pyenv Python version switching

CONCEPTS

pyenv builds each Python from source under ~/.pyenv/versions and switches between them with shims; it does not manage packages or virtualenvs by itself.

python shim in PATH
↓ asks pyenv
Resolve version env, file, global
↓ picks
~/.pyenv/versions <version>/bin/python
↓ runs
Your command right interpreter

Version resolution (first hit wins)

Source	Set by
`PYENV_VERSION`	`pyenv shell`
`.python-version here`	`pyenv local`
`.python-version in a parent`	`pyenv local`
`$(pyenv root)/version`	`pyenv global`
`system Python`	`fallback if none`

SETUP

`export PYENV_ROOT="$HOME/.pyenv"`
Root directory (default ~/.pyenv)

`export PATH="$PYENV_ROOT/bin:$PATH"`
Put the pyenv command on PATH

`eval "$(pyenv init - bash)"`
Enable shims in bash (~/.bashrc)

`eval "$(pyenv init - zsh)"`
Enable shims in zsh (~/.zshrc)

Edit shell startup files, then open a new shell. pyenv shell does nothing without the init eval line.

INSTALL VERSIONS

`pyenv install -l`
List all installable versions

`pyenv install` *`<version>`*
Build and install a version

INSTALL VERSIONS (cont.)

`pyenv install`
Install what .python-version names

`pyenv install -v` *`<version>`*
Show the compile output

`pyenv uninstall` *`<version>`*
Remove a version (asks first)

Install fails with missing zlib, ssl or ffi? Install the distro build dependencies first (see Build dependencies).

CHOOSE A VERSION

`pyenv versions`
List installed, active one marked *

`pyenv version`
Active version and where it came from

`pyenv global` *`<version>`*
Default for the user (~/.pyenv/version)

`pyenv local` *`<version>`*
Write .python-version here

`pyenv shell` *`<version>`*
Set PYENV_VERSION in this shell

FIND AND RUN

`pyenv which` *`<command>`*
Full path pyenv would run

`pyenv whence` *`<command>`*
Which versions provide a command

`pyenv rehash`
Regenerate shims

BUILD DEPENDENCIES

Debian / Ubuntu

```
sudo apt update
sudo apt install make build-essential
libssl-dev zlib1g-dev \
  libbz2-dev libreadline-dev libsqlite3-
dev curl git \
  libncursesw5-dev xz-utils tk-dev
libxml2-dev libxmlsec1-dev \
  libffi-dev liblzma-dev libzstd-dev
```