CSC 280 Introduction to Computer Science: Programming with Python

Lecture7: Functions, Return Values

Prof. Bei Xiao

Fall, 2014

American University

Exercise: Break

- Find a cube root of a perfect cube using two kinds of methods:

- 1. While loop
- 2. For loop with Break statement.

Built-in Functions

- You have already seen some built in functions
- type(34)
- range(0, 5)
- int(“32”)
- abs(-2)

math function

- import math
- math.log10(17.0)
- math. cos(0.25*pi)
- math. exp(2)
- math.sqrt(4)

Define a new function

- Function is an named sequence of statements that performs a desired operation.

```
def name of function (list of arguments):
    body of function
```

Calling a function

- Function can be called using a statement.

```
 def Name (arg1, arg2):
     statements

Name(arg1, arg2)
```

Function: Arguments, Return values

```
def max(x,y):
    if x>y:
        return x
    else:
        return y
maxnum = max(5,6)
print 'max is: ', maxnum
```

Question: What is max(4+6,7)?

How about max(4*7,9-3)?

When a function is called

1. The formal parameters of the function are bound to the resulting values in the expression that calls the function. E.g. calling max(6,4) will bind x to 6 and y to 4

When a function is called

1. The formal parameters of the function are bound to the resulting values in the expression that calls the function. E.g. calling max(6,4) will bind x to 6 and y to 4.
2. The **point of execution** (the next instruction to be executed) moves from the point of invocation to the first statement in the body of the function.

When a function is called

1. The formal parameters of the function are bound to the resulting values in the expression that calls the function. E.g. calling max(6,4) will bind x to 6 and y to 4.
2. The **point of execution** (the next instruction to be executed) moves from the point of invocation to the first statement in the body of the function.
3. The code in the body of the function is executed until either a **return** statement is encountered or there are no more statements to execute.

When a function is called

1. The formal parameters of the function are bound to the resulting values in the expression that calls the function. E.g. calling max(6,4) will bind x to 6 and y to 4.
2. The **point of execution** (the next instruction to be executed) moves from the point of invocation to the first statement in the body of the function.
3. The code in the body of the function is executed until either a **return** statement is encountered or there are no more statements to execute.
4. The value of the invocation is the returned value
5. After the invocation, the point of execution is transferred back to the code after the invocation.

Example: function without argument

Write a function that print the following shape:

```
 | |
--------
 | |
--------
 | |
```

Write another function to call the above function to print:

```
 | |
--------
 | |
--------
 | |
 | |
--------
 | |
--------
 | |
--------
 | |
--------
```

Example, say hello

Define a function to print “ hello” to a specific name.

Invoke the function by

sayHello(“Bob”)

Which should print: Hello, Bob

Example: print a song

- Define a function "happybirthday(name)" which print out the lyrics of the song "happy birthday" to a specific name.
- Ask for a friend's name and print the song "Happy Birthday" to a friend.

Example: write a few arithmetic functions

- Def add(num1, num2):
- Def sub(num1,num2):
- Def mul(num1,num2):
- Def div(num1, num2):

Calling the above function by:

```
add(2,3)
add('input()', 'input()' )
Num1 = 2;
Num2 = 3;
sub(num2,num1)
```

Quiz: Write a calculator

Write a program calculator.py which calling a bunch of arithmetic functions and perform as a calculator:

```
>>>
Welcome to calcualtor.py
input an operator to start, followed by two numbers, press 'q' to quit
 +
 4
 5
4 + 5 = 9
 -
 7
 2
2 - 7 = -5
 q
Thankyou for using calculator.py!
```

Take-home: isIn

- Write a functio isIn that accepts two strings as arguments and returns True if either string ocurs anywhere in the other, and False otherwise.

- Hint: use the built-in str operation in.

Take home readings

Chapter 3 and 5

Chapter 6