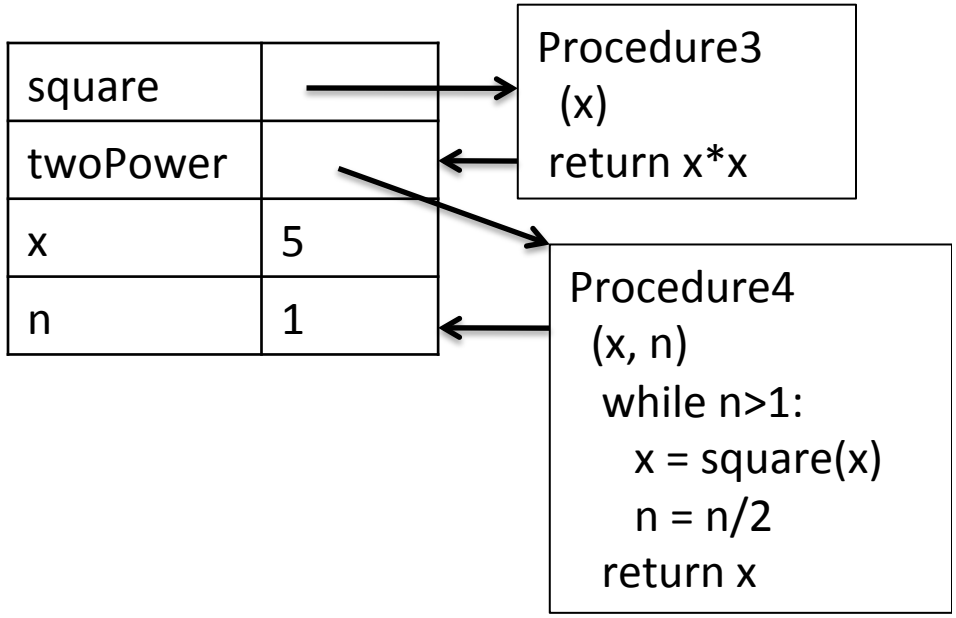
Another example

```
def square(x):
    return x*x

def twoPower(x, n):
    while n > 1:
        x = square(x)
        n = n/2
    return x
```

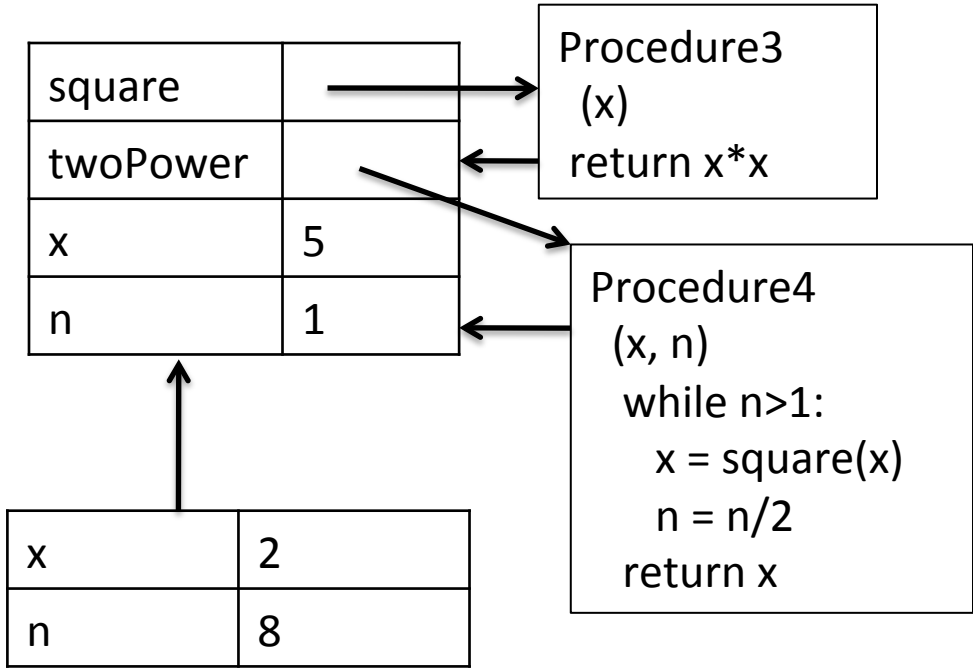
Let's try it out

```
def square(x):
    return x*x

def twoPower(x, n):
    while n > 1:
        x = square(x)
        n = n/2
    return x

x = 5
n = 1
print(twoPower(2,8))
```



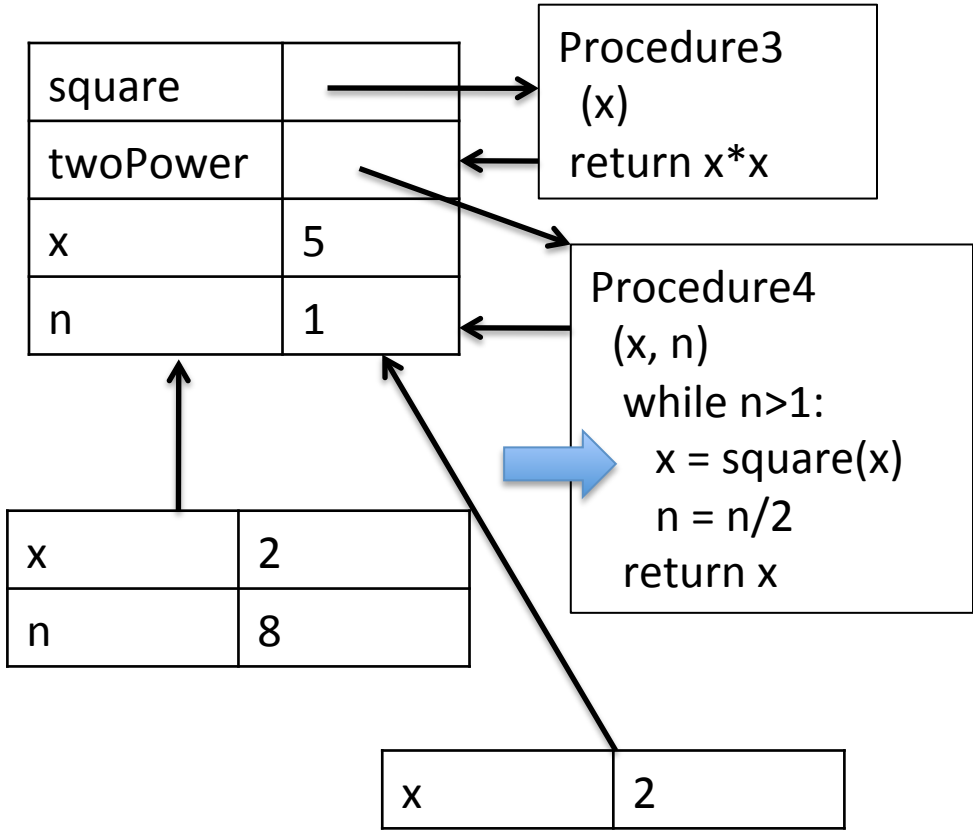
Let's try it out

```
def square(x):
    return x*x

def twoPower(x, n):
    while n > 1:
        x = square(x)
        n = n/2
    return x

x = 5
n = 1
print(twoPower(2,8))
```



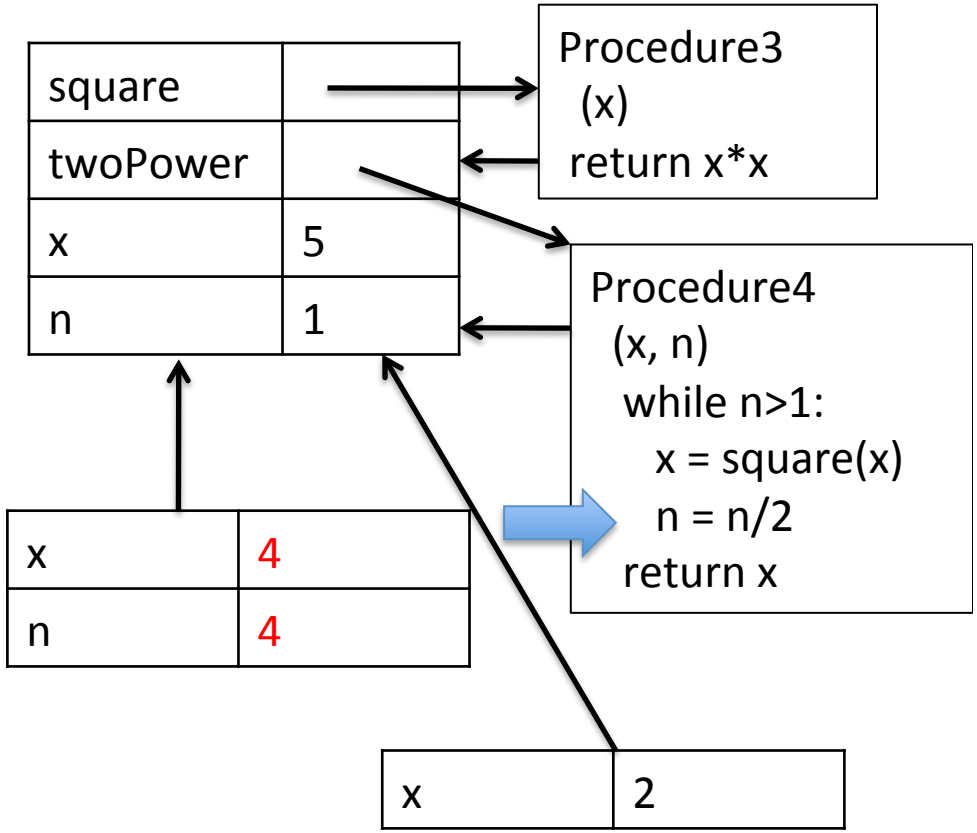
Let's try it out

```
def square(x):
    return x*x

def twoPower(x, n):
    while n > 1:
        x = square(x)
        n = n/2
    return x

x = 5
n = 1
print(twoPower(2,8))
```



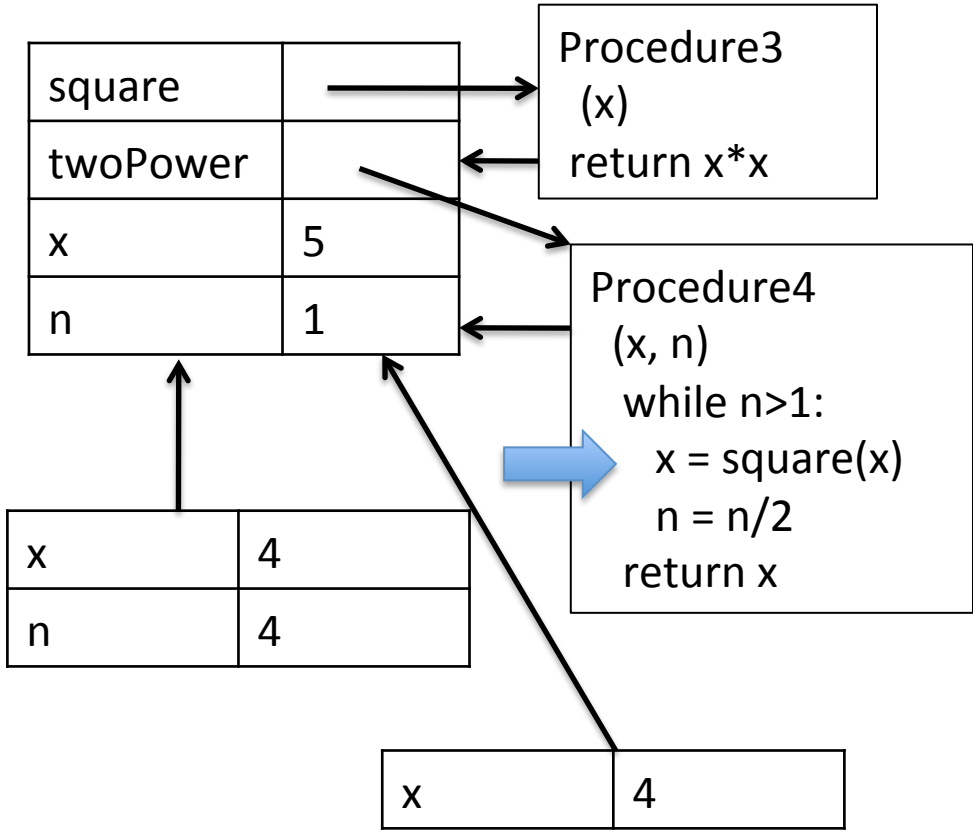
Let's try it out

```
def square(x):
    return x*x

def twoPower(x, n):
    while n > 1:
        x = square(x)
        n = n/2
    return x

x = 5
n = 1
print(twoPower(2,8))
```



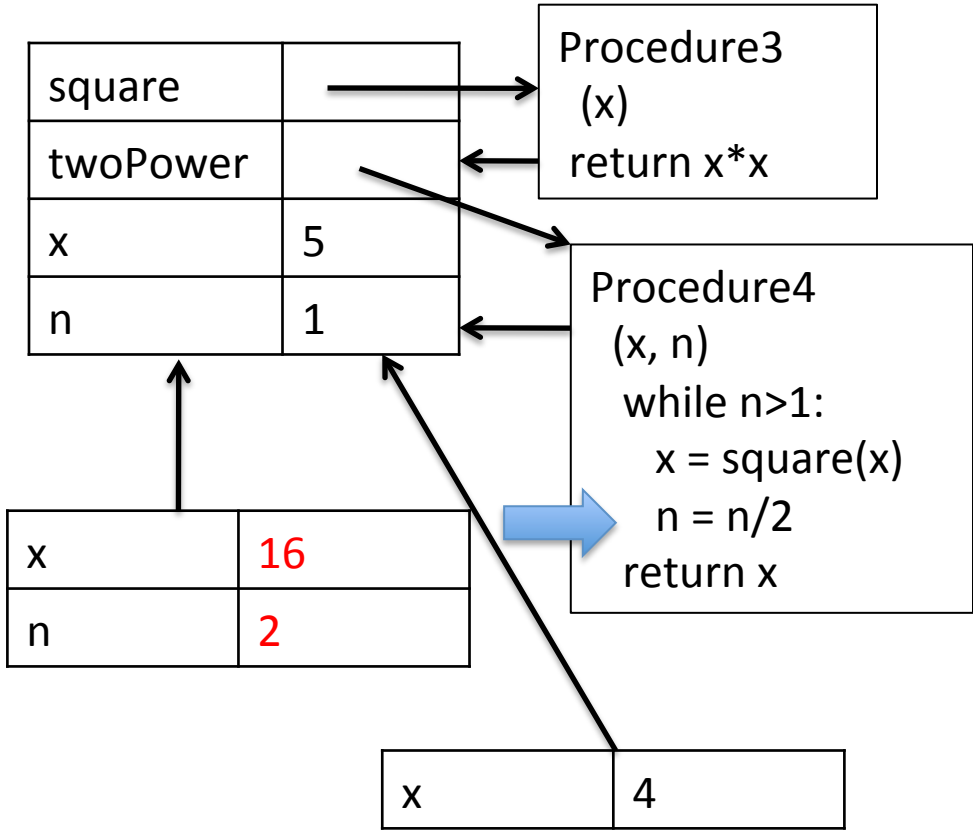
Let's try it out

```
def square(x):
    return x*x

def twoPower(x, n):
    while n > 1:
        x = square(x)
        n = n/2
    return x

x = 5
n = 1
print(twoPower(2,8))
```



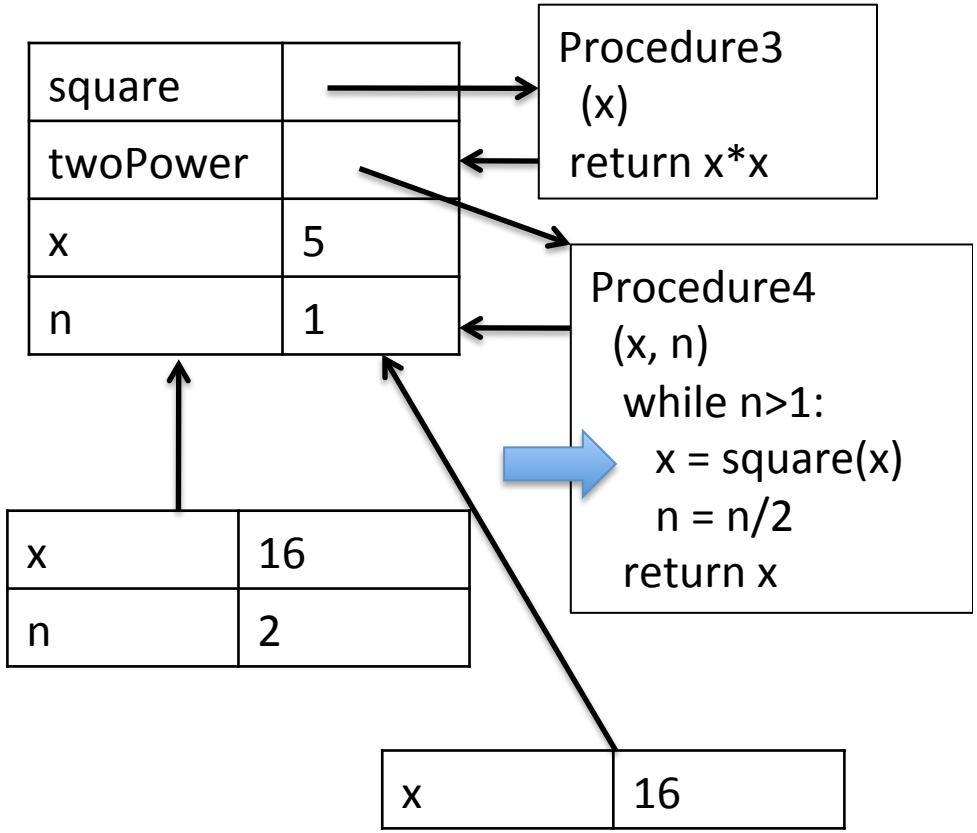
Let's try it out

```
def square(x):
    return x*x

def twoPower(x, n):
    while n > 1:
        x = square(x)
        n = n/2
    return x

x = 5
n = 1
print(twoPower(2,8))
```



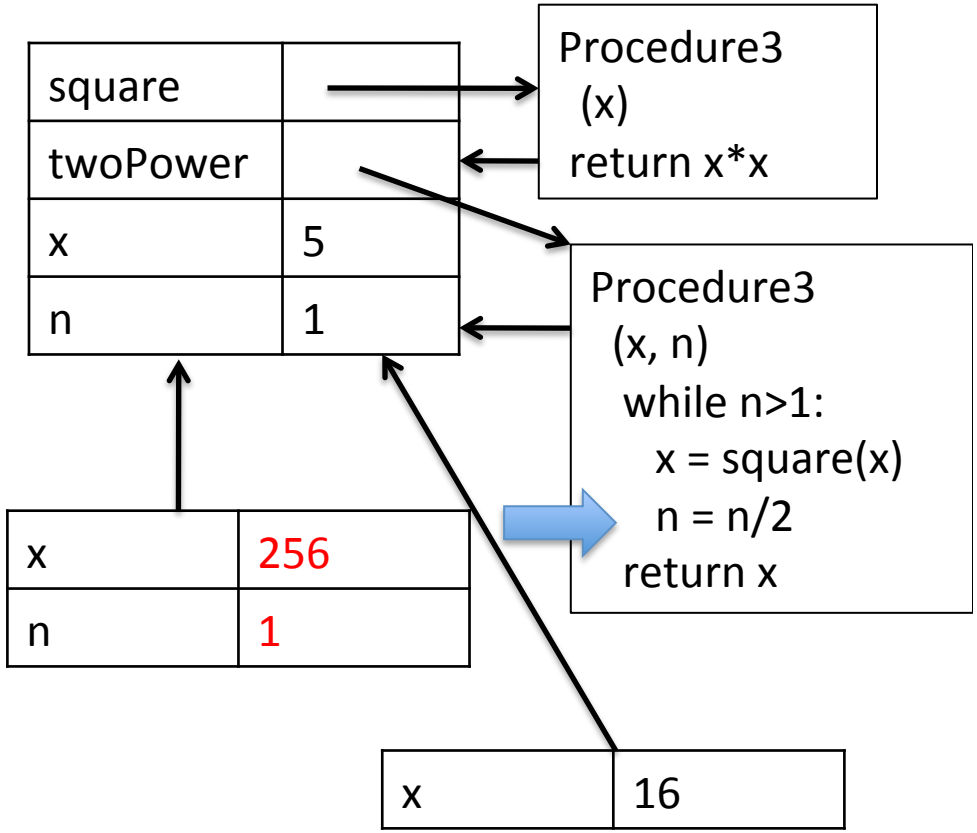
Let's try it out

```
def square(x):
    return x*x

def twoPower(x, n):
    while n > 1:
        x = square(x)
        n = n/2
    return x

x = 5
n = 1
print(twoPower(2,8))
```



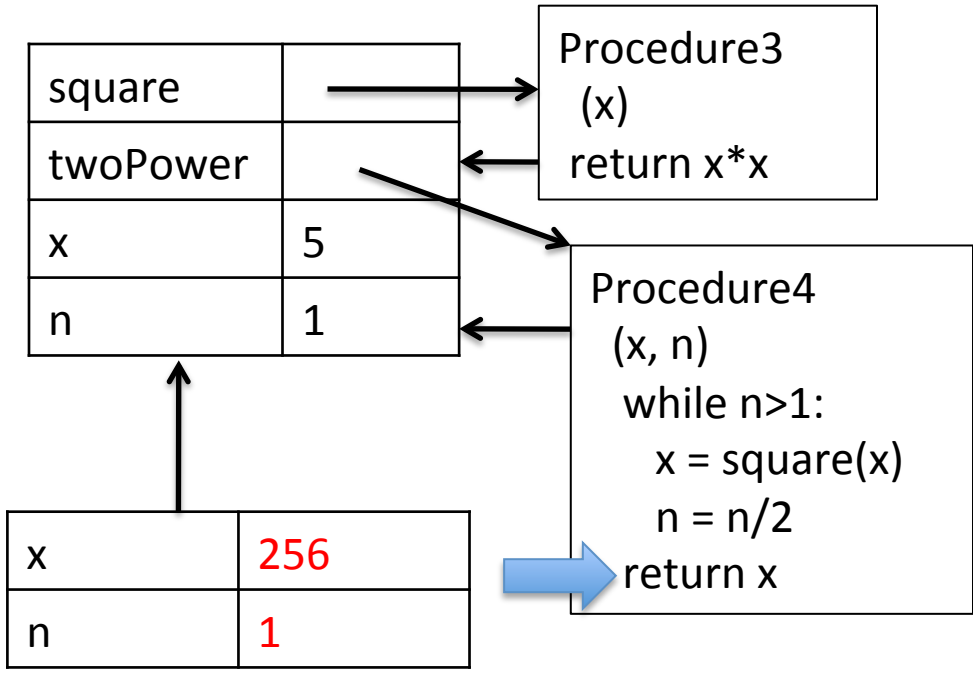
Let's try it out

```
def square(x):
    return x*x

def twoPower(x, n):
    while n > 1:
        x = square(x)
        n = n/2
    return x

x = 5
n = 1
print(twoPower(2,8))
```



Let's try it out

```
def square(x):
    return x*x

def twoPower(x, n):
    while n > 1:
        x = square(x)
        n = n/2
    return x

x = 5
n = 1
print(twoPower(2,8))
256
```



Some observations

- Notice how each call to square created a new frame, with a local binding for x
- The value of x in the global environment was never confused with values within frames from function calls
- The value of x used by the call to square is different from the binding for x in the call to twoPower
- The rules we described can be followed mechanically to determine scoping of variables