Nox Python session runner

CONCEPTS

Sessions are plain Python functions in noxfile.py; each gets its own fresh virtualenv under .nox/.

noxfile.py @nox.session
↓ nox -s
Virtualenv .nox/<name>
↓ session.install
Dependencies pip / uv
↓ session.run
Commands tests, lint, docs

Without -s, every session in nox.options.sessions (or all, if unset) runs in file order.

RUNNING

nox
Run all default sessions

nox -s *<session>* **[***<session>***...]**
Run named sessions

nox -l
List sessions and their Pythons

nox -- *<args>*
Pass args to sessions as posargs

nox -p *<version>*
Only sessions using this Python

nox -f *<file>*
Use another noxfile

nox -t *<tag>* **[***<tag>***...]**
Run sessions with a tag

nox -k "*<expr>***"**
Select by keyword expression

nox -s "tests(dj='5')"
Run one parametrized variant

nox -x
Stop after the first failure

nox --usage *<session>*
Show a session's docstring

nox --extra-pythons *<version>*
Add Pythons beyond the noxfile

nox -P *<version>*
Force this Python for all sessions

nox -j *<N|auto>*
Run sessions in parallel

nox --error-on-missing-interpreters
Fail instead of skipping

RUNNING (cont.)

nox --error-on-external-run
Fail on non-venv programs

ENVIRONMENTS & EXECUTION

nox -r
Reuse existing virtualenvs

nox -R
Reuse venvs and skip installs

nox --install-only
Install deps, skip session.run

nox --no-install
Skip install calls on reused venv

nox -db uv
Default venv backend (uv, virtualenv...)

nox --no-venv
Run in the current interpreter

NOXFILE

import nox
Load the library

@nox.session
Define a session function

@nox.session(python=["3.11", "3.12"])
Run once per Python

session.install("*<pkg>***")**
Install into the session venv

session.install("-e", ".")
Install the project editable

session.run("*<cmd>***", "***<arg>***")**
Run a command in the venv

session.posargs
Args given after --

session.run("pytest", *session.posargs)
Forward posargs to a tool

@nox.session(name="*<name>***")**
Set a name different from the function

@nox.session(tags=["*<tag>***"])**
Group sessions for -t

@nox.parametrize("*<arg>***", [***<v1>***, ***<v2>***])**
Run once per value

@nox.session(venv_backend="uv")
Choose backend per session

NOXFILE (cont.)

@nox.session(reuse_venv=True)
Reuse this session's venv

session.notify("*<session>***")**
Queue another session after this one

session.skip("*<reason>***")**
Skip the session with a message

session.chdir("*<dir>***")**
Change directory for later commands

session.run("*<cmd>***", external=True)**
Allow a program outside the venv

session.run(..., env={"K": "V"})
Set env vars for one command

session.run_install("*<cmd>***")**
Run a setup command, skipped with -R

nox.options.sessions = [...]
Default sessions when none given

nox.options.default_venv_backend = "uv"
Set backend for the whole file

RECIPES

Test matrix with uv

```
import nox

nox.options.default_venv_backend = "uv"

@nox.session(python=["3.11", "3.12", "3.13"])
def tests(session):
    session.install("-e", ".", "pytest")
    session.run("pytest", *session.posargs)
```

Lint and format sessions

```
@nox.session(tags=["style"])
def lint(session):
    session.install("ruff")
    session.run("ruff", "check", ".")

# nox -t style
```

Fast local loop

```
nox -s tests -p 3.12 -R -- -x -k parser

# first time: nox -s tests -p 3.12 -r
```