

# Just python environments

---

Pierce Edmiston

## What is the purpose of developer tooling?

To improve our ability to collaborate and deliver software products<sup>1</sup>.

- version control
- tests
- package managers
- build tools

---

<sup>1</sup>and not to write in JavaScript.

*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 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`.

`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  # ...
pyenv local 3.7.3    # creates .python-version
pyenv global 3.7.3  # creates ~/.pyenv/version
```

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

## pipenv gotchas

1. pipenv + visual-studio-code
2. pipenv install “typo”

### 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.

## 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 in a way that adds functionality.