Python CLI Interpreter and -m tools

CONCEPTS

python3 -m <module> runs a module from the import path, so tools like venv, pip and pytest always match the interpreter you called.

RUN CODE

python3 *<script.py> <args>*
Run a script

python3 -c "*<code>*"
Run code from a string

python3 -m *<module> <args>*
Run a module as a script

python3 -V
Print the version

INTERACTIVE

python3
Start the REPL

python3 -i *<script.py>*
Run script, then stay in the REPL

_
Last result in the REPL

help(*<obj>***)**
Show docs for an object

dir(*<obj>***)**
List attributes of an object

VIRTUAL ENVS AND PACKAGES

python3 -m venv .venv
Create a virtual environment

source .venv/bin/activate
Activate it (Linux, macOS)

deactivate
Leave the environment

.venv/bin/python *<script.py>*
Use a venv without activating

python3 -m pip install *<pkg>*
Install a package

VIRTUAL ENVS AND PACKAGES (cont.)

python3 -m pip install -r requirements.txt
Install from a list

python3 -m pip install -e .
Editable install of this project

DEV SERVERS AND TOOLS

python3 -m http.server
Serve current dir on port 8000

python3 -m http.server *<port>*
Serve on another port

python3 -m json.tool *<file>*
Validate and pretty-print JSON

TEST AND DEBUG

python3 -m pdb *<script.py>*
Debug a script from the first line

breakpoint()
Drop into the debugger from code

n / s / c / q
pdb: next, step, continue, quit

l / ll / w
pdb: list code, whole function, stack

MEASURE

python3 -m timeit "*<stmt>*"
Time a small statement

FLAGS AND BEHAVIOUR

python3 -u *<script.py>*
Unbuffered stdout and stderr

ENVIRONMENT VARIABLES

PYTHONPATH=*<dir>***[:***<dir>***]**
Extra module search directories

PYTHONUNBUFFERED=1
Same as -u, good for logs and containers