tox Test automation

CONCEPTS

tox builds a fresh virtualenv per environment, installs your package and deps, then runs the commands. Same steps locally and in CI.

Package build sdist/wheel
↓ install
Env virtualenv in .tox/
↓ deps
Install deps + package
↓ commands
Run exit code = result

RUNNING

tox
Run the envs in env_list

tox -e *<env>*
Run a single environment

tox -e *<e1>*,*<e2>*
Run several named environments

tox -e *<env>* **--** *<args>*
Forward args to {posargs}

tox -r -e *<env>*
Recreate the env from scratch

PARALLEL

tox p
Run envs in parallel (alias of run-parallel)

INSPECT

tox l
List envs with descriptions

tox c -e *<env>*
Show the resolved config of an env

tox c -e *<env>* **-k** *<key>*
Show only chosen keys

OTHER COMMANDS

tox quickstart
Interactively create a config (alias q)

OTHER COMMANDS (cont.)

tox exec -e *<env>* **--** *<cmd>*
Run any command inside an env

CONFIG FILE

[tox]
Core section in tox.ini

env_list = *<e1>*, *<e2>*
Default environments

[testenv]
Base section for all envs

[testenv:*<env>***]**
Settings for one env

deps = *<pkg>*
Packages to install

commands = *<cmd>* **{posargs}**
Commands to run, one per line

extras = *<name>*
Install optional dependencies

pass_env = *<VAR>*
Let env vars through to the env

set_env = *<VAR>*=*<val>*
Set env vars

allowlist_externals = *<cmd>*
Allow commands from outside the env

SUBSTITUTION & FACTORS

{posargs}
Args after -- on the command line

RECIPES

Minimal tox.ini

```
[tox]
env_list = lint, py313

[testenv]
deps = pytest
commands = pytest {posargs}

[testenv:lint]
skip_install = true
deps = ruff
commands = ruff check .
```