

# A Jr. Developer's Guide to Software Engineering October 2018

- Now: Jr. Developer, Pythonista/Djangonaut
- Then: Accountant & uni instructor w/ an MBA

## Who Else Am I?

- DjangoCon US Organizer
- DEFNA North American Ambassador
- @DEFNADO for more info

Coding?

Why?

How?

# Two Styles of Coding

## Programming

- Individuals
- Author
- Less robust

## Software Engineering

- Teams
- Others
- More robust

## Interjection(!): “Real” coders

- If your code runs & does what you want it to do, you’re a “real” coder.
- Choose your style

## The Issue

- Career changers seek internships or "Jr. Developer" jobs
- Jr. Developer == ???
- “When am I ready?”

Kojo Idrissa

@Transition

The Issue

MOAR Programming!

## Secondary Issue

How do you mentor a new developer?



# What I'll Talk About

- Version Control
- Documentation
- Testing
- Dependency Management/Deployment
- Development Environment

# Version Control: Why?

- Protect yourself from mistakes
- Enable yourself to try new things
- Learn to work on a team

# Version Control: When?

- More code than you want to retype
- Working code you're scared to break

# Version Control: How?

- Solo usage
- Basic feature branch/merge workflow
- Git != Github
- Ask me about a practical example

# Version Control: How?

- Reference
  - Pro Git Chapters 1-3
  - Free!

# Documentation

- Refine your thinking
- Help your future self
- Help your teammates help you

# (!) Self-Documenting Code

- Kinda, not really
- Naming is hard
- Start to think about it now

# Documentation: When?

- Writing code using a new technique
- Writing functions, especially if they call other functions



## Documentation: How

- Docstrings for functions (minimums)
  - What the function does
  - What's the input? (include data type)
  - What's the output? (include data type)

Testing & TDD

So.

Many.

Reasons.

# Why Testing: The Big Picture

- Tests -> Automated tests
- Automated Tests -> CI
- CI -> CD (Continuous Delivery)
- CD -> CD (Continuous Deployment)

# Why should YOU write tests?

- Protection from future changes breaking things
- Assurance past fixes still work

# Testing: When?

- As soon as possible
- Art + Science: experience helps

# Testing: How?

- doctest & unittest: builtin
- pytest: *pip install pytest*

# Dependency Mgmt & Deployment

- Virtual Env(*ironments*)
- Containers
- Virtual Machines (VMs)

# Other

- Development Environment
- A shell language
- Terminal multiplexer



# Finding New Developers

To:

- Engineering Managers
- Dev Leads
- Others trying to hire developers...

# Questions/Comments?

- @Transition on Twitter
- [kojoidrissa.com](http://kojoidrissa.com)
- [https://github.com/kojoidrissa/pygotham\\_2018](https://github.com/kojoidrissa/pygotham_2018)