anatomy_of_a_python_package

August 2, 2023

1 Introduction

A python, or more generally a software, package is just a collection of files and directories that contain code. But, typically what is meant when we call a set of software a 'package' is that it can be easily installed. In the case of Python, this means that the package can be installed with `pip`.

Formally packaging our code makes a number of tools available to us (the developers). As such, the code we will be writing in this course in Python is formally packaged.

2 The package structure

2.1 Stuff you can ignore

For this course, you can ignore the vast majority of files and directories which are in the assignment repositories. These files and directories function to configure the various tools, for instance some of the github or vscode functionality. These are files which you should expect to see, but which you do **not** need to open:

- .devcontainer
- .github
- .pytest_cache
- .venv
- .vscode
- .gitignore: note, it can be sometimes useful to add paths to the .gitignore
- make.bat
- Makefile
- poetry.lock
- pyproject.toml

2.2 Assignment files

In each package, there is by convention a subdirectory with the same name as the repo. In this assignment, that means there is a subdirectory called `python_intro`. In this repo, there is more stuff which you can and should ignore:

- ___pycache___
- ___init___.py
- utils

But, you *should* look at `assignment.py` and `test_assignment.py`:

- `assignment.py` will store the source code which is where you will be doing the majority of your work
- `test_assignment.py` stores the tests which your code should pass. In very limited cases, you will be asked to write or adjust a test.