

# Python para el análisis de datos- Lectura 5

Ing. Pedro Rotta

Universidad de Piura - Vida Universitaria

Enero-2022

# Librerías

En esta clase hablaremos de 2 librerías importantes en python, con las que haremos programas mucho más sofisticados y al mismo tiempo más útiles.

# Librerías

En esta clase hablaremos de 2 librerías importantes en python, con las que haremos programas mucho más sofisticados y al mismo tiempo más útiles.

Una **librería** es un archivo de python o un paquete de archivos que contiene varias funciones predefinidas o un conjunto de objetos que se pueden usar directamente por el usuario.

# Librerías

En esta clase hablaremos de 2 librerías importantes en python, con las que haremos programas mucho más sofisticados y al mismo tiempo más útiles.

Una **librería** es un archivo de python o un paquete de archivos que contiene varias funciones predefinidas o un conjunto de objetos que se pueden usar directamente por el usuario.

Existen varias librerías preinstaladas en python, como **math**, **time**, **os**, pero algunas no están instaladas y se tienen que instalar. Para ello existe **Pip**.

# Pip

Pip es el instalador de librerías de python. Se instala directamente en el código fuente del sistema operativo cuando se instala python en la computadora. Ejecutemos los siguientes comandos en el command system para revisar que tiene instalado pip en su sistema operativo:

```
python --version  
python -m pip --version
```

Para instalar una librería que no tenga, use:

```
pip install librería
```

# Detalles

Los siguientes detalles son importantes para trabajar con paquetes de python:

- ▶ Sintaxis:

```
import libreria.
```

# Detalles

Los siguientes detalles son importantes para trabajar con paquetes de python:

- ▶ Sintaxis:

```
import libreria.
```

- ▶ Para saber sus métodos:

```
dir(librería).
```

# Detalles

Los siguientes detalles son importantes para trabajar con paquetes de python:

- ▶ Sintaxis:

```
import libreria.
```

- ▶ Para saber sus métodos:

```
dir(librería).
```

- ▶ Para importar solo un método específico:

```
from library import method. Ver problema 1
```



# Detalles

Los siguientes detalles son importantes para trabajar con paquetes de python:

- ▶ Sintaxis:

```
import libreria.
```

- ▶ Para saber sus métodos:

```
dir(librería).
```

- ▶ Para importar solo un método específico:

```
from library import method. Ver problema 1
```

- ▶ Deben importarse antes de usar sus funciones.

# Detalles

Los siguientes detalles son importantes para trabajar con paquetes de python:

- ▶ Sintaxis:

```
import libreria.
```

- ▶ Para saber sus métodos:

```
dir(librería).
```

- ▶ Para importar solo un método específico:

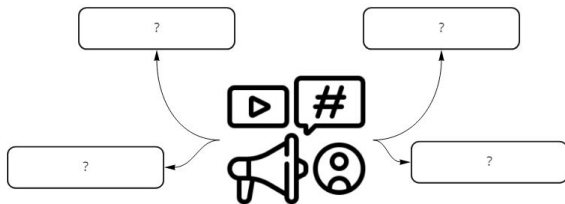
```
from library import method. Ver problema 1
```

- ▶ Deben importarse antes de usar sus funciones.

- ▶ Hay librerías enfocadas para distintos temas (Análisis de datos, IA, Procesamiento de Imágenes, Raspberry, Sistemas macroeconómicos y microeconómicos, Matemáticas, etc).

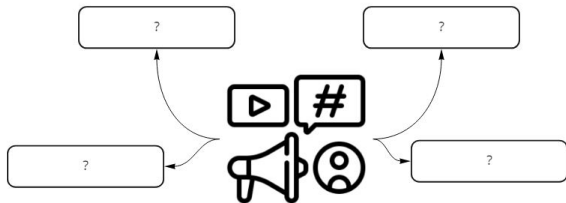
# Numpy

Numpy es la librería computacional más importante de python.  
Brinda un tipo de objeto que se llama ndarray. Este objeto permite coleccionar tipos de datos N-dimensionales.



# Numpy

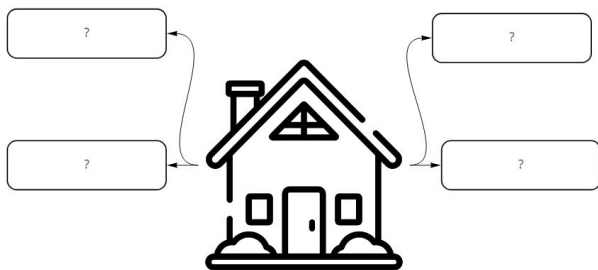
Numpy es la librería computacional más importante de python.  
Brinda un tipo de objeto que se llama ndarray. Este objeto permite coleccionar tipos de datos N-dimensionales.



El nombre, la edad, género, etc, se pueden representar como un tipo de dato N-Dimensional.

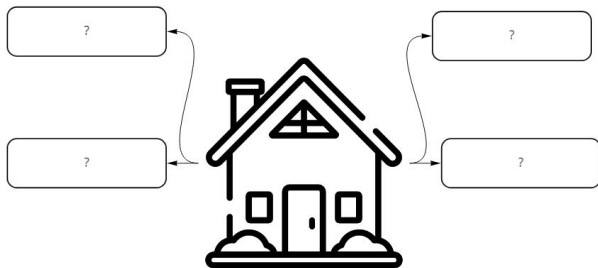
# Numpy

Otro ejemplo sería si deseo comprar una casa, por ejemplo, necesitaría saber algunos datos.



# Numpy

Otro ejemplo sería si deseo comprar una casa, por ejemplo, necesitaría saber algunos datos.



Estos datos también forman parte de un tipo de objeto N-dimensional.

# Numpy

Estos datos que pueden ser guardados en python, forman lo que en matemáticas se conoce como una matriz de datos o un vector de datos.

## Sintaxis :

```
import numpy as np
vector = np.array([11,12,13])
matriz = np.array([[11,12,13],[10,9,8]])
```

Ver problema 2

# Métodos de numpy

El uso de la librería numpy se ve referido a sus métodos. Numpy tiene métodos importantes que permiten realizar cálculos vectoriales y matriciales para optimizar el código.



# Métodos de numpy

El uso de la librería numpy se ve referido a sus métodos. Numpy tiene métodos importantes que permiten realizar cálculos vectoriales y matriciales para optimizar el código.

Métodos numpy para vectores:

- ▶ Para generar vectores en un rango definido:

```
np.arange(5)
```

- ▶ Para generar vectores o matrices de unos:

```
np.ones((2,2))
```

- ▶ Para generar vectores en un rango lineal:

```
np.linspace((0,1,5))
```

- ▶ Para generar vectores o matrices de ceros:

```
np.zeros((2,2))
```

Ver problema 3.

# Métodos de numpy

Para manejar vectores:

- ▶ Para juntar vectores de manera horizontal:  
`np.hstack([x,y])`
- ▶ Para juntar vectores de manera vertical:  
`np.vstack([x,y])`
- ▶ Para juntar vectores de manera vertical-en y:  
`np.c_[x,y]`

Ver problema 4 colab.

# Métodos de numpy

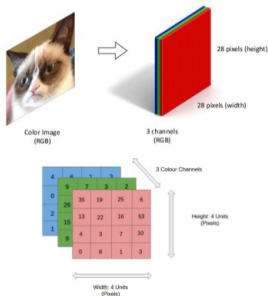
## Métodos de Álgebra lineal:

- ▶ Cambiar el tamaño de una matriz:  
`array.reshape(nfil,ncol)`
- ▶ Obtener una matriz transpuesta:  
`array.transpose()`
- ▶ Matriz identidad:  
`np.eye(n)`
- ▶ Producto Punto:  
`a.dot(b)`
- ▶ El valor máx de un array se encuentra por el método  
`arrayx.max()`
- ▶ El valor min de un array se encuentra por el método  
`arrayx.min()`
- ▶ La suma de los valores de un array se encuentran por el método  
`arrayx.sum()`

# Tensoros

Una imagen, es un objeto que se puede dividir en 3 canales, R,G y B. Así mismo un video es la sucesión de varias imágenes.

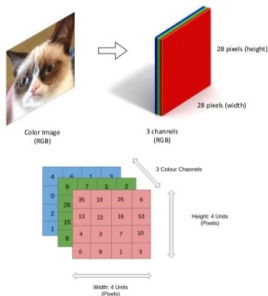
color image is 3rd-order tensor



# Tensoros

Una imagen, es un objeto que se puede dividir en 3 canales, R,G y B. Así mismo un video es la sucesión de varias imágenes.

color image is 3rd-order tensor



A este tipo de datos, que no se pueden agrupar como matrices propiamente dichas, sino como un conjunto de matrices. Se les conoce como tensores. Ver problema 5

# Indexación de vectores y matrices

Para indexar se usa la siguiente sintaxis:

`array[fil, colu]`

Ejemplos:

- ▶ Del array `x`, todas las filas y la primera columna:  
`x[:,0]`
- ▶ Del array `x`, todas las columnas y la primera fila:  
`x[0,:]`
- ▶ Del array `x`, la primera fila con la primera columna:  
`x[0,0]`