

Introducción al Aprendizaje Automático

Guía 0 - Repaso de Python

Durante la cursada de la materia vamos a usar, como lenguaje de programación, Python y las librerías asociadas del ecosistema de datos (Pandas, Matplotlib, Scikit-Learn, etc.). Ya deberían estar familiarizados con este lenguaje, pero por las dudas dejamos algunas pautas y lineamientos que nos servirán para tener un trabajo más organizado.

¿Por qué Python?

Python es un lenguaje de programación de código abierto (open source) con una sintaxis simplificada y flexible que lo convierte en una de las mejores opciones para utilizar. Existen muchos motivos para elegir Python:

- Es fácil de usar, sobre todo cuando lo comparamos con otros lenguajes de programación.
- Es rápido y eficiente (*no entraremos en debate sobre esto*).
- Hay una gran comunidad usándolo, en particular en Ciencia de Datos, Aprendizaje Automático e Inteligencia Artificial.
- Cuenta con una amplia cantidad de librerías específicas para diferentes tareas.

Local o en la nube

Durante la cursada, la mayoría de la ejercitación será presentada en forma de notebooks. Hay dos formas de trabajar con estos notebooks, local (en tu computadora) o en la nube (es decir, corriendo en alguna computadora en algún servidor ubicado en algún lugar del planeta). Cada una tiene sus ventajas y desventajas, que mencionaremos durante la clase. La forma que vamos a utilizar por defecto es la nube, en [Google Colab](#).

Si nunca viste un notebook, [acá dejamos uno de ejemplo ya colgado en Colab](#).

Si quieres trabajar localmente, al final de esta guía te dejamos una serie de instrucciones que puedes seguir para hacerlo. Recomendamos que las sigas, ya que aprender a configurar entornos de trabajo es parte de los aprendizajes que eventualmente debemos hacer.

Repaso de Python

Te dejamos dos notebooks de repaso de Python:

1. En el [primero](#) hacemos un repaso de Python puro, llegando hasta listas y ciclos (loops).
2. En el [segundo](#), vemos operaciones lógicas, estructuras if-else y un repaso de NumPy, una librería que será uno de nuestros caballitos de batalla.

En ambos notebook encontrarás ejercicios sencillos para resolver.

Bibliografía recomendada

Te dejamos algunos recursos para que mires sobre Python y cómo usarlo. Recomendamos que tengas Jupyter Lab abierto con un Notebook vacío - o, si prefieres, Colab - y pruebes los ejemplos que te van mostrando. ¡No tengas miedo de probar y romper! ¡Aprender a programar es aprender de los errores!

- <https://learnxinyminutes.com/docs/python/> - Exclusivo sobre programación en Python, sin mucho contexto y directo al grano. Minimalista, pero detallado. Muy útil si ya tienes una base de programación, pero si no, también. Tal vez el mejor recurso para empezar a probar cosas en tu notebook o en Colab. En algunas cosas profundiza más de lo que vamos a necesitar, pero eso nunca es malo.
- [Python Data Science Handbook](#) - Un recurso muy útil, ya que se trata de un libro de acceso gratuito con una explicación de muchas herramientas que usaremos durante la materia. Además, está orientado a Ciencia de Datos, como su nombre lo indica. Lo recomendamos fuertemente, pero tal vez conviene que lo empiecen a mirar una vez que ya hayan programado un poco.

Pasos para instalación local

1 - Instalación de Python

En primer lugar, debemos instalar Python. Vamos a instalar una distribución particular: [Miniconda](#). Si ya tienes instalado Anaconda está perfecto, no hagas nada. Si tienes otra distribución, te recomendamos que instales Miniconda. Dejamos las instrucciones para los Sistemas Operativos más populares.

WINDOWS

PASO 1: Instalar Python-Miniconda

- 1) Ir al [link](#) y descargar la versión correspondiente a tu sistema operativo. En este caso, [Miniconda3 Windows 64-bit](#).
- 2) ¡Instalar!
- 3) Poner que “sí” a las preguntas que te haga

PASO 2: Comprobar que se instaló correctamente

- 1) Abrir la Anaconda Prompt (que es similar a una terminal). Si no sabes lo que es una terminal, basta con que vayas a la barra de inicio de tu computadora y escribas “terminal”, “prompt” o “cmd” en el buscador. En este caso, abrir “Anaconda Prompt”, que debería figurar entre las opciones.
- 2) Tipear `python` + Enter
- 3) Fijarse qué versión de Python les aparece. Debería aparecer mencionado Anaconda.
- 4) Poner `2+5` y apretar Enter. ¿Qué más pueden hacer?
- 5) Para salir, poner `exit()` + Enter o cerrar la terminal.

Si algún paso falló, puede ser que A) no encuentres la Anaconda Prompt ó B) cuando escribas `python` no reconozca esta instrucción. En ese caso, no te preocupes, no sigas los pasos siguientes y, por ahora, trabaja en la nube. En una clase presencial o virtual puedes pedir ayuda a algún docente o compañero/a.

UBUNTU

PASO 1: Instalar Python-Miniconda

- 1) Ir al [link](#) y descargar la versión correspondiente a tu sistema operativo. En este caso, [Miniconda Linux 64-bit](#).
- 2) ¡Instalar!
- 3) Poner que “sí” a las preguntas que te haga

PASO 2: Comprobar que se instaló correctamente

- 1) Abrir una Terminal nueva (es importante que cierres la terminal donde hiciste la instalación y abras una nueva).
- 2) Tipear `python` + Enter
- 3) Fijarse qué versión de Python les aparece
- 4) Poner `2+5` y apretar Enter. ¿Qué más pueden hacer?
- 5) Para salir, poner `exit()` + Enter o cerrar la terminal.

Si algún paso falló, puede ser que cuando escribas Python no reconozca esa instrucción. En ese caso, no te preocupes, no sigas los pasos siguientes y, por ahora, trabaja en la nube. En una clase presencial o virtual puedes pedir ayuda a algún docente o compañero/a.

macOS (Apple)

PASO 1: Instalar Python-Miniconda

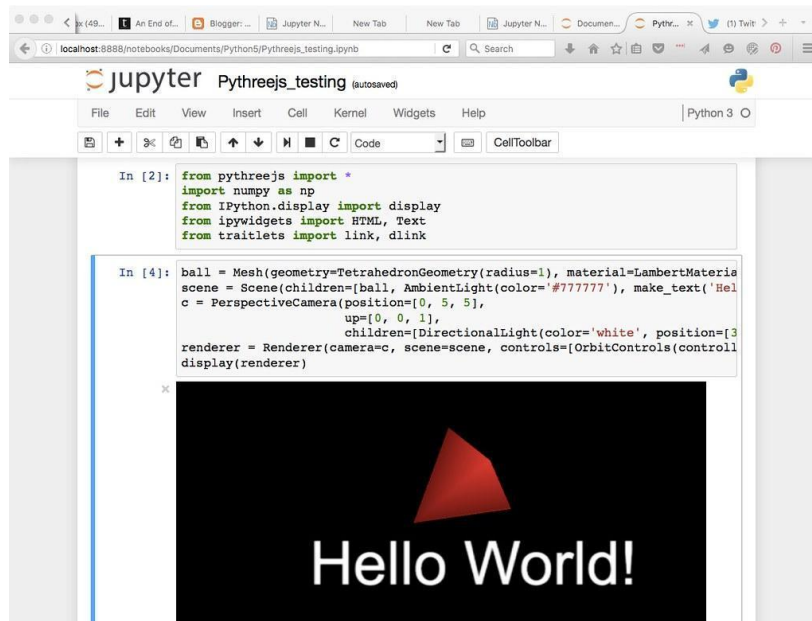
- 1) Ir al [link](#) y descargar la versión correspondiente a tu sistema operativo.
- 2) ¡Instalar!
- 3) Poner que “sí” a las preguntas que les haga

PASO 2: Comprobar que se instaló correctamente

- 1) Abrir una Terminal (Aplicaciones -> Utilidades -> Terminal).
- 2) Tipear `python` + Enter
- 3) Fijarse qué versión de Python les aparece
- 4) Poner `2+5` y apretar Enter. ¿Qué más pueden hacer?
- 5) Para salir, poner `exit()` + Enter o cerrar la terminal.

2 - Instalación de Jupyter

Los siguientes pasos son comunes a todos los sistemas operativos.



Los notebooks de Jupyter permiten intercalar código con texto, fórmulas matemáticas, imágenes y figuras generadas por el propio código. Es una herramienta muy útil de prototipado y comunicación efectiva entre colegas.

Conda nos permite crear ambientes (environments) de trabajo e instalar librerías, las cuales vendrían a ser complementos de un lenguaje de programación. No es obligatorio para trabajar con Python crear un ambiente, pero sí conveniente. A medida que vayamos trabajando con Python y vayamos instalando más librerías, van a ir quedando claras las ventajas de trabajar con ambientes.

1. Abrir una terminal nuevamente y poner `conda create --name iaa`. Les preguntará si proceder, poner que sí (y).
2. Activar el ambiente: `conda activate iaa`
3. Instala Jupyter Lab: `conda install jupyterlab`. Les preguntará si proceder, poner que sí (y).
4. Comprobar que anduvo. Poner `jupyter lab`. Debería abrirles una pestaña en su navegador.
5. Descargar el [notebook que te mostramos de ejemplo](#) y abrirlo. Para ello, pueden buscarlo en sus computadoras usando la sección izquierda del Lab.
6. Si quieren crear un nuevo notebook vacío, pueden hacerlo desde la pestaña File (¡Usar Python 3!) o clickeando en la opción Python 3 debajo de Notebook en el *Launcher*.