Tarea

1) Descargar Git, crear una cuenta en algunos de los repositorios remotos de Git: Bitbucket, GitHub .....

2) Descargar Python 3.

Recomendaciones:

- Si usa linux o mac, hacerlo desde el gestor de paquetes usando pip. (Esto es si usted tiene experiencia en el manejo de la terminal).

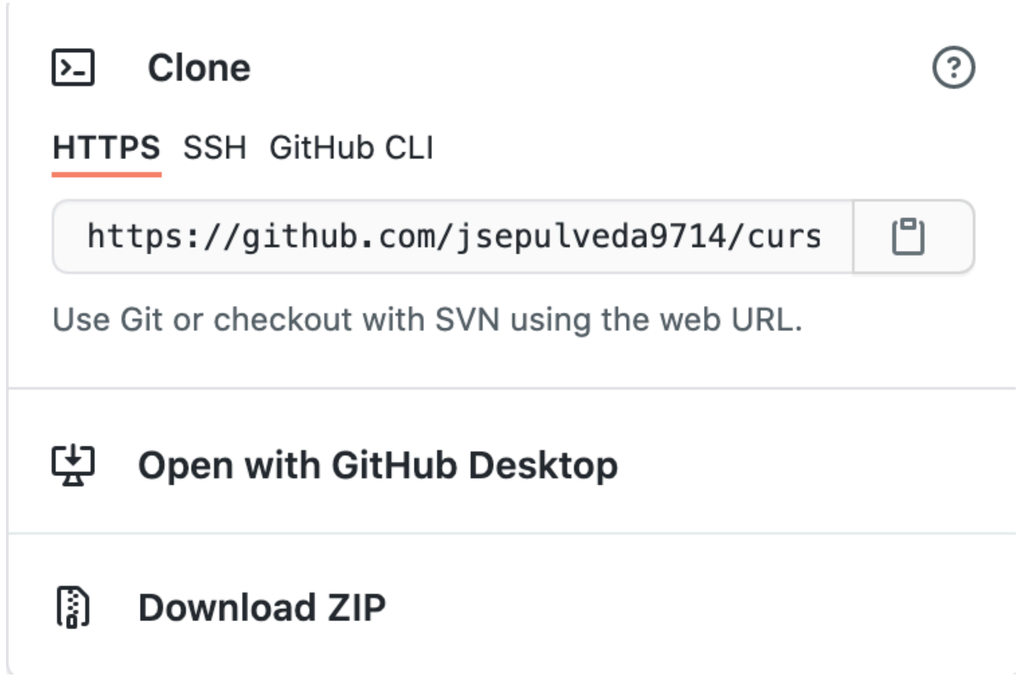
- Si no quiere complicarse una buena opción: Anaconda.

3) Crear un repositorio en Git

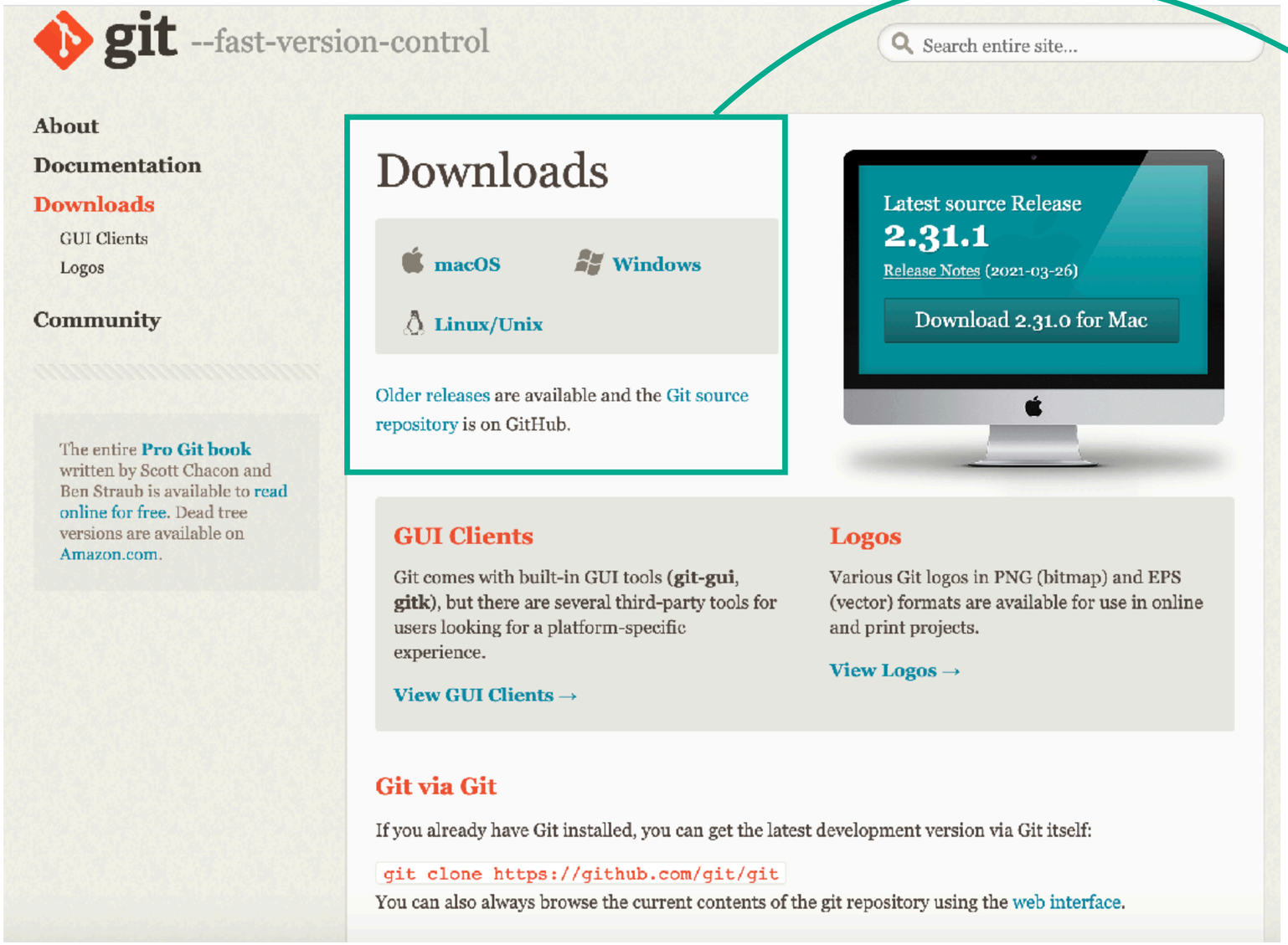
Repositorio del Curso

El contenido del curso se irá actualizando en el siguiente repositorio de GitHub

https://github.com/jsepulveda9714/curso_PythonBasico.git

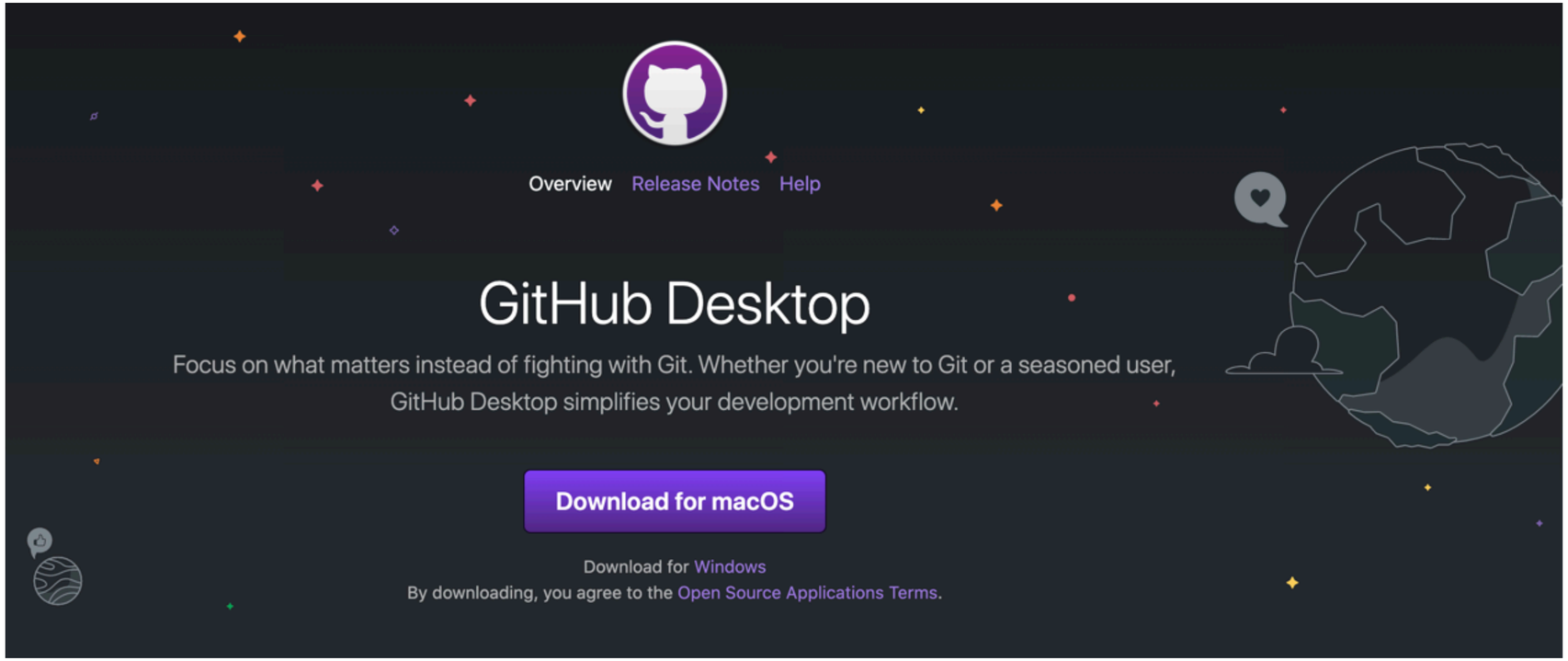


Descargar Git para el OS de preferencia



https://git-scm.com/downloads

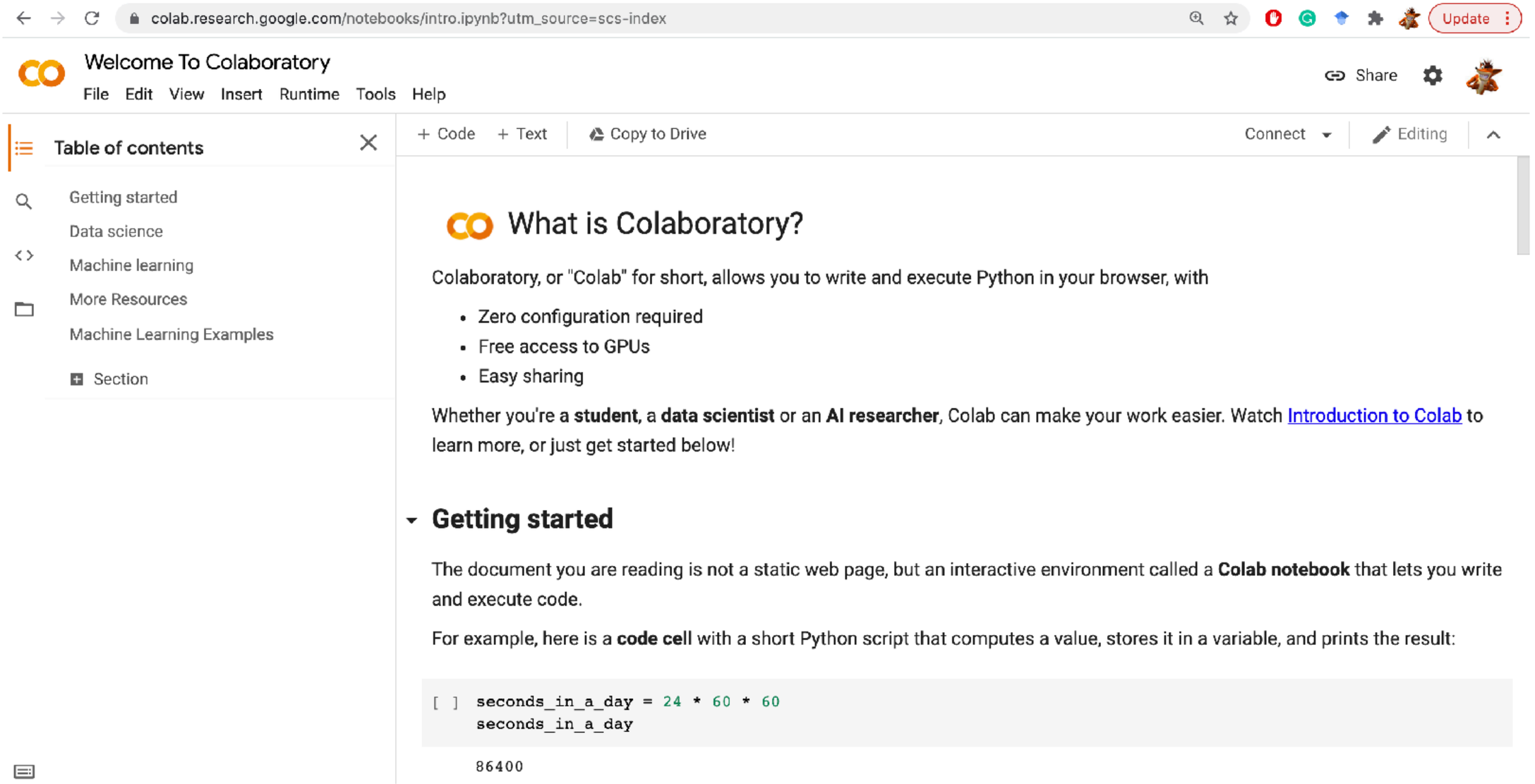
Gestor de escritorio



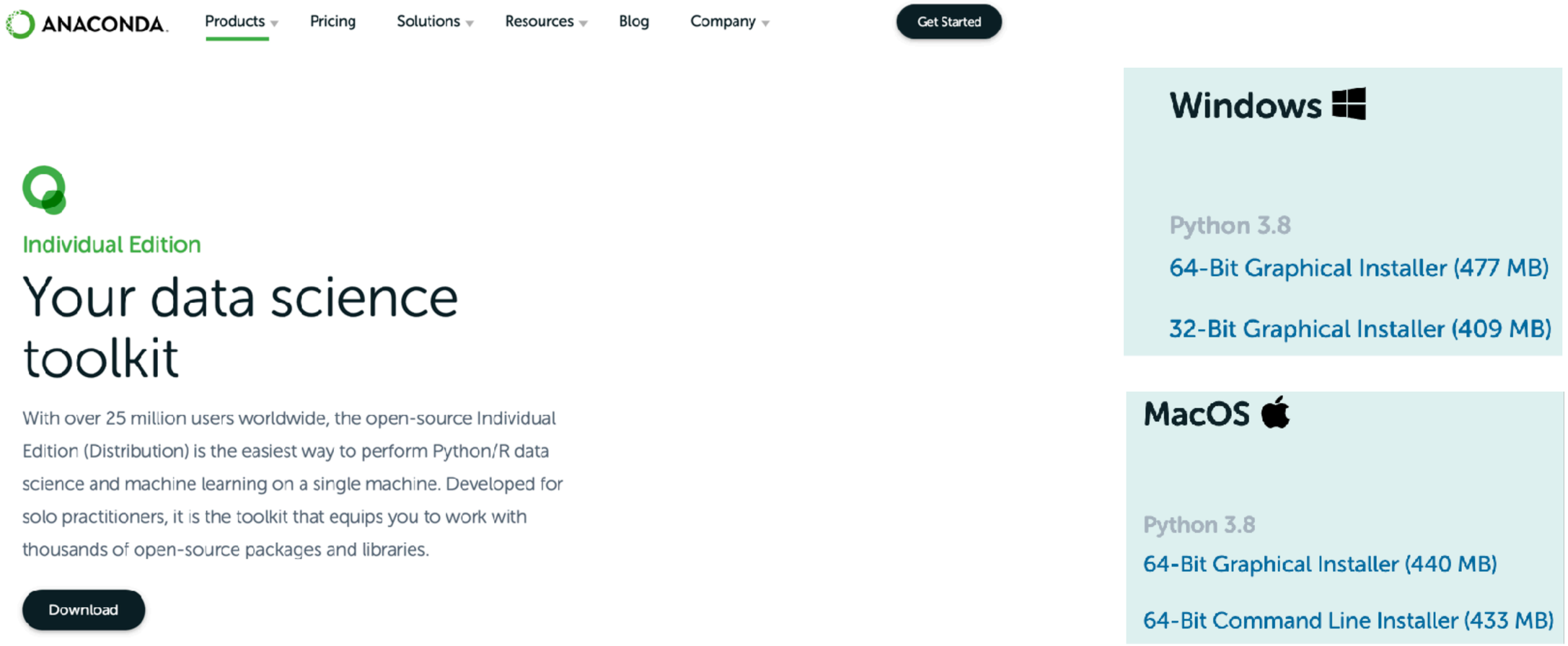
Para evitar algunas complicaciones con el uso de comandos es recomendable una interfaz de escritorio.

https://desktop.github.com/

Inicio Rápido en Python

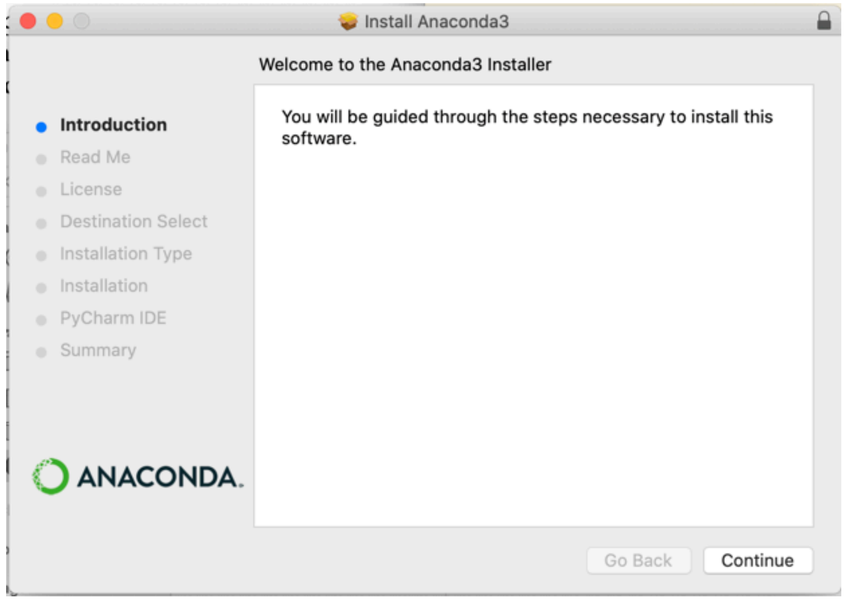


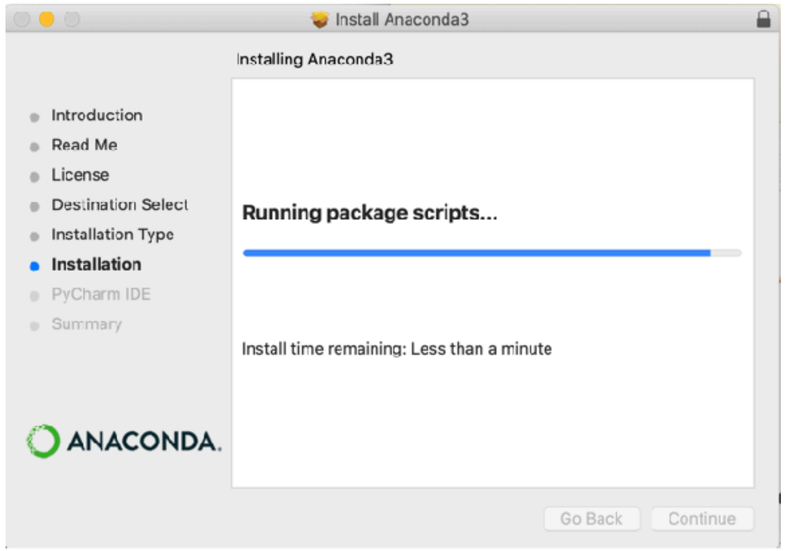
Anaconda: Un pre-compilado muy completo

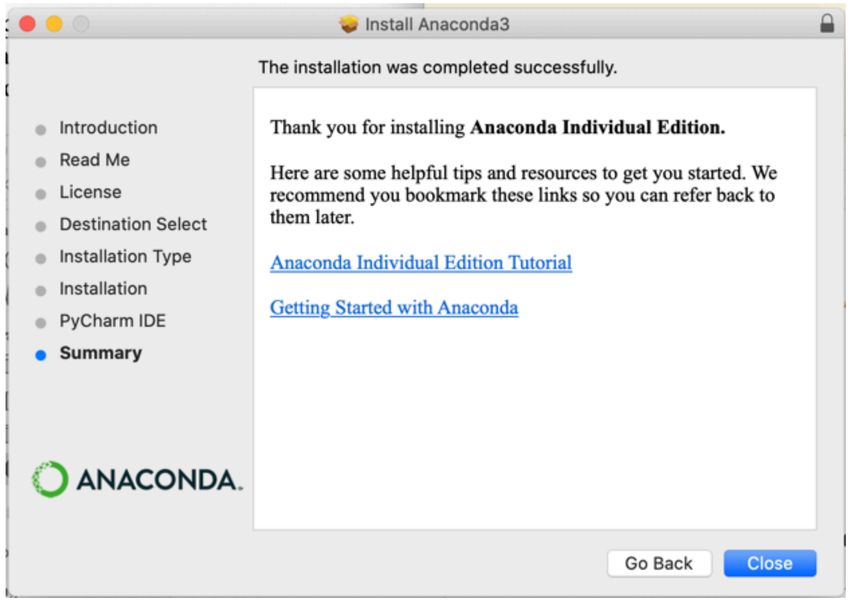


https://www.anaconda.com/products/individual

Proceso de instalación

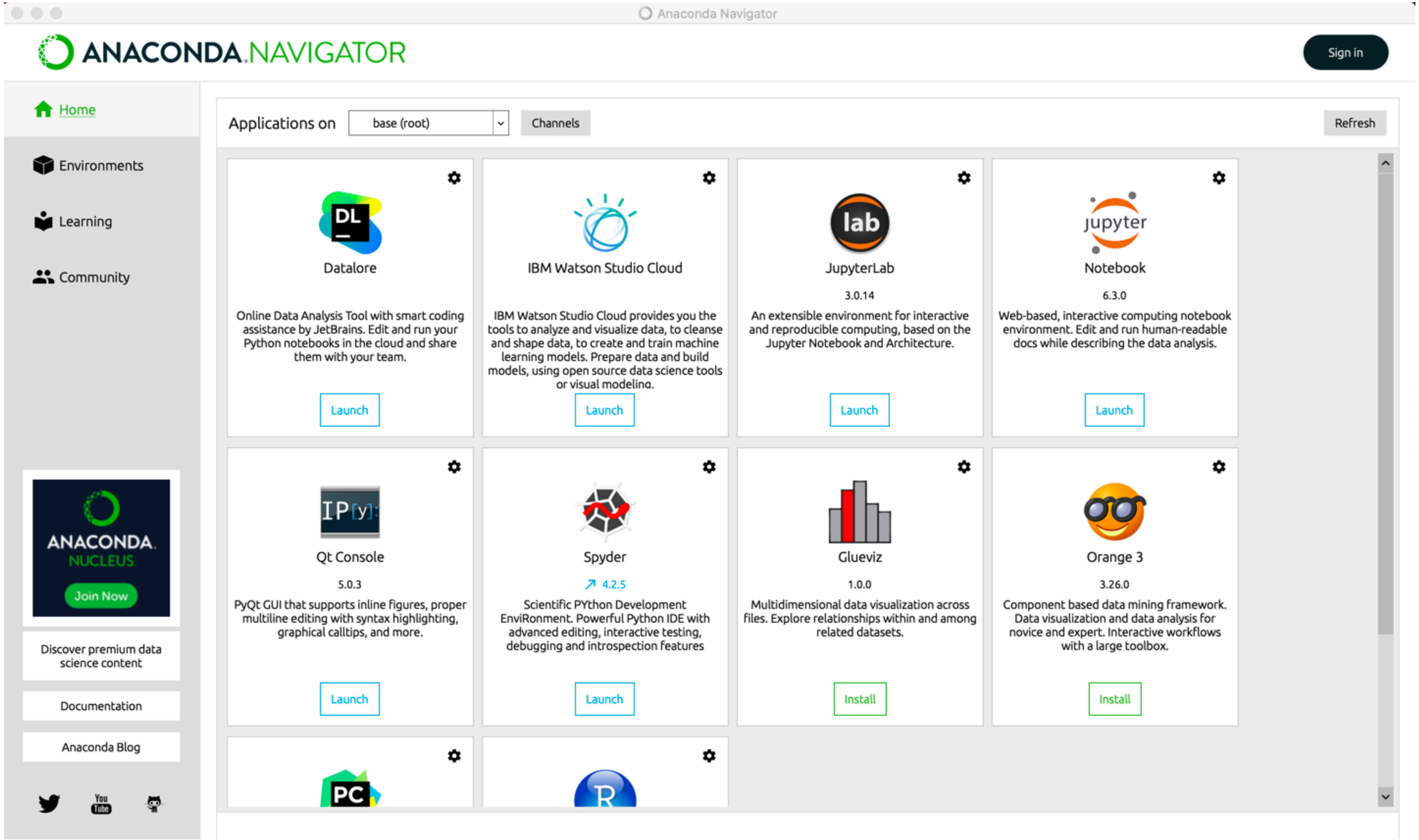




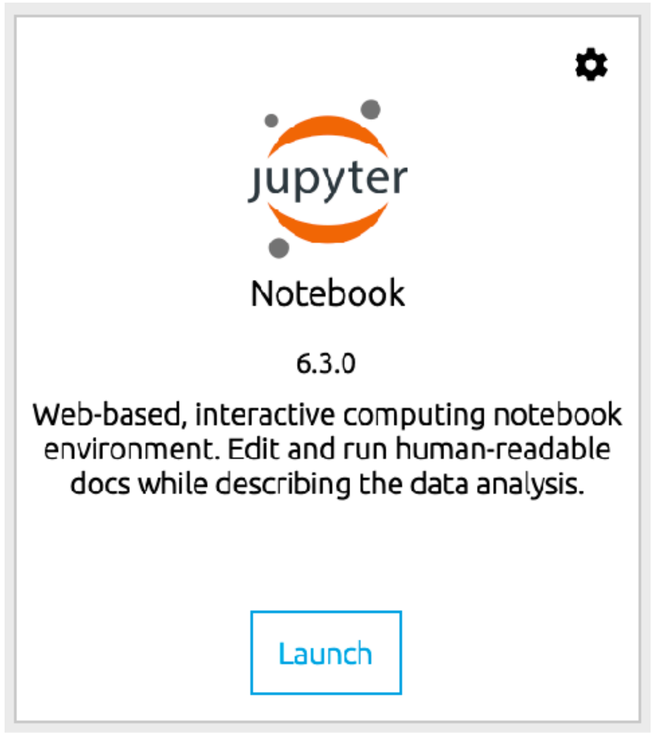


En solamente es descargar la versión continuar ... continuar .... si a todo!

¿Que debería ver cuándo abra Anaconda?



El más usado en este curso: Jupyter



jupyter Fourier_teorico Last Checkpoint: 08/15/2019 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

Transformada de Fourier en python

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import math
import random
```

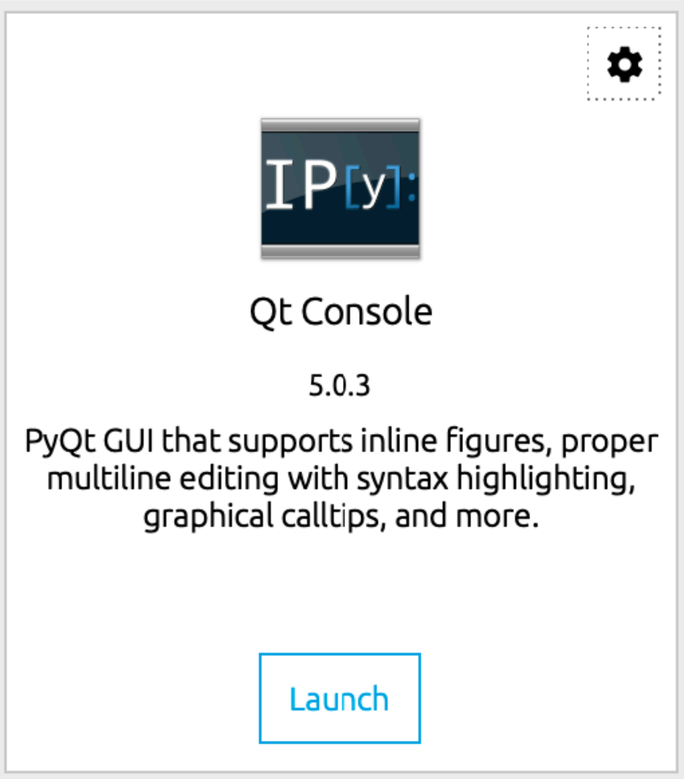
In [2]:
```
def myfft(serie, time_step):
    size_f = len(serie)
    fourier = np.fft.fft(serie)
    power_spec = np.abs(fourier) ** 2.0
    amp = np.abs(fourier)
    freq = np.fft.fftfreq(size_f, time_step)
    period = 1./freq
    return fourier, power_spec, amp, freq, period
```

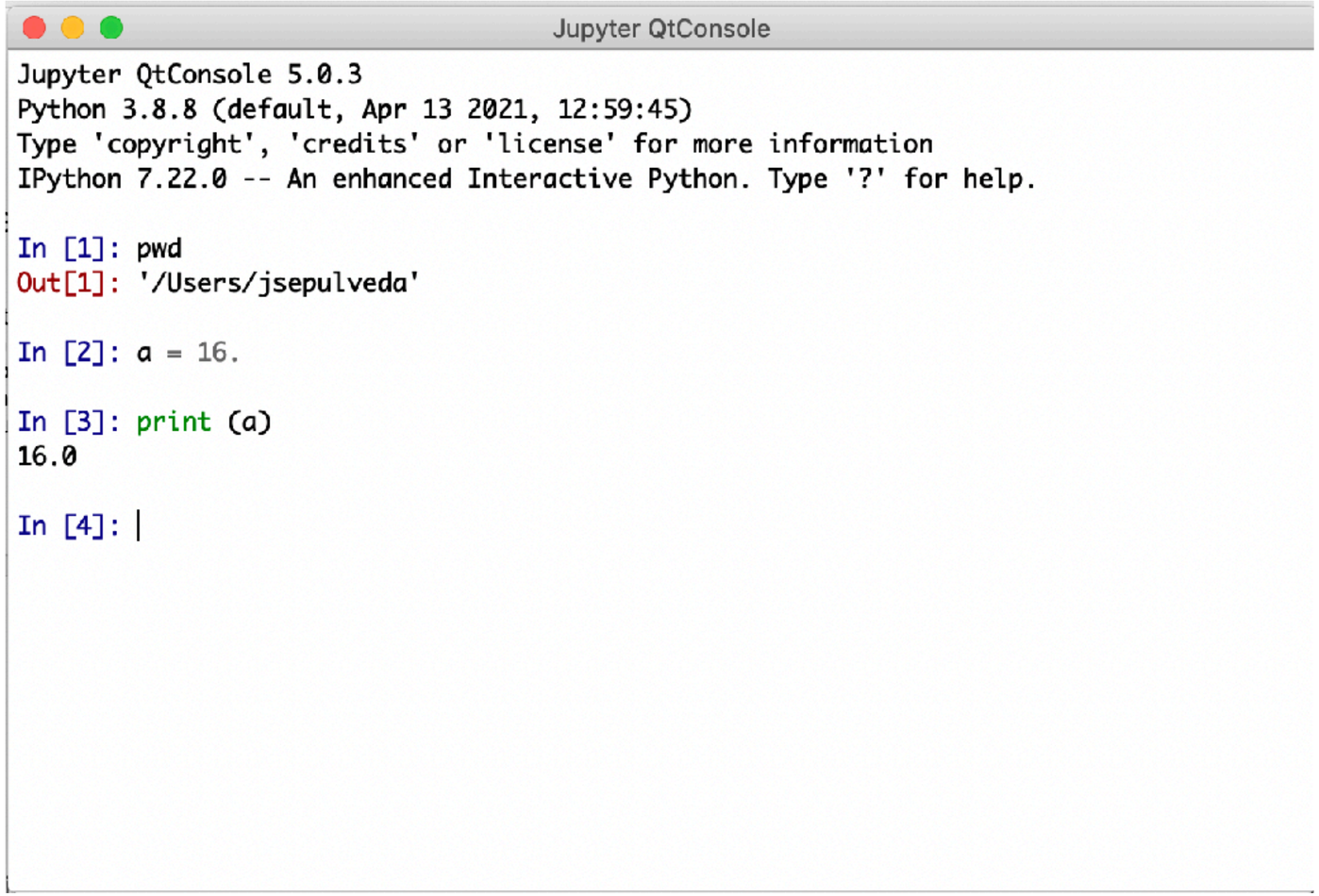
In [3]:
```
time_step = 0.005
t = np.arange(0,10,time_step)
fun = np.sin(2*np.pi*t) + np.sin(2*2*np.pi*t)\
    + np.sin(4*2*np.pi*t)

fourier,power_spec,amp,freq,period = myfft(fun, time_step)
```

```
<ipython-input-2-8915982c9319>:7: RuntimeWarning: divide by zero encountered in true_divide
  period = 1./freq
```

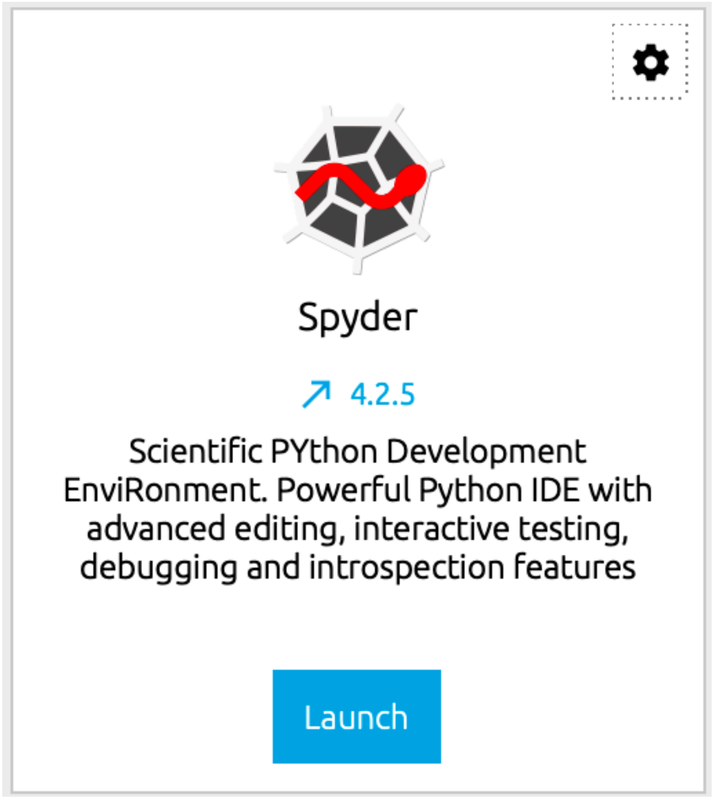
In [4]:
```
plt.close()
plt.cla()
plt.clf()
plt.xlim(-10.,10.)
plt.plot(freq, power_spec)
plt.show()
```

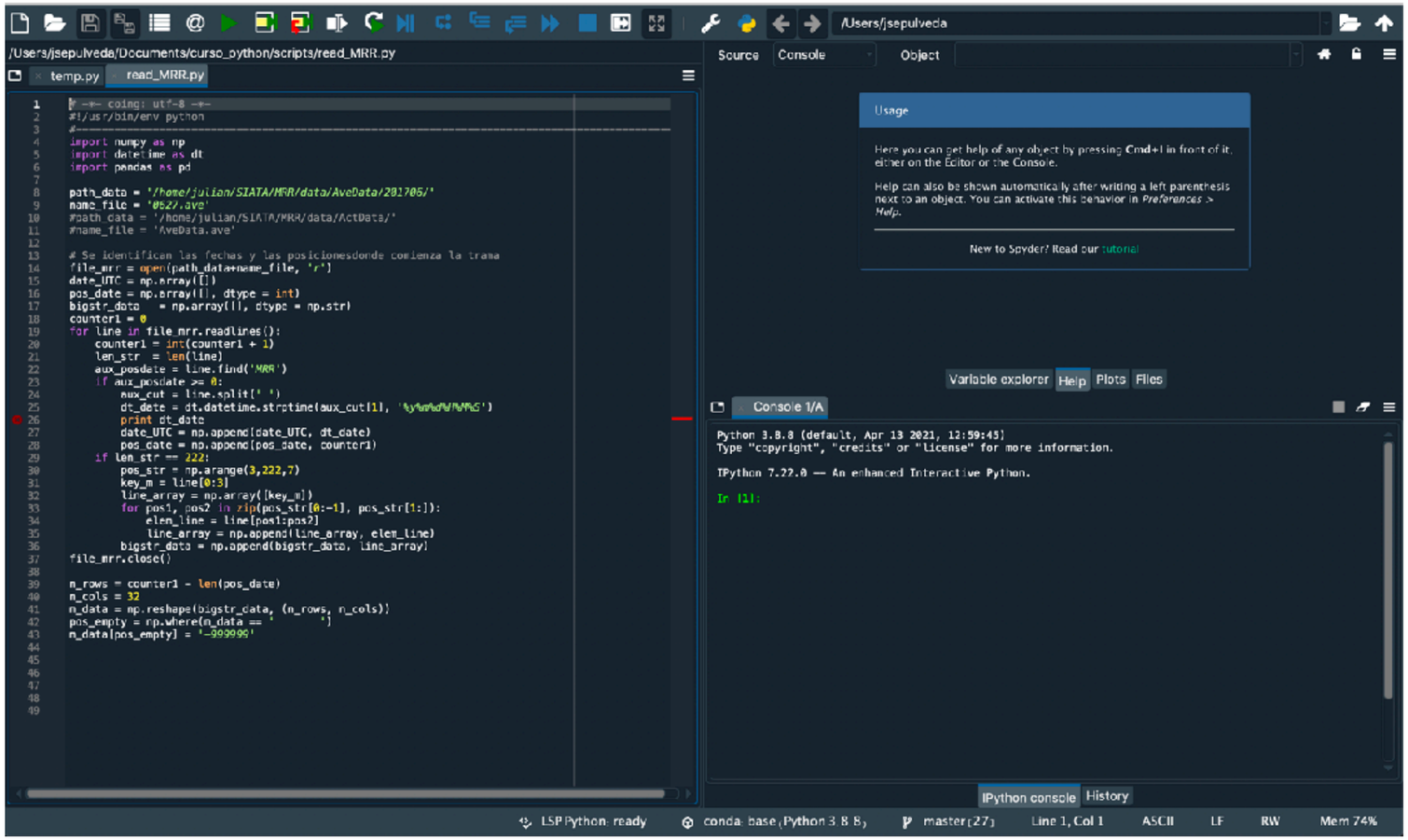
El interprete: Ipython



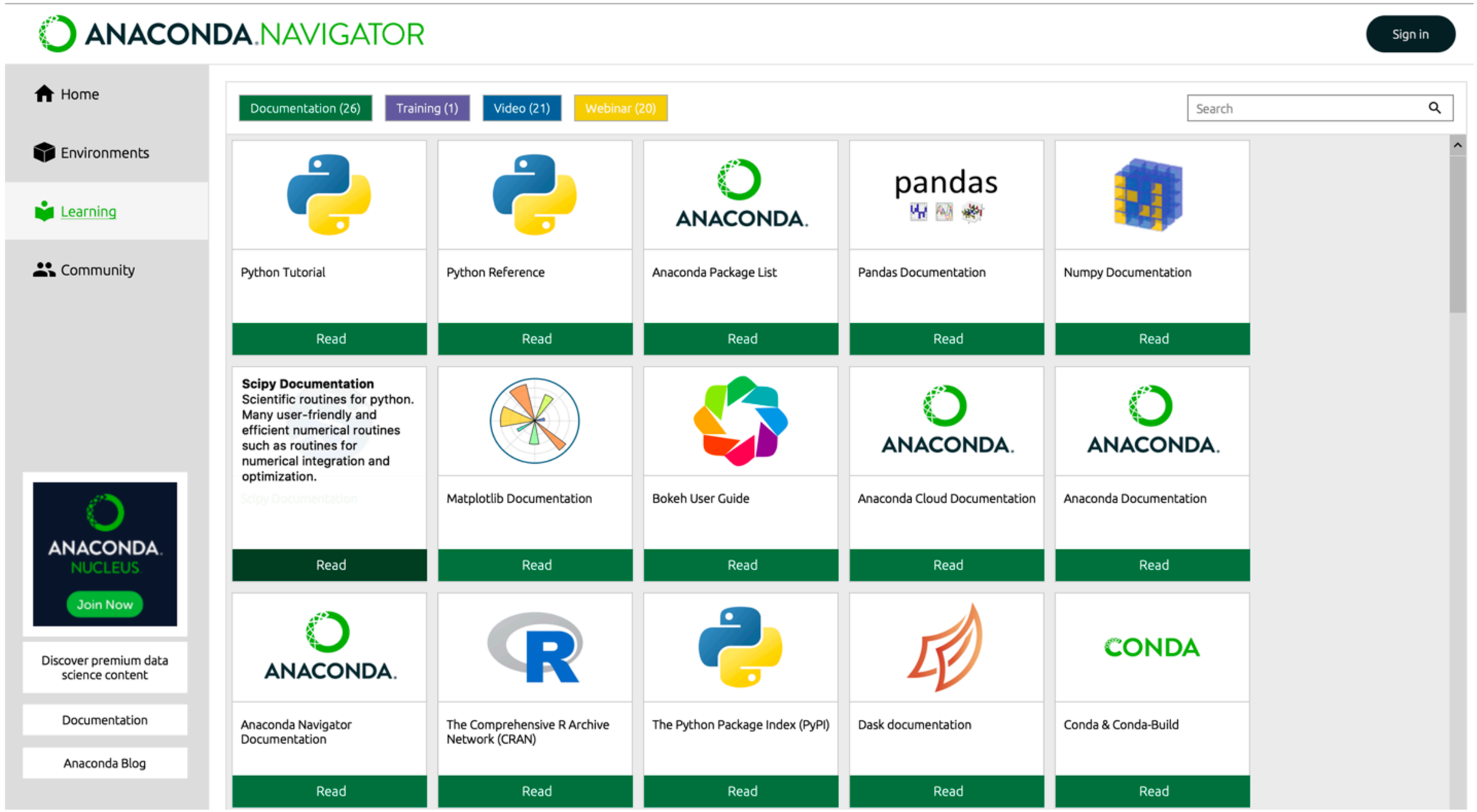


Agradable y completo: Spyder

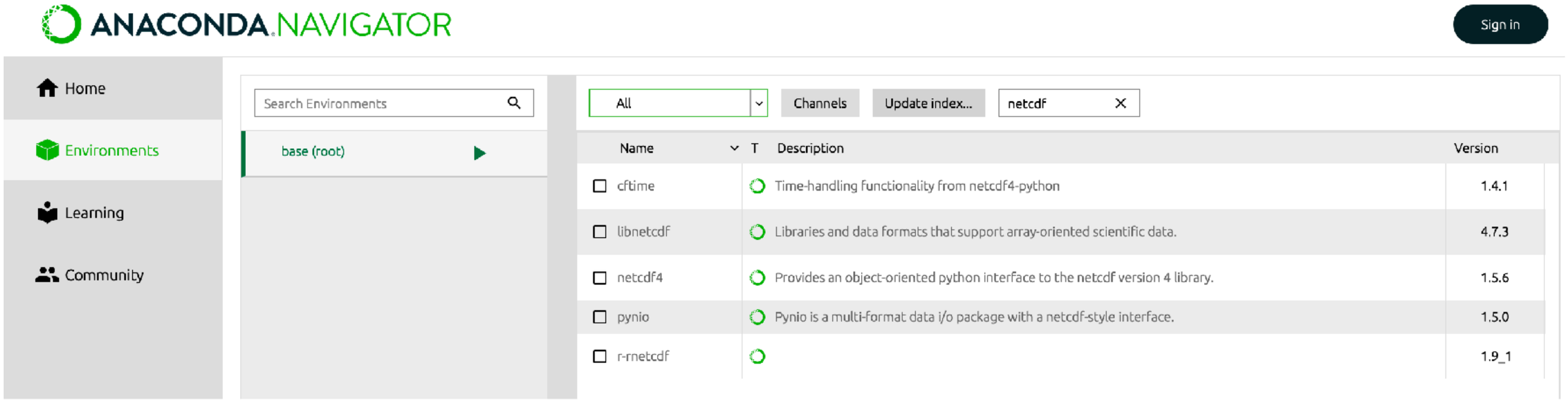




Tutoriales al interior de Anaconda



Instalación de paquetes



Old School

scripts — vim ozono.py — 103×56

```
# -*- coding: utf-8 -*-

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pandas import datetime
from matplotlib import ticker
import datetime as dt

#Funcion para leer excel de variables meteorologicas
def creardf_m(excelfilename,sheetname):
    df = pd.read_excel(path,sheetname = sheetname, usecols=[0,24,26,30,32,34], skipfooter = 0, index_col=0, parse_dates=[0])
    df.index.name = 'Fecha'
    return df
#Leer excel meteorologico
path = "3_2017v1.xlsx"
sheetname = "Consolidado"
df = creardf_m(path, sheetname = sheetname)

#Funcion para leer excel ozono
def creardf_c(excelfilename,sheetname):
    df_c = pd.read_excel(path,sheetname = sheetname, usecols=[0,14], skipfooter = 0, index_col=0, parse_dates=[0])
    df_c.index.name = 'Fecha'
    return df_c
#leer el ozono
df_c = creardf_c(path, sheetname = sheetname)
#pasar a microgramo/metro cubico
df_c['ozono'] = df_c['ozono'].multiply(47.997/24.45)

#crear series horarias y mensuales
O3_3diario = df_c.resample("d").mean()
O3_3mes = df_c.resample("M").mean()

#Quita los valores nan de la serie, no se puede trabajarcon nan para estandarizar una serie
df_csnan=df_c.dropna(0)
#grafico sencillo
ax = O3_3mes['ozono'].plot(legend=True, figsize=(9,5))
ax.set_ylabel(u'Concentración de Ozono [$\mu g/m^3$]',fontsize=12)
ax.set_xlabel('Meses',fontsize=12)
ax.grid()

#Estandarización de la serie diaria
def hourstand(serie): #ingresa dataframe
    serie['stand'] = np.ones(len(serie)) * -999.
    mean_array = np.empty((24))
    std_arr    = np.empty((24))
    for i in np.arange(0,24):
        dumm = serie.loc[serie.index.hour == i]
        mean_h = np.nanmean(dumm.ozono.values)
        std_h  = np.nanstd(dumm.ozono.values)
        mean_array[i] = mean_h
        std_arr[i]    = std_h
```

ozono.py
"ozono.py" [dos] 97L, 3190C

scripts — ipython — 100×54

~/Documents/curso_python/scripts — ipython | ~/Documents/curso_python/notebooks — -zsh

```
Python 3.8.8 (default, Apr 13 2021, 12:59:45)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.22.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: exit
(base) jsepulveda@julians-MacBook-Pro-2 ~ %  cd /Users/jsepulveda/Documents/curso_python/scripts
(base) jsepulveda@julians-MacBook-Pro-2 scripts % ls
ozono.py                read_MRR_improve.py     read_ascii.py
read_MRR.py             read_MRR_improveV2.py   read_nc_wrf.py
(base) jsepulveda@julians-MacBook-Pro-2 scripts % ipython
Python 3.8.8 (default, Apr 13 2021, 12:59:45)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.22.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: run ozono.py
/Users/jsepulveda/Documents/curso_python/scripts/ozono.py:6: FutureWarning: The pandas.datetime class is deprecated and will be removed from pandas in a future version. Import from datetime module instead.
  from pandas import datetime
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~/Documents/curso_python/scripts/ozono.py in <module>
     16 path = "3_2017v1.xlsx"
     17 sheetname = "Consolidado"
---> 18 df = creardf_m(path, sheetname = sheetname)
     19
     20 #Funcion para leer excel ozono

~/Documents/curso_python/scripts/ozono.py in creardf_m(excelfilename, sheetname)
     10 #Funcion para leer excel de variables meteorologicas
     11 def creardf_m(excelfilename,sheetname):
---> 12     df = pd.read_excel(path,sheetname = sheetname, usecols=[0,24,26,30,32,34], skipfooter = 0, index_col=0, parse_dates=[0])
     13     df.index.name = 'Fecha'
     14     return df

~/opt/anaconda3/lib/python3.8/site-packages/pandas/util/_decorators.py in wrapper(*args, **kwargs)
    297                 )
    298                 warnings.warn(msg, FutureWarning, stacklevel=stacklevel)
--> 299             return func(*args, **kwargs)
    300
    301         return wrapper

TypeError: read_excel() got an unexpected keyword argument 'sheetname'

In [2]:
```

La verdadera esencia de la programación