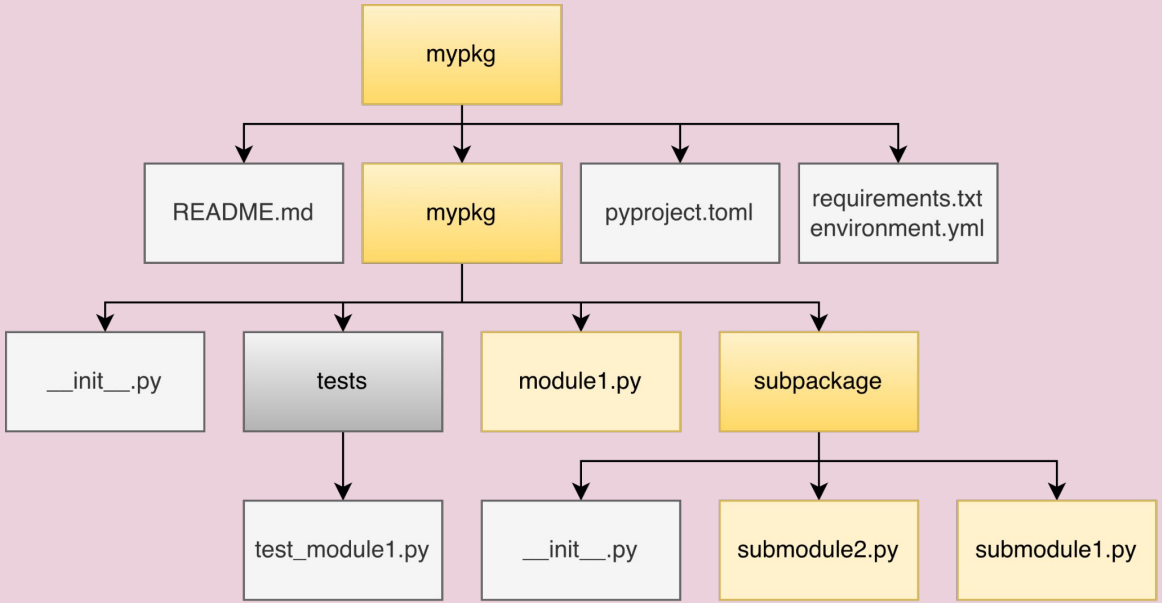
Lab 09

Python Packaging

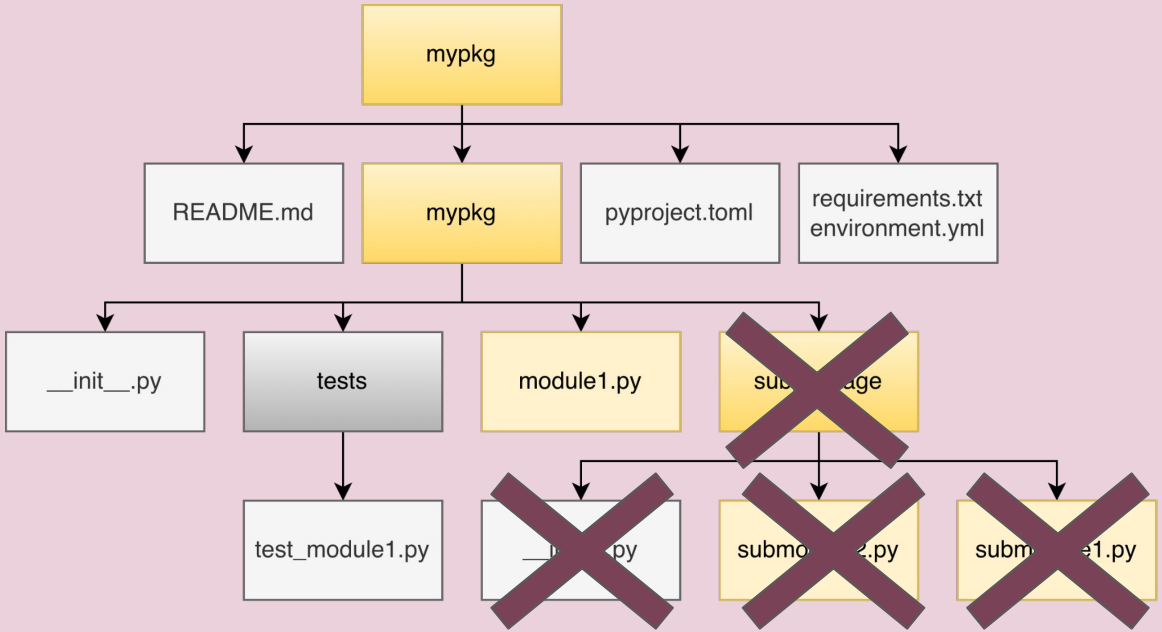
Today's plan

01	**Review of Packaging**	(Python only)
02	**Reading Checkpoint**	
03	**Lab notebook**	Moving from notebooks to packages

Package Structure in Python



Package Structure in Python - For Today's lab



Core Components of a Python Package

Minimal components:

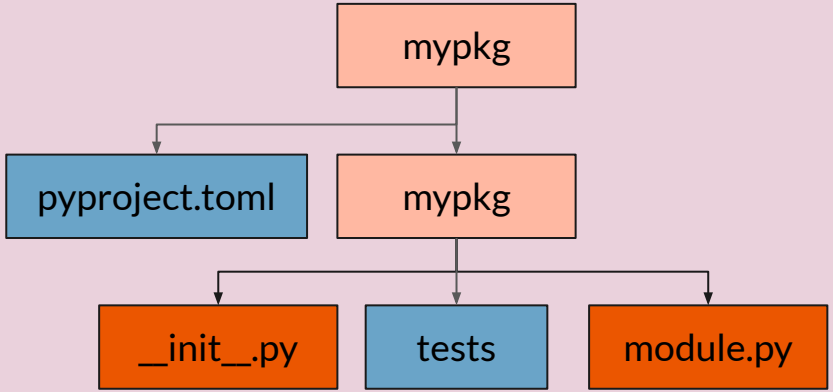
- Directory named after the package
- __init__.py within package directory
- Package code in a module

To make it installable with pip:

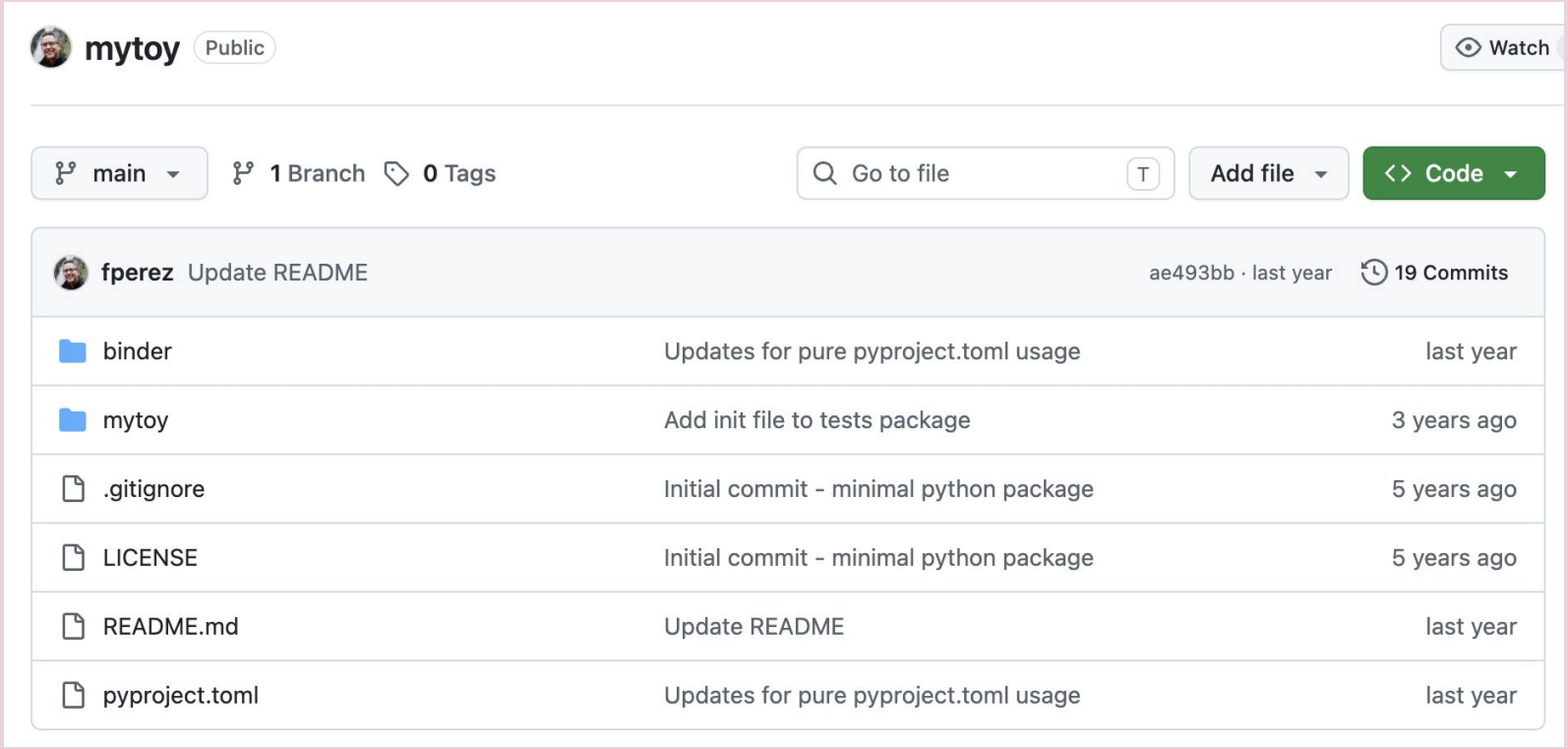
- pyproject.toml

To add tests:

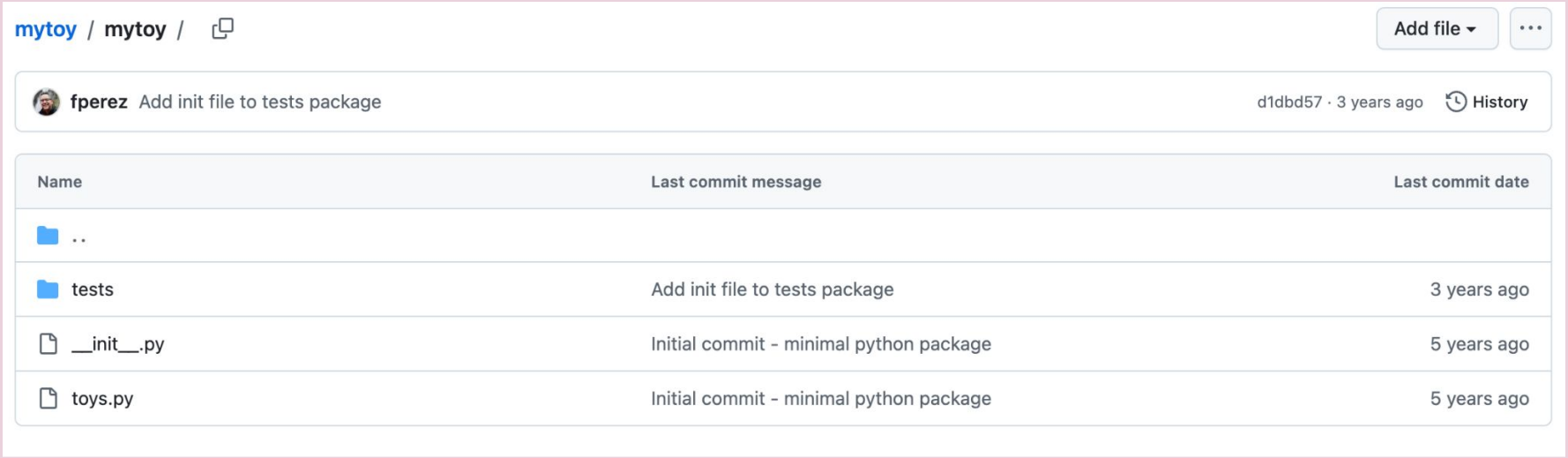
- ‘tests’ directory



Example Package



Example Package



Example Package - pyproject.toml

Code | Blame | 36 lines (29 loc) · 739 Bytes

```toml
[build-system]
requires = ["setuptools >= 70"]
build-backend = "setuptools.build_meta"

[project]
name = "mytoy"
authors = [
  { name="Example Author", email="author@example.com" },
]
description = "A Python library to make toys"
readme = "README.md"

requires-python = ">=3.10"
dependencies = [
    "tqdm",
]

license = "BSD-3-Clause"
license-files = ["LICEN[CS]E*"]
classifiers = [
    "Programming Language :: Python :: 3",
    "Operating System :: OS Independent"
]

keywords = ["tools", "toys"]
dynamic = ["version"]

[project.urls]
Homepage = "https://github.com/fperez/mytoy"
Issues = "https://github.com/fperez/mytoy/issues"

[tool.setuptools]
packages = ["mytoy"]

[tool.setuptools.dynamic]
version = {attr = "mytoy.__version__"}
```

(Reading) Checkpoint #8

You'll return it at the end of class

Work on lab09.ipynb