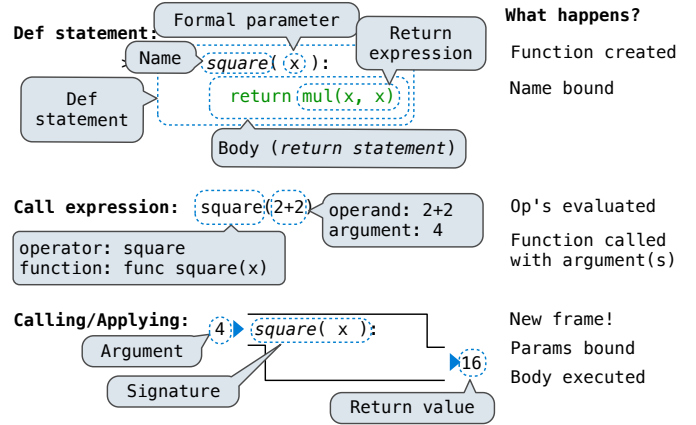


# 61A Lecture 3

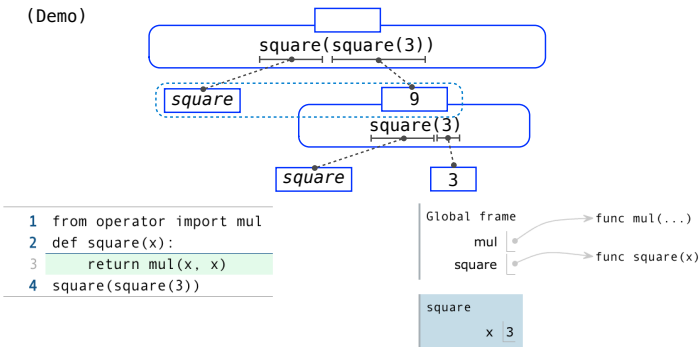
Wednesday, August 29

## Life Cycle of a User-Defined Function



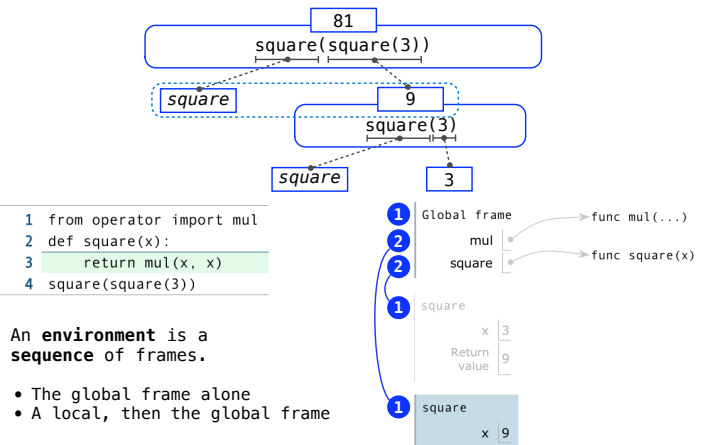
## Multiple Environments in One Diagram!

(Demo)



Example: <http://goo.gl/668fU>

## Multiple Environments in One Diagram!

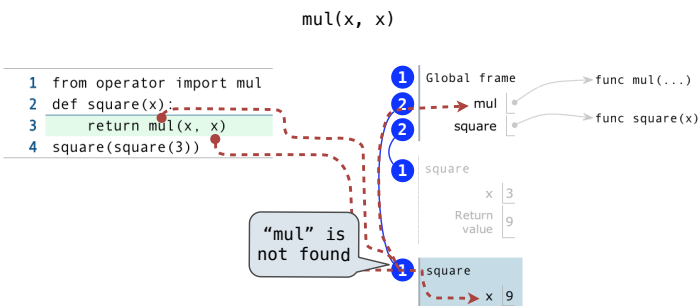


Example: <http://goo.gl/668fU>

## Names Have No Meaning Without Environments

Every expression is evaluated in the context of an environment.

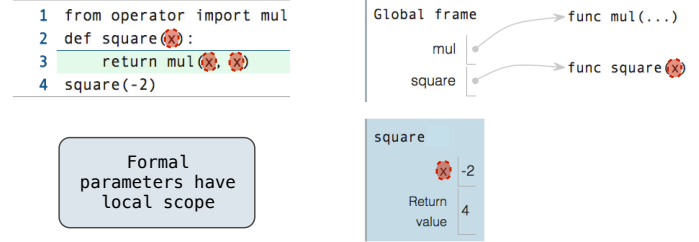
A name evaluates to the value bound to that name in the earliest frame of the current environment in which that name is found.



Example: <http://goo.gl/668fU>

## Formal Parameters

`def square(x):`  
`return mul(x, x)` vs `def square(y):`  
`return mul(y, y)`



(Demo)

Example: <http://goo.gl/0apJa>

## Python Feature Demonstration

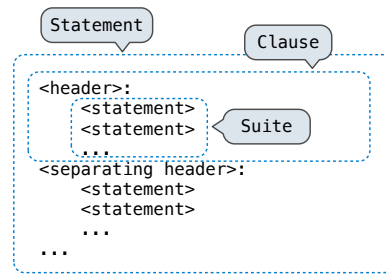
- Operators
- Multiple Return Values
- Docstrings
- Doctests
- Default Arguments
- Statements

7

## Statements

A statement is executed by the interpret to perform an action

### Compound statements:



The first header determines a statement's type

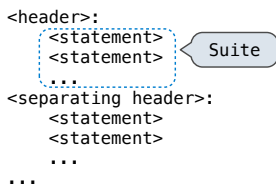
The header of a clause "controls" the suite that follows

def statements are compound statements

8

## 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:

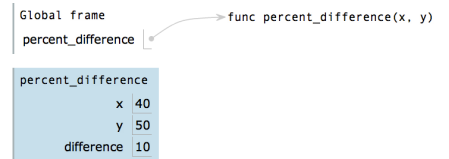
- Execute the first
- Unless directed otherwise, execute the rest

9

## Local Assignment

```

1 def percent_difference(x, y):
2     difference = abs(x-y)
3     return 100 * difference / x
4 diff = percent_difference(40, 50)
    
```



### Execution rule for assignment statements:

1. Evaluate all expressions right of =, from left to right.
2. Bind the names on the left the resulting values in the **first frame** of the current environment.

Example: <http://goo.gl/wcE71>

10

## Conditional Statements

```

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:

- 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.

11

## 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

12

## 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!**

## Iteration



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

Global frame  
i ~~xxx~~ 3  
total ~~xxx~~ 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.