
SETUP

ANACONDA

- ▶ Distribución open source de los lenguajes Python y R (principalmente)
- ▶ Dispone de un gestor de versiones y entornos propio, así como su propio repositorio de paquetes
- ▶ Históricamente anaconda fue innovador ya que permitía dependencias no escritas en Python (esto fue añadido a pip con el uso de wheels)
- ▶ Hay que distinguir dos proyectos distintos pero íntimamente relacionados: Anaconda y Conda

ANACONDA/CONDA

- ▶ Anaconda
 - ▶ Es una distribución completa de Python (u otros lenguajes como R)
 - ▶ Versiones de Python 3.7 a 3.10
 - ▶ Existe miniconda que es una distribución mínima
- ▶ Conda
 - ▶ Es un gestor de paquetes y entornos. Realmente sería un sustituto de apt/yum orientado a recursos de Python

Python version	Name	Size	SHA256 hash
Python 3.10	Miniconda3 Windows 64-bit	52.9 MiB	2e3086630fa3fae7636432a954be530c88d0705fce497120d56e0f5d865bd51
Python 3.9	Miniconda3 Windows 64-bit	53.0 MiB	4b92942fbd70e84a221306a801b3e4c06dd46e894f949a3eb19b4b150ec19171
Python 3.8	Miniconda3 Windows 64-bit	52.5 MiB	9f6ce5307db5da4e391ced4a6a73159234c3fc64ab4c1d6621dd0b64b0c24b5f
Python 3.7	Miniconda3 Windows 64-bit	50.7 MiB	4d48f78d7edbf4db0660f3b3e28b6fa17fa469cdc98c76b94e08662b92a308bd
Python 3.9	Miniconda3 Windows 32-bit	67.8 MiB	4fb64e6c9c28b88beab16994bfba4829110ea3145baa60bda5344174ab65d462
Python 3.8	Miniconda3 Windows 32-bit	66.8 MiB	60cc5874b3cce9d80a38fb2b28df96d880e8e95d1b5848b15c20f1181e2807db
Python 3.7	Miniconda3 Windows 32-bit	65.5 MiB	a6af674b984a333b53aaf99043f6af4f50b0bb2ab78e0b732aa60c47bbfb0704

VENTAJAS/INCONVENIENTES

▶ Ventajas

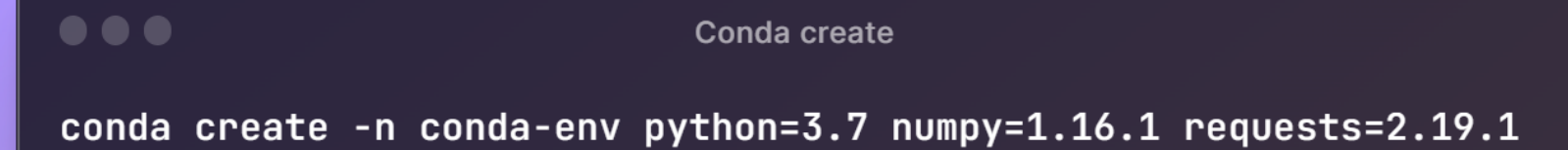
- ▶ Python puede estar más optimizado de base
- ▶ Librerías críticas pueden ser instaladas más fácilmente

▶ Desventajas

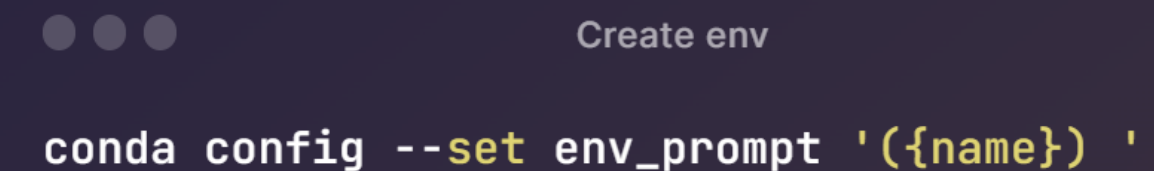
- ▶ Mantener dos gestores de paquetes
 - ▶ [Anaconda | Using Pip in a Conda Environment](#)
- ▶ Los paquetes de conda están por detrás normalmente
- ▶ El gestor de conda es más lento (se puede sustituir por mamba)
- ▶ Dos formas de gestionar entornos virtuales

RECOMENDACIÓN

- ▶ Utilizar Conda como gestor accesorio en tareas/proyectos cortos donde el objetivo sea similar al de demo/poc/comercial
- ▶ Intentar no añadir paquetes posteriores a la creación del entorno. Esto conda lo hace bastante mal y puede dar problemas al hacer downgrades inesperados
- ▶ Se puede modificar el env_prompt con el comando de la derecha



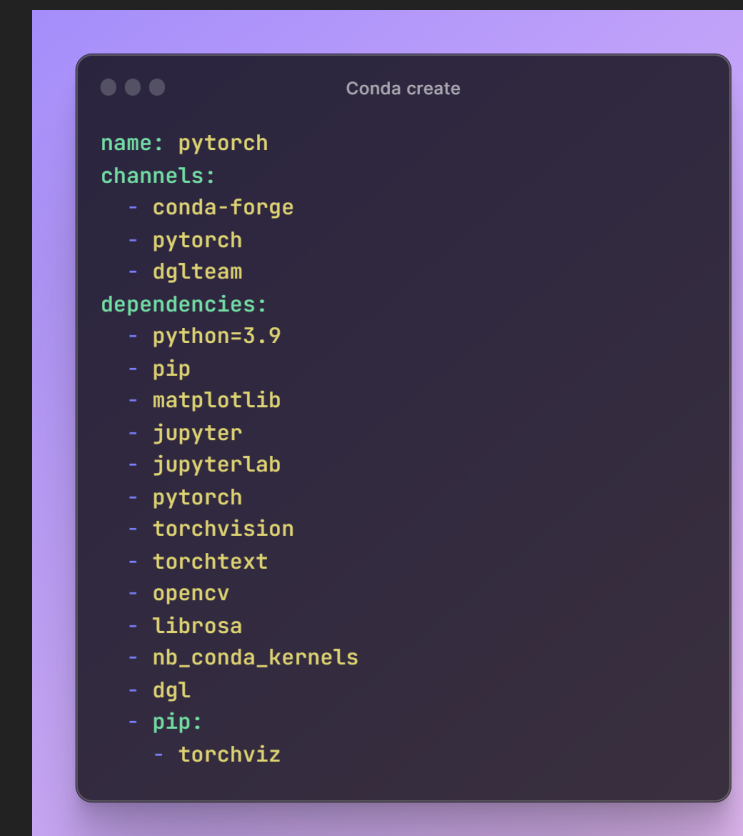
```
conda create -n conda-env python=3.7 numpy=1.16.1 requests=2.19.1
```



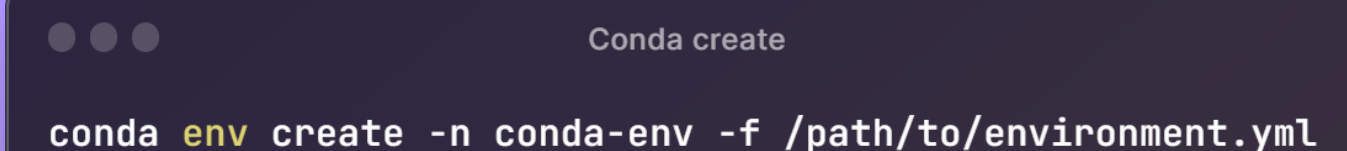
```
conda config --set env_prompt '({name}) '
```

EJEMPLO

- ▶ Vamos a crear un entorno ya configurado por un tercero en un `environment.yml` (podemos historizarlos en repos de gitlab personales)
- ▶ A continuación creamos el entorno (podemos modificar el path donde se creará con el flag `--prefix`)



```
name: pytorch
channels:
  - conda-forge
  - pytorch
  - dglteam
dependencies:
  - python=3.9
  - pip
  - matplotlib
  - jupyter
  - jupyterlab
  - pytorch
  - torchvision
  - torchtex
  - opencv
  - librosa
  - nb_conda_kernels
  - dgl
  - pip:
    - torchviz
```



```
conda env create -n conda-env -f /path/to/environment.yml
```

CONFIGURACIÓN MINICONDA Y OPEN SOURCE CHANNELS

- ▶ [Installation – conda 23.1.0 documentation](#)
- ▶ A continuación configuramos el .condarc
 - ▶ [Using the .condarc conda configuration file – conda 23.1.0 documentation](#)

A screenshot of a code editor window titled ".condarc". The file contains a comment "# This is a sample .condarc file." followed by a configuration block for channels. The block is labeled "channels:" and lists "- conda-forge" as the channel.

```
.condarc  
  
# This is a sample .condarc file.  
  
channels:  
  - conda-forge
```

GESTOR DE VERSIONES PYTHON

- ▶ Como alternativa a la gestión de versiones de Python disponemos de la existencia de varios clis de gestores de versiones. Personalmente recomiendo 2:
- ▶ pyenv (pyenv-win)
- ▶ asdf. Más recomendable pero no existe en Windows



EJERCICIO

- ▶ Instalar las herramientas propuestas
 - ▶ Gestor de versiones (pyenv/asdf)
- ▶ Tras el clonado del repositorio del curso fijar una versión de python de trabajo y generar un entorno virtual aislado (virtualenv)
 - ▶ A partir de aquí realizar las correspondientes instalaciones de paquetes en este entorno