

Environments

Announcements

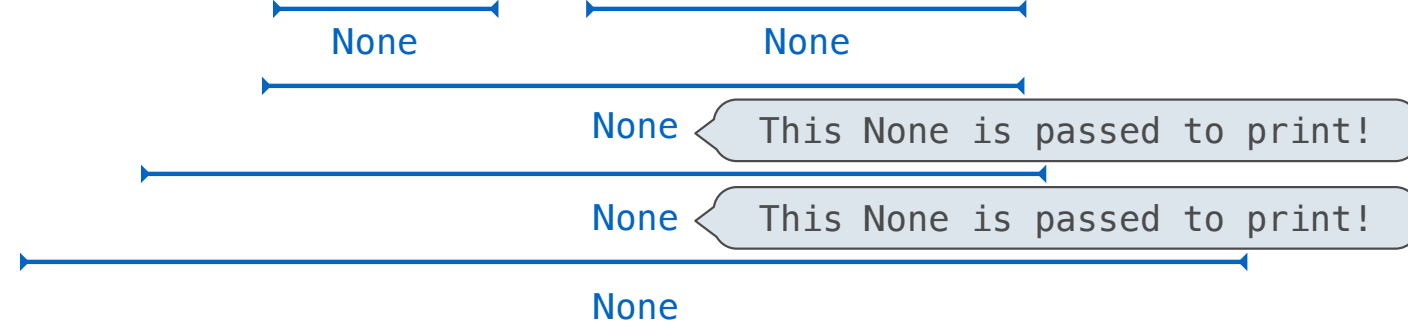
Print and None Review

Fall 2022 CS 61A Midterm 1, Question 1

What does the long expression print?

```
s = "Knock"
```

```
print(print(print(s, s) or print("Who's There?")), "Who?")
```



Knock Knock

Who's There?

None

None Who?

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

To evaluate the expression **<left> or <right>**:

1. Evaluate the subexpression **<left>**.
2. If the result is a true value **v**, then the expression evaluates to **v**.
3. Otherwise, the expression evaluates to the value of the subexpression **<right>**.

Environments for Higher-Order Functions

Names can be Bound to Functional Arguments

```
1 def apply_twice(f, x):  
2     return f(f(x))  
3  
→ 4 def square(x):  
5     return x * x  
6  
→ 7 result = apply_twice(square, 2)
```

Global frame
apply_twice
square

func apply_twice(f, x) [parent=Global]

func square(x) [parent=Global]

Applying a user-defined function:

- Create a new frame
- Bind formal parameters (f & x) to arguments
- Execute the body:
return f(f(x))

```
→ 1 def apply_twice(f, x):  
→ 2     return f(f(x))  
3  
4 def square(x):  
5     return x * x  
6  
7 result = apply_twice(square, 2)
```

2 Global frame

1 f1: apply_twice [parent=Global]

apply_twice
square

func apply_twice(f, x) [parent=Global]

func square(x) [parent=Global]

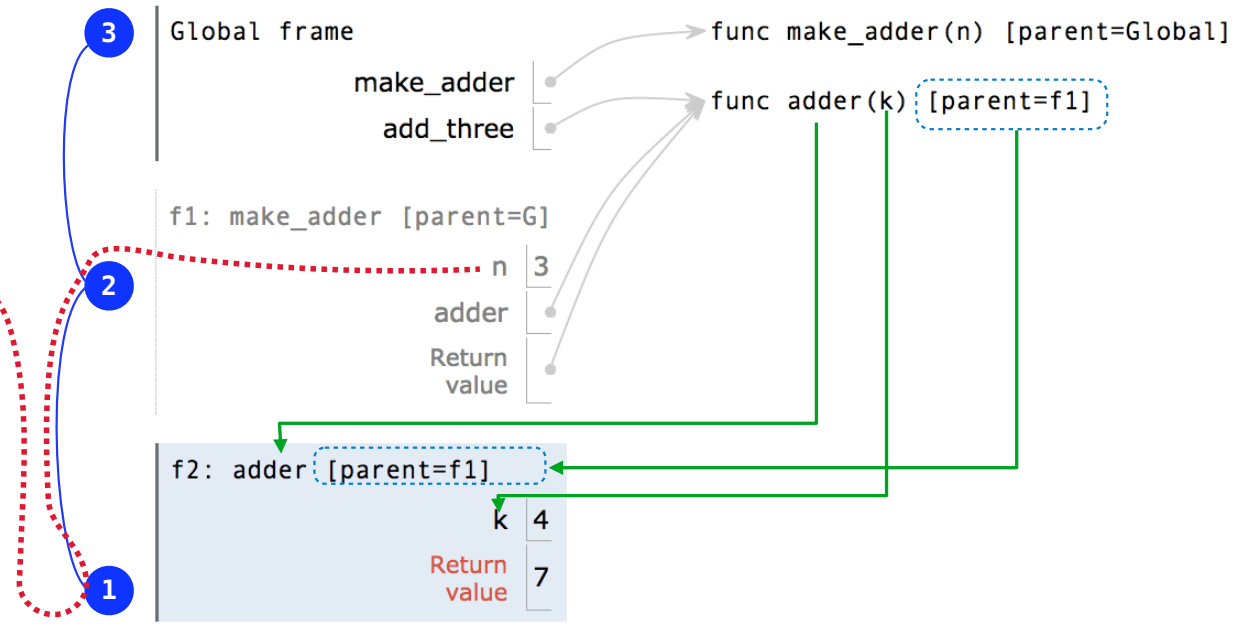
f
x 2

Environment Diagrams for Nested Def Statements

Nested def

```
1 def make_adder(n):
2   def adder(k):
3     return k + n
4   return adder
5
6 add_three = make_adder(3)
7 add_three(4)
```

- Every user-defined function has a parent frame (often global)
- The parent of a function is the frame in which it was defined
- Every local frame has a parent frame (often global)
- The parent of a frame is the parent of the function called

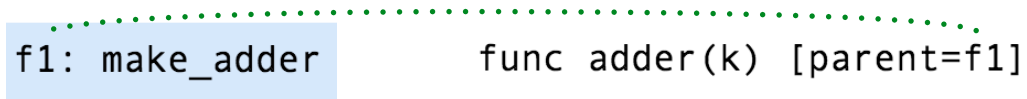


How to Draw an Environment Diagram

When a function is defined:

Create a function value: `func <name>(<formal parameters>) [parent=<label>]`

Its parent is the current frame.



`f1: make_adder` `func adder(k) [parent=f1]`

Bind `<name>` to the function value in the current frame

When a function is called:

1. Add a local frame, titled with the `<name>` of the function being called.
- ★ 2. Copy the parent of the function to the local frame: `[parent=<label>]`
3. Bind the `<formal parameters>` to the arguments in the local frame.
4. Execute the body of the function in the environment that starts with the local frame.

Environment Diagram Practice

