

Aplicações em meteorologia usando Python

Cheat Sheet

Rafael Cesario de Abreu, Natália Machado Crespo, Marcia Akemi Yamasoe, Rita Yuri Ynoue

Conda

Listar pacotes instalados no ambiente atual	<code>conda list</code>
Instalar um pacote	<code>conda install PACOTE</code>
Instalar um pacote de um canal específico	<code>conda install -c CANAL PACOTE</code>
Instalar a versão específica de um pacote	<code>conda install PACOTE=VERSAO</code>
Criar um ambiente novo	<code>conda create --name NOME</code>
Ativar um ambiente	<code>conda activate NOME</code>
Desativar o ambiente atual	<code>conda deactivate</code>
Remover um ambiente	<code>conda env remove --name NOME</code>
Listar ambientes	<code>conda env list</code>

Básico da linguagem

Definição de variáveis	<code>x = 0</code>
Imprimir alguma variável ou expressão	<code>print(x)</code>
Retorna o tipo da variável	<code>type(x)</code>
Lista	<code>x = [1, 2, 3, 7, 10]</code>
Adicionar elemento ao final da lista	<code>x.append(11)</code>
Selecionar elemento da lista (Nota: No python a contagem é feita de 0 a n-1, onde n é o tamanho da lista, ou seja, o índice 2 se refere ao terceiro elemento de x)	<code>x[2]</code>
Tamanho da lista	<code>len(x)</code>

Importar pacotes (Nota: o nome as dá um apelido para o pacote, para não precisarmos digitar o nome inteiro do pacote quando formos usá-lo)	<code>import pacote as pt</code>
Utilizar uma função de um pacote	<code>np.sqrt(x)</code>
Loop for	<code>for i in range(10): print(i)</code>
Loop while (Nota: a expressão <code>i += 1</code> é equivalente a <code>i = i + 1</code>)	<code>x = 10 while i < 10: print(i) i += 1</code>
Condicional (Nota: o comando <code>elif</code> é um encurtamento para a expressão <code>else if</code>)	<code>x = 10 if x == 10: print('x e igual a 10') elif x == 5: print('x e igual a 5') else: print('x nao e nem 5 nem 10')</code>
Função	<code>def minha_funcao(x): x_squared = x * x return x_squared</code>
Criar função anônima	<code>minha_funcao = lambda x: x * x</code>

Nota: Se define os blocos de código através da indentação do código.

Numpy	
Importar o numpy	<code>import numpy as np</code>
Tamanho de uma matriz	<code>x.shape</code>
Número de dimensões da matriz	<code>x.ndim</code>
Criar array	<code>x = np.array([[1., 0., 0.], [0., 1., 2.]])</code>
Criar matriz de 1s	<code>x = np.ones((3, 4))</code>
Criar matriz de 0s	<code>x = np.zeros((3, 4))</code>
Criar matriz de um número até outro com um espaçamento definido	<code>np.arange(INICIO, FIM, ESPACAMENTO)</code>
Criar matriz entre um número até outro com um número de elementos definido	<code>np.linspace(INICIO, FIM, N°ELEMENTOS)</code>

Selecionar primeira linha de uma matriz de duas dimensões (Nota: o : (dois pontos) indica que todos os elementos daquela dimensão serão utilizados)	<code>x[0, :]</code>
Selecionar segunda coluna de uma matriz de duas dimensões	<code>x[:, 1]</code>
Criar filtro com base em uma condição (Nota: Retorna um array de booleanos (True/False))	<code>x > 1</code>
Multiplicação de matrizes ponto a ponto	<code>C = A * B</code>
Multiplicação matricial (ou produto interno)	<code>C = np.dot(A, B)</code>
Calcular média da matriz inteira	<code>x.mean()</code>
Calcular média de cada coluna	<code>x.mean(axis=0)</code>

Matplotlib

Importar o matplotlib	<code>import matplotlib.pyplot as plt</code>
Criar um gráfico de linhas	<code>plt.plot(x,y)</code>
Criar um scatter plot	<code>plt.scatter(x,y)</code>
Cria um gráfico de contornos preenchidos (shaded)	<code>plt.contourf(x, y, z)</code>
Criar uma figura com subplots (duas linhas e duas colunas)	<code>fig, axes = plt.subplots(2, 2)</code>
Cria um gráfico de contornos preenchidos (shaded)	<code>ax.set_title('TITULO')</code>
Adicionar um título no eixo x do subplot	<code>ax.set_xlabel('TITULO')</code> <code>ax.set_ylabel('TITULO')</code>
Adicionar uma legenda ao subplot	<code>ax.legend()</code>

Pandas

Importar o pandas	<code>import pandas as pd</code>
Criar uma série	<code>s = pd.Series([1, 3, 5, np.nan, 6, 8])</code>
Criar um dataframe	<code>df = pd.DataFrame({'A': [3, 4], 'B': [6, 8]}, index=pd.to_datetime(['2021-01-01', '2021-01-02']))</code>

Ler dados em texto	<code>df = pd.read_csv(NOME_DO_ARQUIVO)</code>
Visualizar as primeiras linhas	<code>df.head()</code>
Visualizar as últimas linhas	<code>df.tail()</code>
Acessar uma coluna específica	<code>df['A']</code> ou <code>df.loc[:, 'A']</code>
Acessar uma linha específica	<code>df.loc['2021-01-01']</code>
Fazer média mensal (e.g., 2019-12-31, 2021-01-31, 2021-02-28, ...)	<code>df.resample('M').mean()</code>
Calcular a média dos meses (1, 2, ..., 12)	<code>df.groupby(df.index.month).mean()</code>
Converte os dados em uma tabela do tipo planilha, com um índice, coluna e os valores	<code>df.pivot_table(index='INDEX', columns='COLUMN', values='VALUE')</code>
Aplicar uma função específica em todos os valores (no caso a raiz quadrada)	<code>df.apply(np.sqrt)</code>
Transformar o dataframe em um dataset do xarray	<code>df.to_xarray()</code>
Selecionar todas ocorrências do valor 2 na coluna A	<code>df.query('A==2')</code> ou <code>df[df['A'] == 2]</code>

Xarray

Importar o xarray	<code>import xarray as xr</code>
Ler arquivo netcdf	<code>ds = xr.open_dataset(NOME_DO_ARQUIVO)</code>
Selecionar ponto mais próximo	<code>ds.sel(lat=-23.6, lon=-46, method='nearest')</code>
Selecionar um range de datas	<code>ds.sel(time=slice('2019-01-01', '2019-01-05'))</code>
Calcular média mensal (análogo ao caso do pandas)	<code>ds.resample(time='M').mean(dim='time')</code>
Calcular média dos meses (análogo ao caso do pandas)	<code>ds.groupby('time.month').mean(dim='time')</code>
Calcular a média de todos os pontos para cada tempo	<code>ds.mean(dim=['lat', 'lon'])</code>

Cartopy

Importar o cartopy	<code>import cartopy.crs as ccrs</code>
--------------------	---

Criar uma figura com um subplot e projeção	<pre>fig, ax = plt.subplots(subplot_kw={'projection': ccrs.PlateCarree()}))</pre>
Adicionar linhas de costas	<pre>ax.coastlines('RESOLUTION')</pre>
Ajustando os eixos de longitude e latitude no mapa de acordo com a projeção desejada	<pre>ax.set_xticks(INTERVALOS_DESEJADOS_EM_X, crs=ccrs.PlateCarree()) ax.set_yticks(INTERVALOS_DESEJADOS_EM_Y, crs=ccrs.PlateCarree())</pre>
Adicionando contorno de países	<pre>ax.add_feature(ccrs.feature.BORDERS)</pre>
Inserindo linhas de grade no mapa	<pre>ax.grid()</pre>
Fazendo o plot preenchido	<pre>ax.contourf(LON, LAT, DATA, levels=INTERVALO_DESEJADO, cmap='CÓDIGO_COLORMAP_MATPLOTLIB')</pre>