Solution: Variable Scope

Quiz Solution: Variable Scope

The code snippet on the previous page is repeated here:

```
egg_count = 0

def buy_eggs():
    egg_count += 12 # purchase a dozen eggs

buy_eggs()
```

This code causes an `UnboundLocalError`, because the variable `egg_count` in the first line has global scope. Note that it is not passed as an argument into the function, so the function assumes the `egg_count` being referred to is the global variable.

In the last video, you saw that within a function, we can print a global variable's value successfully without an error. This worked because we were simply accessing the value of the variable. If we try to **change** or **reassign** this global variable, however, as we do in this code, we get an error. Python doesn't allow functions to modify variables that aren't in the function's scope.

A better way to write this would be:

```
egg_count = 0

def buy_eggs(count):
    return count + 12  # purchase a dozen eggs

egg_count = buy_eggs(egg_count)
```

← Previous Next →

Give Page Feedback