conda Package and env manager

CONCEPTS

> Never install into base: give every project its own named environment.

Channels defaults, conda-forge
↓ search + solve
Package cache pkgs/
↓ link
Environment envs/<name>
↓ activate
Shell PATH set

ENVIRONMENTS

conda create -n *<env>* *<package>*...
Create a named environment

conda activate *<env>*
Activate an environment

conda deactivate
Deactivate the current one

conda env list
List all environments

conda env remove -n *<env>*
Delete an environment

conda init *<shell>*
Hook conda into a shell profile

conda run -n *<env>* *<command>*
Run a command without activating

PACKAGES

conda install *<package>*
Install into the active env

conda install *<package>*=*<version>*
Install a given version

conda install -c *<channel>* *<package>*
Install from a channel

conda update *<package>*
Update one package

conda remove *<package>*
Uninstall a package

PACKAGES (cont.)

conda list
List installed packages

conda search *<package>*
Find available versions

CHANNELS AND CONFIG

conda config --add channels conda-forge
Add a channel (highest priority)

conda config --set channel_priority strict
Prefer higher channels strictly

SHARING ENVIRONMENTS

conda env export > environment.yml
Export the active env

conda env export --from-history
Export only what you asked for

conda env create -f environment.yml
Create from a file

conda env update -f environment.yml --prune
Sync env to the file

environment.yml

```
name: myenv
channels:
  - conda-forge
dependencies:
  - python=3.12
  - numpy
  - pip:
    - requests
```

CLEANUP AND DIAGNOSTICS

conda clean --all
Remove caches and unused packages

conda update -n base conda
Update conda itself