

# Teaching Ruby to Kids

Or, how to do your part for world domination.

Sarah Mei

# Ohai

- sarahmei (twitter, github...)
- Software engineer @ Pivotal Labs
- Teaching is my hobby
  - Workshops for adults
  - Workshops for kids

# Why?

**Most programming instructors = FAIL.**

**Programming is becoming part of basic literacy.**

# Why should you teach?

- It's rewarding to spread your excitement about programming to someone else.

# Why should you teach?

- It's rewarding to spread your excitement about programming to someone else. (blah, blah, blah...)
- Teaching leads to learning

# Teaching: not rocket science

- You already have all the tools you need
- You already use all the techniques you need

# Agile teaching

## Look familiar?

- Set goals
- Form a plan, but expect to adapt
- Keep iterations short
- Listen to the customer
- Deploy continuously and listen to feedback

# Set goals

**Specific, immediate, measurable, achievable**



I want Aiden to grow  
up to be a developer.

I want Emma to get  
interested in  
programming.

I want Lily to be so  
excited that she  
explores things on her  
own after we're done.

I want Jack to enjoy programming so much that he looks forward to the next time.

I want Ada to be my  
little clone being so I  
can subcontract work  
out to her when she  
turns 10.

I want the kids to leave  
excited about what they  
can build, with enough  
tools to keep exploring.

# Form a plan

(but expect to adapt)

- Also known as, “what do I start with?”
- Keep your goals in mind.
- Tools:
  - Shoes
  - Hackety Hack
  - Small Ruby

# Notice a theme?

- Kids love anything visual
- They also love anything interactive
- irb: compelling for adults.
- (Maybe.)



# Your plan

- Install all the tools you might use on all the computers the kids have access to.
- Start small
- Use teh intarwebz

# Keep iterations short

- Your “lesson plan” should be a series of very small steps
- 15 minutes or less
- Visual change

# Listen to the customer

- Follow tangents!
- Don't stick to a plan because it's the plan
- Don't worry about "finishing"
- Look for teachable moments
- Look for signs they've turned off

# Deploy continuously

- Teaching is a learned skill.
- Take all the opportunities you can to teach.
  - talks at your local meetup
  - pair programming
  - summer camps, etc., need volunteers
  - National Lab Day
  - In SF, I always need teachers for introductory workshops

**Expect some things you  
try to fall flat.**

**Keep at it.**

# To sum up

- You **should** teach.
- You **can** teach.
- Agile is for more than just development.
- Practice.

# Questions