conda Package and env manager

START HERE

`conda create -n <env> python=3.12`
New environment

`conda activate <env>`
Switch into it

`conda install <package>`
Add a package

`conda list`
What is installed?

`conda env list`
All environments

`conda env export > environment.yml`
Share the setup

`conda env create -f environment.yml`
Rebuild from file

`conda deactivate`
Leave the environment

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

CONCEPTS

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

Packages are hard-linked from a shared cache, so many environments cost little disk.

mamba is a faster drop-in for these commands; recent conda already uses the libmamba solver.

Use pip inside an environment only after conda installs; mixing the other way can break it.

ENVIRONMENTS

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

`conda create -p <path> <package>...`
Create at a path instead of by name

`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 create -n <new> --clone <env>`
Copy an environment

`conda rename -n <env> <new>`
Rename an environment

`conda info`
Show conda setup and active env

`conda init <shell>`
Hook conda into a shell profile
Restart the shell afterwards; conda config --set auto_activate_base false keeps base off.

`conda run -n <env> <command>`
Run a command without activating
Add --no-capture-output to stream output live.

PACKAGES

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

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

`conda install -n <env> <package>`
Install into another env

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

`conda update <package>`
Update one package

`conda update --all`
Update everything

`conda remove <package>`
Uninstall a package

`conda list`
List installed packages

`conda list <regex>`
Filter the installed list

`conda search <package>`
Find available versions

`conda search -c <channel> <package>`
Search one channel

`conda install --dry-run <package>`
Preview without changing

`conda install --file <file>`
Install specs from a text file

`conda install --revision <n>`
Roll back to a revision

`conda list --revisions`
Show environment history

CHANNELS AND CONFIG

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

`conda config --set channel_priority strict`
Prefer higher channels strictly

`conda config --show channels`
Show configured channels

`conda config --show-sources`
Show config files in use

CHANNELS AND CONFIG (cont.)

`conda config --show`
Show effective settings

`conda config --get <key>`
Read one setting

`conda config --set <key> <value>`
Set a setting

`conda config --append channels <channel>`
Add a channel at lowest priority

`conda config --remove channels <channel>`
Remove a channel

`conda config --set auto_activate_base false`
Do not activate base at startup

`conda install -c <channel> --override-channels <package>`
Use only this channel

SHARING ENVIRONMENTS

`conda env export > environment.yml`
Export the active env

`conda env export --from-history`
Export only what you asked for
Portable across platforms; leaves out transitive pins.

`conda env export --no-builds`
Drop build strings

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

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

`conda export --format environment-yaml`
Export in a chosen format

`conda list --explicit > spec-file.txt`
Exact URLs for reproducible builds

`conda create -n <env> --file spec-file.txt`
Rebuild from an explicit list

SHARING ENVIRONMENTS (cont.)

`conda env create -n <env> -f environment.yml`
Override the name in 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 clean --packages`
Remove unused packages

`conda doctor`
Check environment health

`conda --version`
Print the conda version

`conda update -n base conda`
Update conda itself

RECIPES

Reproducible project environment

```
conda create -n proj -c conda-forge python=3.12 numpy
conda activate proj
conda env export --from-history > environment.yml
git add environment.yml
```

Recreate an environment on another machine

```
conda env create -f environment.yml
conda activate proj
conda list
```

Try a change safely, then roll back

```
conda install --dry-run <package>
conda install <package>
conda list --revisions
conda install --revision <n>
```

Free disk space

```
conda clean --dry-run --all
conda clean --all
conda env list
conda env remove -n <old-env>
```