Conda

conda used only for containeriez environments and pip for install packages

1 - Create and activate a conda environment
- conda create --name myenv
- conda activate myenv

→ switch from base conda env to myenv

2 - Install pip using conda
- conda install pip

→ To install my packages on current environment.
pip is specific with Python packages only, but conda not.

3 - Install jupyterlab
- pip install jupyterlab → IDE for notebook

4 - install ipykernel
- pip install ipykernel → the backend of Jupyter which handle the execution of code sent to notebook

5 - Add the environment to jupyter → for each env, we create it's own kernel in Jupyter
- python -m ipykernel install --user --name=myenv --display-name "Python (myenv)"

6 - Launch jupyterlab
- jupyter-lab → launch a notebook

Create clonned environment

1 - conda create --name newenv --clone oldenv

الميزة ان كل ال Packages الي كانت في ال oldenv هتكون موجودة في ال newenv ومش هتحتاج تنزلها تاني

2 - conda activate newenv

3 - python -m ipykernel install --user --name=newenv --display-name "Python (newenv)"

بنعمل لل newenv ال Kernel الخاصة بيها على Jupyter

4 - jupyter-lab
- select the kernal for this newenv in jupyter

لما تفتح notebook من ال web تبقا تختار ال Python(newenv) kernal من الإختيارات

- ليه بعمل الخطوة رقم (٣) عشان طبعاً ال isolation يكون 100% ما بين ال enviroments ويكون independent
- لو عندي env1,2,3,4,5,6 وفي tmux عملت run ل env1 مع ال Kernel بتاعتها وأنا مخطئ كله مكتب وبيستخدم env1-kernel بس لما آجي أعمل run ل another env هلاقي ال env1-kernel مشغولة
فساعتها الأفضل ان كل env يكون ليها ال Kernel الخاصة بيها

* conda env list --> list all your conda environments
* conda list --> list all packaged installed for current environment
* conda deactivate --> switch to base conda environment
* conda remove --name myenv --all --> delete myenv

1- install miniconda from conda website and install it using "bash installer_path" and reboot the terminal

2- when open the terminal again , by default it is opend on base conda

3- from the base you can create environments and switch between them
- create new environment " conda create --name <env_name> "
- activate an environment " conda activate <env_name> "

4- after activation we switched to another environment and has no packages

5- setup our environment to use it
- "conda install pip" --> for installing all my packages for each environment
- "conda install jupyter ipykernel" --> the backend of notebooks to run and debug it
- "python -m ipykernel install --user --name=myenv --display-name "Python (myenv)"
  - replace myenv with my current environmet name
  - so we can choose it from the kernal options in the notebook
- "jupyter notebook" --> to create new notbook
- this setup is made for every new environment "isolated"

JubyterLab is better

6 - we can clone a new environemt with all packages from another