CSC 280 Introduction to Programming Lecture 9

Scoping, review of Control Flow

Bei Xiao

American University

Fruitful functions

- Return values: return immediately from this function and use the following expression as a return value.

```
Import math
def  area(radius):
      temp = math.pi * radius **2
      return temp
# Calling the function:
area_val = area(5)
```

Exercise: multiple return

- Write a function and return 0 if x==y and return -1 if x<y and 1 x>y .

Return & Print inside the Function

- Return: the value is actually returned and the function has an output which is can be assigned to a variable.

- Print: Only print out the results onto the terminal. The value is not accessible outside the function.

Exercise

Write a function that ask the user two put two ages, your age, and your friend's age

```
def  getAges(age1, age2):
        …….
```

And then call the function

```
myAge,myFriendAge = getAges()
```

Exercise

Write another function that compute the birth year given an age

```
def  getBirthYear(age):
        ……..
myAge,myFriendAge = getAges()
Print getBirhtYear(myAge)
```

Exercise

Write another function that Print the birth year given an age:

```
def  printBirthYear(age):
      ……

myAge,myFriendAge = getAges()
getBirhtYear(myAge)
```

Return and Print

- What is the output of the following code:

```
def getBirthYear(age):
        print 75
        return 2014 – age
        print 50

Print getBirthYear(25)
```

Lexical Scoping

- Local variable inside a function only exists inside the function, you cannot use it outside.
- Each function defines a new name space, also called a scope.
- What is the output of the following code:

```
def  f(x):
    y =1
    x = x+y
    print 'x=', x
    return x

x = 3
y = 2

z = f(x)  # x is a parameters
```

Quiz: can you guess the output of the following code?

```
a_var = 'global value’
def a_func():
    a_var = 'local value'
    print(a_var, '[ a_var inside a_func() ]’)
a_func()
print(a_var, '[ a_var outside a_func() ]’)
```

Lexical Scoping: summary

- If a variable is assigned inside a def, it is local to that function.
- If a variable is assigned in an enclosing def, it is nonlocal to the nested function.
- If a variable is assigned outside all defs, it is *global* to the entire file.