



# Python Avanzado

## Día 2

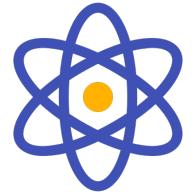
Curso Proteco 2020-2

### Instructores:

- Samuel Garrido Sánchez
- Alicia Carbalindo García
- Armando Rivera



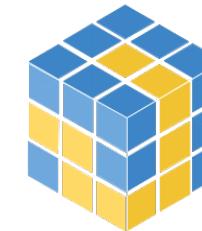
# La ciencia en Python



- Python tiene la característica que el área científica se ha visto muy atraída a su simpleza y su popularidad, por ello paquetes se crean exclusivamente para tratar con cosas numéricas.



- Algunos de ellos son:
  - Numpy
  - SciPy
  - Matplotlib
  - Pandas



**NumPy**

# Instalación de módulos

- La instalación de estos módulos se hará a través de PIP, trate de hacerlo en el siguiente orden:

```
pip install numpy  
pip install pandas  
pip install matplotlib  
pip install scipy
```

Si utiliza alguna de las herramientas como Google Colab o Microsoft Azure Notebooks, estas ya vienen por defecto instaladas.

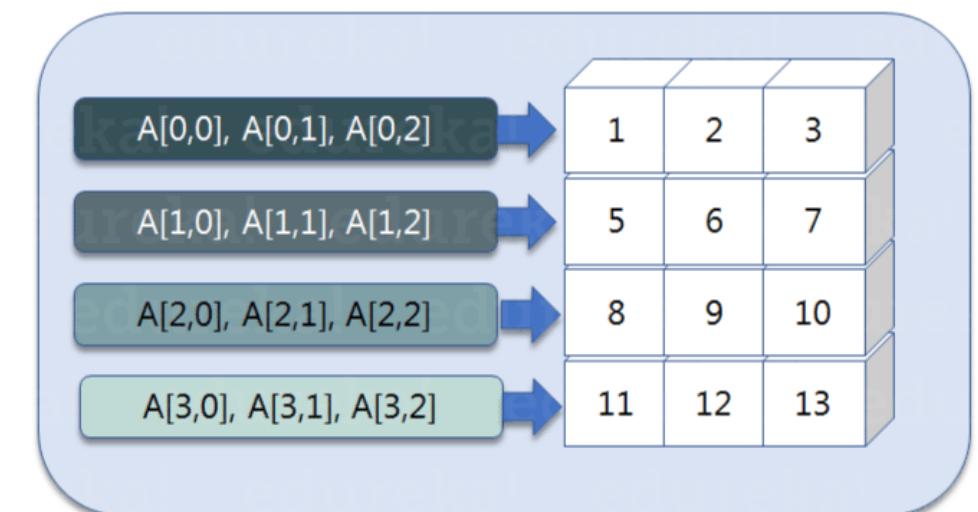
Para el caso de macOS, en lugar de pip poner pip3 para instalar (ya que su SO cuenta Python 2 y 3)



# Numpy: Unas listas algo raras

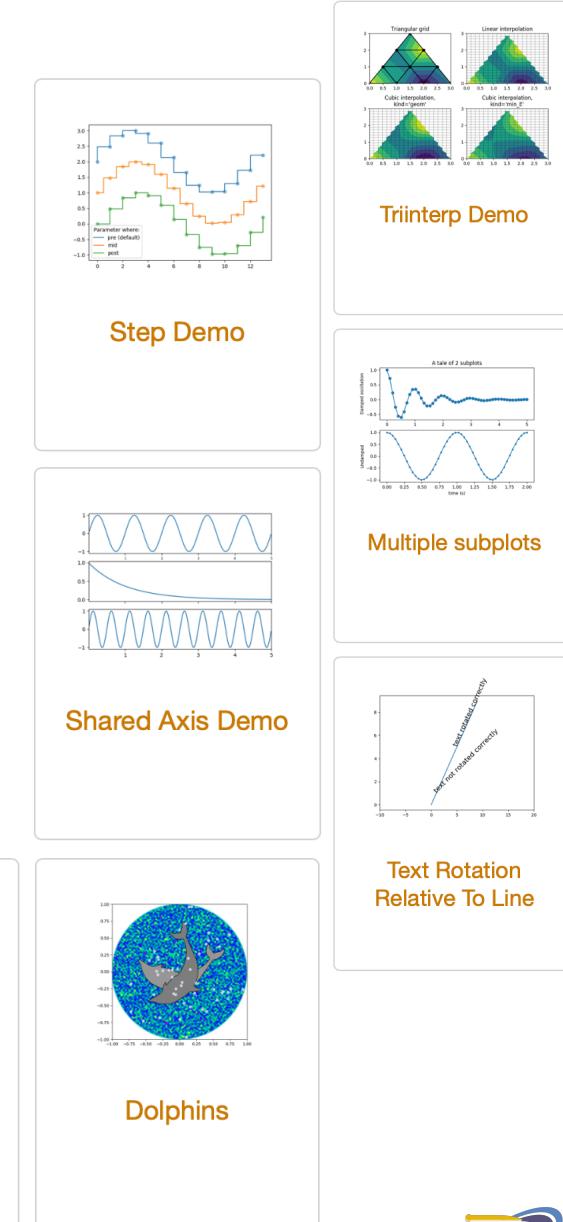
- Como hemos visto ayer, numpy es una herramienta que nos permite poner arreglos matriciales y algunos elementos numéricos.
- Usamos la matriz NumPy en lugar de una lista debido a las siguientes razones:
  - Menos memoria
  - Rápido
  - Conveniente

```
## Arreglo lineal  
a=np.array([1,2,3])  
  
##Arreglo  
multidimensional  
b=np.array([(1,2,3),(4,5,6)])
```

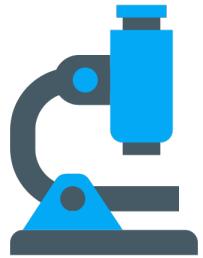


# MatPlotLib: Visualización de datos

- Los humanos comprendemos mejor de manera visual, por ello MatPlotLib nos permite ver datos con gráficas y otros elementos visuales.
- Si alguien es del área de ciencias, esto le puede sentar de maravilla.
- En Jupyter tenemos la ventaja de todas las gráficas se imprimen justo debajo del código, dejando un cuaderno muy impecable.



# SciPy: Sabes, yo también soy “100tífiko”



- Ecosistema de software de código abierto basado en Python para matemáticas, ciencias e ingeniería. Incluye Pandas, Numpy y Sympy.
- Para qué vamos a poder utilizar SciPy:
  - Integración
  - Derivación
  - Cálculo Vectorial
  - Análisis Numérico
  - Ecuaciones Diferenciales
  - Optimización
  - Sistemas Dinámicos
  - O para imprimir la cara de un mapache 🐾

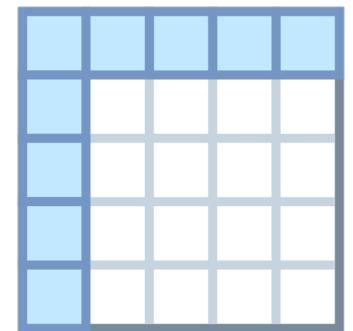
```
In [27]: from scipy import misc  
import matplotlib.pyplot as plt  
  
face = misc.face()  
plt.imshow(face)  
plt.show()
```



# Pandas : Un vistazo a la ciencia de datos



- Pandas es una herramienta de manipulación de datos de alto nivel desarrollada por Wes McKinney.
- Es construido con el paquete Numpy y su estructura de datos clave es llamada el DataFrame. El DataFrame te permite almacenar y manipular datos tabulados en filas de observaciones y columnas de variables.
- Con este podremos tratar datos dentro de tablas de datos o mejor llamadas datasets.



# Pandas para manipular, datos.

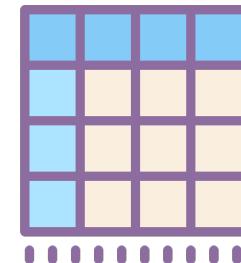


- Podemos convertir un diccionario en un Panda DataFrame

```
In [4]: import pandas as pd
diccionario = {
    "País": ["Brasil", "Rusia", "India", "China", "Sudáfrica"],
    "Capital": ["Brasilia", "Moscú", "Nueva Dehli", "Beijing", "Pretoria"],
    "Área": [8.516, 17.10, 3.286, 9.597, 1.221],
    "Población": [200.4, 143.5, 1252, 1357, 52.98] }

paisesDF = pd.DataFrame(diccionario)
print(paisesDF)
```

	País	Capital	Área	Población
0	Brasil	Brasilia	8.516	200.40
1	Rusia	Moscú	17.100	143.50
2	India	Nueva Dehli	3.286	1252.00
3	China	Beijing	9.597	1357.00
4	Sudáfrica	Pretoria	1.221	52.98



# Abrir archivos con Pandas

- Dependiendo el tipo de archivo que queramos abrir(CSV, EXCEL,JSON), pandas nos ofrecerá funciones para tratar con estos de manera automática:

```
import pandas as pd
```

```
## Estas variables ya son de tipo DataSet, que podremos manipular con Pandas
```

```
miJson = pd.read_json('prueba.json')
```

```
miExcel = pd.read_excel('prueba.xlsx')
```

```
miCSV = pd.read_csv('prueba.csv')
```

```
print(miJson)
```

```
print(miExcel)
```

```
print(miCSV)
```

