

GEONETCast-Americas Training for the Eastern Caribbean States

Day 5 - May 12th

Session 2:

Hands-on

Introduction to Visual Studio Code



Diego Souza
diego.souza@inpe.br

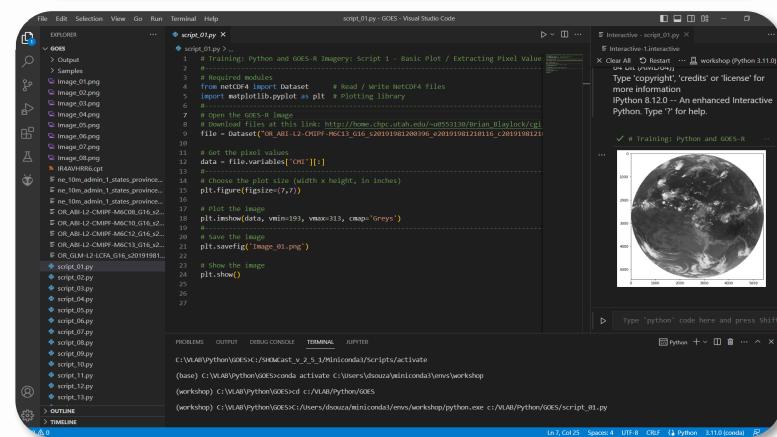
DISSM - Meteorological Satellites and Sensors' Division
CGCT - General Coordination of Earth Sciences
INPE - National Institute for Space Research



Caribbean Institute for Meteorology and Hydrology



- Running Scripts Locally (Review)
- Installing Visual Studio Code
- Configuring and Running Scripts
- Extra Features



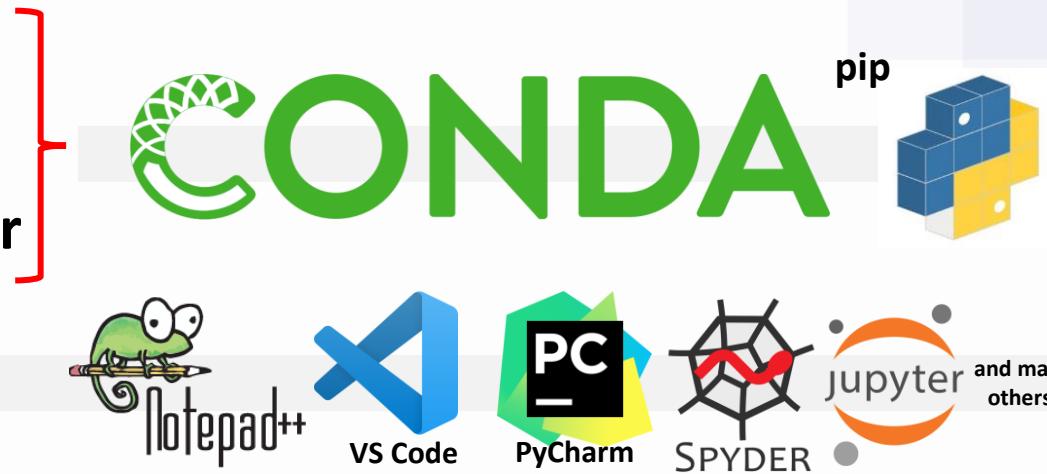
A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with several Python scripts (script_01.py, script_02.py, etc.) and image files (Image_01.png, Image_02.png, etc.). The main editor area displays a Python script named "script_01.py" with the following code:

```
1 # Training: Python and GOES-R Imagery: Script 1 - Basic Plot / Extracting Pixel Value
2
3 # Import required modules
4 from netCDF4 import Dataset
5 import matplotlib.pyplot as plt # Plotting library
6
7 # Open the GOES-R Image
8 nc = Dataset("http://noaa-ces.sci.gsfc.nasa.gov/noaa15/10min/2011/01/10/101020.nc")
9 file = dataset("NOAA-15-CMIP5-PCMCIA-G16_x2_201101101010200000_g00110101201101_20110101201101.nc")
10
11 # Get the pixel values
12 data = file.variables["C01"][:]
13
14 # Choose the plot size (width x height, in inches)
15 plt.figure(figsize=(7,7))
16
17 # Plot the image
18 plt.imshow(data, vmin=193, vmax=311, cmap='Greys')
19
20 # Save the image
21 plt.savefig("Image_01.png")
22
23 # Show the image
24 plt.show()
```

The right side of the interface shows the "Interactive" tab with a Python shell and a preview of the generated satellite imagery plot.

Running Scripts Locally

- Python
- Package Manager
- Virtual Environment Manager



- Text Editor / IDE

(Integrated Development Environment)

- Sample Data



A variety of mechanisms available

Running Scripts Locally

password: workshop

<https://geonetcast.wordpress.com/2023/03/19/getting-started-with-python-and-satellite-imagery/>

Objectives of the Activity

GNC-A
GEONETCast-Americas
Delivering Environmental Data to Users in The Americas

BLOG

[Home](#) [Blog Content](#) [The GEONETCast System](#) [GEONETCast-Americas](#) [Documentation](#)

[Data Processing](#) [User Information](#) [GNC-A Forum](#) [Contact](#) [About](#)

Posted on 2023-03-01 by Diego Souza

[← Previous](#) [Next →](#)

Getting Started With Python and Satellite Imagery



April 24 - May 05 | 2023

Contact: If you have any questions, please contact:

- E-mail: diego.souza@inpe.br

Learning Objectives: By the end of this activity, participants will:

- Become familiar with some basic tools to start manipulating satellite images with Python
- Understand how to make basic operations such as:

- Reading a GOES-R (GOES-16, 17 or 18) NetCDF files containing a GOES-R plot and extract pixel values (brightness temperatures / reflectance)
- Change color scales, add a title and a colorbar to the plot
- Add coordinates, countries, states / provinces (and other shapes)
- Overlay GRIB data with ABI
- Create an RGB Composite

Estimated duration of the pre-course activity: 2-6 h

PREREQUISITES

For this exercise, we'll need the following:

- Python 3.10 (our programming language)
- A "Package Manager" (to install libraries)
- An IDE (Integrated Development Environment) to separate our projects
- A Text Editor (to write our code)
- GOES-R imagery samples (the data to be manipulated)

For the first three items ("Python 3.10", "Package Manager" and "Environment Manager"), the "Miniconda" tool will be sufficient. As for the text editor, there are many options available (Visual Studio Code, Spyder, PyCharm, Atom, Jupyter, etc), but for simplicity (and avoiding problems with editors), we will use "Notepad++".

For the GOES-R imagery samples, we'll download them directly from the cloud (Amazon Web Services). You may also get samples from your GNC-A station or other receive mechanisms (like GRB, PDA and LDM).

EXECUTING SCRIPTS LOCALLY X EXECUTING SCRIPTS IN THE CLOUD

It is possible to run the Python scripts that we will see both locally (on your own

Installation of Tools

INSTALLATION STEPS

1. Download and install Miniconda for Python 3.10 at the following link (approximately 60 MB):

<https://docs.conda.io/en/latest/miniconda.html>



— Downloading Miniconda

Notes – Windows installation:

- During the installation, it is not necessary to check "Add Anaconda to my Path environment variable".
- You may check "Register Anaconda as my default Python 3.10"

Notes – Linux installation:

- In the Linux installation, it is possible to choose the installation directory with the "-p" parameter. Choose the appropriate directory on your machine!

```
curl https://repo.anaconda.com/miniconda/miniconda3-latest-Linux-x86_64.sh -o /my_directory/miniconda3-latest-Linux-x86_64.sh
```

Notes – Windows and Linux installation:

- The installation will take approximately 5 minutes

CREATING A PYTHON VIRTUAL ENVIRONMENT AND INSTALLING LIBRARIES.

2. Let's create a Python environment called "workshop". We will call it "workshop", but you may give any name you want to your environment, as long as you use this name during the activation as we will see. We will install the following libraries and their dependencies in this environment.

- matplotlib: library for creating plots
- netcdf4: read and write NetCDF files
- xarray: library for other analysis of geospatial data
- boto3: integration with Amazon cloud services
- gdal: reprojection/manipulation of geospatial data
- scikit: array manipulation and analysis
- pandas: data manipulation and analysis
- pygrib: read and write GRIB files
- metpy: reading, visualization and calculations with meteorological data
- imggeo: read and write a wide variety of image data

In order to create the environment, open the recently installed "Anaconda Prompt" as an Administrator and insert the following command:

```
conda create --name workshop -c conda-forge matplotlib netcdf4 cartopy
```



Access to Samples (GNC or Cloud)

Note: For the screenshots on this procedure, we've downloaded files for JULY 17, 2019, 12:00 UTC.

GOES-16/17 on Amazon Download Page

[View All Assets](#) [Download](#) [Upload](#)

Source: AWS
Satellite: GOES-16/17
Domain: All Data
Product: ABI-12_Cloud_and_Humidity_Imagery
Date: 17/07/2019
Hour (UTC): 12

Sample Assets

Image #1	17/07/2019 12:00:00 UTC
Image #2	17/07/2019 12:00:00 UTC
Image #3	17/07/2019 12:00:00 UTC
Image #4	17/07/2019 12:00:00 UTC
Image #5	17/07/2019 12:00:00 UTC
Image #6	17/07/2019 12:00:00 UTC
Image #7	17/07/2019 12:00:00 UTC
Image #8	17/07/2019 12:00:00 UTC
Image #9	17/07/2019 12:00:00 UTC
Image #10	17/07/2019 12:00:00 UTC
Image #11	17/07/2019 12:00:00 UTC
Image #12	17/07/2019 12:00:00 UTC
Image #13	17/07/2019 12:00:00 UTC
Image #14	17/07/2019 12:00:00 UTC
Image #15	17/07/2019 12:00:00 UTC
Image #16	17/07/2019 12:00:00 UTC

— Downloading files from Amazon Web Services using a graphical interface

The NetCDF will be downloaded.

Put the NetCDF in the work folder you just created (e.g.: C:\VLAB\Python\GOES)

You may also retrieve some samples from your GEONETCast-Americas stations and use them during this procedure! In the stations of the project, you will find NetCDFs at the following folder:

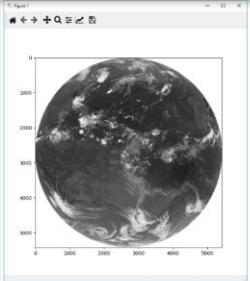
```
icasdata/stations/faust/GOES-R-CMI-Imagery
```

```
icasdata@faust-OptiPlex-709:~/icasdata/stations/faust$ ls
icasdata@faust-OptiPlex-709:~/icasdata/stations/faust$ ls -l
total 124
drwxr-xr-x  8 Faust  faust    2864 Mar  1  2019 79_Mer_12:45_CIMSS
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 18782358_Mer_12:45_EUMETSAT
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 32048254_Mer_12:45_GEOS-R-DOCS
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 31800000_Mer_12:45_GEOS-R-Products
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 3208_Mer_12:45_GEOS-R_Level-2_Products
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 4096_Mer_12:45_Convective
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 73724_Mer_12:45_IMER
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 30000000_Mer_12:45_IMER-DOCS
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 3979132_Mer_12:45_IMER-2_CLIMATE
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 27742408_Mer_12:45_IMER-DOCS
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 824328_Mer_12:45_IMER-GRIB2
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 449312_Mer_12:45_IMER-MARAD
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 22349414_Mer_12:45_IMER-SURFACE
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 6393152_Mer_12:45_IMER-WIND
drwxr-xr-x  2 Faust  faust     64 Mar  1  2019 4887_Mer_12:45_HIML-1_Salvador
drwxr-xr-x  4 Faust  faust    1280 Mar  1  2019 icasdata
```

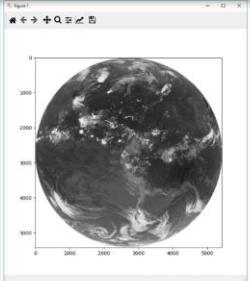
— One foot plot

Checking the Brightness Temperatures

Move the mouse pointer over the plot, and you will see the Band 13 pixel value in Brightness Temperatures (K) in the lower left part of the screen. In the example image below, this particular cloud top temperature is 227 K.

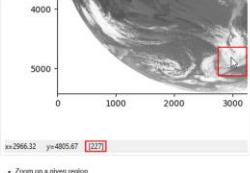


Local Data Processing



— Checking the Brightness Temperatures

Move the mouse pointer over the plot, and you will see the Band 13 pixel value in Brightness Temperatures (K) in the lower left part of the screen. In the example image below, this particular cloud top temperature is 227 K.



x=2966.32 y=4805.67 [227]

— Zoom on a given region

In order to zoom on a given region, just click on the magnifier icon in the upper part of the screen and select the region you want to zoom in.



To go back to the full view, click at the "Home" icon. Apart from the Visualization screen, a PNG image called "image_01.png" has been saved to your working directory.



Running Scripts Locally: Summary

<https://geonetcast.wordpress.com/2023/03/19/getting-started-with-python-and-satellite-imagery/>

Installing Miniconda

Miniconda																					
Miniconda is a free minimal installer for conda. It is a small, bootstrap version of Anaconda that includes only conda, Python, the packages they depend on, and a small number of other useful packages, including pip, zlib and a few others. Use the <code>conda install</code> command to install 720+ additional conda packages from the Anaconda repository.																					
See if Miniconda is right for you.																					
Windows installers																					
<table border="1"> <thead> <tr><th>Python version</th><th>Name</th><th>Size</th><th>SHA256 hash</th></tr> </thead> <tbody> <tr><td>Python 3.8</td><td>Miniconda3 Windows 64-bit</td><td>55.7 MB</td><td>1f47f17e011215b00f0ff14fc109ca1091002d400104211c16c17423548</td></tr> <tr><td></td><td>Miniconda3 Windows 32-bit</td><td>49.6 MB</td><td>435022134099421a79ef7e127099100000c107a0748cc1c16c17423548</td></tr> <tr><td>Python 2.7</td><td>Miniconda2 Windows 64-bit</td><td>54.1 MB</td><td>69730244032244947497f0d02014000000000000000000000000000000000000</td></tr> <tr><td></td><td>Miniconda2 Windows 32-bit</td><td>47.7 MB</td><td>38900242700310507c109947221f7a3f7a602210000000000000000000000000</td></tr> </tbody> </table>		Python version	Name	Size	SHA256 hash	Python 3.8	Miniconda3 Windows 64-bit	55.7 MB	1f47f17e011215b00f0ff14fc109ca1091002d400104211c16c17423548		Miniconda3 Windows 32-bit	49.6 MB	435022134099421a79ef7e127099100000c107a0748cc1c16c17423548	Python 2.7	Miniconda2 Windows 64-bit	54.1 MB	69730244032244947497f0d02014000000000000000000000000000000000000		Miniconda2 Windows 32-bit	47.7 MB	38900242700310507c109947221f7a3f7a602210000000000000000000000000
Python version	Name	Size	SHA256 hash																		
Python 3.8	Miniconda3 Windows 64-bit	55.7 MB	1f47f17e011215b00f0ff14fc109ca1091002d400104211c16c17423548																		
	Miniconda3 Windows 32-bit	49.6 MB	435022134099421a79ef7e127099100000c107a0748cc1c16c17423548																		
Python 2.7	Miniconda2 Windows 64-bit	54.1 MB	69730244032244947497f0d02014000000000000000000000000000000000000																		
	Miniconda2 Windows 32-bit	47.7 MB	38900242700310507c109947221f7a3f7a602210000000000000000000000000																		
MacOSX installers																					
<table border="1"> <thead> <tr><th>Python version</th><th>Name</th><th>Size</th><th>SHA256 hash</th></tr> </thead> <tbody> <tr><td>Python 3.8</td><td>Miniconda3 Mac OSX 64-bit bash</td><td>53.2 MB</td><td>963801f400804842e1a81712349f842f7680020502ff25613aa03c197c1</td></tr> <tr><td></td><td>Miniconda3 Mac OSX 64-bit pkg</td><td>61.3 MB</td><td>348607c735346971203d27709000000000000000000000000000000000000000</td></tr> <tr><td>Python 2.7</td><td>Miniconda2 Mac OSX 64-bit bash</td><td>40.3 MB</td><td>9426514290222913349764932000000000000000000000000000000000000000</td></tr> </tbody> </table>		Python version	Name	Size	SHA256 hash	Python 3.8	Miniconda3 Mac OSX 64-bit bash	53.2 MB	963801f400804842e1a81712349f842f7680020502ff25613aa03c197c1		Miniconda3 Mac OSX 64-bit pkg	61.3 MB	348607c735346971203d27709000000000000000000000000000000000000000	Python 2.7	Miniconda2 Mac OSX 64-bit bash	40.3 MB	9426514290222913349764932000000000000000000000000000000000000000				
Python version	Name	Size	SHA256 hash																		
Python 3.8	Miniconda3 Mac OSX 64-bit bash	53.2 MB	963801f400804842e1a81712349f842f7680020502ff25613aa03c197c1																		
	Miniconda3 Mac OSX 64-bit pkg	61.3 MB	348607c735346971203d27709000000000000000000000000000000000000000																		
Python 2.7	Miniconda2 Mac OSX 64-bit bash	40.3 MB	9426514290222913349764932000000000000000000000000000000000000000																		

Read the Docs

v latest

Creating a Virtual Environment called "workshop"

```
Anaconda Prompt (Miniconda3) - conda create --name workshop -c conda-forge matplotlib netcdf4 cartopy boto3 gdal scipy pandas
(base) D:\Users\dsouza conda create --name workshop -c conda-forge matplotlib netcdf4 cartopy boto3 gdal scipy pandas
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: D:\Users\dsouza\miniconda3\envs\workshop

added / updated specs:
- boto3
- cartopy
- gdal
- matplotlib
- netcdf4
- pandas
- scipy

The following packages will be downloaded:

```

package	build	size	source
boost-cpp-1.74.0	h54f0996_2	16.1 MB	conda-forge
boto3-1.17.16	pyhd8ed1ab_0	70 KB	conda-forge
botocore-1.20.16	pyhd8ed1ab_0	4.5 MB	conda-forge
cartopy-0.7.0	py39hb82d6ee_1001	369 KB	conda-forge
bz2file-0.1.0	hf7e7307_4	149 KB	conda-forge
certifi-2020.12.5	hs545a59_0	1.7 KB	conda-forge
cairo-1.16.0	hba1bbd2_1007	2.2 MB	conda-forge

Installing a Text Editor

Downloads	
Notepad++ 7.9: Stand with Hong Kong	
Notepad++ 7.8.9: Stand with Hong Kong	
Notepad++ 7.8.8 release	
Notepad++ 7.8.7 release	
Notepad++ 7.8.6 release	

Downloads

Notepad++ 7.9: Stand with Hong Kong

Notepad++ 7.8.9: Stand with Hong Kong

Notepad++ 7.8.8 release

Notepad++ 7.8.7 release

Notepad++ 7.8.6 release

Accessing Sample Imagery

GOES-16/17 on Amazon Download Page

Less@0153-dpsg fazt7 ls -l
total 72566

Source: AWS OCC

Satellite: GOES-16/East GOES-17/West

Domain: Full Disk

Product: ABI L2 Cloud and Moisture Imagery

Date: 17/07/2019

Hour (UTC): 12

Submit

Band 01	00	10	20	30	40	50
Band 02	00	10	20	30	40	50
Band 03	00	10	20	30	40	50
Band 04	00	10	20	30	40	50

Required modules
`from netCDF4 import Dataset` # Read / Write NetCDF files
`import matplotlib.pyplot as plt` # Plotting library

Open the GOES-R image
Download files at this link: http://home.chpc.utah.edu/~mabu12-CMIPF-M6C13_G16_s201919812

Get the pixel values
`data = file.variables['CM1'][:]`

Choose the plot size (width x height, in inches)
`plt.figure(figsize=(7,7))`

Plot the image
`plt.imshow(data, vmin=193, vmax=313, cmap='Greys')`

Save the image
`plt.savefig('Image_01.png')`

Show the image
`plt.show()`

Running Scripts

Until Now, We Have Used Notepad++ (Locally)

C:\VLAB\Python\GOES\script_01.py - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins Janela ?

script_01.py

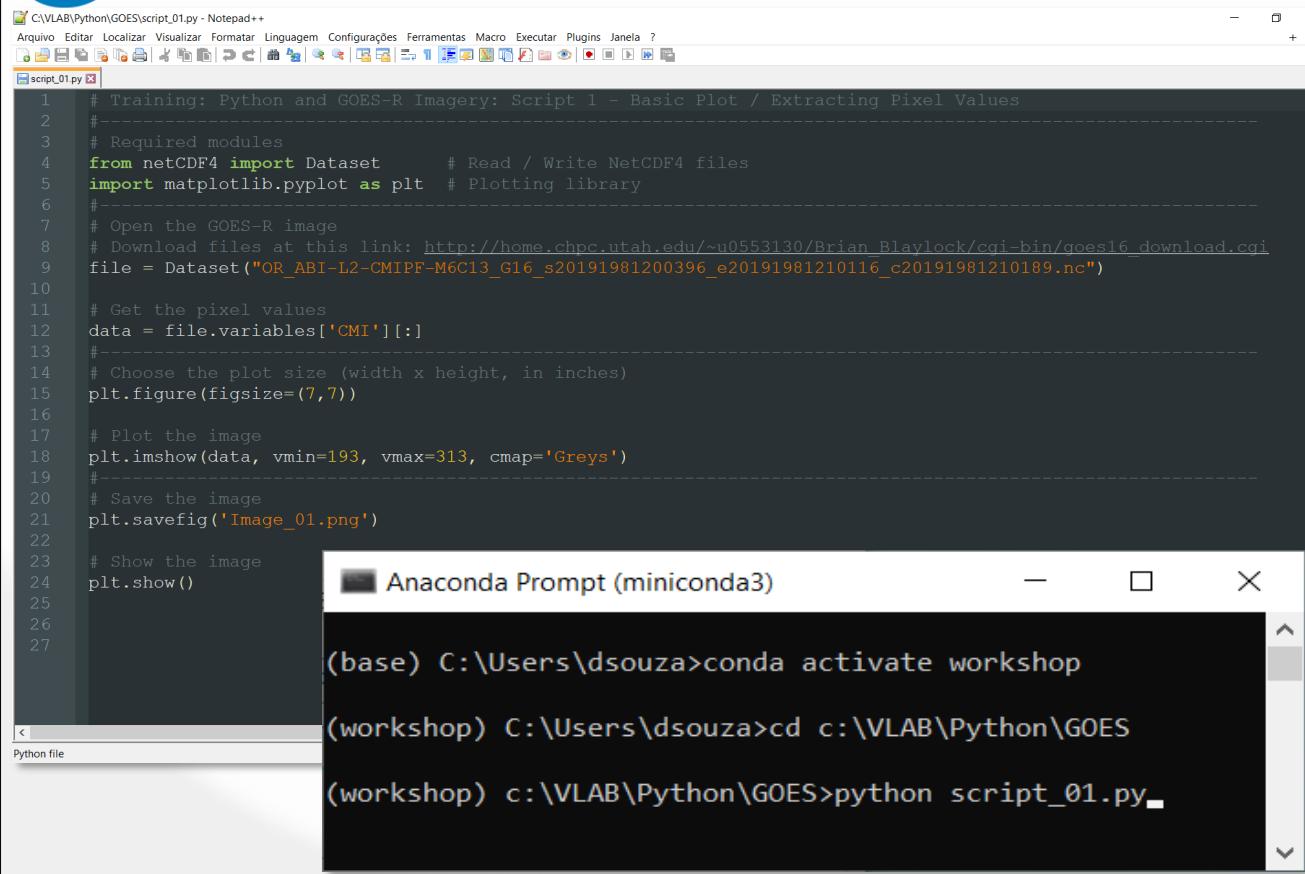
```
1 # Training: Python and GOES-R Imagery: Script 1 - Basic Plot / Extracting Pixel Values
2 #-----
3 # Required modules
4 from netCDF4 import Dataset      # Read / Write NetCDF4 files
5 import matplotlib.pyplot as plt  # Plotting library
6 #-----
7 # Open the GOES-R image
8 # Download files at this link: http://home.chpc.utah.edu/~u0553130/Brian\_Blaylock/cgi-bin/goes16\_download.cgi
9 file = Dataset("OR_ABI-L2-CMIPF-M6C13_G16_s20191981200396_e20191981210116_c20191981210189.nc")
10
11 # Get the pixel values
12 data = file.variables['CMI'][:]
13 #-----
14 # Choose the plot size (width x height, in inches)
15 plt.figure(figsize=(7,7))
16
17 # Plot the image
18 plt.imshow(data, vmin=193, vmax=313, cmap='Greys')
19 #
20 # Save the image
21 plt.savefig('Image_01.png')
22
23 # Show the image
24 plt.show()
25
26
27
```

length : 1,195 lines : 27 Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 INS

For simplicity and to avoid configuration issues not related to scripts, so far we have worked with Notepad++, a simple but nice option.

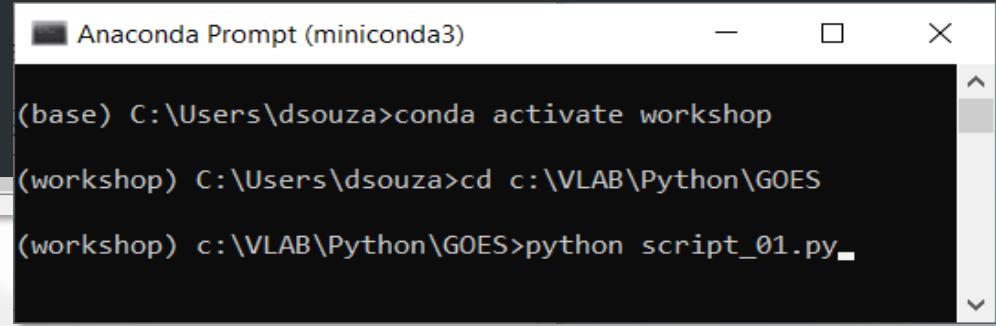


Until Now, We Have Used Notepad++ (Locally)



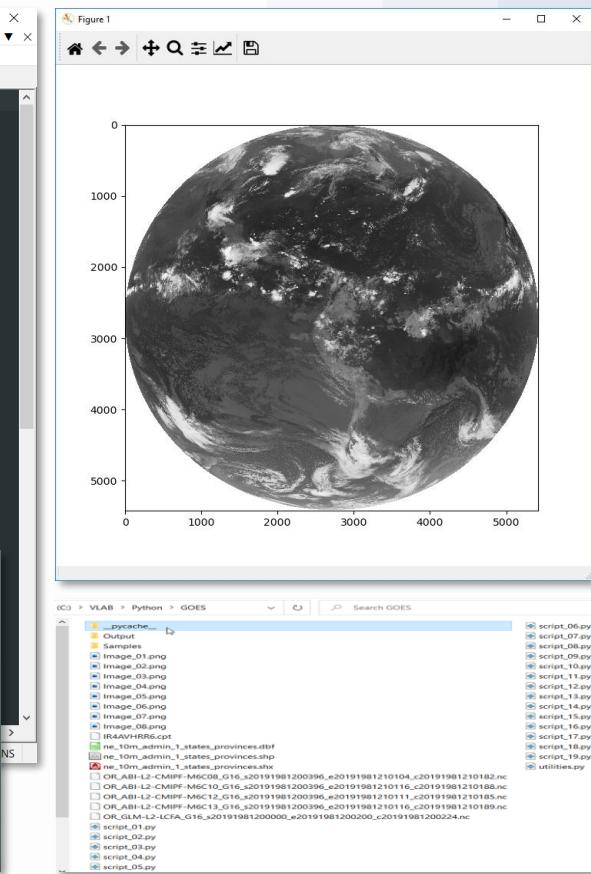
The screenshot shows a Notepad++ window with a Python script titled "script_01.py". The script uses the netCDF4 library to read a GOES-R image file and matplotlib.pyplot to plot it. The resulting image is a grayscale satellite view of Earth, centered on the Atlantic Ocean.

```
C:\VLAB\Python\GOES>script_01.py - Notepad++
Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins Janela ?
script_01.py
1 # Training: Python and GOES-R Imagery: Script 1 - Basic Plot / Extracting Pixel Values
2 #
3 # Required modules
4 from netCDF4 import Dataset      # Read / Write NetCDF4 files
5 import matplotlib.pyplot as plt  # Plotting library
6 #
7 # Open the GOES-R image
8 # Download files at this link: http://home.chpc.utah.edu/~u0553130/Brian_Blaylock/cgi-bin/goes16_download.cgi
9 file = Dataset("OR_ABI-L2-CMIPF-M6C13_G16_s20191981200396_e20191981210116_c20191981210189.nc")
10
11 # Get the pixel values
12 data = file.variables['CMI'][:]
13 #
14 # Choose the plot size (width x height, in inches)
15 plt.figure(figsize=(7,7))
16
17 # Plot the image
18 plt.imshow(data, vmin=193, vmax=313, cmap='Greys')
19 #
20 # Save the image
21 plt.savefig('Image_01.png')
22
23 # Show the image
24 plt.show()
25
26
27
```



The screenshot shows an Anaconda Prompt window running under miniconda3. It displays the command to activate the "workshop" environment, change the directory to "c:\VLAB\Python\GOES", and run the script "script_01.py".

```
Anaconda Prompt (miniconda3)
(base) C:\Users\dsouza>conda activate workshop
(workshop) C:\Users\dsouza>cd c:\VLAB\Python\GOES
(workshop) c:\VLAB\Python\GOES>python script_01.py
```



The screenshot shows a Windows File Explorer window displaying the contents of the "GOES" folder. It lists several files: "script_01.py", "Output", "Samples", "Image_01.png", "Image_02.png", "Image_03.png", "Image_04.png", "Image_05.png", "Image_06.png", "Image_07.png", "Image_08.png", "IR4WHR6.grd", "ne_10m_adm1_states_provinces.gdb", "ne_10m_adm1_states_provinces.shp", "ne_10m_adm1_states_provinces.shx", "OR_ABI-L2-CMIPF-M6C08_G16_s20191981200396_e20191981210104_c20191981210182.nc", "OR_ABI-L2-CMIPF-M6C12_G16_s20191981200396_e20191981210111_c20191981210185.nc", "OR_ABI-L2-CMIPF-M6C13_G16_s20191981200396_e20191981210116_c20191981210189.nc", "OR_GeoTIFF-LCFA_G16_s20191981200000_e20191981200200_c20191981200224.nc", "script_01.py", "script_02.py", "script_03.py", "script_04.py", and "script_05.py".

Let's See How to Start Using an IDE: Visual Studio Code

File Edit Selection View Go Run Terminal Help

script_01.py - GOES - Visual Studio Code

EXPLORER

GOES

- > Output
- > Samples
- Image_01.png
- Image_02.png
- Image_03.png
- Image_04.png
- Image_05.png
- Image_06.png
- Image_07.png
- Image_08.png
- IR4AVHRR6 cpt
- ne_10m_admin_1_states_province...
- ne_10m_admin_1_states_province...
- ne_10m_admin_1_states_province...
- OR_ABI-L2-CMIPF-M6C08_G16_s2...
- OR_ABI-L2-CMIPF-M6C10_G16_s2...
- OR_ABI-L2-CMIPF-M6C12_G16_s2...
- OR_ABI-L2-CMIPF-M6C13_G16_s2...
- OR_GLM-L2-LCFA_G16_s20191981...
- script_01.py
- script_02.py
- script_03.py
- script_04.py
- script_05.py
- script_06.py
- script_07.py
- script_08.py
- script_09.py
- script_10.py
- script_11.py
- script_12.py
- script_13.py

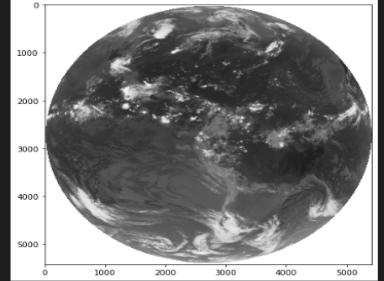
script_01.py > ...

```
1 # Training: Python and GOES-R
2 #
3 # Required modules
4 from netCDF4 import Dataset      # Read / Write NetCDF4 files
5 import matplotlib.pyplot as plt   # Plotting library
6 #
7 # Open the GOES-R image
8 # Download files at this link: http://home.chpc.utah.edu/~u0553130/Brian_Blaylock/cgi-bin/getfile?file=OR_ABI-L2-CMIPF-M6C13_G16_s20191981200396_e20191981210116_c20191981210116
9 file = Dataset("OR_ABI-L2-CMIPF-M6C13_G16_s20191981200396_e20191981210116_c20191981210116")
10 #
11 # Get the pixel values
12 data = file.variables['CMI'][:]
13 #
14 # Choose the plot size (width x height, in inches)
15 plt.figure(figsize=(7,7))
16 #
17 # Plot the image
18 plt.imshow(data, vmin=193, vmax=313, cmap='Greys')
19 #
20 # Save the image
21 plt.savefig('Image_01.png')
22 #
23 # Show the image
24 plt.show()
```

IDE: Integrated Development Environment
Everything in a single interface!

Type 'copyright', 'credits' or 'license' for more information
IPython 8.12.0 -- An enhanced Interactive Python. Type '?' for help.

Training: Python and GOES-R



Type 'python' code here and press Shift + Enter to execute it.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
C:\VLAB\Python\GOES>c:/SHOWCast_v_2_5_1/Miniconda3/Scripts/activate
(base) C:\VLAB\Python\GOES>conda activate C:\Users\dsouza\miniconda3\envs\workshop
(workshop) C:\VLAB\Python\GOES>cd c:/VLAB\Python\GOES
(workshop) C:\VLAB\Python\GOES>C:/Users/dsouza/miniconda3/envs/workshop/python.exe c:/VLAB\Python\GOES/script_01.py
```

Ln 7, Col 25 Spaces: 4 UTF-8 CRLF Python 3.11.0 (conda)

IDE: Everything in a Single Interface

The screenshot shows the Visual Studio Code interface with several panes highlighted by yellow boxes:

- EXPLORER**: Shows a file tree with various Python scripts and image files.
- SCRIPT**: Displays the content of `script_01.py`, which reads a GOES-R image and saves it as `Image_01.png`.

```
script_01.py > ...
1 # Training: Python and GOES-R Imagery: Script 1 - Basic Plot / Extracting Pixel Value
2 #
3 # Required modules
4 from netCDF4 import Dataset      # Read / Write NetCDF4 files
5 import matplotlib.pyplot as plt   # Plotting library
6 #
7 # open the GOES-R image
8 # Download files at this link: http://home.chpc.utah.edu/~u0553130/Brian_Blaylock/cgi
9 file = Dataset("OR_ABI-L2-CMIPF-M6C13_G16_s20191981200396_e20191981210116_c20191981210116")
10
11 # Get the pixel values
12 data = file.variables['CMI'][:]
13 #
14 # Choose the plot size (width x height, in inches)
15 plt.figure(figsize=(7,7))
16
17 # Plot the image
18 plt.imshow(data, vmin=193, vmax=313, cmap='Greys')
19
20 # Save the image
21 plt.savefig('Image_01.png')
22
23 # Show the image
24 plt.show()
```

- OUTPUT**: Displays a circular grayscale image of Earth's visible disk.
- TERMINAL**: Shows the command line history for running the script.

```
C:\VLAB\Python\GOES>c:/SHOWCast_v_2_5_1/Miniconda3/Scripts/activate
(base) C:\VLAB\Python\GOES>conda activate C:\Users\dsouza\miniconda3\envs\workshop
(workshop) C:\VLAB\Python\GOES>cd c:/VLAB/Python/GOES
(workshop) C:\VLAB\Python\GOES>C:/Users/dsouza/miniconda3/envs/workshop/python.exe c:/VLAB/Python/GOES/script_01.py
```



1. Download and Install Visual Studio Code (VSCode)

<https://code.visualstudio.com/>

Visual Studio Code - Code Editing +

code.visualstudio.com

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn

Join us for [VS Code Day](#) on April 26th!

Code editing.
Redefined.

Free. Built on open source. Runs everywhere.

[Download for Windows](#)
Stable Build

Web, Insiders edition, or other platforms

By using VS Code, you agree to its
[license and privacy statement](#).

EXTENSIONS: MARKETPLACE

- @sortinstalls
- Python 2019.6.24221 549M 45 Microsoft [Install](#)
- GitLens — Git super... 9.8.5 23.1M 5 Eric Amadio [Install](#)
- C/C++ 0.24.0 23M 3.5 Microsoft [Install](#)
- ESLint 1.9.0 21.9M 4.5 Integrates ESLint JavaScript into VS... Dirk Baumer [Install](#)
- Debugger for Ch... 4.11.6 20.6M 4 Microsoft [Install](#)
- Language Supp... 0.47.0 18.7M 4.5 Java Linting, Intellisense, formatting, ... Red Hat [Install](#)
- vscode-icons 8.8.0 17.2M 5 Icons for Visual Studio Code VS Code Icons Team [Install](#)
- Vetur 0.21.1 17M 4.5 Vue tooling for VS Code Pine Wu [Install](#)
- C# 12.1.0 15.6M 4 C# for Visual Studio Code (powered ... Microsoft [Install](#)

File Edit Selection View Go Debug Terminal Help serviceWorker.js - create-react-app - Visual Studio Code - In...

JS App.js JS index.js JS serviceWorker.js

```
src > JS serviceWorker.js > register > window.addEventListener('load') callback
checkValidServiceWorker(swUrl, config);
// Add some additional logging to localhost, p...
navigator.serviceWorker.ready.then(() => {
  product
  products
  removeSiteSpecificTrackingException
  removeWebWideTrackingException
  requestMediaKeySystemAccess
  sendBeacon
  serviceWorker (property) Navigator.serviceWorker
  storage
  storeSiteSpecificTrackingException
  storeWebWideTrackingException
} userAgent
vendor
```

TERMINAL ... 1: node + ☰ ^ x

You can now view `create-react-app` in the browser.

Local: http://localhost:3000/
On Your Network: http://10.211.55.3:3000/

Note that the development build is not optimized.

Ln 43, Col 19 Spaces: 2 UTF-8 LF JavaScript ☰ 🔍

[Download](#)

1. Download and Install Visual Studio Code (VSCode)

Download Visual Studio Code - N × +

<https://code.visualstudio.com/Download>

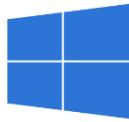
code.visualstudio.com/Download

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn Search Docs Download

Join us for [VS Code Day](#) on April 26th! 🎉

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

[Windows](#) [.deb](#) [.rpm](#) [Mac](#)

Windows 8, 10, 11 | Debian, Ubuntu | Red Hat, Fedora, SUSE | macOS 10.11+

User Installer | .deb | .zip | CLI
System Installer | .deb | .rpm | .tar.gz | Snap | CLI
.zip | .deb | .rpm | .tar.gz | Snap Store | .zip | Intel chip | Apple silicon | Universal
CLI | x64 | x86 | Arm64 | x64 | x86 | Arm64



When Opening for the First Time, This is What You See

File Edit Selection View Go Run Terminal Help

Welcome - Visual Studio Code

Welcome x

Visual Studio Code
Editing evolved

Start

- New File...
- Open File...
- Open Folder...

Recent

- GOES C:\VLAB\Python
- Python (Workspace) C:\VLAB\Python
- TRMM C:\VLAB

Walkthroughs

- Get Started with VS Code
Discover the best customizations to make VS Code yours.
- Learn the Fundamentals
- Boost your Productivity
- Get Started with Python Development Updated
- Get Started with Jupyter Notebooks Updated

Show welcome page on startup

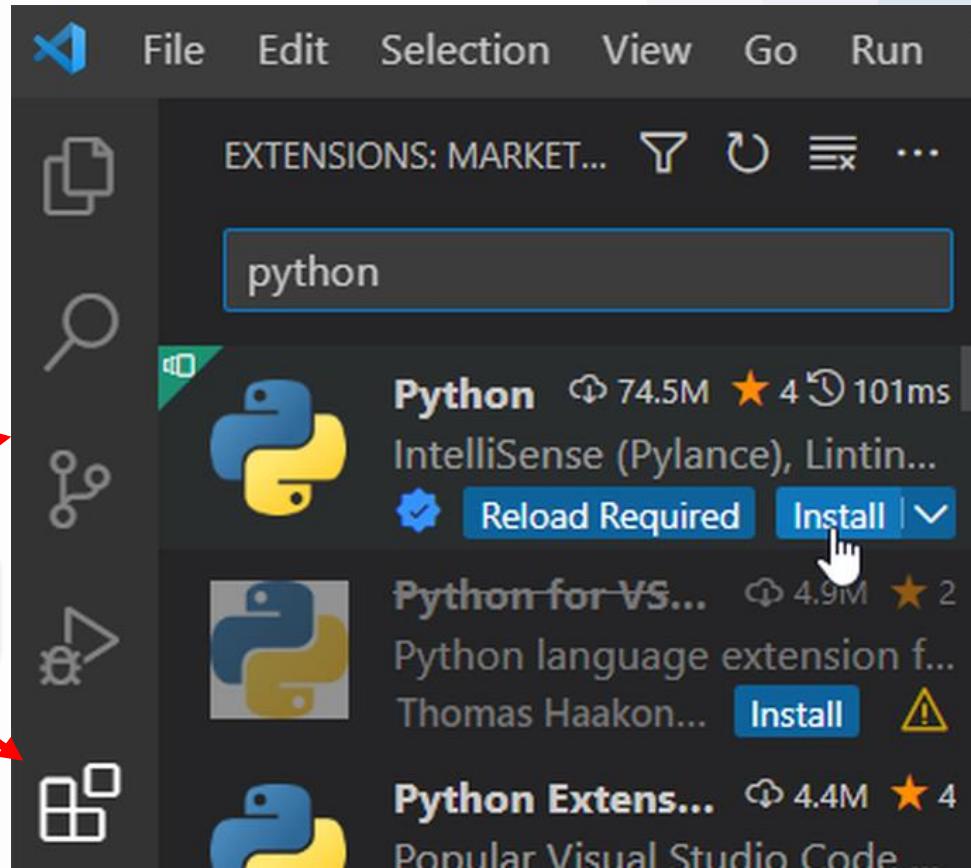
Restricted Mode 0 0 0



2. Install the Python Extension for VSCode



In the **left toolbar**, select “Extensions”, search for “Python” and click “Install”



3. Insert the Conda Path and Default Interpreter Path

3.a. Go to:

“File” > “Preferences” > “Settings” > “Extensions” > “Python”

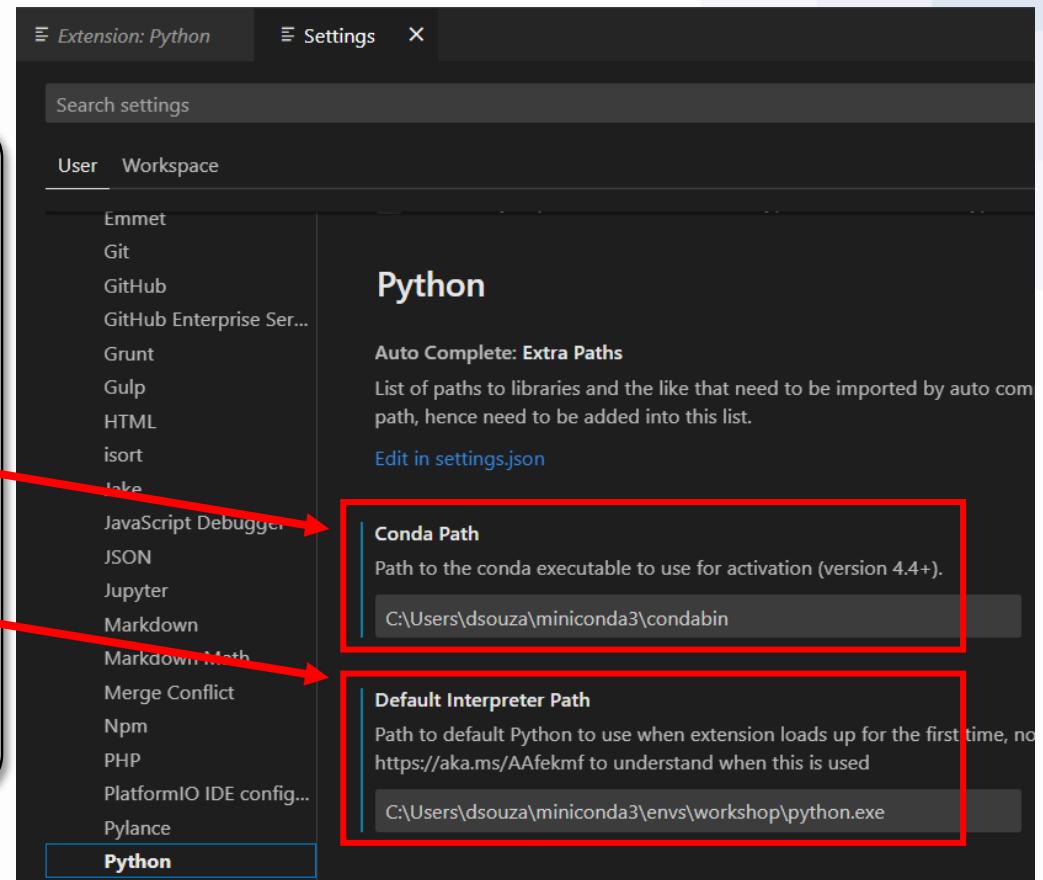
3.b. In “CondaPath” insert the path to the conda executable, in my case:

C:\Users\dsouza\miniconda3\condabin

3.c. In “Default Interpreter Path”, insert the default Python to use, in my case:

C:\Users\dsouza\miniconda3\envs\workshop\python.exe

Now we are ready to create our Python script that will be executed with our conda env.



4. Select Your Working Folder

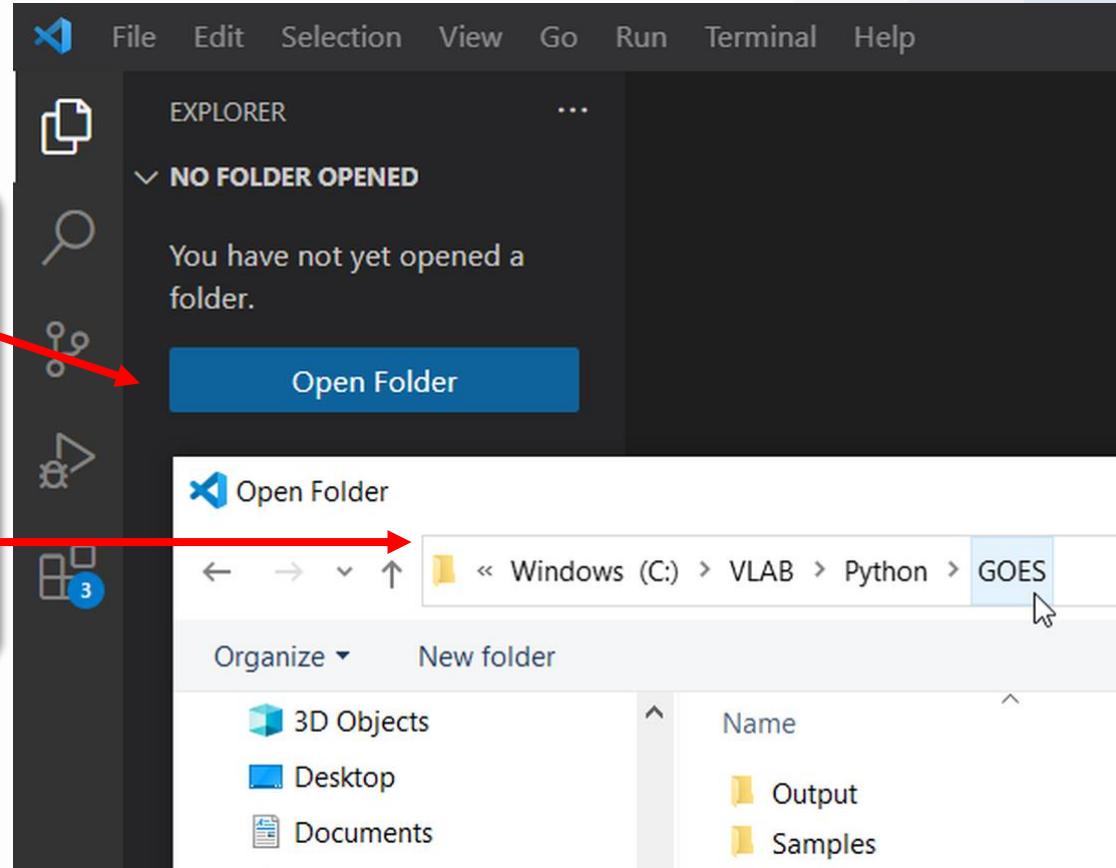
In the left toolbar, select “Explorer” (first icon), click at “Open Folder”

and select the folder where you have your scripts / files:

In this example, we'll select:

C:\VLAB\Python\GOES

But it may be **any folder you want.**



5. Open Your Script

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a file tree with a folder named 'GOES' containing various image files (Image_01.png, Image_02.png, etc.) and shapefiles (ne_10m_admin_1_states_provinces.dbf, ne_10m_admin_1_states_provinces.shp, ne_10m_admin_1_states_provinces.shx). A red box highlights the 'script_01.py' file, which is currently selected. The main editor area shows the content of 'script_01.py'. The status bar at the bottom indicates the script is 3.11.0 (conda) and the current line is Ln 11, Col 23.

```
# Training: Python and GOES-R Imagery: Script 1 - Basic Plot / Extracting Pixel Values
# Required modules
from netCDF4 import Dataset      # Read / Write NetCDF4 files
import matplotlib.pyplot as plt   # Plotting library
# Open the GOES-R image
# Download files at this link: http://home.chpc.utah.edu/~u0553130/Brian_Blaylock/cgi-bin/goes16_download.cgi
file = Dataset("OR_ABI-L2-CMIPF-M6C13_G16_s20191981200396_e20191981210116_c20191981210189.nc")
# Get the pixel values
data = file.variables['CMI'][:]
# Choose the plot size (width x height, in inches)
plt.figure(figsize=(7,7))
# Plot the image
plt.imshow(data, vmin=193, vmax=313, cmap='Greys')
# Save the image
plt.savefig('Image_01.png')
# Show the image
plt.show()
```

The files in your **working folder** will appear in the **explorer**. Select the **python script** and it will appear on a new tab at the right side.

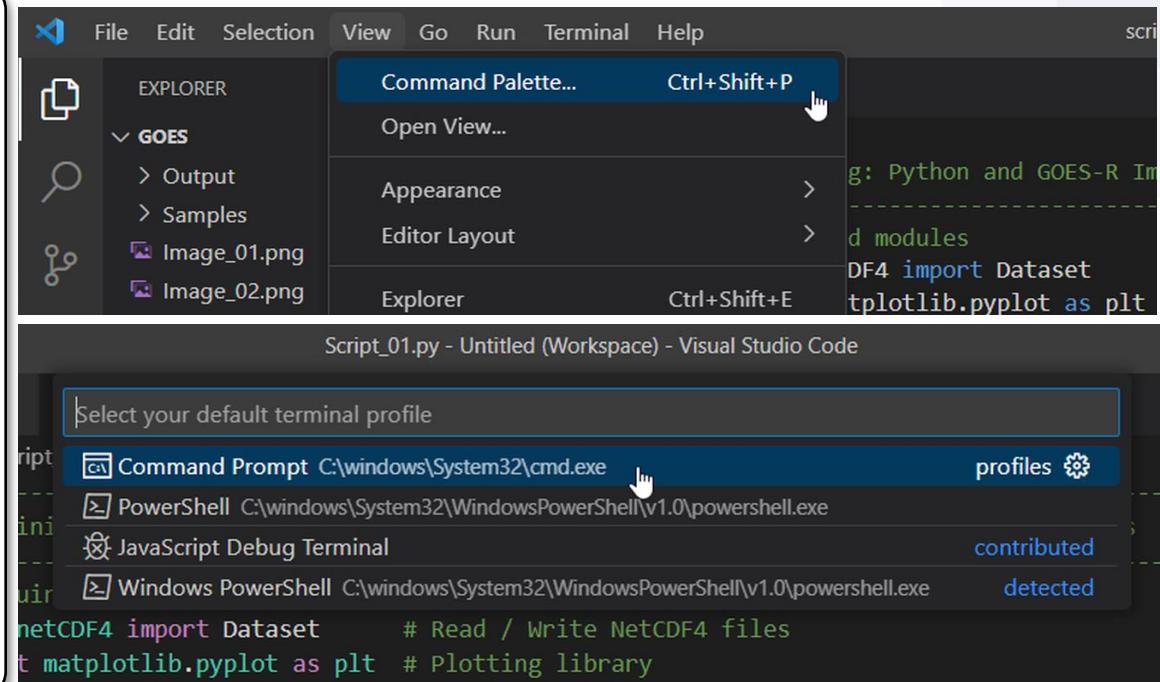
6. Changing the Default Terminal

If we want to execute conda commands (to list the installed libraries for example) directly in the terminal available in Visual Studio Code, we need to change the default terminal to the Windows **Command Prompt** ([reference](#)):

1. Open the command palette:
View > “Command Palette...”
(or **Control+Shift+P**)

2. Search for
“Terminal: Select Default Profile”

3. Select the desired default terminal
(**“Command Prompt”** in my case).



7. Running Scripts With the Conda Environment

After all these steps, if you click “Run Python File” (play button) in the upper right, the selected python script will be executed with the default conda env selected in Step 5.

You will see your default conda environment (“workshop” in this case) being activated and the python script being executed automatically in the terminal.

The screenshot shows the Visual Studio Code interface. A red arrow points from the text above to the play button in the top right corner of the code editor. Another red arrow points from the text below to the terminal window at the bottom.

Help script_01.py - GOES - Visual Studio Code

```
script_01.py ...
1  # Training: Python and GOES-R Imagery: Script 1 - Basic Plot / Extracting Pixel Values
2  #-----
3  # Required modules
4  from netCDF4 import Dataset      # Read / Write NetCDF4 files
5  import matplotlib.pyplot as plt  # Plotting library
6  #-----
7  # Open the GOES-R image
8  # Download files at this link: http://home.chpc.utah.edu/~u0553130/Brian_Blaylock/cgi-
9  file = Dataset("OR_ABI-L2-CMIPF-M6C13_G16_s20191981200396_e20191981210116_c20191981210
10
11 # Get the pixel values
12 data = file.variables['CMI'][:]
13 #
14 # Choose the plot size (width x height, in inches)
15 plt.figure(figsize=(7,7))

20 # Save the image
21 plt.savefig('Image_01.png')
22
23 # Show the image
24 plt.show()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Microsoft Windows [Version 10.0.19044.2846]
(c) Microsoft Corporation. All rights reserved.

(base) C:\VLAB\Python\GOES>conda activate C:\Users\dsouza\miniconda3\envs\workshop
(workshop) C:\VLAB\Python\GOES>cd c:/VLAB/Python/GOES
(workshop) C:\VLAB\Python\GOES>C:/Users/dsouza/miniconda3/envs/workshop/python.exe c:/VLAB/Python/GOES/script_01.py
```

Ln 12, Col 32 Spaces: 4 UTF-8 CRLF Python 3.11.0 (conda)

8. The Plot Will Appear

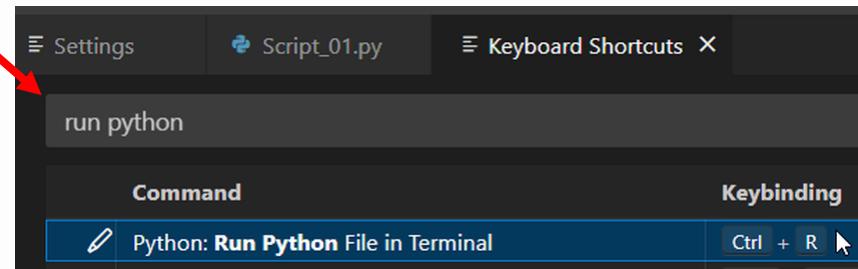
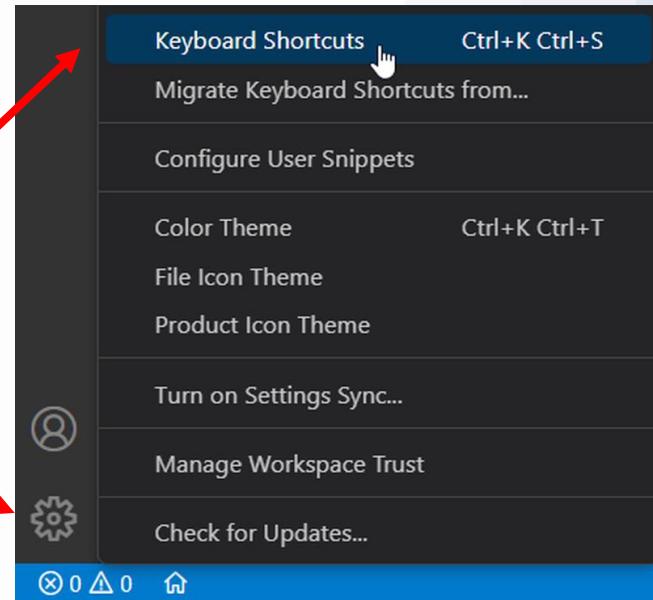
The screenshot shows a Visual Studio Code interface with the following details:

- EXPLORER:** Shows a folder named "GOES" containing various files like "Image_01.png" through "Image_08.png", "IR4AVHRR6.cpt", and several shapefiles (dbf, shp, shx) and geotiffs (G16_s2019198...).
- CODE:** The active file is "script_01.py". The code uses netCDF4 and matplotlib.pyplot to read a GOES-R image dataset, extract pixel values, and save a grayscale plot titled "Image_01.png".
- TERMINAL:** Displays the command-line output of running the script in a Windows environment, showing the activation of a conda environment and the execution of the script.
- PLOT:** A separate window titled "Figure 1" displays a global satellite image of Earth, centered on the Atlantic Ocean, showing cloud cover and landmasses.

9. Extra: Using a Keyboard Shortcut to Run Python Scripts

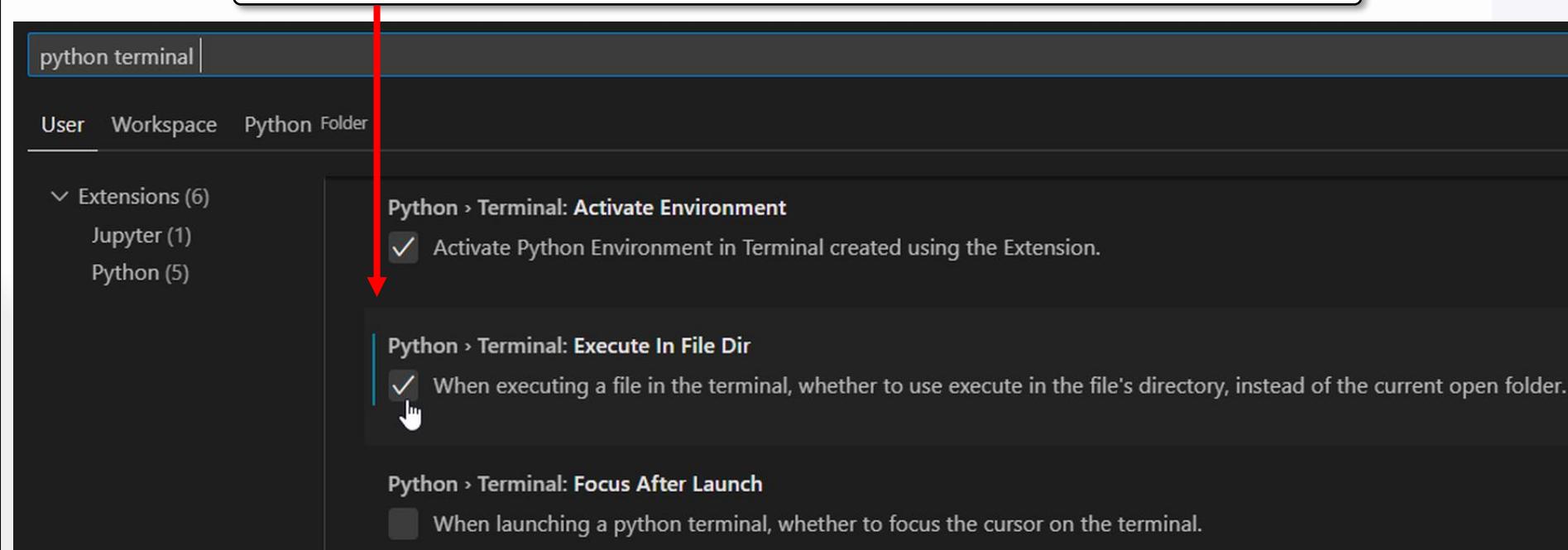
By default, you need to click on the “Run Python File” (Play button) in the upper right corner, as seen on the previous step. If you want to use a shortcut, click on the **gear icon** in the lower left and select “Keyboard shortcuts” ([reference](#)).

Search for “run python in terminal”, select “Python: Run Python File in Terminal” and assign the desired shortcut (“Ctrl+R”, in this example).



10. Extra: Reading Files in the Same Work Folder

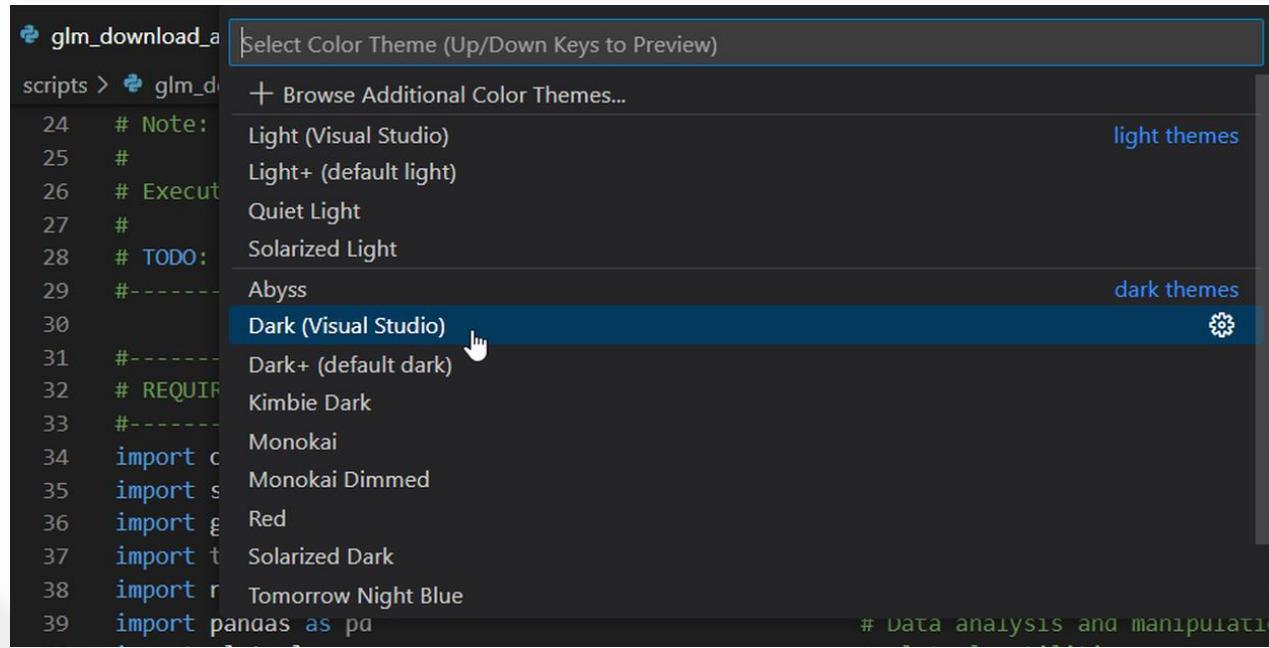
By default you need to specify the full path where a given file is. If you want to put the file in the same folder and **just insert the file name in the code**, access the **User Settings (Gear icon > Settings)**, search for “**python terminal execute in file dir**” and check this option ([reference](#)).



11. Extra: Changing the Editor Color Scheme

Personally, I don't like the default color scheme of the editor, called the "Dark+ (default dark)".

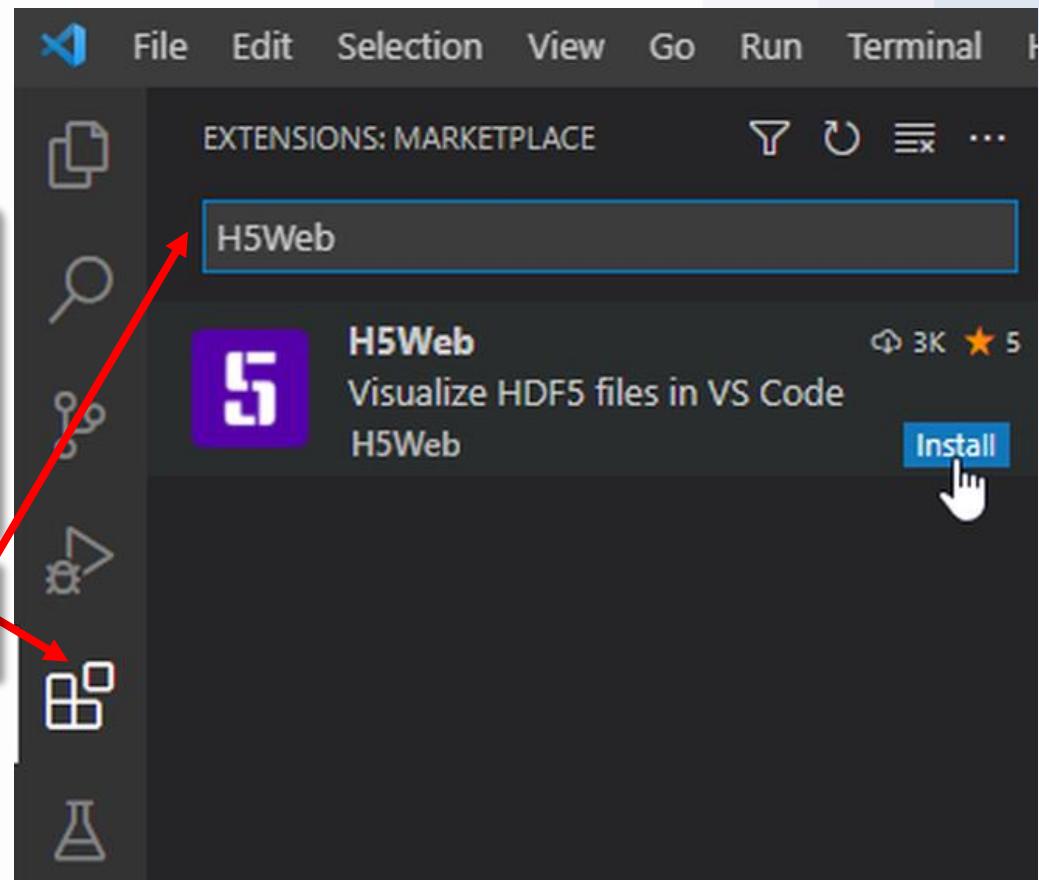
If you want to change the color scheme, go to **File > Preferences > Theme > Color Theme** and select your preferred theme.



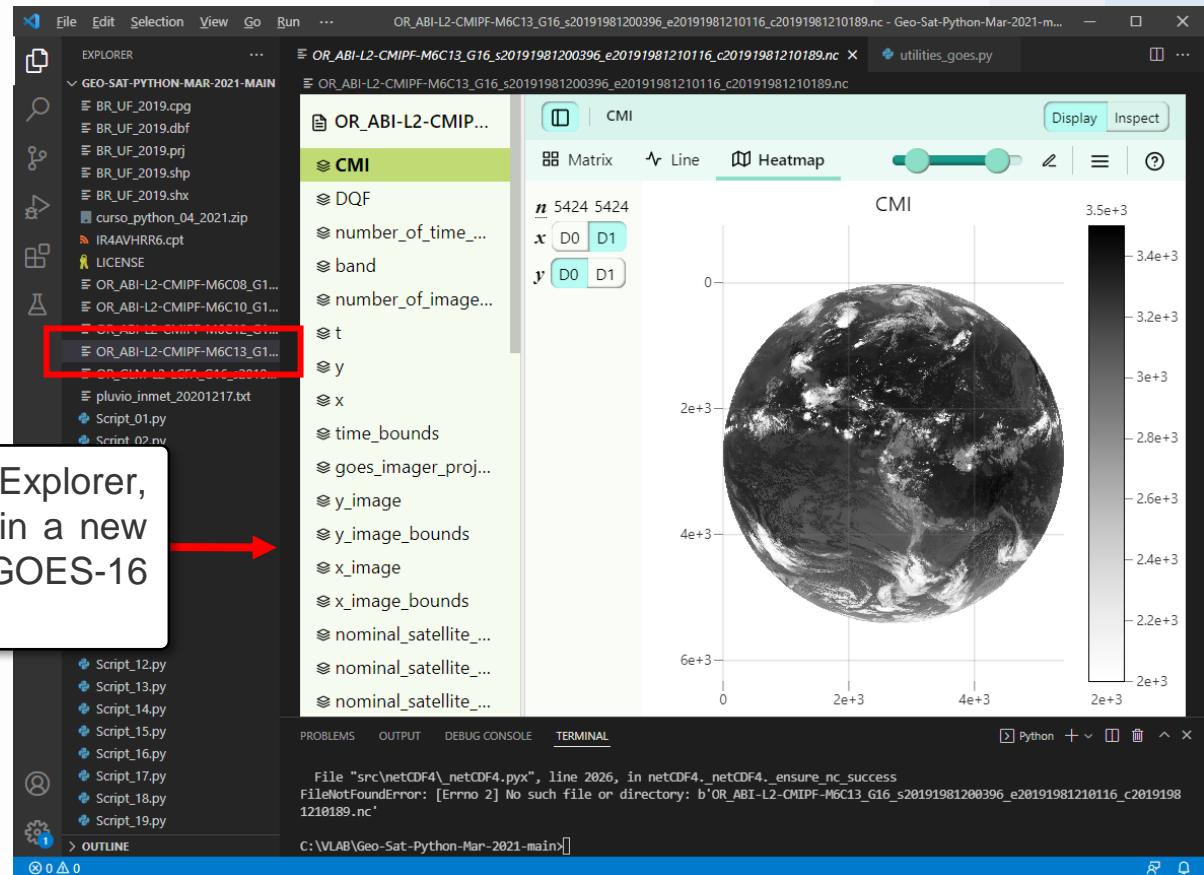
12. Extra: Installing Extensions

The [H5Web extension](#) for **VSCode** allows you to explore and visualize **HDF5 / NetCDF** files **directly in the IDE**, which is a great feature to have during the development of new scripts. It really makes things faster, as you don't need to use a separate tool like [NASA's Panoply](#) for example.

To install it, access the “**Extensions**” in the VSCode left toolbar and search for “**H5Web**”.



12. Extra: Installing Extensions



After the installation, in the VSCode Explorer, click on a file and it will be visualized in a new tab. In this example, I'm displaying a GOES-16 NetCDF (CMI dataset).

12. Extra: Installing Extensions

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "GEO-SAT-PYTHON-MAR-2021-MAIN" containing various files like "BR_UF_2019.cpg", "BR_UF_2019.dbf", "BR_UF_2019.prj", "BR_UF_2019.shp", "BR_UF_2019.shx", "curso_python_04_2021.zip", "IR4AVHRR6 cpt", "LICENSE", "OR_ABI-L2-CMIPF-M6C08_G1...", "OR_ABI-L2-CMIPF-M6C10_G1...", "OR_ABI-L2-CMIPF-M6C12_G1...", "OR_ABI-L2-CMIPF-M6C13_G1...", "OR_GLM-L2-LCFA_G16_s2019...", and "pluvio_inmet_20201217.txt".
- Code Editor:** Displays the file "goes_imager_projection.py" with the following code:

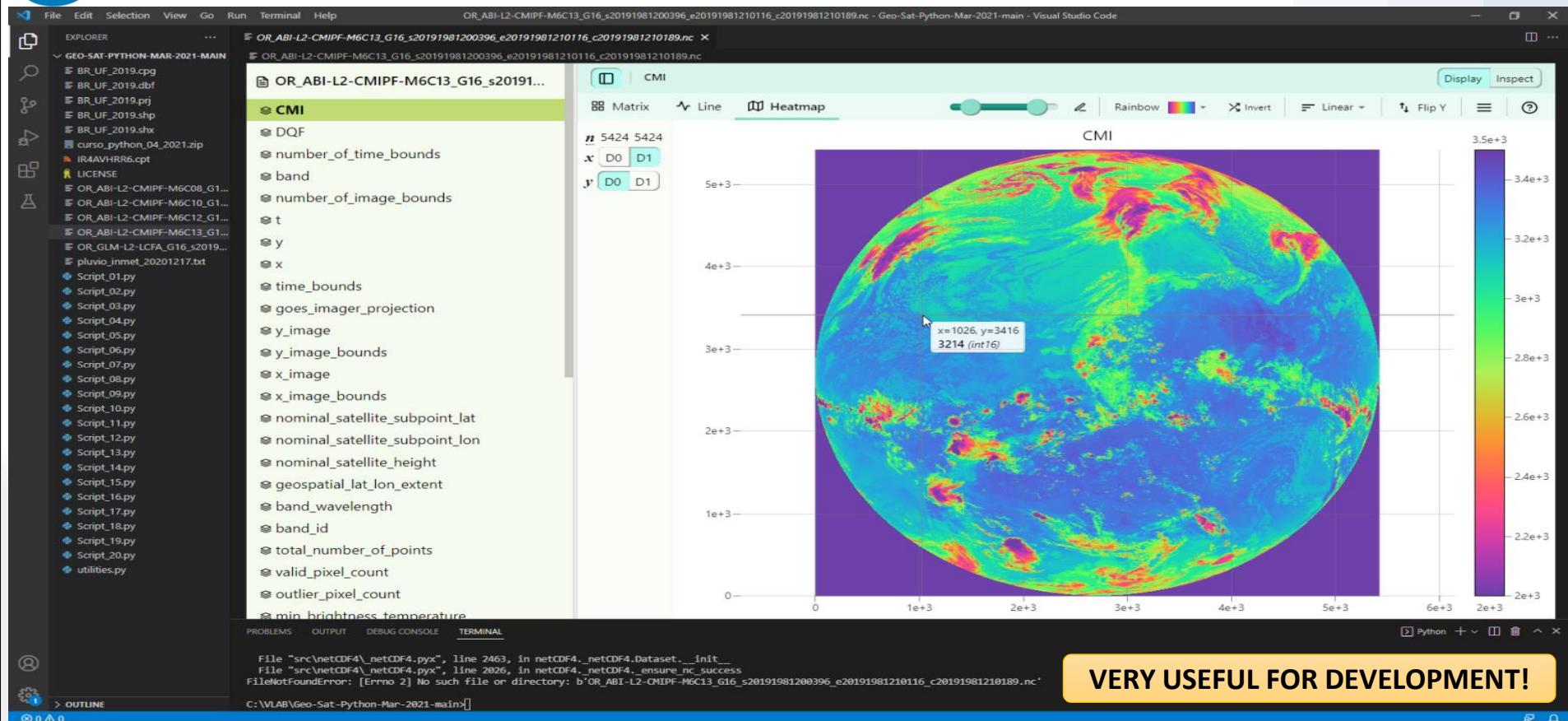
```
File "src\netCDF4\netCDF4.pyx", line 2026, in netCDF4.netCDF4._ensure_nc_success
FileNotFoundError: [Errno 2] No such file or directory: b'OR_ABI-L2-CMIPF-M6C13_G16_s20191981200396_e20191981210116_c20191981210189.nc'
```
- Properties Panel:** Shows the dataset "goes_imager_projection" with the following attributes:

Path	/goes_imager_projection
Name	goes_imager_projection
Type	Integer, 32-bit, little-endian
Shape	Scalar
Raw	▶ Inspect
- Attributes Panel:** Shows the attributes of the dataset:

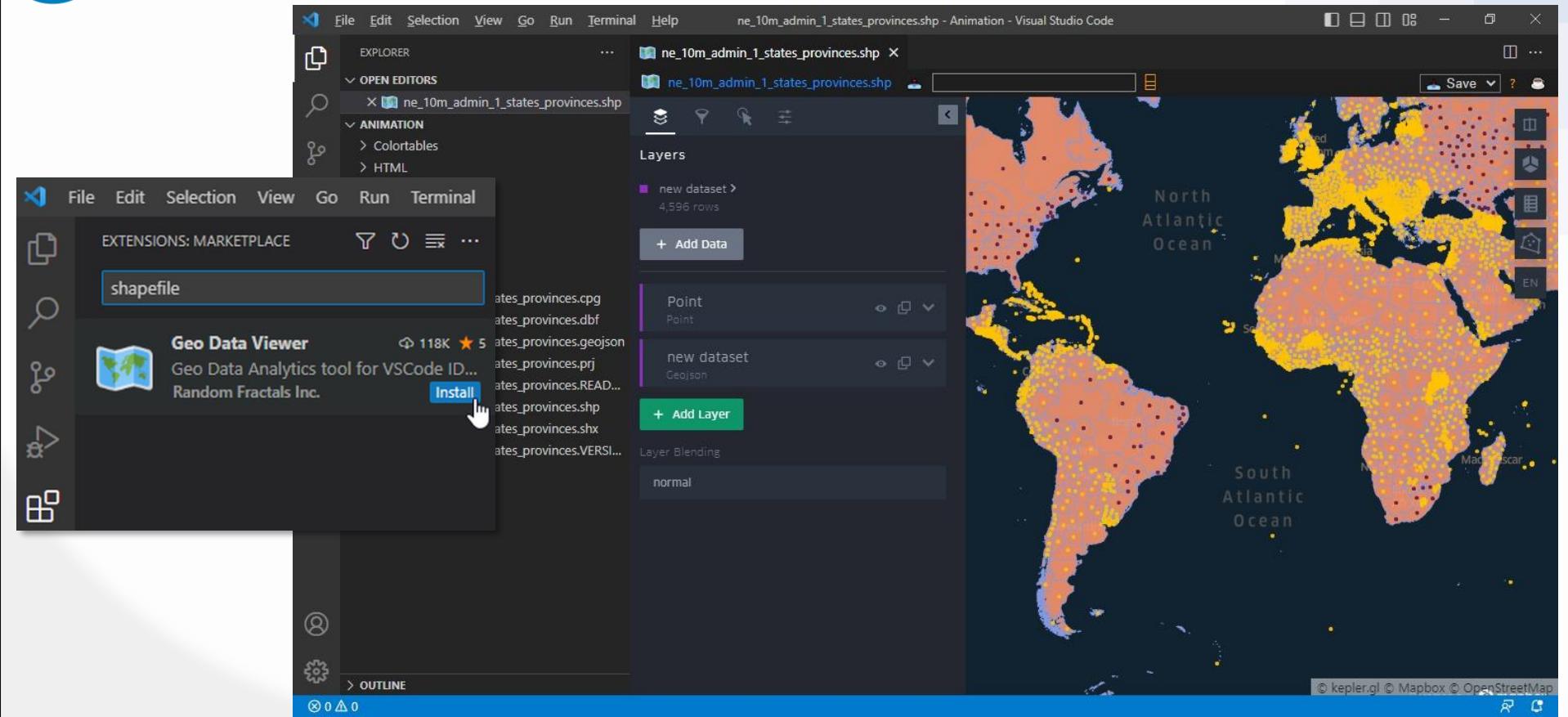
Attribute	Value
long_name	"GOES-R ABI fixed grid projection"
grid_mapping_name	"geostationary"
perspective_point_height	[35786023]
semi_major_axis	[6378137]
semi_minor_axis	[6356752.31414]
inverse_flattening	[298.2572221]
latitude_of_projection_origin	[0]

You can also inspect the dataset attributes instead of displaying it.

12. Extra: Installing Extensions



12. Extra: Installing Extensions



13. Extra: Running Script in Interactive Window

The screenshot shows a Visual Studio Code interface with the following components:

- Left Sidebar:** Shows a file tree under the 'GOES' folder, including files like 'Image_01.png' through 'Image_13.png', 'IR4AVHRR6.cpt', and various script files from 'script_01.py' to 'script_13.py'.
- Terminal:** Displays command-line output related to activating a conda environment and running a script:

```
C:\VLAB\Python\GOES>C:/SHOWCast_v2_5_1/Miniconda3/Scripts/activate  
(base) C:\VLAB\Python\GOES>conda activate C:\Users\dsouza\miniconda3\envs\workshop  
(workshop) C:\VLAB\Python\GOES>cd c:/VLAB/Python/GOES  
(workshop) C:\VLAB\Python\GOES>C:/Users/dsouza/miniconda3/envs/workshop/python.exe c:/VLAB/Python/GOES/script_01.py
```
- Code Editor:** Shows the content of 'script_01.py'. The code reads a NetCDF file ('OR_ABI-L2-CMIPF-M6C13_G16_s20191981200396_e20191981210116_c20191981210116.nc') and creates a grayscale plot of Earth's surface using matplotlib.
- Interactive Window:** A context menu is open over the code editor, with the option 'Run Current File in Interactive Window' highlighted in blue.
- Output Panel:** Shows the output of the script execution, including the path to the generated image 'Image_01.png'.
- Figure View:** Displays the resulting grayscale map of Earth's surface.

Note: necessary to install ipykernel
conda install -c anaconda ipykernel

14. Extra: Split Editor

File Edit Selection View Go Run Terminal Help

script_02.py - GOES - Visual Studio Code

EXPLORER

- > GOES
 - > Output
 - > Samples
 - Image_01.png
 - Image_02.png
 - Image_03.png
 - Image_04.png
 - Image_05.png
 - Image_06.png
 - Image_07.png
 - Image_08.png
 - IRAVHRR6 cpt
 - ne_10m_admin_1_stat...
 - ne_10m_admin_1_stat...
 - ne_10m_admin_1_stat...
 - OR_ABI-L2-CMIPF-M...
 - OR_ABI-L2-CMIPF-M...
 - OR_ABI-L2-CMIPF-M...
 - OR_ABI-L2-CMIPF-M...
 - OR_GLM-L2-LCFA_G1...
- script_01.py
- script_02.py
- script_03.py
- script_04.py
- script_05.py
- script_06.py
- script_07.py
- script_08.py
- script_09.py
- script_10.py
- script_11.py
- script_12.py
- script_13.py

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
C:\VLAB\Python\GOES>C:/SHOWCast_v_2_5_1/Miniconda3/Scripts/activate  
(base) C:\VLAB\Python\GOES>conda activate C:\Users\dsouza\miniconda3\envs\workshop  
(workshop) C:\VLAB\Python\GOES>conda install -c anaconda ipykernel  
Collecting package metadata (current_repodata.json): done  
Solving environment: done
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.0 (conda)

script_01.py > ...

```
1 # Training: Python and GOES-R Imagery: Script 1
2 #
3 # Required modules
4 from netCDF4 import Dataset      # Read / Write
5 import matplotlib.pyplot as plt  # Plotting library
6 #
7 # Open the GOES-R image
8 # Download files at this link: http://home.chpc.utah.edu/~chpcdata/goes/ABI/L2/CMIPF/M6C13/G16_S20190101T00Z.nc
9 file = Dataset("OR_ABI-L2-CMIPF-M6C13_G16_S20190101T00Z.nc")
10
11 # Get the pixel values
12 data = file.variables['CMI'][:]
13 #
14 # Choose the plot size (width x height, in inches)
15 plt.figure(figsize=(7,7))
16
17 # Plot the image
18 plt.imshow(data, vmin=193, vmax=313, cmap='Greys_r')
19 #
20 # Save the image
21 plt.savefig('Image_01.png')
22
23 # Show the image
24 plt.show()
25
26
27
```

SCRIPT 1

script_02.py > ...

```
1 # Training: Python and GOES-R Imagery: Script 2
2 #
3 # Required modules
4 from netCDF4 import Dataset      # Read / Write
5 import matplotlib.pyplot as plt  # Plotting library
6 from datetime import datetime    # Basic time functions
7 #
8 # Open the GOES-R image
9 # Download files at this link: http://home.chpc.utah.edu/~chpcdata/goes/ABI/L2/CMIPF/M6C13/G16_S20190101T00Z.nc
10 file = Dataset("OR_ABI-L2-CMIPF-M6C13_G16_S20190101T00Z.nc")
11
12 # Get the pixel values
13 data = file.variables['CMI'][:] - 273.15
14 #
15 # choose the plot size (width x height, in inches)
16 plt.figure(figsize=(7,7))
17
18 # Plot the image
19 plt.imshow(data, vmin=-80, vmax=40, cmap='Greys_r')
20
21 # Add a colorbar
22 plt.colorbar(label='Brightness Temperature')
23
24 # Extract the date
25 date = (datetime.strptime(file.time_coverage_start, '%Y-%m-%dT%H:%M:%S.%f') + timedelta(hours=4)).strftime('%Y-%m-%d %H:%M:%S')
26
27 # Add a title
28 plt.title(date, fontstyle='italic', fontweight='bold', fontsize=10, loc='left')
29
30
```

SCRIPT 2

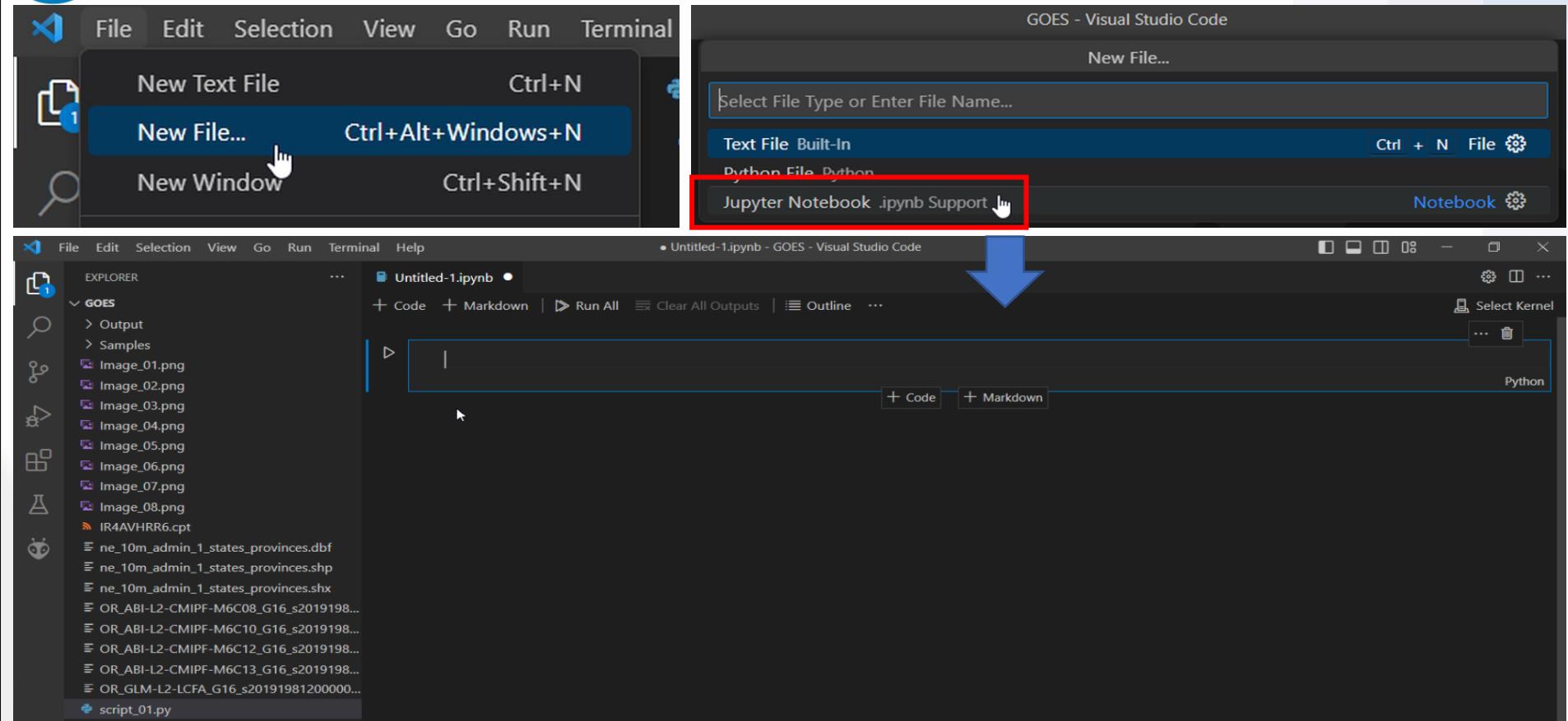
Interactive - script_01.py

Interactive-1.interactive

Clear All ... workshop (Python 3.11.0)
[MSC v.1929 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license'
for more information
IPython 8.12.0 -- An enhanced
Interactive Python. Type '?' for help.

Training: Python and ...

Creating a Jupyter Notebook



Creating a Jupyter Notebook

A screenshot of a Jupyter Notebook interface. At the top, there's a toolbar with a file icon, the title "Untitled-1.ipynb", and several buttons: "Code", "Markdown", "Run All", "Clear All Outputs", "Outline", and a "..." button. Below the toolbar, a code cell contains the Python code `print("Hello World!")`. To the left of the code cell is a button with a play icon and a hand cursor icon. A red arrow points from the text below to the dropdown menu for selecting a kernel. The dropdown menu shows "Select kernel for 'Untitled-1.ipynb'" at the top, followed by "workshop (Python 3.11.0) ~\miniconda3\envs\workshop\python.exe" which is highlighted with a blue background, and "Select Another Kernel..." at the bottom.

When running a Jupyter Notebook for the first time, it is necessary to choose the conda env:

Select kernel for 'Untitled-1.ipynb'

workshop (Python 3.11.0) ~\miniconda3\envs\workshop\python.exe

Select Another Kernel...

Creating a Jupyter Notebook

The screenshot shows the Visual Studio Code interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. A status bar at the top right indicates "Untitled-1.ipynb - GOES - Visual Studio Code". The left sidebar has icons for Explorer, Search, Open, and others. The main area displays a Jupyter Notebook cell with the title "This is a text cell". Below it is a code cell containing the Python command `print("Hello World! This is a Python command.")`. The output of the cell is "Hello World! This is a Python command." with a green checkmark and "0.0s" execution time. At the bottom right of the code cell are buttons for "+ Code" and "+ Markdown", with a cursor hovering over the "+ Markdown" button.

File Edit Selection View Go Run Terminal Help

Untitled-1.ipynb - GOES - Visual Studio Code

EXPLORER ...

GOES

- > Output
- > Samples
- Image_01.png
- Image_02.png
- Image_03.png
- Image_04.png
- Image_05.png
- Image_06.png
- Image_07.png
- Image_08.png
- IR4AVHRR6 cpt
- ne_10m_admin_1_states_provinces dbf
- ne_10m_admin_1_states_provinces shp
- ne_10m_admin_1_states_provinces shx
- OR_ABI-L2-CMIPF-M6C08_G16_s2019198...
- OR_ABI-L2-CMIPF-M6C10_G16_s2019198...
- OR_ABI-L2-CMIPF-M6C12_G16_s2019198...
- OR_ABI-L2-CMIPF-M6C13_G16_s2019198...
- OR_GLM-L2-LCFA_G16_s20191981200000...
- script_01.py
- script_02.py
- script_03.py

This is a text cell

▶ print("Hello World! This is a Python command.")

[2] ✓ 0.0s

... Hello World! This is a Python command.

+ Code + Markdown

IDE: Everything in a Single Interface

The screenshot shows the Visual Studio Code interface with several panes highlighted by yellow boxes:

- EXPLORER**: Shows a file tree with various Python scripts and image files.
- SCRIPT**: Displays the content of `script_01.py`, which reads a GOES-R image and saves it as `Image_01.png`.

```
script_01.py > ...
1 # Training: Python and GOES-R Imagery: Script 1 - Basic Plot / Extracting Pixel Value
2 #
3 # Required modules
4 from netCDF4 import Dataset      # Read / Write NetCDF4 files
5 import matplotlib.pyplot as plt   # Plotting library
6 #
7 # open the GOES-R image
8 # Download files at this link: http://home.chpc.utah.edu/~u0553130/Brian_Blaylock/cgi
9 file = Dataset("OR_ABI-L2-CMIPF-M6C13_G16_s20191981200396_e20191981210116_c20191981210116")
10
11 # Get the pixel values
12 data = file.variables['CMI'][:]
13 #
14 # Choose the plot size (width x height, in inches)
15 plt.figure(figsize=(7,7))
16
17 # Plot the image
18 plt.imshow(data, vmin=193, vmax=313, cmap='Greys')
19
20 # Save the image
21 plt.savefig('Image_01.png')
22
23 # Show the image
24 plt.show()
```

- OUTPUT**: Displays a circular grayscale image of Earth's visible disk.
- TERMINAL**: Shows the command line history for running the script.

```
C:\VLAB\Python\GOES>c:/SHOWCast_v_2_5_1/Miniconda3/Scripts/activate
(base) C:\VLAB\Python\GOES>conda activate C:\Users\dsouza\miniconda3\envs\workshop
(workshop) C:\VLAB\Python\GOES>cd c:/VLAB/Python/GOES
(workshop) C:\VLAB\Python\GOES>C:/Users/dsouza/miniconda3/envs/workshop/python.exe c:/VLAB/Python/GOES/script_01.py
```



You Can Use VSCode With Many Languages

EXTENSIONS: MARKETPLACE

C/C++

C/C++ ms-vscode.cpptools

Microsoft | ⚡ 13.6M ★ 3.5

C/C++ IntelliSense, debugging, and code browsing.

C/C++ Extension Pack 1.0.0 ⚡ 290

Popular extensions for C++ development in Visual Studio Code.

C/C++ Themes 1.0.0 ⚡ 43K

UI Themes for C/C++ extension.

Install

Details Feature Contributions Changelog

EXTENSIONS: MARKPLACE

fortran

fortran

An extension for VS Code which provides support for the Fortran language.

Xavier Hahn ⚡ 141K ★ 4.5

Install

Modern Fortran

Fortran language support, syntax highlighting, Language Server support, Debugging, and more.

The Fortran Programming Language ⚡ 296K ★ 4.5

Install

EXTENSIONS: MARKPLACE

R

R

R Extension for Visual Studio Code

REditorSupport ⚡ 1M ★ 5

Install

R Debugger

R Debugger for VS Code

R Debugger ⚡ 183K ★ 5

Install

EXTENSIONS: MARKPLACE

matlab

MATLAB

Edit MATLAB code with syntax highlighting, linting, navigation support, and more.

MathWorks ⚡ 7K ★ 5

Install

Matlab Unofficial

MATLAB support for Visual Studio Code

Xavier Hahn ⚡ 562K ★ 4.5

Install

EXTENSIONS: MARKPLACE

C#

C#

C# for Visual Studio Code (powered by OmniSharp).

Microsoft ⚡ 21.5M ★ 3

Install

EXTENSIONS: MARKPLACE

java

Extension Pack for Java

Popular extensions for Java development that provides Java IntelliSense, debugging, and more.

Microsoft ⚡ 19.3M ★ 4

Install

Debugger for Java

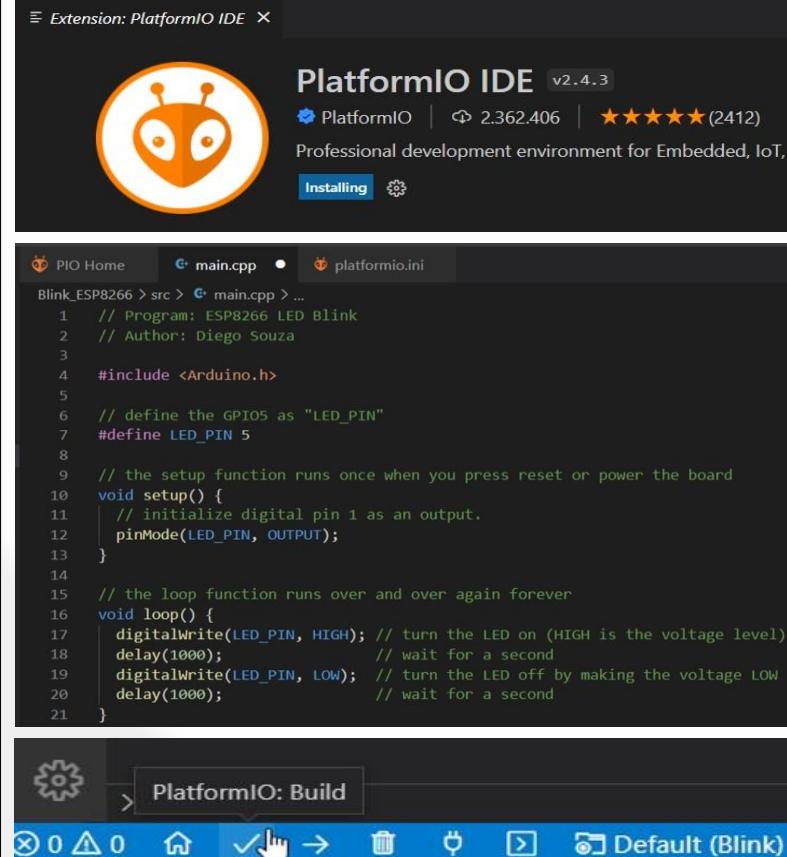
A lightweight Java debugger for Visual Studio Code

Microsoft ⚡ 21.9M ★ 4

Install

You Can Use VSCode For Many Applications

Extension: PlatformIO IDE X

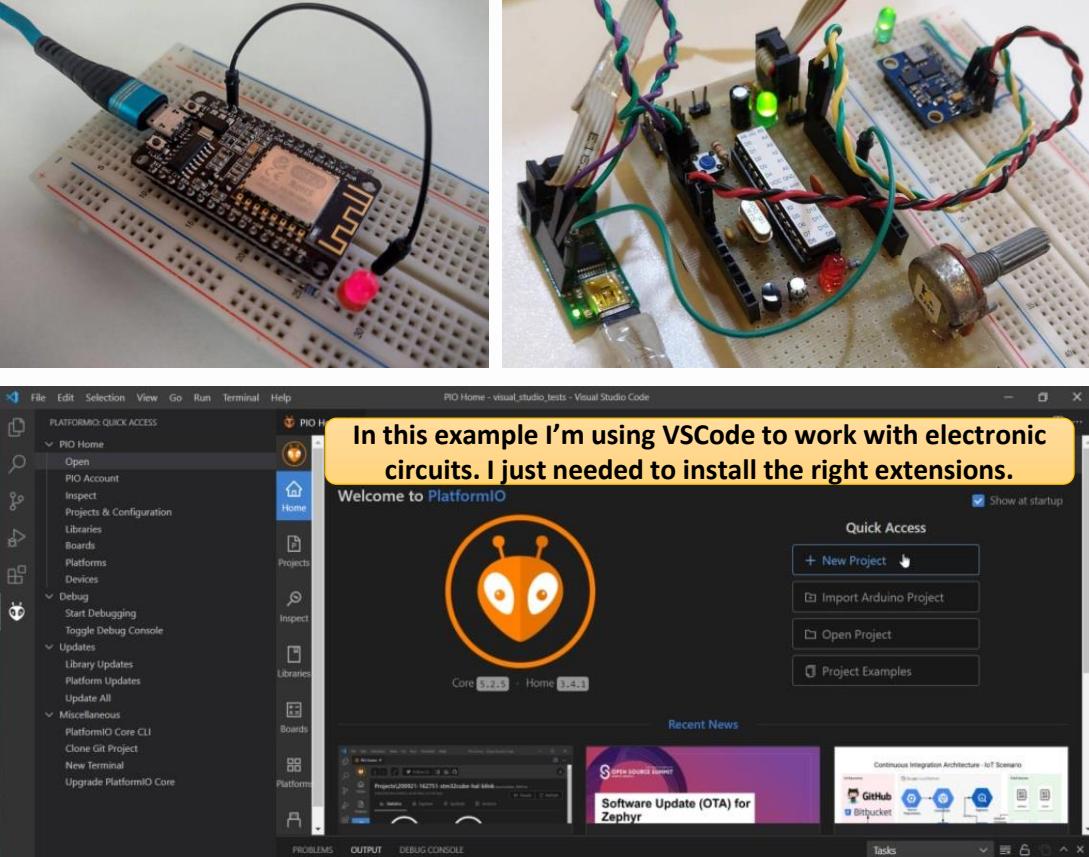


PlatformIO IDE v2.4.3
PlatformIO | 2.362.406 | ★★★★★ (2412)
Professional development environment for Embedded, IoT,
Installing

```
PIO Home main.cpp platformio.ini
Blink_ESP8266 > src > main.cpp > ...
1 // Program: ESP8266 LED Blink
2 // Author: Diego Souza
3
4 #include <Arduino.h>
5
6 // define the GPIO5 as "LED_PIN"
7 #define LED_PIN 5
8
9 // the setup function runs once when you press reset or power the board
10 void setup() {
11     // initialize digital pin 1 as an output.
12     pinMode(LED_PIN, OUTPUT);
13 }
14
15 // the loop function runs over and over again forever
16 void loop() {
17     digitalWrite(LED_PIN, HIGH); // turn the LED on (HIGH is the voltage level)
18     delay(1000);               // wait for a second
19     digitalWrite(LED_PIN, LOW); // turn the LED off by making the voltage LOW
20     delay(1000);               // wait for a second
21 }
```

PlatformIO: Build

0 0 0 ✓ → 🗑️ 🔋 Default (Blink)



In this example I'm using VSCode to work with electronic circuits. I just needed to install the right extensions.

File Edit Selection View Go Run Terminal Help

PLATFORMIO: QUICK ACCESS

- Open
- PIO Home
- Inspect
- Projects & Configuration
- Libraries
- Boards
- Platforms
- Devices
- Debug
- Start Debugging
- Toggle Debug Console
- Updates
- Library Updates
- Platform Updates
- Update All
- Miscellaneous
- PlatformIO Core CLI
- Clone Git Project
- New Terminal
- Upgrade PlatformIO Core

Welcome to PlatformIO

Quick Access

- + New Project
- Import Arduino Project
- Open Project
- Project Examples

Recent News

Core 5.2.5 · Home 3.4.3

PROBLEMS OUTPUT DEBUG CONSOLE

Software Update (OTA) for Zephyr

Continuous Integration Architecture - IoT Scenario

Github Bitbucket

Tasks

GEONETCast-Americas Training for the Eastern Caribbean States

Day 5 - May 12th

Session 2:

Hands-on

Introduction to Visual Studio Code

THANK YOU! QUESTIONS?



Diego Souza
diego.souza@inpe.br

DISSM - Meteorological Satellites and Sensors' Division
CGCT - General Coordination of Earth Sciences
INPE - National Institute for Space Research

This is made possible by the generous support of the American people through the United States Agency for International Development (USAID)

