

Supplement contents for Wk05_Lec05_Functions

Clarifications for Common Mysteries of "return"

(1) "Why does a function need to return a value? Why not just output the value?"

Answer: For some functions, we want the function to return us a result so that we can proceed doing flexible things with the result.
At a time we use the result in this way, at another time in another way.

```
# calculate the discount
# maximum 5 dollars

def calc_discount(price, discount):
    discounted = price * (discount / 100)
    if discounted > 5.0:
        return 5.0
    else:
        return discounted

print("You have saved", calc_discount(40, 10), "dollars!!")
print("Saving is", calc_discount(30, 20), "dollars!!")
```

(2) "Why does a function need to use a return statement to return a value?"

Answer: If a function is to return a value, there must be a syntax to do so.
Here: *return [result]*.

```
def a_long_function(input1, input2):
    ...
    ...
    x1 = ...
    ...
    ...
    x2 = ...
    ...
    ...
    x3 = ...
    ...
    ...
```

- * the above function has sophisticated logic and several variables
- * if there is no return [result], then what value will the caller receive?

(3) "Can a function return one result and then return another result?"

Answer: When executing a function, it can return at most one result:

When we call a function to run, python execute the function code step by step, when it runs a "return" statement, the value in the return statement is returned, and the function stops (will not continue running any remaining code).

As the function is stopped, it will definitely not return another value.

When a function is executed, if it does not step through a "return" statement, then the final returned value is "none"

(4) "I saw people writing "return" only, without any value to be returned. What does that mean?"

Answer: "return" without value simply means to stop the function.

Example:

```
def a_useful_function(input1, input2):
    if ____: # something wrong with input1, input2
        print("please run again with proper input arguments")
        return # function is stopped. remaining code will not run.

    ... # the code to handle normal cases
    ...
```