Announcements

- Quiz 1 scores will be posted eventually, but you already know what you'll get
  - 0/3: Please talk to your TA for advice on how to proceed
  - 1/3: Make sure to spend time understanding all lab & discussion questions
  - 2/3: Practice is extremely helpful in learning how to solve CS problems
- Guerrilla Section 1 on higher-order functions is on Saturday 1/31 in 271 Soda
  - Optional discussion to promote mastery of core concepts (prepares you for midterms)
  - 2pm - 4pm is the vanguard section (you commit to helping teach the main section)
  - 4pm - 6pm is the main section
  - Please do not bring questions about homework or projects to guerrilla sections
- Small-group tutoring begins next week! Apply online by Sunday if you want a (free) tutor
  - Homework 2 (which is small) is due Monday 2/2 at 11:59pm
  - Project 1 (which is BIG) is due Thursday 2/5 at 11:59pm
  - Midterm 1 on Monday 2/9 7pm-9pm
    - Conflict? Fill out the conflict form today! http://goo.gl/2P55Kp

Environments Enable Higher-Order Functions

Functions are first-class: Functions are values in our programming language

Higher-order function: A function that takes a function as an argument value or
A function that returns a function as a return value

Environment diagrams describe how higher-order functions work!

(Demo)

Environments for Nested Definitions

Names can be Bound to Functional Arguments

```
def apply_twice(f, x):
    return f(f(x))

def square(x):
    return x * x

result = apply_twice(square, 2)
```

Environment Diagrams for Nested Def Statements

```
def make_adder(n):
    def adder(k):
        return k + n
    return adder

add_three = make_adder(3)
add_three(4)
```

How to Draw an Environment Diagram

When a function is defined:
Create a function value:   func <name>(<formal parameters>) [parent=<label>]
Its parent is the current frame.

```
    fi: make_adder    func adder(k) [parent=fi]
```

Bind <name> to the function value in the current frame

When a function is called:
1. Add a local frame, titled with the <name> of the function being called.
2. Copy the parent of the function to the local frame: [parent=<label>]
3. Bind the formal parameters to the arguments in the local frame.
4. Execute the body of the function in the environment that starts with the local frame.
Local Names

Local Names are not Visible to Other (Non-Nested) Functions

- An environment is a sequence of frames.
- The environment created by calling a top-level function (no def within def) consists of one local frame, followed by the global frame.

Interactive Diagram

Function Composition

The Environment Diagram for Function Composition

Lambda Expressions

Lambda Expressions

Lambda Expressions Versus Def Statements

Lambda expressions are not common in Python, but important in general.

Lambda expressions in Python cannot contain statements at all!
Function Currying

```python
def make_adder(n):
    return lambda k: n + k

>>> make_adder(2)(3)
5
```

There's a general relationship between these functions

Curry: Transform a multi-argument function into a single-argument, higher-order function