

# Introducción a Python



```
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

self.logger
if path:
    self.file = open(os.path.join(path, 'requirements.txt'), 'w')
    self.file.seek(0)
    self.fingerprints.update(e.reqreq() for e in self.reqreq())

@classmethod
def from_settings(cls, settings):
    debug = settings.getbool('DEBUG_FILTER_JOBS')
    return cls(job_dir(settings), debug)

def request_seen(self, request):
    fp = self.request_fingerprint(request)
    if fp in self.fingerprints:
        return True
    self.fingerprints.add(fp)
    if self.file:
        self.file.write(fp + os.linesep)
```

## Objetivos

El objetivo de esta práctica es introducir y repasar algunos conceptos necesarios para empezar a trabajar con redes neuronales en Python.

## Temas

- Repaso de Programación con Python
- Uso Jupyter Notebooks y Google Colab
- Bibliotecas Numpy y Pandas

## Jupyter Notebooks y Google Colab

Jupyter Notebooks y Google Colab son herramientas similares (la segunda está basada en la primera). La diferencia más importante es que **Jupyter NB** permite trabajar off-line al instalarse de forma local (ej: con Anaconda) mientras que **Colab** permite trabajar on-line al alojarse en los servidores de Google Cloud. A manera de introducción pude ver los siguientes video donde se muestra el funcionamiento básico de los entornos:

<https://www.youtube.com/watch?v=KajSbrEBZ5k>

[https://www.youtube.com/watch?v=8VFYs3Ot\\_aA](https://www.youtube.com/watch?v=8VFYs3Ot_aA)

### Ejercicio 1

Investigue y defina:

- ¿Qué es un cuaderno (notebook) Jupyter? ¿Que es una celda? ¿Que tipos de celdas existen?
- ¿Qué es un entorno de ejecución? ¿Cuánto tiempo dura una sesión de un entorno?

### Ejercicio 2

Las celdas de código de un cuaderno no solo permiten ejecutar instrucciones de Python. Utilizando el símbolo ! (signo de admiración) es posible ejecutar los comandos disponibles desde la línea de comandos de su sistema operativo. De esta manera **!ls** o **!dir** (dependiendo el sistema operativo) listan los archivos de la carpeta actual y **!pip** o **!conda** permiten administrar los paquetes python.

Experimente la ejecución de varios comandos del sistema operativo a través de celdas de código. Entre las pruebas incluya la invocación de:

- python para determinar la versión instalada.
- pip show **nombre** (pandas, numpy, tensorflow, etc.) de paquete para saber la versión instalada.
- comandos que permitan crear y eliminar carpetas.

### Ejercicio 3

Dado que el entorno de ejecución de un cuaderno Colab tiene un límite de duración, es importante descargar o salvar fuera del mismo los archivos que se generan.

Conecte su cuenta de Google Drive con Google Colab:

- Acceda a la url <https://colab.research.google.com/> y auténtíquese con su usuario Google. Cree un nuevo cuaderno (notebook).
- Asocie Drive con Colab. Compruebe que Drive queda montado como una carpeta.
- Suba un pequeño archivo de texto a Drive (NO a Colab) y ábralo desde una celda de código Colab utilizando el siguiente código:

```
ruta_arch = '....'      # ruta y nombre a archivo a LEER desde su drive
f = open(ruta_arch, 'r') # abre archivo para leer
print(f.readlines())    # imprime contenido en pantalla
f.close()               # cierra archivo
```

- Genere el siguiente archivo y guárdelo en su carpeta Drive, comprobando que efectivamente se ha creado con el contenido esperado:

```
ruta_arch = '....'          # ruta y nombre a archivo a ESCRIBIR en su drive
f = open(ruta_arch, 'w')    # abre archivo para escribir
texto = 'Esta es una línea de texto\nEsta es otra línea de text'
f.writelines(texto)         # escribe contenido en archivo
f.close()                   # cierra archivo
```

## Repaso de Python

En esta sección tiene como objetivo hacer un pequeño repaso y proveer algunos recursos que faciliten el desarrollo con Python. Para un tutorial desde cero puede consultarse el siguiente video <https://www.youtube.com/watch?v=zAIWnwqHGok>. Para repasar estructuras de datos más utilizadas puede consultarse <https://www.youtube.com/watch?v=CCUNuqqn7PQ>. Para una referencia rápida, el sitio [https://ipgp.github.io/scientific\\_python\\_cheat\\_sheet/](https://ipgp.github.io/scientific_python_cheat_sheet/) o <https://www.pythoncheatsheet.org/> son buenas alternativas. Aquí puede repasar temas como operadores, funciones, listas, tuplas, diccionarios, conjuntos, manejo de strings y lectura/escritura de archivos, entre muchas otras.

### Ejercicio 1

Investigue/repase que son las listas, tuplas, conjuntos y diccionarios nativos de Python (puede consultar <https://www.youtube.com/watch?v=CCUNuqqn7PQ>). Utilizando los constructores para cada tipo de dato genere códigos de ejemplo y recórralos imprimiendo sus valores.

### Ejercicio 2

Genere el código necesario para recorrer simultáneamente 2 listas con la misma cantidad de elementos e imprima los mismos utilizando un único for (tip: función zip).

### Ejercicio 3

Implemente una función que a partir de la lista que recibe cómo parámetro, retorne una nueva lista sin elementos repetidos. Compruebe su correcto funcionamiento.

### Ejercicio 4

Implemente una función que calcule la distancia entre 2 puntos (2D). Utilice la función sqrt del paquete **math** para implementarla y compruebe el correcto funcionamiento de la misma.

### Ejercicio 5

Investigue y escriba código que demuestre el funcionamiento de los “slices” en listas.

## Biblioteca Numpy

Numpy es una biblioteca especializada para cálculo numérico y análisis de datos. Permite representar, a través del tipo de datos array, colecciones de datos homogéneos (mismo tipo) en múltiples dimensiones y provee funciones eficientes para su manipulación. Para una referencia rápida puede acceder al video: <https://www.youtube.com/watch?v=WxJr143Os-A>

### Ejercicio 1

Practique la creación de vectores, matrices y tensores y responda:

- ¿Qué diferencias hay entre los constructores, array, empty, full, zeros, ones, identity?
- ¿Qué tipos de datos pueden utilizarse? ¿En qué se diferencian? ¿Cuál es el tipo que se toma por defecto? ¿Es siempre el mismo?
- ¿Qué funciones se pueden utilizar para generar arreglos con números aleatorios?

### Ejercicio 2

Investigue y ejemplifique las funciones relacionadas al tamaño de los arrays de Numpy:

- ¿Para qué sirven las funciones shape, len, ndim, size?
- ¿Qué tipos de datos pueden utilizarse? ¿En qué se diferencian? ¿Cuál es el tipo que se toma por defecto? ¿Es siempre el mismo?
- ¿Qué funciones se pueden utilizar para generar arreglos con números aleatorios?

### Ejercicio 3

Practique funciones de agregación (sum, min, max, etc.) sobre vectores, matrices y tensores. Enumere y pruebe todas las funciones que encuentre y responda:

- ¿Estas funciones se aplican a todos los datos del array o pueden realizarse sobre dimensiones particulares? Ejemplifique.

### Ejercicio 4

Investigue y realice ejemplos que utilicen funciones para manipular elementos de arreglos (append, insert, delete, etc.) y arreglos entre sí (vstack, hstack, concatenate, etc.)

### Ejercicio 5

Los arrays de numpy (así como las listas) proveen de un mecanismo versátil para hacer o referenciar una sección de los mismos. Practique este mecanismo de acceso con vectores, matrices y tensores imprimiendo y modificando distintas regiones de los mismos.

## Biblioteca Pandas

Pandas es un paquete de Python que proporciona estructuras de datos rápidas, flexibles y expresivas, diseñadas para trabajar con datos **relacionales** o **etiquetados**. Para comprender el objetivo de Pandas puede revisar el video <https://www.youtube.com/watch?v=gimfTyCNfGw> a manera de introducción. Para realizar los ejercicios prácticos puede consultar el video <https://www.youtube.com/watch?v=5S01zSgE9GA>.

### Ejercicio 1

Investigue el funcionamiento del Dataframe de Pandas y cree uno con la información de la siguiente tabla:

Nombre	Edad	País
Juan	20	Argentina
María	26	Peru
Pedro	18	Brasil
José	22	Chile

Realice las siguientes operaciones:

- Imprimir los nombres de las columnas.
- Agregar a la tabla a Pablo que tiene 30 años y es originario de Colombia. Agregarlo de 2 formas diferentes.
- Eliminar de la tabla al Pedro repetido.
- Modificar los atributos de países que dicen “Peru” (sin acento) y reemplazarlos por “Perú” (con acento).

### Ejercicio 2

Guarde en disco el dataframe del ejercicio anterior en los siguientes formatos:

- archivo con separación por delimitadores (tabulador como separador).
- archivo con separación por delimitadores (punto y coma como separador).
- archivo excel.
- archivo json.