Writing Production-grade Python Code

Nick Russo

NETWORK ENGINEER

@nickrusso42518 www.njrusmc.net

Agenda

Version control with git

In-line Python error checking

Making code pretty with formatters

Checking for bugs with linters

Simplifying execution with Makefiles

My Thoughts on Git

Time travel

Concurrent development

Complex to learn

Demo

Time travel with git

Some Ideas for Error Checking

- CLI arguments
- File I/O
- Arithmetic
- Parsing
- Communications
- Format/structure

Demo

Improving durability with error checking

Beautification with Formatters

Eliminate "snowflake" code

Inter-project consistency

Not many features by design

Demo

Formatting Python code with "black"

Bug Checking with Linters

Find syntax and styling issues

Fast and secure

Widely used for many languages

Demo

Linting Python code with "pylint"

Makefiles to Simplify Operations

Dependency-aware build tool

Break up execution phases

Different "targets" for different tasks

Demo

Using Makefiles with Python projects

The Clean up Module Review

Makes your life easier

Makes other people's lives easier

Makes your code stronger

Final Thoughts

Many suitable solutions

No single best answer

Thank you!