

# Development Environments and Package Management in Python 3

## Managing Python Packages with Pip



**Reindert-Jan Ekker**

@rjekker | <https://nl.linkedin.com/in/rjekker>



# Version Check



**This version was created by using:**

- Python 3.10
- Pip 23.0.1
- Poetry 1.4.2
- Pipenv 2023.3.20
- Pipx 1.2.0



# Version Check



**This course is 100% applicable to:**

- Python 3.x
- Pip 22+
- Pipenv 2023.x and later
- Poetry 1.3+
- Pipx 1.x



# Overview



## Pip basics recap

### Installing packages

- Specific versions
- Version operators
- Dependencies

### Packages not hosted on Pypi

- Github or other VCS
- Local
- Editable

### Installing outside of a project

### Configuring pip

- Network proxy, custom repository, etc.



## Demo



### Pip basics recap

- PyPi package repository
- Install and remove packages
- Inspecting installed packages



## Demo



## Installing

- Specific versions
- Version comparisons
- Upgrading



## Demo



### Packages with dependencies

- Definition
- Extras



## Demo



### Installing packages that are not on Pypi

- Github
- Local files



## Demo



### Editable installs

- Installing while developing



# Installing Packages – Requirements Specifiers

```
# Download and install latest release of arrow  
pip install arrow
```

```
# Request a specific version  
pip install "arrow==1.2.3"
```

```
# Version comparison (Operators: <, >, >=, <=)  
pip install "arrow<1.2"  
pip install "arrow>=1.2"
```

```
# Excluding a version  
pip install "arrow!=1.2.1"
```

```
# Combining operators: comma means AND  
pip install "Flask>=1.0.0, !=1.1.0, <=2.0.0"
```



# Installing Packages – Requirements Specifiers 2

```
# Compatible release:  
# Same major version, minor version equal or greater  
pip install "requests ~=2.24.0"  
  
# Equivalent  
pip install "requests >=2.24, ==2.*"  
  
# Environment marker  
pip install "asyncio; python_version=='3.4'"  
pip install "typing; python_version<'3.5'"  
  
pip install "asyncio; os_name=='posix'"          # Mac OS, Linux, UNIX  
pip install "asyncio; os_name='nt'"                # Windows  
pip install "asyncio; sys_platform='darwin'"      # Mac OS
```



# Installing Packages – Requirements Specifiers 3

```
# This will not override existing installation  
pip install requests
```

```
# Force overwrite  
pip install --upgrade requests
```

```
# Optional extras  
pip install "fastapi[all]"  
pip install "fastapi[docs,test]"
```



# Installing from Other Sources

```
# Install from Github URL
pip install "MyProject @ git+ssh://git.example.com/MyProject"
pip install "MyProject @ git+https://git.example.com/MyProject"

# Install from commit, tag or branch
pip install "MyProject @ git+https://git.example.com/MyProject@master"
pip install "MyProject @ git+https://git.example.com/MyProject@da39b"
pip install "MyProject @ git+https://git.example.com/MyProject@v2.1"

# Install a local project or wheel file
pip install path/to/my/project
pip install myproject-1.1.whl

# Install editable
pip install -e path/to/my/project
pip install -e .
```



## Demo



### Installing packages outside a project

- Utilities
- pip --user
- pipx



## Demo



### Configuring pip

- Hosting your own package repo
- Using a proxy server
- Custom SSL certificates



# Installing Outside of a Project

```
# Pipx: https://pypa.github.io/pipx/
# Install packages inside their own venv
pipx install black

# Make sure PATH is configured correctly
pipx ensurepath

# Alternative: use pip with --user switch
# This does NOT automatically setup a venv
# And you need to do PATH setup yourself
pip install --user black

# Install for the entire system (Linux)
# Probably a bad idea!
sudo pip install black
```



# Resources Referenced in this Course

Pip Documentation: <https://pip.pypa.io/>

Pip install examples: [https://pip.pypa.io/en/stable/cli/pip\\_install/#examples](https://pip.pypa.io/en/stable/cli/pip_install/#examples)

Pip config file: <https://pip.pypa.io/en/stable/topics/configuration/>

sys.platform environment marker (incl. link to os.name):

<https://docs.python.org/3/library/sys.html#sys.platform>



# Resources Referenced in this Course 2

**Quickstart for pyproject.toml**

<https://setuptools.pypa.io/en/latest/userguide/quickstart.html>

**Install executables with pipx:** <https://pypa.github.io/pipx/>

**Manage multiple Python installs with pyenv:** <https://github.com/pyenv>



# Summary



## Installing packages

- Specific versions
- Version operators
- Dependencies

## Packages not hosted on Pypi

- Github or other VCS
- Local
- Editable

## Installing outside of a project

## Configuring pip

- Network proxy, custom repository, etc.



**Up Next:**

# **Setting up your Project with Virtual Environments**

---

