One-on-One Sessions

- Still ongoing: 1/2-hour one-on-one sessions
  - To help prepare you for the assignment
  - Primarily for students with little experience
- There are still some spots available
  - Sign up for a slot in CMS
- Will keep running after September 17
  - Will open additional slots after the due date
  - Will help students revise Assignment 1

A Motivating Example

Function Definition

```python
def foo(a,b):
    """Do something"
    Param a: number
    Param b: number"
    x = a
    y = b
    return x*y+y
```

Function Call

```python
>>> x = 2
>>> foo(3,4) x ?
```

What is in the box?

A: 2  
B: 3  
C: 16 
D: Nothing! 
E: I do not know

How Do Functions Work?

- Function Frame: Representation of function call
- A conceptual model of Python

Example: to_centigrade(50.0)

1. Draw a frame for the call
2. Assign the argument value to the parameter (in frame)
3. Execute the function body
   - Look for variables in the frame
   - If not there, look for global variables with that name
4. Erase the frame for the call

Example: to_centigrade(50.0)

1. Draw a frame for the call
2. Assign the argument value to the parameter (in frame)
3. Execute the function body
   - Look for variables in the frame
   - If not there, look for global variables with that name
4. Erase the frame for the call

Text (Section 3.10) vs. Class

**Textbook**

def to_centigrade(x):
    return 5*(x-32)/9.0

to_centigrade x -> 80.0

**This Class**

def to_centigrade(x):
    return 5*(x-32)/9.0

Example:

to_centigrade(50.0)

Initial call frame (before exec body)

1. Draw a frame for the call
2. Assign the argument value to the parameter (in frame)
3. Execute the function body
   - Look for variables in the frame
   - If not there, look for global variables with that name
4. Erase the frame for the call

Example:

to_centigrade(50.0)

Executing the return statement

1. Draw a frame for the call
2. Assign the argument value to the parameter (in frame)
3. Execute the function body
   - Look for variables in the frame
   - If not there, look for global variables with that name
4. Erase the frame for the call

Return statement creates a special variable for result
Example: to_centigrade(50.0)

1. Draw a frame for the call
2. Assign the argument value to the parameter (in frame)
3. Execute the function body
   - Look for variables in the frame
   - If not there, look for global variables with that name
4. Erase the frame for the call

```
def to_centigrade(x):
    return 5*(x-32)/9.0
```

But don’t actually erase on an exam

---

Call Frames vs. Global Variables

The specification is a lie:
```
def swap(a,b):
    """Swap global a & b""
    tmp = a
    a = b
    b = tmp
```

But don’t actually erase on an exam

```
>>> a = 1
>>> b = 2
>>> swap(a,b)
```

---

Visualizing Frames: The Python Tutor

- Global Space
- Call Frame

Function Access to Global Space

- All function definitions are in some module
- Call can access global space for that module
  - math.cos: global for math
  - temperature.to_centigrade uses global for temperature
- But cannot change values
  - Assignment to a global makes a new local variable!
  - Why we limit to constants

Exercise Time

```
def foo(a,b):
    """Do something"
    Param x: a number
    Param y: a number""
    x = a
    y = b
    return x*y+y
```

What does the frame look like at the start?