Announcements

- Homework 1 is due Wednesday 1/28 @ 11:59pm
  - Homework party on Tuesday 1/27 5–6:30pm in 2050 VLSB
  - Please declare your partner on ok.cs61a.org
- Take-home Quiz 1 released Wednesday 1/28 is due Thursday 1/29 @ 11:59pm
  - 3 points, similar in format to homework, but graded for correctness
  - If you score 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, etc.
  - Closed-help: Please don't talk to your classmates, search for answers, etc.
- Project 1 due Thursday 2/5 @ 11:59pm
- Midterm 1 on Monday 2/9 @ 7pm
Multiple Environments
Life Cycle of a User-Defined Function

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

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

**Call expression:**
- **Operand**: 2+2
- **Operator & operands evaluated**: 4
- **Function (value of operator) called on arguments**: 4

**Calling/Applying:**
- **Argument**: 2+2
- **Signature**: Square(x)
- **Body is executed in that new environment**: 16
- **Return value**: 16

A new frame is created!
Parameters bound to arguments
Multiple Environments in One Diagram!

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

Interactive Diagram
Multiple Environments in One Diagram!

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

Interactive Diagram
Multiple Environments in One Diagram!

An environment is a sequence of frames.

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

Interactive Diagram
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

Interactive Diagram
Names Have Different Meanings in Different Environments

A call expression and the body of the function being called are evaluated in different environments.

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

Interactive Diagram

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.
Miscellaneous Python Features

Operators
Multiple Return Values
Docstrings
Doctests
Default Arguments

(Demo)
Conditional Statements
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:

<header>:
    <statement>
    <statement>
    ...

<separating header>:
    <statement>
    <statement>
    ...

A suite is a sequence of statements

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)

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

Execution Rule for Conditional Statements:

Each clause is considered in order.

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.

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

Reading: http://composingprograms.com/pages/15-control.html#conditional-statements
Iteration
While Statements

(Demo)

1. Evaluate the header’s expression.

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

#### 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.