Lecture 6

Visualizing Functions
Announcements for this Lecture

Last Call

- Quiz: About the Course
- Take it by tomorrow
- Also remember survey

Assignment 1

- Assignment 1 is live
  - Posted on web page
  - Due Thur, Sep. 17th
  - Due in place of Lab 4
- Lab 3 will help a lot
  - Testing is a major part
  - Try to finish it first
  - But start this Saturday!
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
A1: The Module urllib2

- Module urllib2 is used to read web pages
  - Function urlopen creates a url object
    - `u = urllib2.urlopen('http://www.cornell.edu')`

- url has a method called read()
  - Returns contents of web page
  - **Usage**: `s = u.read()`  # s is a string
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)
9
```

What is in the box?
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
```

```python
>>> foo(3, 4)
```

What is in the box?

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

9/10/15  Visualizing Functions  6
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)
```

What is in the box?

A: 2  CORRECT
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

Draw parameters as variables (named boxes)

- Number of statement in the function body to execute next
- Starts with 1

function name  |  instruction counter
--- | ---
parameters  |  local variables (later in lecture)
Textbook vs. This Class

**Textbook**

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

def to_centigrade(x):
    x -> 50.0

**This Class**

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

def to_centigrade(x):
    x 50.0

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

**Call:** to_centigrade(50.0)
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

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

Initial call frame (before exec body):

- `x`: 50.0

Next line to execute: `return 5*(x-32)/9.0`
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

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

Executing the return statement

Return statement creates a special variable for result

`to_centigrade`

x 50.0  RETURN 10.0

9/10/15 Visualizing Functions 11
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

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

Executing the return statement

```
50.0 10.0
```

The return terminates; no next line to execute
Example: \texttt{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

```python
def to_centigrade(x):
    return 5\,(x-32)/9.0
```

But don’t actually erase on an exam
The specification is a **lie**:

```
def swap(a,b):
    """Swap global a & b""
    tmp = a
    a = b
    b = tmp
```

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

---

**Global Variables**

- a: 1
- b: 2

**Call Frame**

- swap: 1
- a: 1
- b: 2
Call Frames vs. Global Variables

The specification is a lie:

```python
def swap(a,b):
    """Swap global a & b"""
    tmp = a
    a = b
    b = tmp
```

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

Global Variables

Call Frame

Visualizing Functions
Call Frames vs. Global Variables

The specification is a lie:

```python
def swap(a,b):
    """Swap global a & b"""
    tmp = a
    a = b
    b = tmp
```

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

Global Variables

Call Frame
Call Frames vs. Global Variables

The specification is a **lie**:

```python
def swap(a, b):
    """Swap global a & b"""
    tmp = a
    a = b
    b = tmp
```

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

- **Global Variables**
  - `a` = 1
  - `b` = 2

- **Call Frame**
  - `a` = 2
  - `b` = 1
  - `tmp` = 1
The specification is a lie:

```python
def swap(a, b):
    """Swap global a & b""
    tmp = a
    a = b
    b = tmp
```

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

Global Variables

Call Frame

ERASE THE FRAME
Visualizing Frames: The Python Tutor

```python
1 def max(x, y):
2     if x > y:
3         return x
4     return y
5
6 a = 1
7 b = 2
8 max(a, b)
```

Frames

```
Global frame

max
a 1
b 2
```

Objects

```
max
x 1
y 2
```

9/10/15 Visualizing Functions 19
Visualizing Frames: The Python Tutor

![Diagram showing Python code and frames]

```
def max(x, y):
    if x > y:
        return x
    return y
```

Global Space

Call Frame
Visualizing Frames: The Python Tutor

Variables from second lecture go in here

Global Space

Call Frame

def max(x, y):
    if x > y:
        return x
    return y

a = 1
b = 2
max(a, b)
Visualizing Frames: The Python Tutor

```
def max(x, y):
    if x > y:
        return x
    return y
```

```
a = 1
b = 2
max(a, b)
```

Frames

Objects

Global frame

max

```
x 1
y 2
```
Visualizing Frames: The Python Tutor

```
def max(x, y):
    if x > y:
        return x
    return y

a = 1
b = 2
max(a, b)
```

Line number marked here (sort-of)

Missing line numbers!
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

```python
# globals.py

"""Show how globals work""

a = 4  # global space

def show_a():
    print(a)  # shows global
```

```python
show_a()  # shows global
```

Global Space
(for globals.py)

```
a  4
```

```
show_a  1
```
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

```python
# globals.py
"""Show how globals work""

a = 4  # global space

def change_a():
    a = 3.5  # local variable
```

```python
# globals.py
"""Show how globals work""

a = 4  # global space

def change_a():
    a = 3.5  # local variable
```
Exercise Time

Function Definition

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

Function Call

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

What does the frame look like at the start?
Which One is Closest to Your Answer?

A:

```
foo 0
a 3 b 4
```

B:

```
foo 1
a 3 b 4
```

C:

```
foo 1
a 3 b 4
x 3
```

D:

```
foo 1
a 3 b 4
x y
```
Which One is Closest to Your Answer?

A:  
```
+---+---+
| foo | 0 |
+---+---+
| a 3 | b 4 |
+---+---+
```

B:  
```
+---+---+
| foo | 1 |
+---+---+
| a 3 | b 4 |
+---+---+
```

C:  
```
+---+---+
| foo |   |
+---+---+
| a 3 | x 3 |
+---+---+
```

D:  
```
+---+---+
| foo |   |
+---+---+
| x 3 | y 4 |
+---+---+
```

E:  
```
(ツ)_/\ 
```
Exercise Time

Function Definition

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

Function Call

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

B:

```
+---+---+---+
| foo| a | b |
+---+---+---+
|   | 3 | 4 |
+---+---+---+
1   2   3
```

9/10/15
**Function Definition**

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

**Function Call**

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

**B:**

What is the **next step?**
Which One is Closest to Your Answer?

A:  
```
<table>
<thead>
<tr>
<th>foo</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>3</td>
</tr>
<tr>
<td>b</td>
<td>4</td>
</tr>
<tr>
<td>x</td>
<td>3</td>
</tr>
</tbody>
</table>
```

B:  
```
<table>
<thead>
<tr>
<th>foo</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>3</td>
</tr>
<tr>
<td>b</td>
<td>4</td>
</tr>
<tr>
<td>x</td>
<td>3</td>
</tr>
</tbody>
</table>
```

C:  
```
<table>
<thead>
<tr>
<th>foo</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>3</td>
</tr>
<tr>
<td>b</td>
<td>4</td>
</tr>
<tr>
<td>x</td>
<td>3</td>
</tr>
</tbody>
</table>
```

D:  
```
<table>
<thead>
<tr>
<th>foo</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>3</td>
</tr>
<tr>
<td>b</td>
<td>4</td>
</tr>
<tr>
<td>x</td>
<td>3</td>
</tr>
<tr>
<td>y</td>
<td></td>
</tr>
</tbody>
</table>
```
Exercise Time

Function Definition

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

Function Call

```
>>> x = foo(3, 4)
```

C:
Exercise Time

Function Definition

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

Function Call

>>> x = foo(3, 4)

C:

What is the next step?
Which One is Closest to Your Answer?

A:

```
+-----+-----+-----+
| foo |     | 3   |
| a   | 3   | b   |
| x   | 3   | y   |
|     |     | 4   |
+-----+-----+-----+
```

B:

```
+-----+-----+-----+
| foo |     | 3   |
| a   | 3   | b   |
| x   | 3   | y   |
|     |     | 4   |
+-----+-----+-----+
```

C:

```
+-----+-----+-----+
| foo |     |     |
| a   | 3   | b   |
| x   | 3   | y   |
|     |     | 4   |
|     |     | 16  |
+-----+-----+-----+
```

D:

```
+-----+-----+-----+
| foo |     | 3   |
| a   | 3   | b   |
| x   | 3   | y   |
|     |     | 4   |
|     |     |     |
+-----+-----+-----+
```

RETURN

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

>>> x = foo(3, 4)

A:
Exercise Time

Function Definition

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

Function Call

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

A:

What is the next step?
Which One is Closest to Your Answer?

A:

```
foo
3
RETURN
16
```

B:

```
foo
3
a
3
b
4
x
3
y
4
RETURN
16
```

C:

```
foo
3
a
3
b
4
x
3
y
4
RETURN
16
```

D:

```
foo
3
a
3
b
4
x
3
y
4
RETURN
16
```

**ERASE THE FRAME**
**Exercise Time**

## Function Definition

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

## Function Call

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

```
C:
```

```
<table>
<thead>
<tr>
<th>foo</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
</tr>
<tr>
<td>b</td>
</tr>
<tr>
<td>x</td>
</tr>
<tr>
<td>y</td>
</tr>
<tr>
<td>RETURN</td>
</tr>
</tbody>
</table>
```
def foo(a, b):
    """Do something
    Param x: a number
    Param y: a number"
    x = a
    y = b
    return x*y+y

>>> x = foo(3,4)

C:

What is the next step?
Which One is Closest to Your Answer?

A: foo
   RETURN 16

B: ERASE THE FRAME

C: foo
   x 16

D: ERASE THE FRAME
   x 16
Exercise Time

**Function Definition**

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

**Function Call**

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

D:

9/10/15

Visualizing Functions
Exercise Time

### Function Definition

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

### Function Call

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

D:

Variable in global space

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x = a</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>y = b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>return x*y+y</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>