

Python Review

Luís Pedro Coelho

Programming for Scientists

February 3, 2009



University of Pittsburgh

Carnegie Mellon

What's The Bug?

```
def mean(values):  
    i = 0  
    S = 0.0  
    while i <= len(values):  
        S += values[i]  
        i += 1  
    return S/len(values)
```

Corrected Version

```
def mean(values):  
    '''  
    M = mean(values)  
  
    Return the mean value of a sequence.  
    '''  
    if not values:  
        raise ValueError(  
            'mean: Mean of empty set not defined')  
    S = 0.0  
    for val in values:  
        S += val  
    return S/float(len(values))
```

Write a function that takes another function f and a number N and computes

$$\sum_{i=0}^N f(i)^2.$$

What's the Bug?

```
def cull_herd(bacteria):  
    for i in xrange(len(bacteria)):  
        if random.random() < bacteria[i].P_dead():  
            del bacteria[i]
```

```
def cull_herd(bacteria):  
    i = 0  
    while i < len(bacteria):  
        if random.random() < bacteria[i].P_dead():  
            del bacteria[i]  
        else:  
            i += 1
```

```
hello.py
```

```
print 'Hello World'
```

```
main.py
```

```
import hello  
import hello
```

What does `python main.py` print out?

- (a) Hello World
Hello World
- (b) *nothing*
- (c) Hello World

```

class Bacterium(object):
    def __init__(self):
        print 'Bacterium'
    ...

class EvolveBacterium(Bacterium):
    def __init__(self):
        print 'EvolveBacterium'
    ...

bac = EvolveBacterium()

```

(a)	(b)	(c)	(d)
Bacterium	EvolveBacterium	Bacterium	EvolveBacterium
		EvolveBacterium	Bacterium

```
A = range(20)
B = A[10:20]
del B[0]
del B[0]
del B[0]
print len(A)
```

(a)
20

(b)
17

(c)
16

(d)
19

```
A = range(20)
B = A[10:20]
B[0] = -1
print A[10]
```

(a)
-1

(b)
10

(c)
9

(d)
11

```
def swap(x,y):  
    x, y = y,x
```

```
a,b = 0,1  
swap(a,b)  
print a, b
```

(From: *How to Think Like a Computer Scientist*)

(a)	(b)
0,1	1,0

```
def swap(A):  
    A[0],A[1] = A[1],A[0]
```

```
A = [0,1]  
swap(A)  
print A[0],A[1]
```

(a)
0,1

(b)
1,0

Write a **generator** that iterates through a list in **reverse** order:

```
for val in reversed(range(4)):  
    print val
```

should print

```
3  
2  
1  
0
```

Trick Question

```
A = [0, 1, 2]
B = A
A += [3]
print B
```

What does this print?

(a)

[0, 1, 2]

(b)

[0, 1, 2, 3]

```
def mystery(*args):  
    if not args:  
        yield ()  
    else:  
        lst = args[0]  
        rest = args[1:]  
        for val in lst:  
            for tp in mystery(*rest):  
                yield (val,) + tp
```

```
def mystery(*args):
    if not args:
        yield ()
    else:
        lst = args[0]
        rest = args[1:]
        for val in lst:
            for tp in mystery(*rest):
                yield (val,) + tp
```

Hint: Consider

```
for T in mystery(range(2), range(4)):
    print T
```

```

def f(x):
    print 'f(%s)' % x
    return x + 2

def g(x):
    print 'g(%s)' % x
    return log(f(x-1))

def h(x):
    print 'h(%s)' % x
    return g(x)-g(x-1)

print h(0)

```

		(c)	
		h(0)	(d)
(b)	(b)	g(0)	h(0)
h(0)	h(0)	f(-1)	g(0)
g(0)	g(0)	g(-1)	f(-1)
OverflowError:	f(-1)	f(-1)	g(-1)
math range error	OverflowError:	math range error	f(-1)
	math range error	OverflowError:	
		math range error	