Lab 7 GRS

Functions, scope, bit operations

Announcements

- My Thursday office hours this week are **cancelled**
  - I will have office hours on Wednesday from 5 pm to 6 pm this week

What are functions?

- A function is a chunk of code that runs when it's **called (or invoked)**
- You've used functions before
  - upper(), lower(), join(), etc.

Why use functions?

- Allow us to reuse code
  - Don’t have to waste a bunch of code to do the same task
- Abstraction
  - We **don’t know** *exactly* how `split()` was implemented, but we do know that **it works,** and **what it does**
  - Driving a car
- Readability

Functions

- What does this do?

```
def do_something(x):
    if ord(x) < 65 or (ord(x)>=91 and ord(x)<=96) or ord(x) > 122:
        return x

    if ord(x)>=65 and ord(x)<=90:
        return chr(ord(x) + 32)

    return x
```

Functions

- This is (probably) the implementation python uses for `lower()`
  - Spoiler alert: characters are actually integers

```
def lower(x):
    if ord(x) < 65 or (ord(x)>=91 and ord(x)<=96) or ord(x) > 122:
        return x

    if ord(x)>=65 and ord(x)<=90:
        return chr(ord(x) + 32)

    return x
```

Functions

- Return is **optional**
  - By default, python returns a value called **None**
    - Equivalent to null, nullptr, etc
- When passing parameters, a copy of the variable is passed

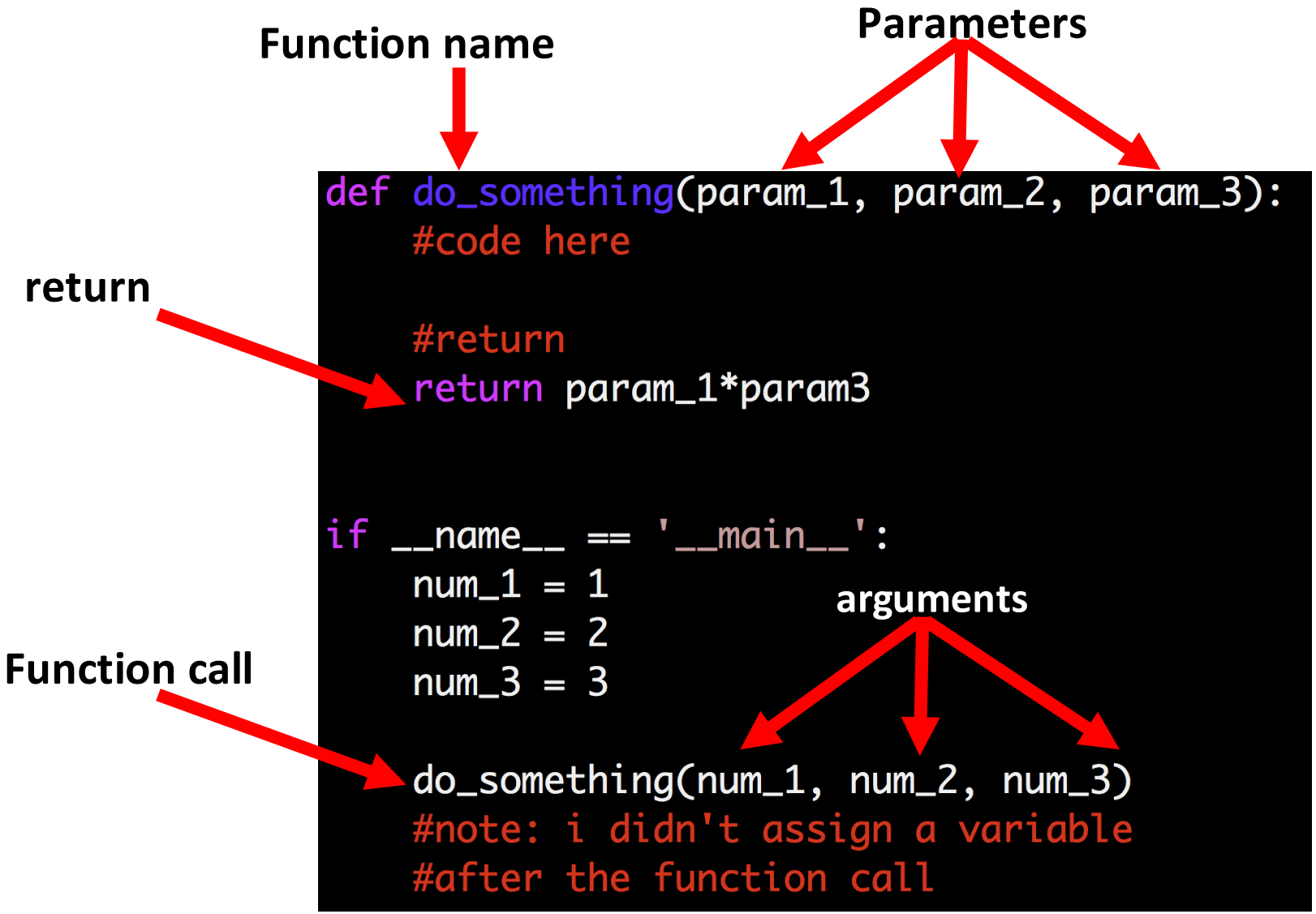
```
def do_something(num1, num2, num3):
    num4 = num1 + num2
    return num4

if __name__ == "__main__":
    do_something(1, 2, 3)
    num4 += 5
```

Passing lists as parameters

- Lists are **pass by reference**
  - Meaning, when you pass them to a function as a parameter, **the actual list is passed, not a copy like other variables**

Structure of a function



Scope

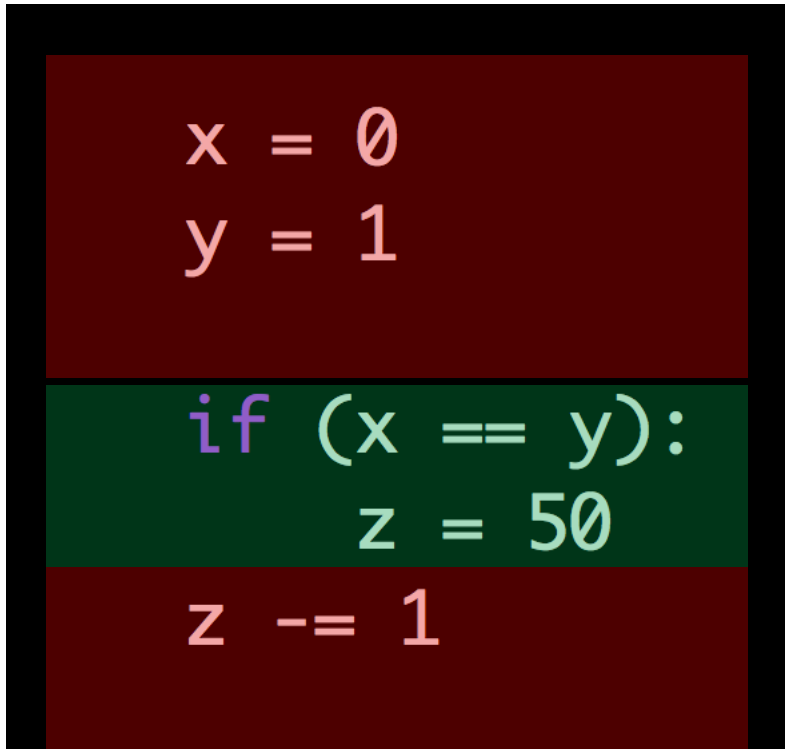
- Variables (and other pieces of data) are only accessible in certain regions
- What happens when the following runs?

```
x = 0
y = 1

if (x == y):
    z = 50
z -= 1
```

Scope

- The variable z exists in the if-statement, **not outside**



Scope

- Also applies to functions
- Functions variables only exist within the function

```
def do_something(param_1, param_2, param_3):
    #code here

    #return
    return param_1*param3


if __name__ == '__main__':
    num_1 = 1
    num_2 = 2
    num_3 = 3

    do_something(num_1, num_2, num_3)
    #note: i didn't assign a variable
    #after the function call
```

Activity

- Copy the activity file from my public directory
- **cp** /afs/umbc.edu/users/a/g/agatha3/pub/201_grs/lab7/activity.py **.**
  - **Don't forget the period**
- Expected output is on my github