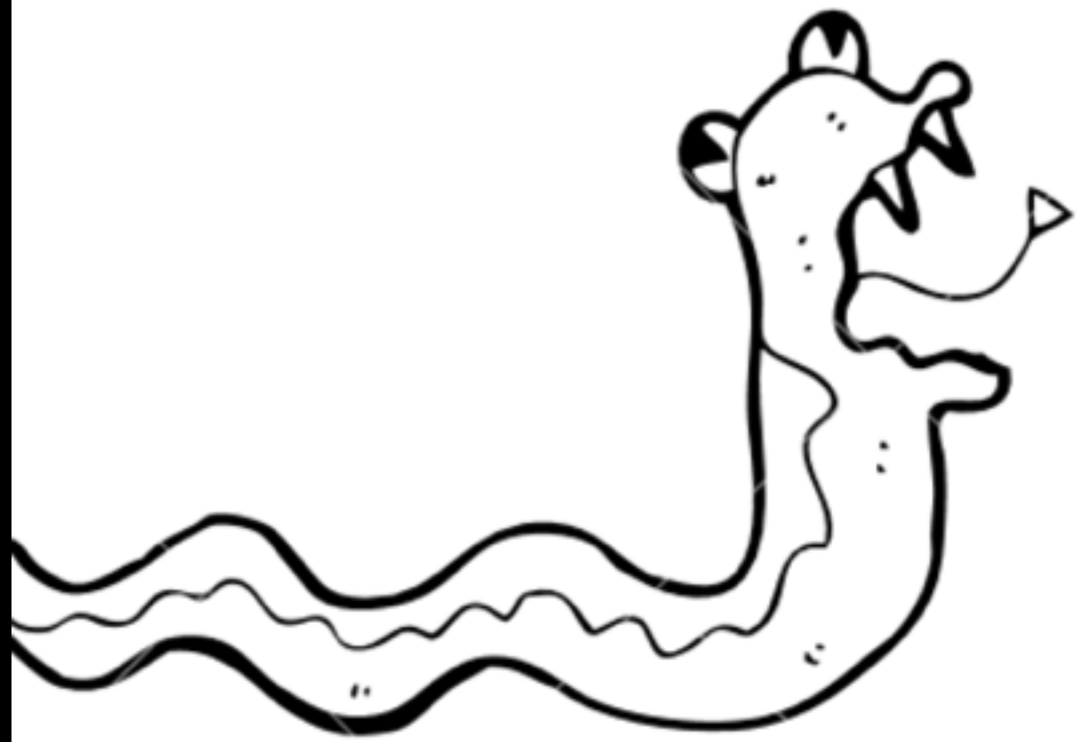
Rapid Prototyping

MICHAEL BERNSTEIN

CS 247

THANKS TO SCOTT KLEMMER AND MANEESH AGRAWALA

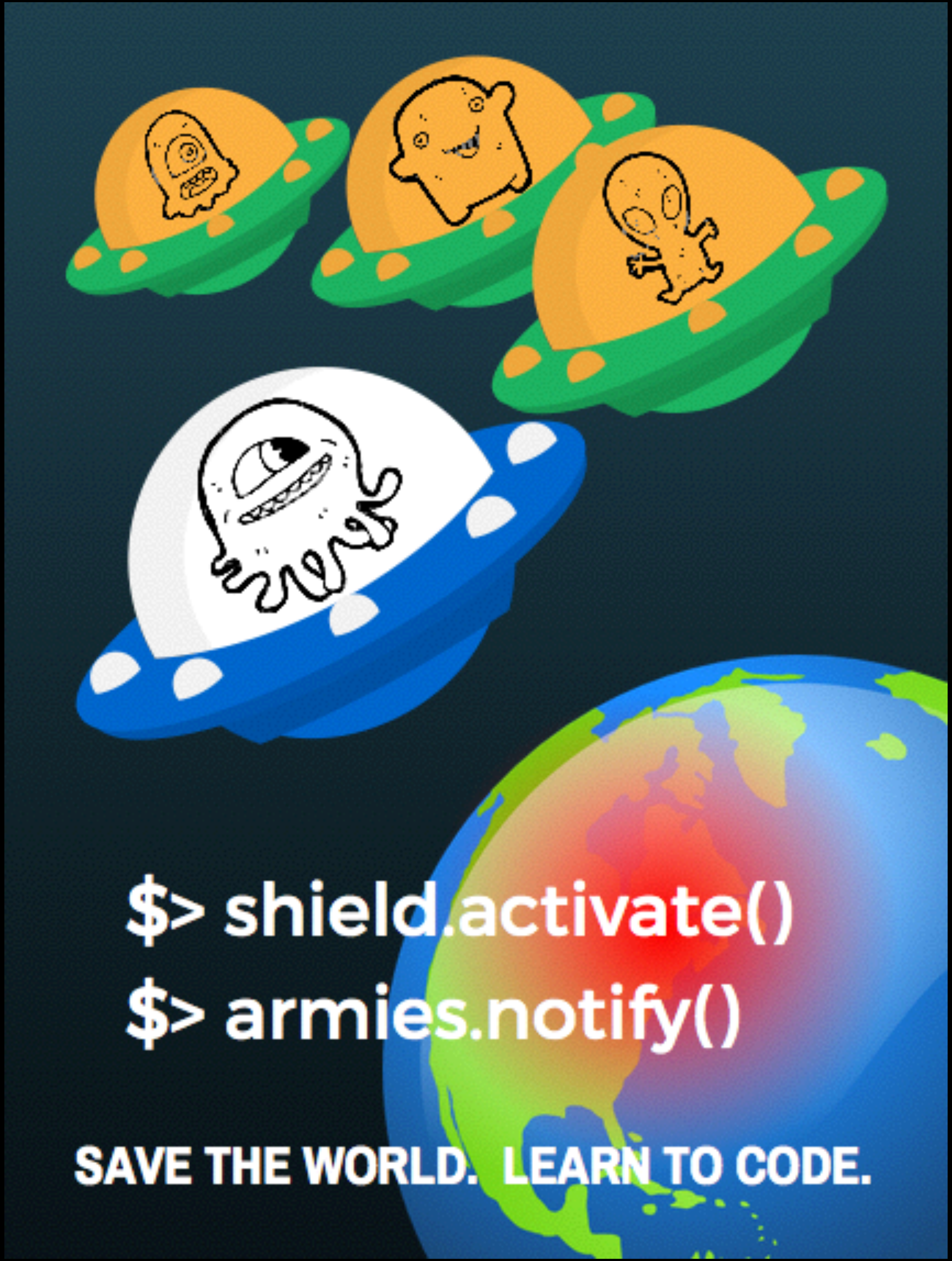
SNAKES AREN'T SO SCARY



Conquer your fears,

learn Python.

2/13, Gates 400



```
$> shield.activate()
$> armies.notify()
```

CODE:

A SYSTEMATIC COLLECTION OF LAWS OR REGULATIONS

DON'T BE A PART OF THE SYSTEM.

HACK THE SYSTEM.

Real Life Super Powers!

learn.code.gov

Some challenges
are greater than others.
Learn to code.
EVERLAST
MICROSOFT
Microsoft Windows
MICROSOFT
IBM

Recall: your prototype’s question

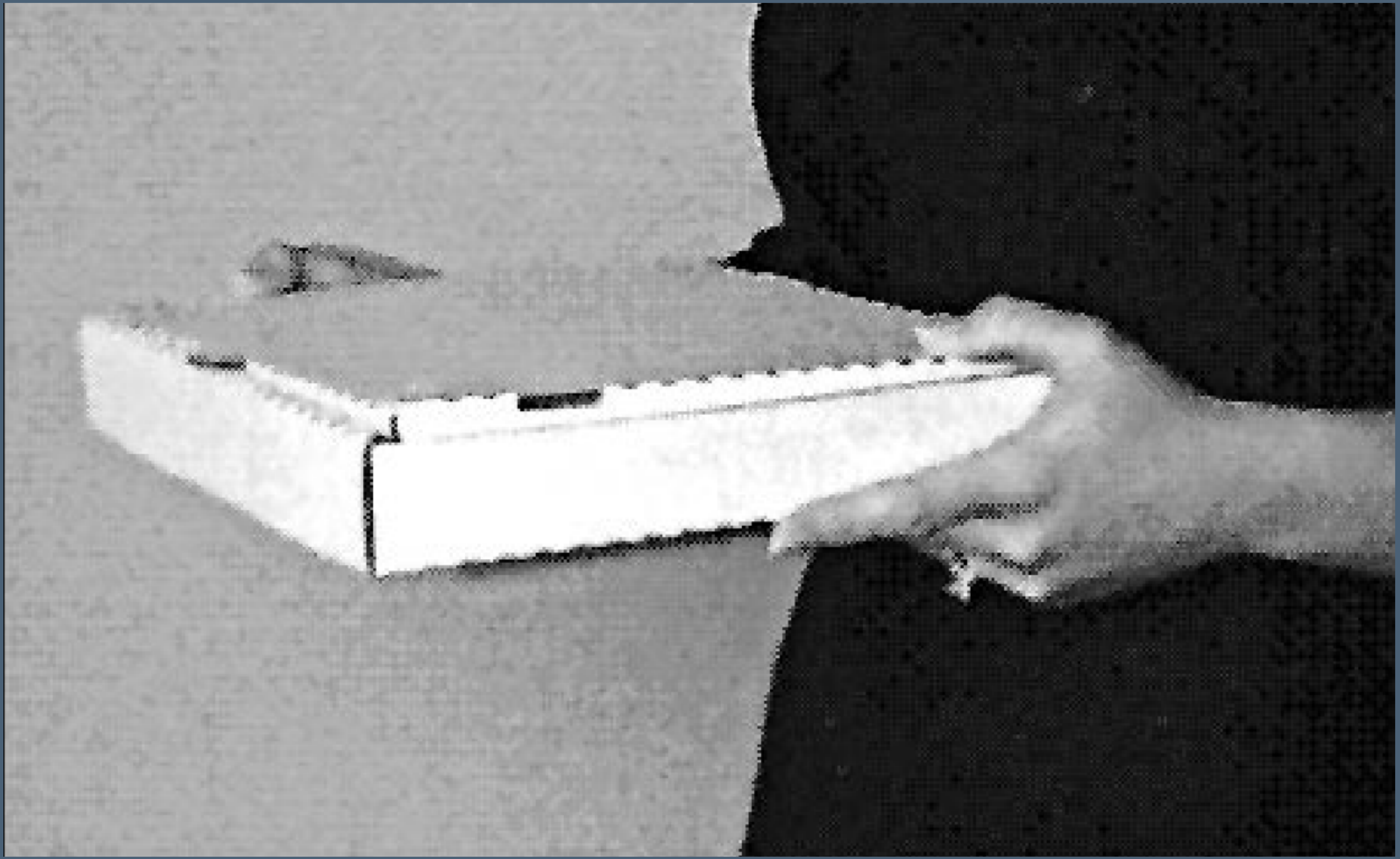
- All prototypes can only ask a single question.
- Ask yourself: what’s the core question you need to answer about your idea?
- Flesh out the elements relevant to that question, and get rid of or fake everything else.

Use the question to focus

- Typically you're trying to convey an experience, not a piece of software

- This means you can wizard-of-oz, fake, and cardboard your way through many aspects of a prototype

Prototype as rapidly as possible



Apple Architect’s Laptop
Houde & Hill 2000

Prototype with a deadline

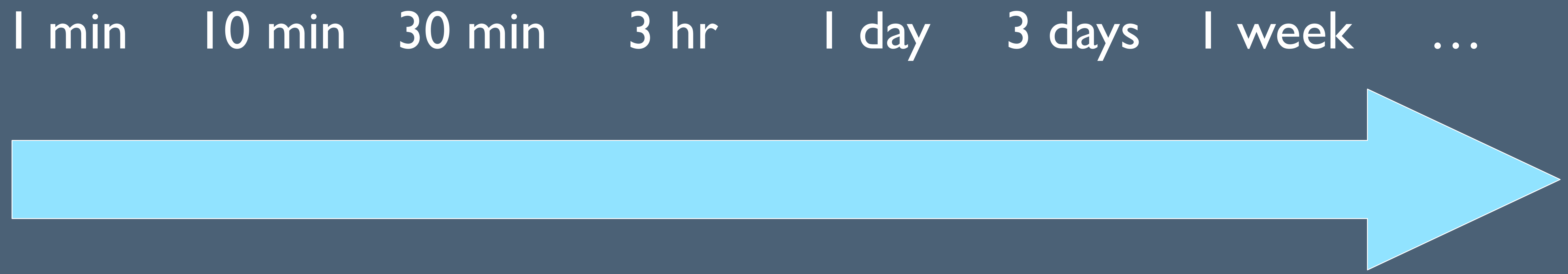
- CS 147: take a week (~10,000 minutes)
- CS 247: take fifteen minutes
- About 650 times faster.
- By necessity, less deep, but I promise you will learn nearly as much.

Today: rapid prototypes

- Today’s exercise will prove to you that you can produce compelling experience prototypes under a strict time limit.

- Never settle for slower prototypes.

Repeatedly prototype at increasing fidelity



1 min 10 min 30 min 3 hr 1 day 3 days 1 week …