📦 UV (Universal Virtualenv) — Quick Guide

🔧 Installing UV

```
pip install uv
```

🎯 Installing Specific Python Versions

Single version:

```
uv python install 3.12
```

Multiple versions:

```
uv python install 3.11 3.12
```

Alternative implementation (e.g., PyPy):

```
uv python install pypy@3.10
```

🔄 Reinstalling Python

Reinstall uv-managed Python versions:

```
uv python install --reinstall
```

📃 Viewing Installed Python Versions

```
uv python list
```

⬆️ Upgrading Python Versions

Upgrade a specific version:

```
uv python upgrade 3.12
```

Upgrade all uv-managed versions:

```
uv python upgrade
```

🐌 Requesting a Python Version

Use the --python flag to request a specific Python version during environment creation:

```
uv venv --python 3.11.6
```

🟢 Activating Virtual Environment

After creating the virtual environment, activate it using:

```
uv venv
```

```
venv\Scripts\activate # For Windows
```

💡 On macOS/Linux, use:

```
source .venv/bin/activate
```

🚀 Running Scripts Without Dependencies

```
uv run example.py
```

🧱 Creating a New Project

Option 1: With project name

```
uv init example-app cd example-app
```

Option 2: In current directory

```
mkdir hello-world cd hello-world uv init
```

Project Structure Created:

```
├── .gitignore ├── .python-version ├── README.md ├── main.py └── pyproject.toml
```

Run your first script:

```
uv run main.py
```

📦 Managing Dependencies

📥 Importing from requirements.txt

```
uv add -r requirements.txt
```

❌ Removing a Dependency

```
uv remove httpx
```

🔄 Changing a Dependency

```
uv add "httpx>0.1.0"
```

🏃‍♂️ Running Commands in Projects

The project installs into .venv, isolated from your current shell. So, use:

```
uv run python -c "import example"
```

🏗️ Building & Publishing a Package

📦 Build a Package

```
uv build
```

🔴 Mark Internal Packages as Private Inside pyproject.toml:

```
[project] classifiers = ["Private :: Do Not Upload"]
```

🔐 This prevents accidental publishing to PyPI. Use per-project PyPI tokens for added safety.

🚀 Publish the Package

```
uv publish
```

📥 Installing & Testing Your Package

```
uv run --with --no-project -- python -c "import "
```

🧠 Tip: If the package was recently installed, avoid caching:

```
--refresh-package
```

📄 This guide is based on instructions from the UV GitHub repository and its official documentation.

🔗 Source: Official UV Documentation - Github, (Documentation)