CS1110

Lecture 8: More Frames; Conditionals

Announcements

A1 iscurrency spec change (posted online): for iscurrency:

```
"""Returns: True if <currency> is a valid 3-letter code for a currency,
False otherwise

Precondition: <currency> is a string."""
```

Want to review last lecture? Additional materials on the course homepage are: solution slides with animations, and code you can paste into the Online Python Tutor

Readings for next time: 10.0-10.2, 10.4-10.6, 10.8-10.13

Slides by D. Gries, L. Lee, S. Marschner, W. White

Frames and objects are real

Q: What do these drawings on paper have to do with real programming?

A: Frames, objects, and variables are *exactly* what's being created in Python.

- The Online Python Tutor shows them to you
- The very curious can look at the Python module `traceback`; this is used by our cunittest module to print frame information.

So it's good to have a notation to talk about them.

The power of the "true name"

lt_speed 3×10^8

id3

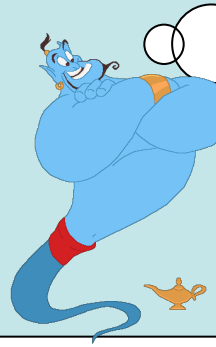
const_list

lt_speed	3×10^8
pi	3.14159
h	6×10^{-34}

violate_physics: **3**

new 42

constants **id3**



"lt_speed = new" wouldn't change the global lt_speed.

But I have the *true name* id3 stored in constants. "constants.lt_speed = new" *does* change the value in id3.

image source (surely not original):
http://4.bp.blogspot.com/_c4VqqkgR0xY/TLHr8YE7RxI/AAAAAAAAAzg/hlPsnjG_Q4k/s1600/GENIE-1.gif

function definition (in lec07.py)

```
def new_rescale(pt):
    """Demo.

       Precond: pt is a Point object"""
    1    norm  = 5.0
    2    pt.y = pt.y / norm
```

code with function call

```
import point
import lec07
p = point.Point(0,3,4)
lec07.new_rescale(p) # what does this line do?
```

Stack of frames: When functions call functions

function definitions

```
def g(m):
    """Returns: energy equivalent of mass m"""
1   E = f(m, lt_speed)
2   return E
```

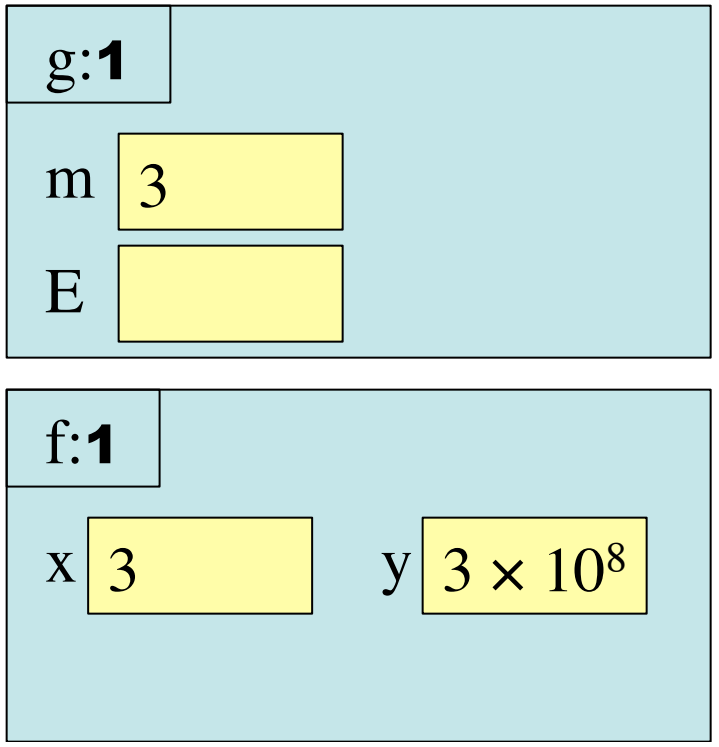
```
def f(x, y):
    """Returns: x times square of y"""
1   return x * (y**2)
```

lt_speed 3×10^8

code with function call

```
lt_speed = 3e8
print g(3)
```

Two "live" frames
(stack shown growing downwards, here)



Error messages show stack info

function definitions

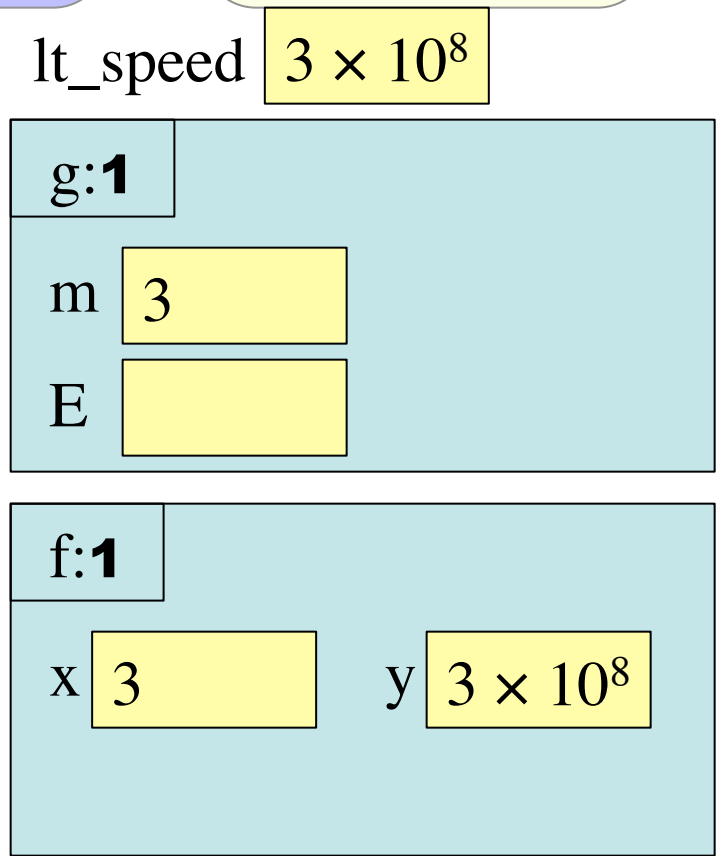
```
def g(m):
    """Returns: energy equivalent of mass m"""
1   E = f(m, lt_speed)
2   return E
```

```
def f(x, y):
    """Returns: x times square of y"""
1   return x * (speed**2)
```

code with function call

```
lt_speed = 3e8
print g(3)
```

```
Traceback (most recent call last):
... line 2, in <module>:  print g(3)
... line 1, in g: E = f(m, lt_speed)
... line 1, in f: return x * (speed **2)
NameError: global name 'speed' is not defined
```



Conditionals

Example for Conditionals (Valentine's Day Special)