEqual(
59             response.context['latest_question_list'],
60             ['<question: Past question.>']
61         )
62
63     def test_index_view_with_two_past_questions(self):
64         """
65         Questions with a pub_date in the past should be displayed on
66         the index page.
67         """
68         create_question(question_text="Past question.", days=-30)
69         create_question(question_text="Past question.", days=-30)
70         response = self.client.get(reverse('polls:index'))
71         self.assertQuerysetEqual(
72             response.context['latest_question_list'],
73             ['<question: Past question.>', '<question: Past question.>']
74         )
75
76     def test_index_view_with_no_questions(self):
77         """
78         When there are no questions present, the user should see a message
79         encouraging them to ask a question.
80         """
81         response = self.client.get(reverse('polls:index'))
82         self.assertContains(response, "No polls are available.")
83         self.assertQuerysetEqual(response.context['latest_question_list'], [])
```

Qual devo utilizar?

Depende...

- do tipo de projeto a ser realizado
- da necessidade de colaboração
- da necessidade de uso remoto/local
- da sua familiaridade com a plataforma
- ...

Qual devo utilizar?

Para a disciplina...

Você poderá utilizar qualquer ambiente!

Atente-se ao requisito de tipo de arquivo solicitado em cada atividade ou trabalho. Ex:

- *Notebook* .ipynb
- *Python Script* .py