Getting started...

- Pre-workshop survey.
- Setup:
  - Projector
  - Wifi
  - Power outlets
  - Text editor
  - Python
  - *Windows difficulties*
- RCC cluster access:
  - Request RCC account
  - Yubikeys

- Coffee, snacks.
- Breaks, stretches.
- Etherpad.
  - *Introductions*
- Introduce yourself to your neighbors.
- Ask us questions
  - *Keyboard shortcuts*
- Pace, experience levels.
- Post-workshop survey.
- Feedback.
There is no best tool—use whatever works for you.
Workshop schedule

Day 1

• The Unix shell (a.m.)  
  — Yuxing Peng

• Programming in Python (p.m.)  
  — Hossein Pourreza

Day 2

• High-performance computing (HPC) using Python (a.m.)  
  — Jonathan Skone

• Version control using Git (p.m.)  
  — Peter Carbonetto
The Software Carpentry approach

1. Learning through “live coding.”
   - Especially learning from our mistakes!
2. Hands on—*using your own computer.*
3. Lateral knowledge transfer.
4. Collaborative note-taking (Etherpad).
Workshop aims

1. Improve comfort level in basic computing skills.

2. Introduce tools & terminology so that you can explore more on your own.

3. Help develop an effective computing environment.
   - Including a high-performance computing (HPC) environment.

4. Keep you engaged!
When you get stuck

1. Talk to your neighbors.

2. Ask for help (Yellow Stickies).

3. Pair up with your neighbor, and wait for the break.

4. Google the name of the program plus a few words related to what you want to do. Follow any instructions.
Other recommended resources

- **Software Carpentry**  
  http://software-carpentry.org/lessons

- **Data Carpentry**  
  http://www.datacarpentry.org/lessons

- **Practical Computing for Biologists**  
  http://practicalcomputing.org

- **Effective Computation in Physics**  
  http://physics.codes