61A Lecture 3

Friday, September 6
Announcements

• Homework 1 is due next Tuesday at 5pm (no email when you submit).
  • Homework is graded for effort.

• Take-home quiz released next Wednesday 9/11 at 1pm, due Thursday 9/12 at 11:59pm.
  • 3 points, graded for correctness.
  • Similar in format to a homework assignment.
  • If you receive 0/3, you will need to talk to the course staff or be dropped.

  Open-computer: You can use the Python interpreter, watch course videos, and read the online text (http://composingprograms.com).

  No external resources: Please don't search for answers, talk to your classmates, etc.

• Project 1 posted this Friday, due Thursday 9/19 at 11:59pm.
  • Demo during next lecture
Multiple Environments
Life Cycle of a User-Defined Function

Def statement: 
- Name: `square(x):`
- Body: `return mul(x, x)`

What happens?
- A new function is created!
- Name bound to that function in the current frame

Call expression:
- Operator: `square(2+2)`
- Argument: 4
- Signature: 4
- Function (value of operator) called on arguments (values of operands)

Calling/Applying:
- Argument: 4
- Signature: 16
- Return value
- A new frame is created!
- Parameters bound to arguments
- Body is executed in that new environment
Multiple Environments in One Diagram!

```python
from operator import mul

def square(x):
    return mul(x, x)

square(square(3))
```

Example: [http://goo.gl/XVtEms](http://goo.gl/XVtEms)
Multiple Environments in One Diagram!

```python
1 from operator import mul
2 def square(x):
3     return mul(x, x)
4     square(square(3))
```

Example: [http://goo.gl/XVtEms](http://goo.gl/XVtEms)
An environment is a sequence of frames.

- The global frame alone
- A local, then the global frame

Example: [http://goo.gl/XVtEms](http://goo.gl/XVtEms)
Names Have No Meaning Without Environments

Every expression is evaluated in the context of an environment.

A name evaluates to the value bound to that name in the earliest frame of the current environment in which that name is found.

An environment is a sequence of frames.

- The global frame alone
- A local, then the global frame

Example: [http://goo.gl/XVtEms](http://goo.gl/XVtEms)
Miscellaneous Python Features

Operators
Multiple Return Values
Docstrings
Doctests
Default Arguments

(Demo)
Conditional Statements
A *statement* is executed by the interpreter to perform an action.

**Compound statements:**

- The first header determines a statement’s type.
- The header of a clause “controls” the suite that follows.
- `def` statements are compound statements.
Compound Statements

Compound statements:

A suite is a sequence of statements

 Suite

To “execute” a suite means to execute its sequence of statements, in order

Execution Rule for a sequence of statements:

• Execute the first statement

• Unless directed otherwise, execute the rest
Conditional Statements

(Demo)

```python
def absolute_value(x):
    """Return the absolute value of x."""
    if x < 0:
        return -x
    elif x == 0:
        return 0
    else:
        return x
```

1 statement, 3 clauses, 3 headers, 3 suites

**Execution rule for conditional statements:**

1. Evaluate the header's expression.
   
2. If it is a true value, execute the suite & skip the remaining clauses.

**Syntax Tips**

1. Always starts with "if" clause.
2. Zero or more "elif" clauses.
3. Zero or one "else" clause, always at the end.
Boolean Contexts

```python
def absolute_value(x):
    """Return the absolute value of x."""
    if x < 0:
        return -x
    elif x == 0:
        return 0
    else:
        return x
```

George Boole
def absolute_value(x):
    """Return the absolute value of x."""
    if x < 0:
        return -x
    elif x == 0:
        return 0
    else:
        return x

Boolean Contexts

False values in Python: False, 0, '', None  (more to come)

True values in Python: Anything else (True)

Read Section 1.5.4!

George Boole
Iteration
While Statements

George Boole

Execution rule for while statements:

1. Evaluate the header’s expression.
2. If it is a true value, execute the (whole) suite, then return to step 1.

Example: [http://goo.gl/0d2cJF](http://goo.gl/0d2cJF)
def choose(total, selection):
    """Return the number of ways to choose SELECTION items from TOTAL.

    choose(n, k) is typically defined in math as:  n! / (n-k)! / k!
    >>> choose(5, 2)
    10
    >>> choose(20, 6)
    38760
    ""
    ways = 1
    selected = 0
    while selected < selection:
        selected = selected + 1
        ways, total = ways * total // selected, total - 1
    return ways

Example: http://goo.gl/38ch3o