



Python Avanzado

Día 1

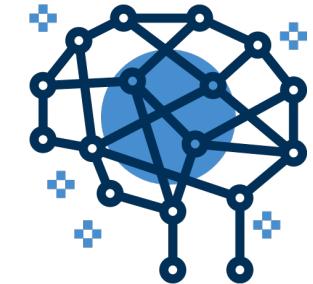
Curso Proteco 2020-2

Instructores:

- Samuel Arturo Garrido Sánchez
- Alicia Carbalindo García
- Armando Rivera



Python para niños grandes



- Python no solamente se limita a crear hola mundos sino a muchísimas otras cosas :
 - Desarrollo web con Django
 - Usado por Instagram, MIT, Spotify, Youtube, Dropbox, Mozilla, etc.
 - Desarrollo web con Flask
 - Inteligencia Artificial:
 - Pytorch, TensorFlow, SciKitLearn, Keras.
 - Hasta para desarrollar apps móviles... peeeeero no se casen con Python todavía. Cada lenguaje tiene sus características específicas y es por esto que JavaScript, PHP, Swift, Kotlin o C# **NO** deben ser descartados.



Herramientas de trabajo

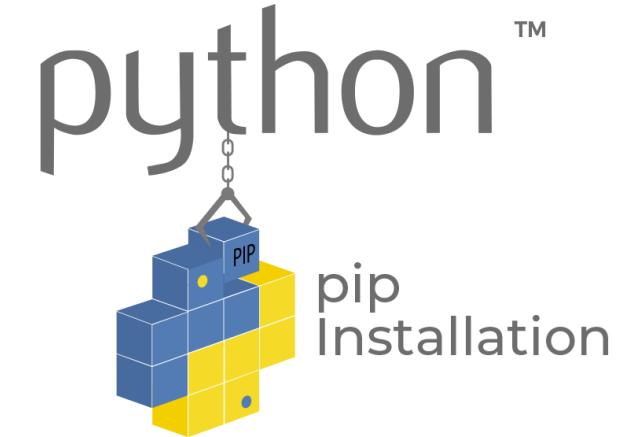


- Gestor de paquetes PIP ([¿Qué será un gestor de paquetes?](#))
- Anaconda Navigator.
 - Interfaz gráfica de usuario que permite iniciar aplicaciones y administrar fácilmente paquetes, entornos y canales.
 - Dentro de este viene:
 - PyCharm IDE
 - Jupyter Notebook
 - Spider IDE
 - Orage para Data mining
 - JupyterLab
 - Rstudio
- O bien para más elegante, ya que solo usaremos Jupyter notebook, colocar en su terminal:
pip install notebook



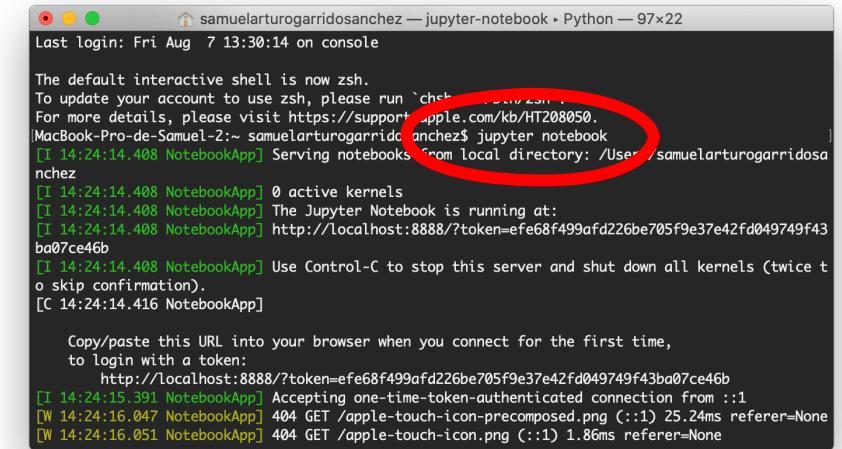
Herramientas de trabajo

- Paquetes de PIP importantes
 - Matplotlib
 - Numpy
 - Folium
 - Googlemaps
 - Gmaps
 - Pandas
 - TensorFlow - pip install tensorflow



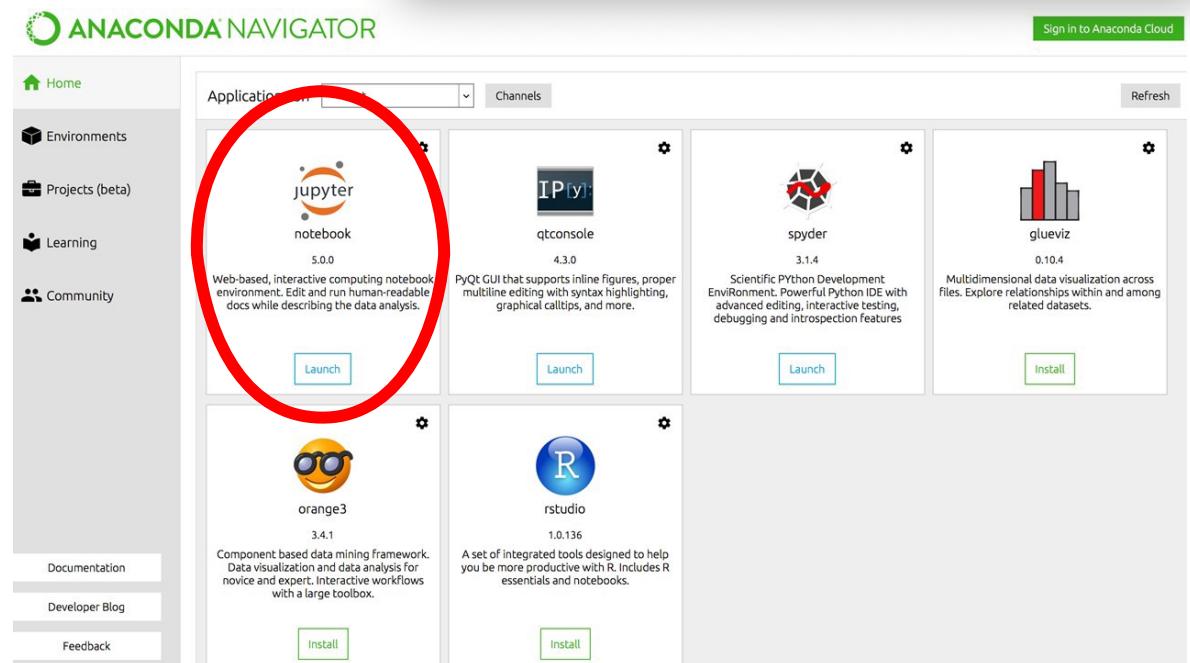
¿Cómo se trabaja en libretas?

- Si ha instalado Jupyter con PIP
 - Escribir en su terminal:
jupyter notebook
- Si ha instalado Jupyter con Anaconda
 - Abrir Anaconda Navigator y hacer click en Juyter para abrir (LAUNCH)

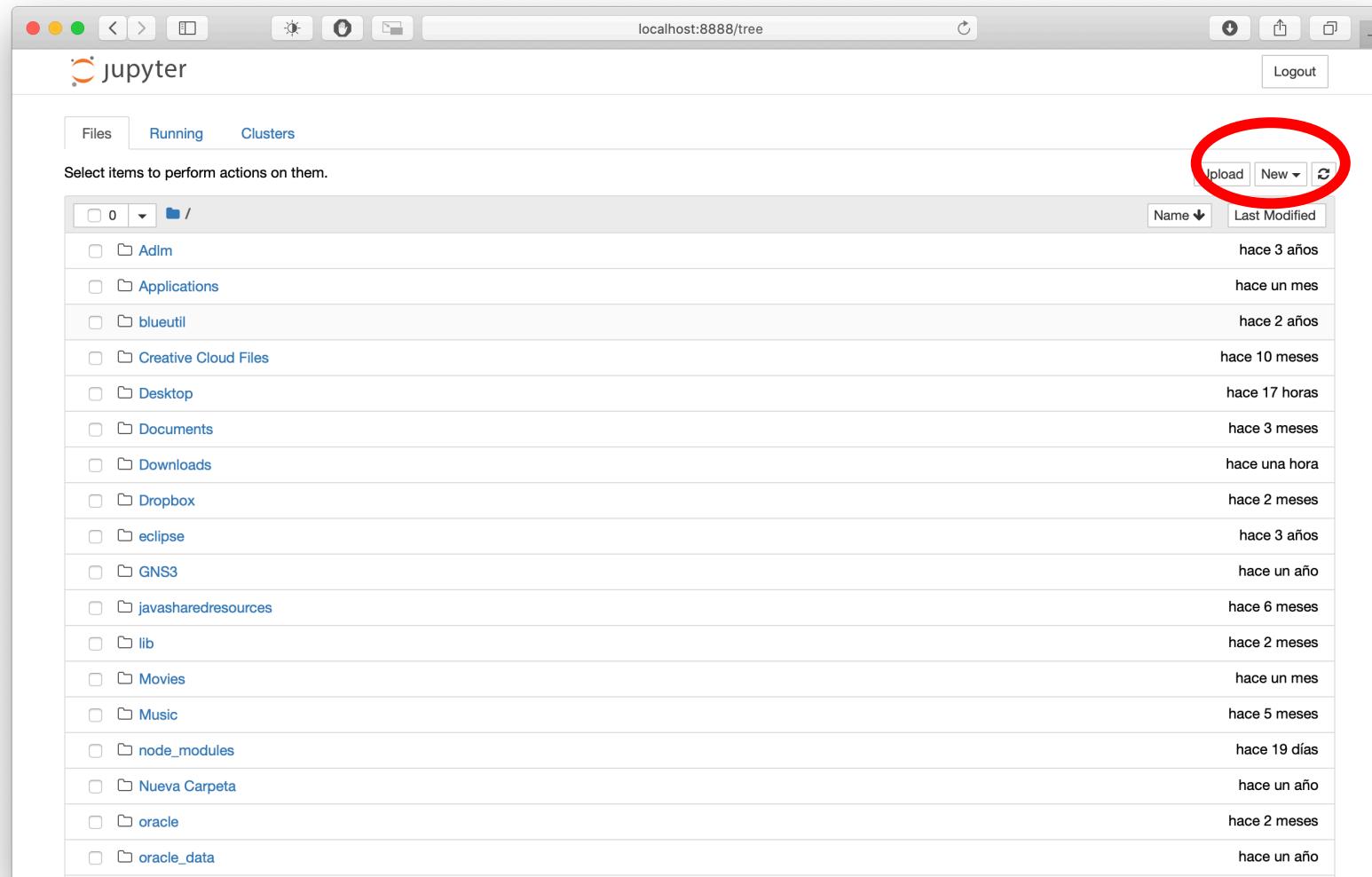


The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit <https://support.apple.com/kb/HT208050>.
[MacBook-Pro-de-Samuel-2:~ samuelarturogarridosanchez\$ jupyter notebook
[14:24:14.408 NotebookApp] Serving notebooks from local directory: /Users/samuelarturogarridosanchez
[14:24:14.408 NotebookApp] 0 active kernels
[14:24:14.408 NotebookApp] The Jupyter Notebook is running at:
[14:24:14.408 NotebookApp] http://localhost:8888/?token=efe68f499af226be705f9e37e42fd049749f43ba07ce46b
[14:24:14.408 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 14:24:14.416 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=efe68f499af226be705f9e37e42fd049749f43ba07ce46b
[14:24:15.391 NotebookApp] Accepting one-time-token-authenticated connection from ::1
[W 14:24:16.047 NotebookApp] 404 GET /apple-touch-icon-precomposed.png (::1) 25.24ms referer=None
[W 14:24:16.051 NotebookApp] 404 GET /apple-touch-icon.png (::1) 1.86ms referer=None

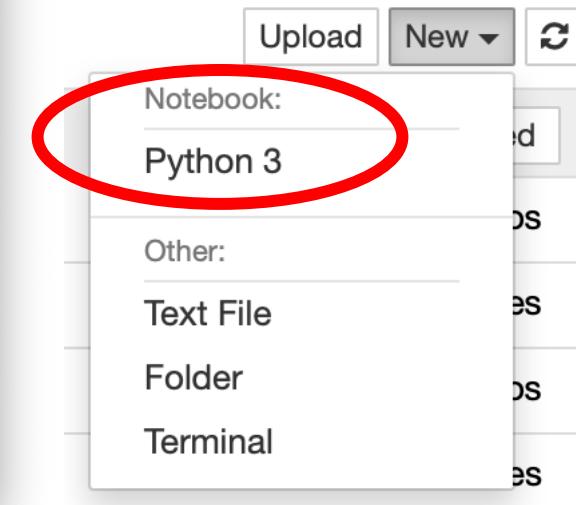


Entorno de trabajo



Ubíquese en la carpeta dentro de su equipo que desee

New -> Python3



Entorno de trabajo

The screenshot shows a Jupyter Notebook window with the following details:

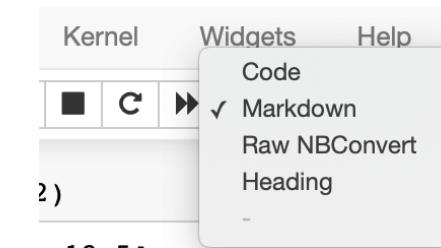
- Title Bar:** jupyter Regionalización - GGMAPS Last Checkpoint: 13/04/2020 (unsaved changes)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, Python 3
- Cell 1 (In [1]):** Code cell containing Python imports for pandas, matplotlib, json, numpy, folium, gmaps, and ipywidgets.
- Cell 2 (In [2]):** Code cell containing code to read a CSV file, filter for 'ALMACENES' clinics, and save the results to a Firebase JSON file.
- Output 2:** Text output showing the counts of total clinics (204) and clinics with CONBAJAS (223).
- Cell 3 (In [3]):** Code cell containing the variable AlmacenesTotal.
- Output 3:** Text output showing the schema of the AlmacenesTotal DataFrame, which includes columns: ID, CLUES, NOMBREENTIDAD, CLAVE-ENTIDAD, NOMBREMUNICIPIO, CLAVE-MUNICIPIO, NOMBRE-LOCALIDAD, and CLAVE-LOCALIDAD.

Note que Jupyter trabaja por CELDAS

Dentro podemos encontrar las funciones principales:

Guardar, agregar, cortar, pegar, arriba, abajo, eje, parar, volver a ejecutar

Además encontramos el seleccionar el tipo de celda, ya sea MarkDown, Code, Raw NBConverter, Heading



MarkDown y LaTEX

- Una libreta de Jupyter sirve para no poner código a la brava sino tener una mejor documentación, esto es muy útil para el área de investigación por ejemplo.
- Markdown se usa a menudo para formatear archivos README, para escribir mensajes en foros de discusión en línea y para crear texto enriquecido con un editor de texto sin formato.



The screenshot shows a Jupyter Notebook interface with the file name "README.md" at the top. The notebook contains the following content:

```
<p>
    
</p>

# El lenguaje de programación
Python

<p>
    
</p>

Holas a todos y bienvenidos sean al curso de programación con Python 2020-2. De parte de todos los colaboradores les agradecemos su preferencia y esperamos que el curso le sea útil en su vida profesional. ❤

#### 🚀 Equipo:

- Alicia Carballido García **(Titular Básico e Intermedio)**
- Samuel Arturo Garrido Sánchez **(Titular Avanzado)**
- Armando Rivera **(Adjunto Avanzado)**
- Ana Paola Lagunas **(Auxiliar)**
- Alejandro Barreiro Valdez **(Auxiliar)**
- Mario Álvares Salmerón **(Auxiliar)**

### 📅 Fechas y horarios

| **Nivel** | Fecha y hora |
| :-----: | :-----: |
| **Básico** | 10 al 14 de Agosto de 2:30 pm a 4:30 pm |
| **Intermedio** | 17 al 21 de Agosto de 2:30 pm a 4:30 pm |
| **Avanzado** | 24 al 28 de Agosto de 2:30 pm a 4:30 pm |
```

El lenguaje de programación Python

Holas a todos y bienvenidos sean al curso de programación con Python 2020-2. De parte de todos los colaboradores les agradecemos su preferencia y esperamos que el curso le sea útil en su vida profesional. ❤

Equipo:

- Alicia Carballido García (**Titular Básico e Intermedio**)
- Samuel Arturo Garrido Sánchez (**Titular Avanzado**)
- Armando Rivera (**Adjunto Avanzado**)
- Ana Paola Lagunas (**Auxiliar**)
- Alejandro Barreiro Valdez (**Auxiliar**)
- Mario Álvares Salmerón (**Auxiliar**)

Fechas y horarios

Nivel	Fecha y hora
Básico	10 al 14 de Agosto de 2:30 pm a 4:30 pm
Intermedio	17 al 21 de Agosto de 2:30 pm a 4:30 pm
Avanzado	24 al 28 de Agosto de 2:30 pm a 4:30 pm

Temario

Básico - Semana 1

- Historia
- Síncrono

MarkDown y LaTEX

- Dentro de la libreta de Jupyter podemos escribir Markdown siempre y cuando seleccionemos la opción de celda adecuada.
- Además podemos hacer uso de LaTEX para escribir ecuaciones para tener mejor documentado nuestro trabajo.
- *Si saben todos lo que es LaTEX?*

Método de la regla falsa

El método de regla falsa o interpolación se basa en tener 2 limitantes o rango que sepamos que contienen a la raíz de la función. Su ecuación es:

\$\$r \in [a, b]\$\$

\$\$x_i = \frac{f(a_i)b_i - f(b_i)a_i}{f(a_i) - f(b_i)}\$\$

Método de la regla falsa

El método de regla falsa o interpolación se basa en tener 2 limitantes o rango que sepamos que contienen a la raíz de la función. Su ecuación es:

$$r \in [a, b]$$

$$x_i = \frac{f(a_i)b_i - f(b_i)a_i}{f(a_i) - f(b_i)}$$

Tipos de celda: Python + MarkDown + LaTEX

- Al cambiar entre tipo de celdas podemos tener algoritmos como el mostrado.
- Existen comandos rápidos o teclas rápidas para realizar las acciones de guardado, selección de celda y otras propiedades

Método de la regla falsa

El método de regla falsa o interpolación se basa en tener 2 limitantes o rango que sepamos que contienen a la raíz de la función. Su ecuación es:

$$r \in [a, b]$$

$$x_i = \frac{f(a_i)b_i - f(b_i)a_i}{f(a_i) - f(b_i)}$$

```
In [7]: import sympy as sy
from sympy.functions import sin,cos
import math
import matplotlib.pyplot as plt
x=sy.symbols('x')
Px = lambda x: x**2 - 2*x - 10
## Fija los rangos de a y b donde buscarán
a = 2
b = 6
tolerancia = 0.1
l = (Px(a)*b - Px(b)*a)/(Px(a)-Px(b))
print(l)
```

3.666666666666665

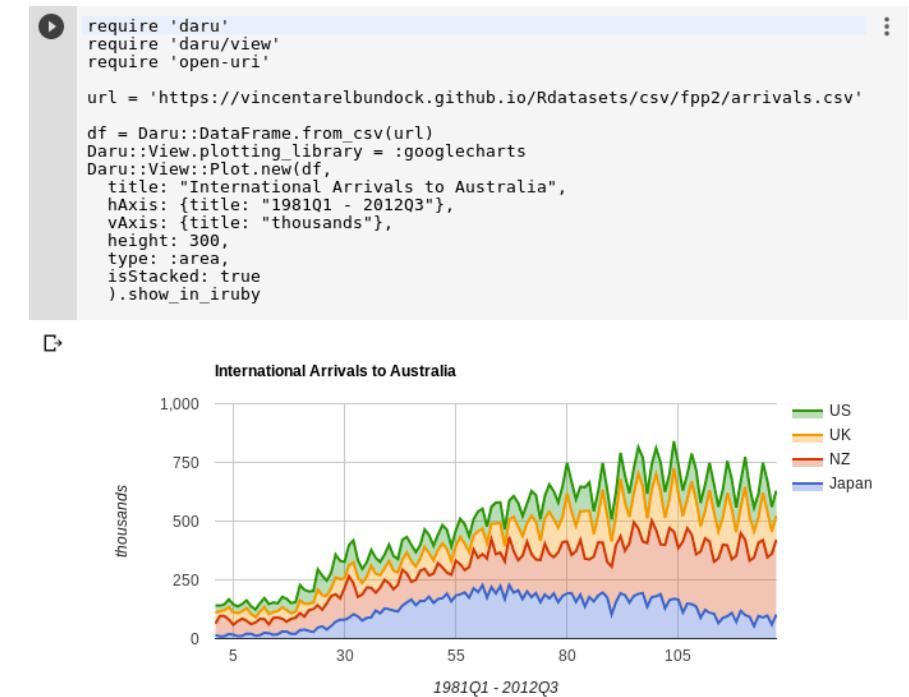
Algunas opciones adicionales

- En caso de no contar con las anteriores existen 2 opciones recomendadas para poder trabajar Python a través de un navegador web:

- Google Colaboratory alias Google Colab

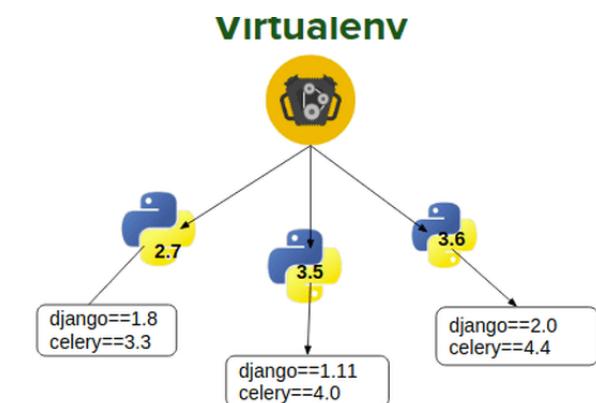


- Microsoft Azure Notebooks



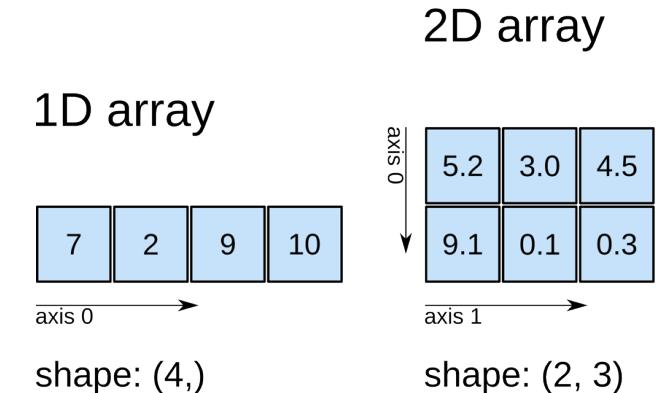
Ambientes virtuales

- Las aplicaciones a veces necesitan una versión específica de una biblioteca, debido a que dicha aplicación requiere que un bug particular haya sido solucionado o bien la aplicación ha sido escrita usando una versión obsoleta de la interfaz.
- Así, los ambientes virtuales es un mecanismo que me permite gestionar programas y paquetes **python** sin tener permisos de administración, es decir, cualquier usuario sin privilegios puede tener uno o más "espacios aislados"



Primero lo primero: Numpy

- Numpy es una biblioteca que agrega mayor soporte para vectores y matrices.
- El objeto principal de NumPy es la matriz multidimensional homogénea. Es una tabla de elementos (generalmente números), todos del mismo tipo, indexados por una tupla de enteros no negativos. En NumPy las dimensiones se llaman ejes.



Algunos ejemplos con Numpy

- Trato con vectores
 - Es importante primero importar numpy.
 - `import numpy as np`
 - Tenemos una lista de elementos, los convertimos a vector numpy
 - Sumamos +1 a cada elemento
 - Multiplicamos x 5 a cada elemento
 - Suma de sus elementos
 - Media
 - Suma de vectores

```
In [1]: import numpy as np
lista=[25,12,15,66,12.5]
vector=np.array(lista)
print(vector)

print(vector)

print(vector+1)

print(vector*5)

print(np.sum(vector))

print(np.mean(vector))

print(vector+vector)

vector2=np.array([11,55,1.2,7.4,-8])

print(vector+vector2)
```

[25. 12. 15. 66. 12.5]
[25. 12. 15. 66. 12.5]
[26. 13. 16. 67. 13.5]
[125. 60. 75. 330. 62.5]
130.5
26.1
[50. 24. 30. 132. 25.]
[36. 67. 16.2 73.4 4.5]