

Python Packaging

From the basics to publishing
packages on PyPI. Plus a bit
of **Swagger**

Daniel Nicolas Gisolfi

What's it all for? 🤔

- Creates a simpler process for distributing packages (hopefully)
- Allows developers to build upon tools
- Forces you to follow at least some of the python standard
- Follow [PEP8!](#)

The Basics - `__init__.py`

Q: What's an `__init__.py` for?

```
coffee
├── LICENSE
├── README.md
└── coffee
    └── __init__.py
└── setup.py
```

A: Is required to import the directory as a package

[Python Docs](#)

The Basics - `setup.py`

Q: What's a `setup.py` for?

```
coffee
├── LICENSE
├── README.md
└── coffee
    └── __init__.py
        └── setup.py
```

A: is the build script for your package. It tells setuptools about your package and which code files to include.

The Basics - `setup.py`

```
from setuptools import setup, find_packages

if __name__ == "__main__":
    setup(
        name=coffee.__title__,
        version=coffee.__version__,
        author=coffee.__author__,
        author_email=coffee.__author_email__,
        description=coffee.__description__,
        license=coffee.__license__,
        packages=find_packages(),
    )
```

Defining Dependencies

A Requirements.txt file is very simple Ex:

```
setuptools >= 21.0.0
```

How can we define dependencies for:

- Testing
- Development
- Production

Requirements.txt

```
coffee
├── LICENSE
├── README.md
├── coffee
│   └── __init__.py
└── test
    └── __init__.py
├── requirements.txt
└── test-requirements.txt
    └── dev-requirements.txt
└── setup.py
```

This is getting messy...

Pipfiles

```
[[source]]
name = "pypi"
url = "https://pypi.org/simple"
verify_ssl = true

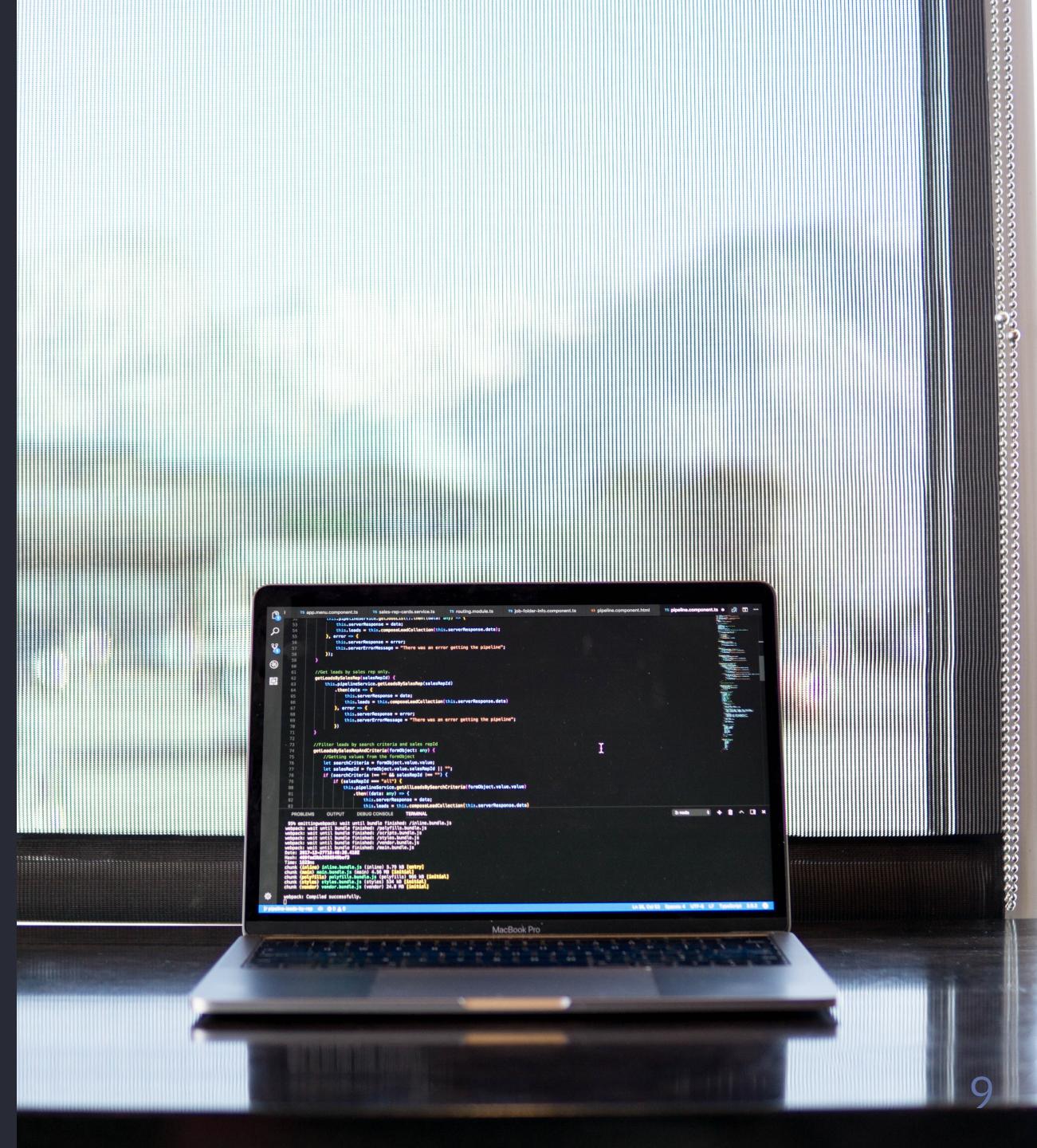
[dev-packages]
twine = "*"

[packages]
click = "*"

[requires]
python_version = "3.7"
```

What makes this better?

- You can define separate collections of requirements
- Packages can be specified from separate pypi repositories
- Creates a virtual env and manages it for you



Demo

** Let's take a look at the example coffee package!**

[Check it out here](#)



Swagger

Q: Why are you derailing this talk to focus on swagger?

“ A specification for machine-readable interface files for describing, producing, consuming, and visualizing RESTful web services. - Wikipedia ”

A: It's important...trust me

Swagger Editor

Twine

Now that we have a package,
how do we publish it?

```
python3 -m twine upload dist/*
```

Upload to a private or public
PyPI repository.



Questions?

Advice

- Write code.
- Write a tool that is useful to you in some capacity
 - [MKP](#) - The multidimensional 0–1 knapsack problem solved using a greedy algorithm.
 - [RotatingProxyBot](#) - A Bot that uses a Rotating Proxy to simulate many clients making a request to a single server