

PEP: 227 - Statically Nested Scopes

Louis Bouddhou, Alex Campbell, Josh Fermin

December 8, 2014

What is the problem?

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

- Cannot reference a variable in a higher order function (nested).
- Static scoping does not work within nested functions.

Example - Without Statically Nested Scopes

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

```
def bank_account(initial_balance):  
    balance = [initial_balance]  
    def deposit(amount):  
        balance[0] = balance[0] + amount  
        return balance  
    return deposit
```

Introduced changes in this PEP

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

- Gives nested functions the scope of outer functions.
- This allows for variables within the parent function to be inherited by the nested function.

Problems this PEP addresses: Utility

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

- Limited utility of nested functions.

Example

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

```
from Tkinter import *  
root = Tk()  
Button(root, text="Click here",  
        command=lambda root=root: root.test.configure()
```

Problems this PEP addresses: Non-lexical

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

- Confusion among new users who are used to lexical scoping.

Example

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

```
def make_adder(base):  
    def adder(x):  
        return base + x  
    return adder  
add5 = make_adder(5)  
add5(6)
```

11

Resolving Free Variables

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

- Nested functions have access to variables in other scopes.

Namespaces

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

- Local
- Global
- Builtin

Local Namespace

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

- Local Namespace `“python def function(num): foo = num
* num return var`

`function(2) “`

Global Namespace

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

```
bar = 10
def function(num)
    foo = bar * num
    return foo
function(2)
```

Builtin Namespace

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

```
def function(num)
    myList = list()
    for n in list:
        n = n * num
    return list
```

```
function(2)
```

Bounds

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

```
def function(num)
    foo = bar * num
    return foo
function(2)

bar = 10
```

Name Search

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

```
myList = list()
def function(num):
    list = myList
    list.append(1)
    list.append(2)
    list.append(3)
    for n in list:
        n = n * num
    print n
    print list
    return list
function(2)
```

Discussion

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

- The PEP works under all circumstances except for the following cases:
 - 1 Name in class scope is not accessible
 - 2 Global statement short-circuits the normal rules

Discussion - Name in Class Scope

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

- Names in a class scope:
- Resolved in the innermost (nested) function

Discussion - Short Circuit

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

- Global statement is unaffected by change

```
myvariable = 5
def func():
    global myvariable
    myvariable = 6    #changes global var
    print myvariable #prints 6

func()
print myvariable    #prints 6
```

Problems - Backwards Compatibility

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

- Two kinds of compatibility problems caused:

- ① Code behavior
- ② Syntax errors

Example - Code Behavior

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

```
x = 1
def f1():
    x = 2
    def inner():
        print x
    inner()
```

Example - Syntax Errors

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

```
y = 1
def f():
    exec "y = 'gotcha'" # or from module import *
    def g():
        return y
```

Conclusion

PEP: 227 -
Statically
Nested Scopes

Louis
Bouddhou,
Alex
Campbell,
Josh Fermin

- FILL THIS IN PLS