

# multiple Python versions on Ubuntu

You can install and manage multiple Python versions on Ubuntu **using the `pyenv` tool or by using the Deadsnakes PPA (Personal Package Archive) with `apt`**. The `pyenv` method is generally recommended as it is safer and user-level, while the PPA method is good for system-wide access.

## Method 1: Using `pyenv` (Recommended for development)

`pyenv` allows you to manage multiple Python versions and switch between them easily without affecting the system's default Python installation.

**Install system dependencies:** These packages are necessary to build Python from source, which `pyenv` does.

### bash

```
sudo apt update
sudo apt install git curl make build-essential libssl-dev zlib1g-dev libbz2-dev libreadline-dev libsqlite3-dev wget llvm libncurses5-dev libncursesw5-dev xz-utils tk-dev libffi-dev liblzma-dev python3-openssl
```

**Install `pyenv`:** Run the automatic installer script provided by `pyenv`.

### bash

```
curl https://pyenv.run | bash
```

**Configure your shell environment:** Add the following lines to your `~/.bashrc` (or `~/.zshrc` if you use Zsh) file to ensure `pyenv` loads correctly.

### bash

```
echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bashrc
echo 'command -v pyenv >/dev/null || export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bashrc
echo 'eval "$(pyenv init -)'" >> ~/.bashrc
```

**Reload your shell:** Close and reopen your terminal, or run the following command to apply the changes to your current session:

## bash

```
source ~/.bashrc
```

**Install desired Python versions:** Now you can install specific Python versions using `pyenv`. For example, to install Python 3.8.0 and 3.12.0:

## bash

```
pyenv install 3.8.0  
pyenv install 3.12.0
```

**Switch between versions:**

To set a global default for your user: `pyenv global 3.12.0`.

To set a local version for a specific project directory: `pyenv local 3.8.0`.

## Method 2: Using the Deadsnakes PPA (System-wide installation)

The Deadsnakes PPA offers various Python versions packaged for Ubuntu, allowing installation via `apt` without manually compiling from source. This method installs Python versions system-wide.

**Add the Deadsnakes PPA:**

## bash

```
sudo add-apt-repository ppa:deadsnakes/ppa  
sudo apt update
```

**Install specific Python versions:** Install the desired versions (e.g., Python 3.8 and 3.12).

## bash

```
sudo apt install python3.8  
sudo apt install python3.12
```

**Verify installations:** The executables will be available as `python3.8` and `python3.12`.

## bash

```
python3.8 --version  
python3.12 --version
```

Install `pip` for each version: `pip` must be installed separately for each version.

## bash

```
sudo apt install python3.8-distutils # Install distutils first if needed
curl -sS https://bootstrap.pypa.io/get-pip.py | python3.8
```

```
sudo apt install python3.12-distutils # Install distutils first if needed
curl -sS https://bootstrap.pypa.io/get-pip.py | python3.12
```

**Important Note:** Avoid modifying the default system Python symlink ( `/usr/bin/python3` ) directly, as system tools rely on it and changing it can cause your operating system to break. The methods above allow different versions to coexist safely. For project-specific dependencies, always use Python virtual environments

---

Revision #1

Created 26 January 2026 06:20:45 by AI Channel

Updated 26 January 2026 06:23:01 by AI Channel