Note 1-6 of cs61a: Essensial steps in make understanding about the processing of the python code.

Author: ***FlappyHimself*** *edited by June.21th*
And this is my first blog and I hope this could be a great start for later studying!

Purpose: Have a basic understanding about how python interpreter processes use an example.

Example:

```
1       def compose1(f, g):
2           return lambda x: f(g(x))
3
4       f = compose1(lambda x: x * x,
5                    lambda y: y + 1)
6       result = f(12)
7   #日你哥的刚学python给我来这么多嵌套返回方程，
8   #让我这个脑子来理解interpreter真的是太高估我了。。。
```

Comprehension:

Step one:

Skip through all the def (Python skips the function implementation until the function was employed.) until we meet expression to evaluate. Now the global has exists compose1. Move to the fourth line until we met obstacle on line 4.

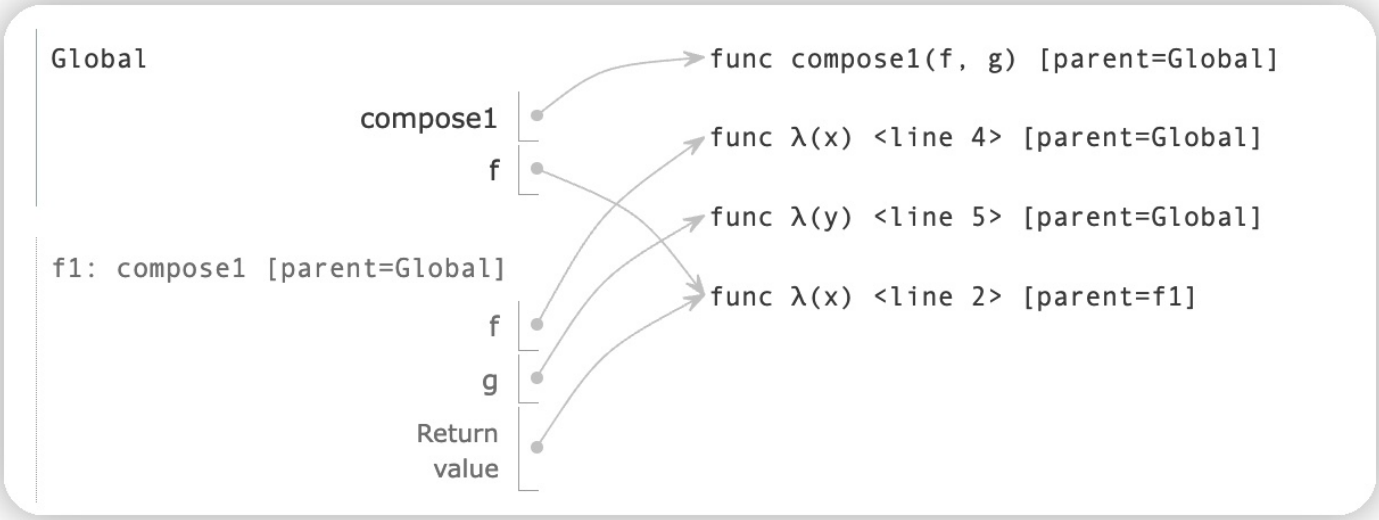
Step Two:

Python firstly evaluates the operator of the whole functions, since the interpreter meet the compose1 function. So the first thing created by memory is the **operand** after the assignment statement.Everything in the operand is firstly evaluated. If there is still not yet evaluated, then we dig deeper to evaluate the deeper operand, just like **recursion**.

Now we move to the line 1 to check function compose1 and create a local environment compose1 with two arguments (**both are functions, which is magic and exciting to me in python when I firstly know that python can designed to have variable name for functions(because those "real" functions are intrinsic)**): f, g, variable names, each of which points to $\lambda(x) < line4 >$ and $\lambda(y) < line5 >$ within the global frame. (**Remember, both of the f, g points to functions in global environment, while function it returned is in f1 local frame.**)

why? My interpretation for comprehension: Any function as an argument needs to be predefined outside the function it's in, in this case, $func\ \lambda(x) < line4 >$ and $func\ \lambda(y) < line5 >$is in the global frame.

The function $compose1$ returns a function, which takes an argument x, returns the $func\ \lambda = \ f(g(x))$. And returned $func\ \lambda(x)$ was assigned to the name f (global frame).

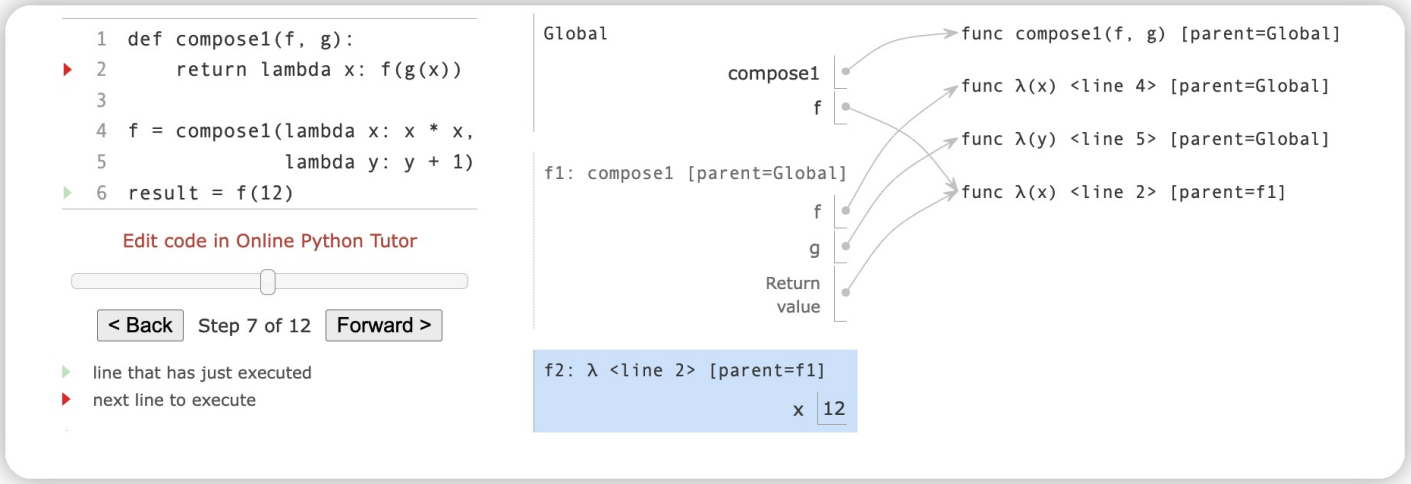


Annotation:*The code below is equivalent to the function of line1 to line2*

```
1  def compose1(f, g):
2      def h(x):
3          return f(g(x))
4  return h
```

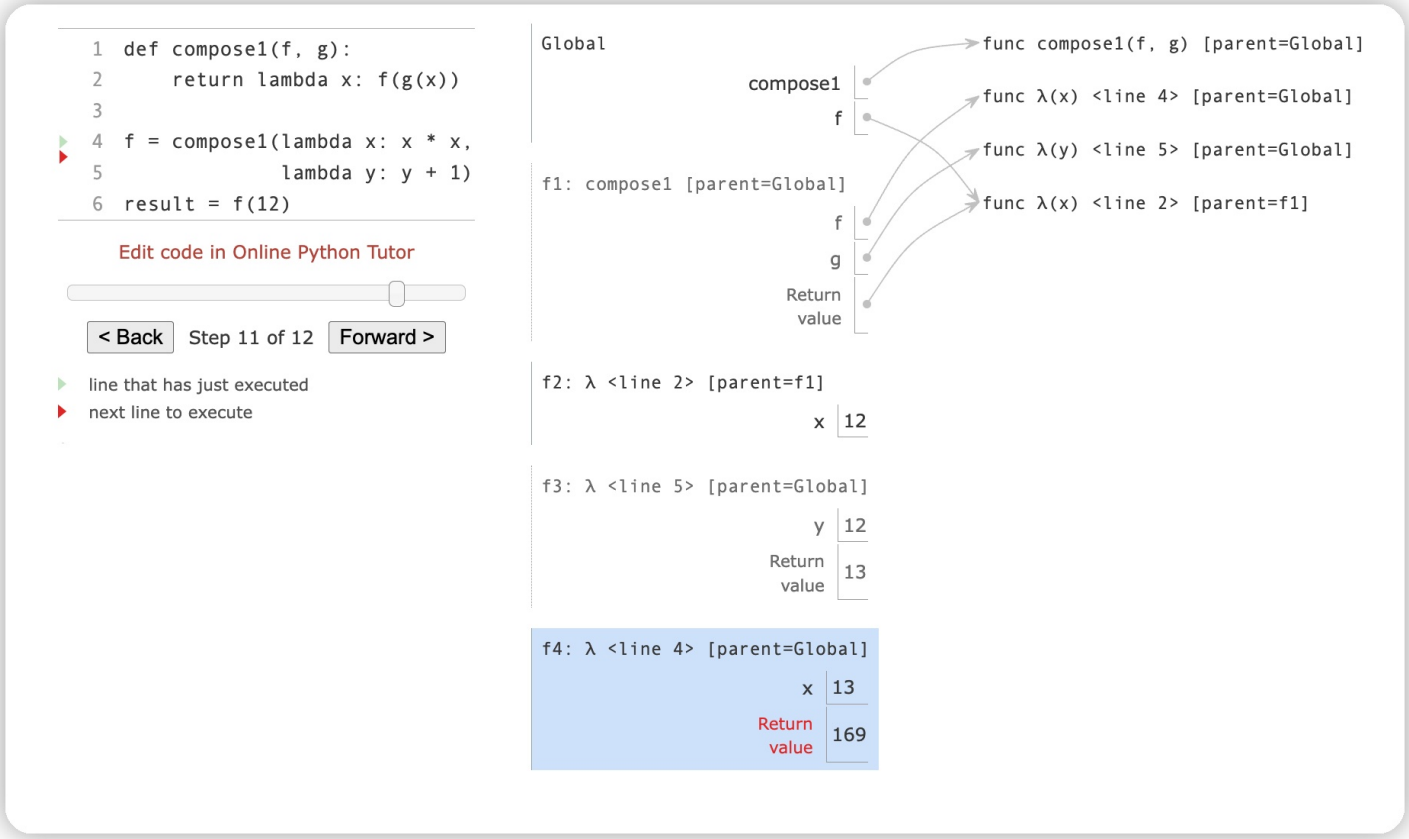
Step Three:

The pointer moves to last line, meeting f, which has already pointed to function $\lambda(x) < line2 >$, as we illustrated last step. Then it created another local frame. λ frame **(It's interesting to see that every language would implement their own way of managing their frames, how variables are interrelated with the environment they are in, computer scientists are fabulous!)**



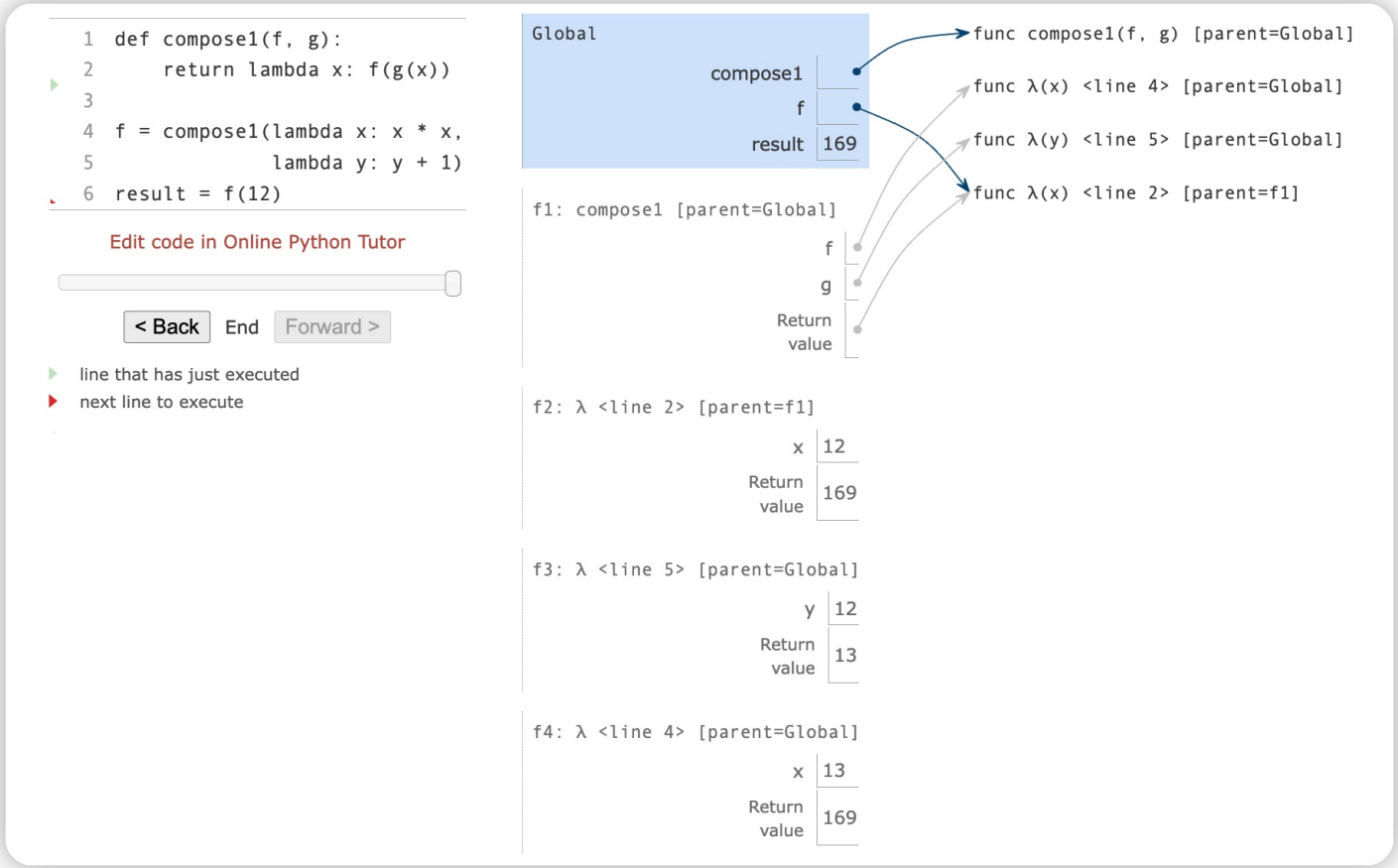
Step Four:

Since We've known f is $func\ \lambda(x) < line2 >$. We examine the x to frame f3, while use the $func\ \lambda(x) < line5 >$, getting the return value, then move to the next operator $func\ \lambda(x) < line4 >$, gaining the return value of 169.



Final Step:

Returning the value 169 to $func\ \lambda(x) < line2 >$, then return back this **value** to variable **result** in global frame. (Pass by value)



The tool: comprehension of python interpreter:

"python tutor" the code tracer
https://pythontutor.com/composingprograms.html#mode=edit

Suggestions in future lectures learning:

1. **Type more and understand the whole content from a blank paper**

2. **Read the notes carefully and slowly, don't be too confident as well as fast. Understanding the principles are more important and more interesting as well.**

Keep going, keep going...