



Aula 2

Tutorial de Instalação da Anaconda (Windows)

Priscila M. Kai



Windows

- 1) Link: <https://www.anaconda.com/>

Individual Edition is now

ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform



- 2) Vá até a pasta Downloads e dê duplo clique para iniciar a instalação.

Links para outros Sistemas Operacionais

Link: <https://www.anaconda.com/products/distribution>

Anaconda Installers

Windows

Python 3.9

64-Bit Graphical Installer (621 MB)

MacOS

Python 3.9

64-Bit Graphical Installer (688 MB)

64-Bit Command Line Installer (681 MB)

64-Bit (M1) Graphical Installer (484 MB)

64-Bit (M1) Command Line Installer (472 MB)

Linux

Python 3.9

64-Bit (x86) Installer (737 MB)

64-Bit (Power8 and Power9) Installer (360 MB)

64-Bit (AWS Graviton2 / ARM64) Installer (534 MB)

64-bit (Linux on IBM Z & LinuxONE) Installer (282 MB)

Windows

- 3) Clique em Avançar.
 - Clique em Concordo.

- 4) Clique em Avançar.
 - Selecione uma pasta de destino
 - De acordo com o site oficial, não é recomendado adicionar o Anaconda à variável de ambiente PATH, podendo interferir em outros softwares.
 - Clique em Install

- 5) Ao terminar a instalação, verifique o Anaconda
 - Opção 1: Anaconda Navigator
 - Iniciar > Anaconda Navigator
 - O Navigator será aberto se a instalação for bem-sucedida.
 - Opção 2: Conda
 - Iniciar > Anaconda Prompt
 - Iniciado o promp, digite: `conda list`
 - Se a instalação estiver funcionando, uma lista de pacotes será exibida.

Instalando o Spyder



The
Scientific
Python
Development
Environment

A screenshot of the Spyder Python IDE interface. The interface is divided into several panels. On the left is a file explorer showing a project structure with folders like 'Plots' and 'plot_example.py'. The main central panel displays a Python script named 'plugin.py' with code for a SpyderDockablePlugin. On the right, there is a 'Variable Explorer' panel showing a table of variables and their values. Below that is a 'Plots' panel displaying a 3D surface plot and a corresponding polar plot. The bottom status bar shows the current environment as 'conda: spyder.dev, Python 3.7.10' and other details like 'master', 'Line 1, Col 1', and 'UTF-8'.

Name	Type	Size	Value
a	foo	1	foo object of __main__ module
filename	str	53	/Users/juanitagomez/spyder/spyder/tests/test_dont_use.py
l	bool	1	True
my_set	set	3	{1, 2, 3}
r	float	1	6.48567886443
t	tuple	5	('abcd', 745, 2.23, 'efgh', 70.2)
thisdict	dict	3	{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
tinylist	list	2	[123, 'efgh']
x	Array of Int64	(2,)	[1 2]
y	timedelta	1	2 days, 0:00:00

```
1 # -*- coding: utf-8 -*-
2
3 # Copyright © Spyder Project Contributors
4 # Licensed under the terms of the MIT License
5 # (see spyder/_init_.py for details)
6
7 """
8 # Plots Plugin.
9 """
10
11 # Third party imports
12 from qtpy.QtCore import Signal
13
14 # Local imports
15 from spyder.api.plugins import Plugins, SpyderDockablePlugin
16 from spyder.api.translations import get_translation
17 from spyder.plugins.plots.widgets.main_widget import PlotsWidget
18
19 # Localization
20 _ = get_translation('spyder')
21
22
23 class Plots(SpyderDockablePlugin):
24     """
25     Plots plugin.
26     """
27     NAME = 'plots'
28     REQUIRES = (Plugins.IPythonConsole)
29     TABIFY = (Plugins.VariableExplorer, Plugins.Help)
30     WIDGET_CLASS = PlotsWidget
31     CONF_SECTION = NAME
32     CONF_FILE = False
33     DISABLE_ACTIONS_WHEN_HIDDEN = False
34
35     # --- SpyderDockablePlugin API
36     #
37     def get_name(self):
38         return ('Plots')
39
40     def get_description(self):
41         return _('Display, explore and save console generated plots.')
42
43     def get_icon(self):
44         return self.create_icon('hist')
45
46     def register(self):
47         # Plugins
48         ipyconsole = self.get_plugin(Plugins.IPythonConsole)
49
50         # Signals
51         ipyconsole.sig_shellwidget_changed.connect(self.set_shellwidget)
52         ipyconsole.sig_shellwidget_created.connect(
53             self.add_shellwidget)
54         ipyconsole.sig_shellwidget_deleted.connect(
55             self.remove_shellwidget)
```

Instalando o Spyder

Instalação com Conda

Para uma instalação completa e dependências opcionais, execute no Prompt do Anaconda:

```
conda create -c conda-forge -n spyder-env spyder numpy scipy  
pandas matplotlib sympy cython
```

Para instalação básica, execute:

```
conda create -c conda-forge -n spyder-env spyder
```

O Spyder será instalado em um novo ambiente chamado spyder-env, usando o canal Conda-Forge.

Instalando o Spyder

Para ativar o ambiente spyder-env:

```
conda activate spyder-env
```

E executar:

```
spyder
```

O guia de instalação (em inglês) pode ser acessado pelo link:

<https://docs.spyder-ide.org/current/installation.html>



C:\Users\untitled0.py

untitled0.py* x

```
1  # -*- coding: utf-8 -*-
2  """
3  TUTORIAL
4
5  Carregando imagens da Sentinel-2 com Python
6  """
7
8
```

Editor

C:\Users\



Nam	Type	Size	Value
-----	------	------	-------

Help Variable Explorer Plots Files

Console I/A x

Python 3.8.13 | packaged by conda-forge | (default, Mar 25 2022, 05:59:45) [MSC v.1929 64 bit (AMD64)]

Type "copyright", "credits" or "license" for more information.

IPython 7.33.0 -- An enhanced Interactive Python.

In [1]:

Console

IPython Console History

Instalando o Spyder

Caso algum pacotes não esteja instalado no ambiente criado, podemos buscá-lo no site do [Anaconda.org](https://anaconda.org), copiar o trecho correspondente à instalação e colar no ambiente via Prompt:

Exemplo: Instalação do pacote Geopandas

```
conda install -c conda-forge geopandas
```



Where packages, notebooks, projects and environments are shared.

SEARCH PACKAGES



⬅ Favorites ▾ Downloads ⬅ Package (owner / package)

38

2198135

 **conda-forge / geopandas** 0.121

Geographic pandas extensions



conda install ?

To install this package run one of the following:

```
conda install -c conda-forge geopandas
```

Carregando imagens do Sentinel-2

Importando bibliotecas:

```
import geopandas as gpd  
import matplotlib.pyplot as plt  
import rasterio  
from os import listdir  
from os.path import isfile, join  
from rasterio.mask import mask
```