

Announcements

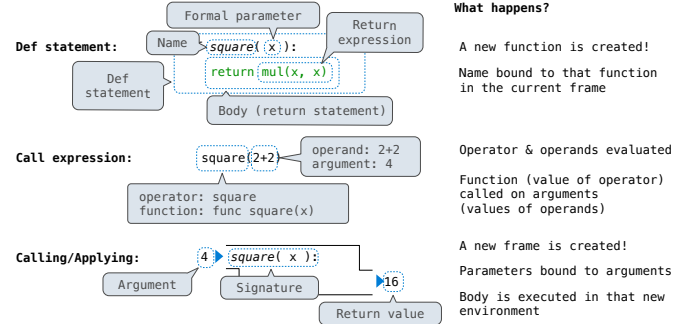
- Homework 1 is due next Tuesday at 5pm (no email when you submit).
 - Homework is graded for effort.
- Take-home quiz released next Wednesday 9/11 at 1pm, due Thursday 9/12 at 11:59pm.
 - 3 points, graded for correctness.
 - Similar in format to a homework assignment.
 - If you receive 0/3, you will 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

61A Lecture 3

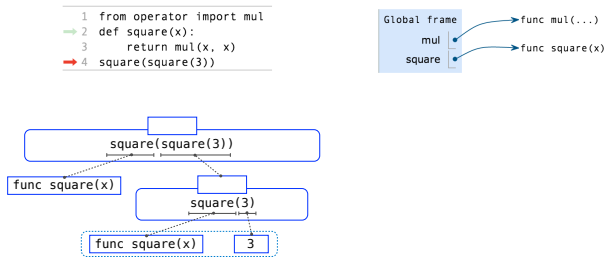
Friday, September 6

Multiple Environments

Life Cycle of a User-Defined Function

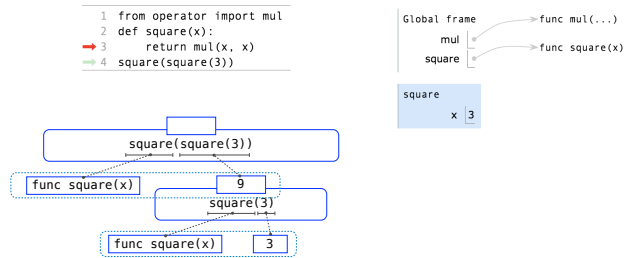


Multiple Environments in One Diagram!



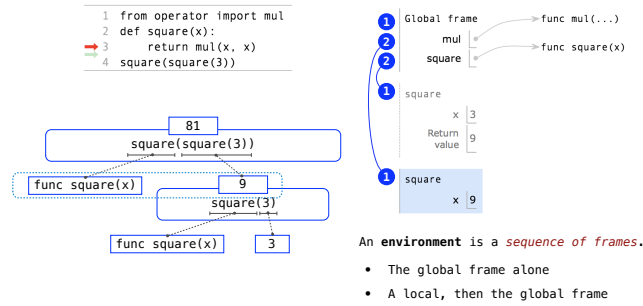
Example: <http://zooq.sl/2018ms>

Multiple Environments in One Diagram!

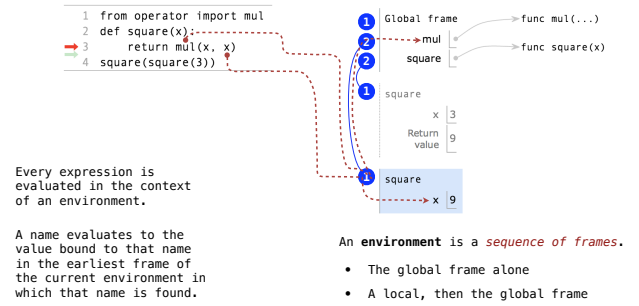


Example: <http://zooq.sl/2018ms>

Multiple Environments in One Diagram!



Names Have No Meaning Without Environments



Miscellaneous Python Features

Operators
 Multiple Return Values
 Docstrings
 Doctests
 Default Arguments

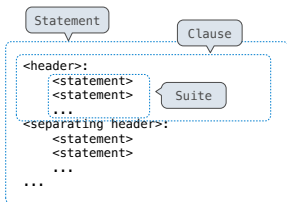
(Demo)

Conditional Statements

Statements

A **statement** is executed by the interpreter to perform an action

Compound statements:



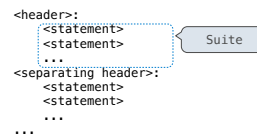
The first header determines a statement's type

The header of a clause "controls" the suite that follows

def statements are compound statements

Compound Statements

Compound statements:



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)

1 statement,
3 clauses,
3 headers,
3 suites

```
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.

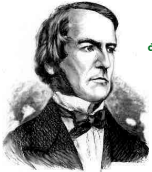
Boolean Contexts

```
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

Boolean Contexts



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
```

Two 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://composingprogram.com/pages/15-control.html#conditional-statements>

Iteration

While Statements

(Demo)



George Boole

```
1 i, total = 0, 0
2 while i < 3:
3     i = i + 1
4     total = total + i
```

Global frame
i ✖ ✖ ✖ 3
total ✖ ✖ ✖ 6

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://zoo.gsl/g2k1f>

Discussion Question

Complete the following definition by placing an expression in _____.

```
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 - 1
    return ways
```

$$\frac{n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot (n-k+1)}{k \cdot (k-1) \cdot (k-2) \cdot \dots \cdot 2 \cdot 1}$$

Example: <http://zoo.gsl/38ch3g>