Just python environments

Pierce Edmiston

Developer environments

What is the purpose of developer tooling?

To improve our ability to collaborate and deliver software products[1].

- version control
- tests
- package managers
- build tools

[1] and not to write in JavaScript.

Just developer environments

A Theory of Justice by John Rawls

> *Society should be structured so that the greatest possible amount of liberty is given to its members – Wikipedia*

Stepping behind the veil of ignorance

What would you expect of your tools if you weren't sure of what you were building?

- version control
- tests
- package managers
- build tools
- **open source**

Tooling for flow

Tooling shouldn't slow development!

1. Tools should improve collaboration
2. Tools should make it easy to switch contexts
3. **Tools should be easy to configure on a new computer**

Congratulations!

Your laptop passed away last night. Here is your new computer.

> You received `1 macOS Mojave`.

Captain's Log

madison-python/environments/log.md

Python development tools

pyenv Python version manager. Installs different distributions of python. cf. rbenv in ruby, nvm in node

pipenv Python virtualenv manager. Installs Python packages with pip in different project environments. Uses Pipfile/Pipfile.lock. cf. Gemfile/Gemfile.lock, package.json/yarn.lock

pytest Python test runner and framework.

pyenv basics

```
pyenv install --list | less
pyenv install --list | grep 3.8
pyenv install 3.8-dev  # error 1
pyenv install 3.7.3    # error 2
pyenv install anaconda3-2019.03  # works but not perfect
pyenv local 3.7.3  # creates .python-version
pyenv global 3.7.3 # creates ~/.pyenv/version
```

pipenv basics

```
pipenv install --python 3.7.3
pipenv install flask
pipenv install jinja2==2.10.1
pipenv install --dev pytest
pipenv --rm
```

pipenv advanced

```
pipenv lock
pipenv install jupyter --skip-lock
pipenv sync
```

pipenv tricks

```
pipenv run ./myscript.py
pipenv run ipython
pipenv shell
```

pipenv reads .env files!

```
# contents of .env
MY_ENV_VAR=100
```

```
$ pipenv run echo $MY_ENV_VAR
```

Why I use pipenv

1. pipenv install –python 3.6.7
2. CMD-T + pipenv shell

```
$ heroku create
Creating app... done, sheltered-bayou-58915
https://sheltered-bayou-58915.herokuapp.com/ | https://git
$ heroku run bash --app sheltered-bayou-58915
Running bash on sheltered-bayou-58915... up, run.5201 (Free
~ $ python3 --version
Python 3.6.7
```

pipenv gotchas

1. pipenv run code .
2. pipenv install typo

Live demo

Roman numeral conversion as a Service (RaaS)

- Write a library function `roman_to_number` that takes in a roman numeral and returns a number.
- Deploy it as an API web service.

Test-driven development

Red-green-refactor

- Can't write any production code until you have a failing test.
- Can't write more production code than is required to pass the test.
- Can't refactor beyond existing functionality.