Motivations/Mechanics

- I would like a study group that tackles a topic in depth, so that we can teach each other about it.
- It fits somewhere between socializing and a full talk.
- Short meetings along the lines of a stand up
- short presentations about:
  - first round: Anything we had difficulty on or questions about.
  - second round (optional) Anything we had to learn to more fully understand a topic

Suggestion for first topic

This series of books that are free to download:
https://softwarefoundations.cis.upenn.edu/index.html

I took the course Software Foundations - covered most of the first 3 books. I Have wanted to work through them again.

Topics:

1. Use of a theorem proover
2. functional programming
3. programming languages
4. formal specification
5. mechanical verification

Quick Introduction to rocq (formally know as coq)

Just to give some foreshadowing

1. theorem provers are based on propositional logic/calculus. There are several good youtube videos and wikipedia pages not to mention a discrete math textbook.
2. The underlying logic in rocq does not use the rule of excluded middle ($\neg p \vee p$)

Quick Demo

Intro standup

What I've done so far:

1. Couldn't get the syntax correct for the last proofs in Basics.v (Theorem lower_letter_lowers). Left as Admitted.
2. Continued with Induction.v
3. created a Dockerfile and tested the configuration with Basics.v and Induction.v
4. Went down the rabbit hole looking up the precise definition for computational equivalence "=?" - Will present this