

CSC 280 Introduction to Computer Science: Programming with Python

Lecture 5 : Functions, Fruitful Functions  
September 13 , 2015

Prof. Bei Xiao

Fall, 2015

American University

# Outline

- Review Function definition
- Variables and Parameters are local
- Fruitful Functions
- The **return** statement
- Exercises

# Example: function without argument

Write a function that print the following shape:

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

Write another function to call the above function to print:

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


# Parameters and arguments

## Function Call

Command to do the function

```
greet('Walker')
```

Function  
Header



## Function Definition

Defines what function does

```
def greet(n):  
    print "hello " +n + "!"
```

- **Parameter**: variable that is listed within the parentheses of a method header.
- **Arguments**: a value to assign to the method parameter when it is called

# Parameters and arguments

## Function Call

Command to do the function

```
greet('Walker')
```

Function  
Header

## Function Definition

Defines what function does

```
def greet(n):  
    print "hello " +n + "!"
```

Function  
Body

- **Parameter**: variable that is listed within the parentheses of a method header.
- **Arguments**: a value to assign to the method parameter when it is called

# Parameters and arguments

## Function Call

Command to do the function

```
greet('Walker')
```

Function  
Header

## Function Definition

Defines what function does

```
def greet(n):  
    print "hello " + n + "!"
```

Declaration of  
parameter n

Function Body

- **Parameter**: variable that is listed within the parentheses of a method header.
- **Arguments**: a value to assign to the method parameter when it is called

# Parameters and arguments

## Function Call

Command to do the function

```
greet('Walker')
```

argument to assign to n

Function  
Header

## Function Definition

Defines what function does

```
def greet(n):  
    print "hello " + n + "!"
```

Declaration of  
parameter n

Function Body

- **Parameter**: variable that is listed within the parentheses of a method header.
- **Arguments**: a value to assign to the method parameter when it is called

# Anatomy of a Function Definition

name

parameters

```
def greet(n):
```

Function Header

```
    """Prints a greeting to the name n
```

```
        Parameter n: name to greet
```

```
        Precondition: n is a string"""
```

Docstring  
Specification

```
    print 'Hello '+n+'!'
```

```
    print 'How are you?'
```

Statements to  
execute when called



# Global and Local Variable

Consider the following example:

```
def f():
```

```
    print (s)
```

```
s = "I love Paris in the summer!"
```

```
f()
```

What will be the output?

# Global and Local Variable

Consider the following example:

```
def f():  
    print (s)  
s = "I love Paris in the summer!"  
f()
```

What will be the output?

# Global and Local Variable

Consider the following example:

```
def f():
```

```
    s = " I love London!"
```

```
    print (s)
```

```
s = "I love Paris!"
```

```
f()
```

```
print (s)
```

What will be the output?

# Global and Local Variable

Consider the following example:

```
def f():
```

```
    s = " I love London!" → local variable
```

```
    print (s)
```

```
s = "I love Paris!" → global variable
```

```
f()
```

```
print (s) → global variable
```

What will be the output?

I love London!

I love Paris!

Why?

# Global and Local Variable

Consider the following example:

```
def f():  
    print(s)  
    s = "I love London!"  
    print (s)  
s = "I love Paris!"  
f()
```

# Global and Local Variable

Local variables of functions can't be accessed from outside, when the function has finished

```
def f():  
    s = "I am globally not known"  
    print (s)  
  
f()  
print(s)
```

# Procedure vs. Fruitful Functions

## Procedures

- Functions that do something
- Call them as statement
- Example: `greet('Walker')`

## Fruitful Function

- Functions that gives a value
- Call them in an expression
- Example: `x = round(2.56,1)`

**Historically “function” = “fruitful functions”  
But now we use “function” to refer to both**

# The **return** statement

Fruitful functions require a **return statement**

**Format:** return <expression>

- Provides value when call is used in an expression
- Also stops executing the function!
- Any statements after a **return** are ignored

**Example:** temperature converter function



# The `return` statement

Fruitful functions require a `return` statement

Format: `return <expression>`

```
def to_centrigrade(x):
```

```
    """Returns: x converted to centigrade
```

```
    """
```

```
    return 5*(x-32)/9.0
```

# Print vs. Return

## Print

- Displays a value on screen
  - used primarily for **testing**
  - Not useful for calculations

```
def print_plus(n):  
    print (n+1)
```

```
x = print_plus(2)
```

```
3
```

```
>>>
```

x



nothing  
here!

## Return

- Define the functions' value
  - Important for **calculations**
  - But does not display anything

```
def return_plus(n):  
    return (n+1)
```

```
x = return_plus(2)
```

```
>>>
```

x



# Module Demo: temperature.py

```
# temperature.py
""" Conversions functions between fahrenheit and centigrade """

def to_centigrade(x):
    """Returns: x converted to centigrade
    """
    return 5*(x-32)/9.0

def to_fahrenheit(x):
    """Returns: x converted to fahrenheit
    """
    return 9*x/5.0+32

# Constants
FREEZING_C = 0.0 # Temperature water freezes in centigrade

FREEZING_F = to_fahrenheit(FREEZING_C)
```

**Style guideline:**

Two blank lines between  
function definition

# Demo: age and birth year

```
def getAges(): # This function doesn't take any arguments
    myAge = int(raw_input("What is your age? "))
    friendAge = int(raw_input("What is your friend's age? "))
    return (myAge, friendAge)

myAge, myFriendAge = getAges()
```

# Exercise: function with **return**

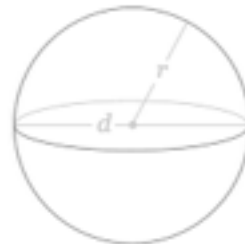
Define a function called  $F(x) = x^3$  that compute cube of a number.

Use “return “ to return  $F(x)$

Call this function to compute the volume of a sphere given the radius  $r$  and print the result:

$$V = \frac{4}{3}\pi r^3$$

$r$  Radius



# Quiz

**Can you guess the output of the following code:**

```
def f():  
    x = 3  
    print x  
def g(x)  
    print x  
    x = 1
```

```
x = 3  
print x  
f()  
x = 3  
g(x)
```

# Take home reading

- Read Chapter 3.7-13.

Functions exercises

- <http://learnpythonthehardway.org/book/ex18.html>
- <http://learnpythonthehardway.org/book/ex13.html>

# Next lecture (Thursday)

- Wed: Lab
- More on function with returns
- Modules
- Lab 1 is due on Wed before class. No need to hand in the code. Just hand in the sheets.