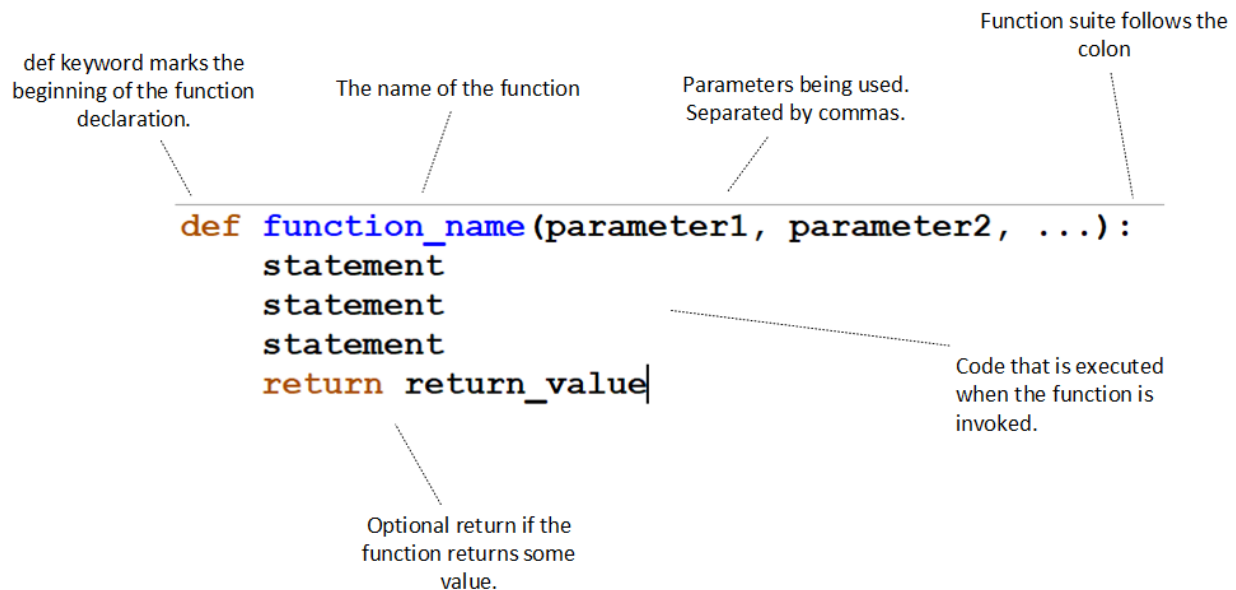


Lab Week 4 - Functions

Skills Needed to complete this Lab

- Create functions
- Invoke functions
- Iterate over strings
- All the skills you've learned in previous weeks

Creating Functions



Functions make it much easier to debug a program as well. You can call the function from the command line multiple times to make sure it functions correctly. Once you are satisfied, you can call the functions in your program and be confident that the results will be consistent.

Slots

In this program, you are going to simulate a slot machine in Pierson Hall. The slot machine has 3 reels; each reel will have numbers ranging from 1 to 10. The user can choose how many chips they start out with. They are allowed to wager between 1 to the amount of bank they currently have on each spin. If they match 3 numbers, they will win 10 times their wager. If they match 2 numbers, they will win 3 times their wager.

Lab04.py

You will start out with the lab04.py program. Notice that it has been stubbed out with functions and much of the main program has been written. Most of your work will be in making each function work correctly. This can help you see how you can break a program down into smaller sub-problems which makes design and testing much easier.

All of these functions have been stubbed out to return a literal value. You can actually run the program and see it running.

For testing below, make sure that any functions are before the `if __name__ == "__main__"` section, and anything in the main portion of the program is indented under this if statement.

Slot_UnitTests.py

Slot_UnitTests will run your functions and check for valid results. They don't test everything, but if you get all the tests passing, then you'll have a pretty good indication that you've written most of the functions effectively.

You can edit and run the Slot_UnitTests.py in python just like your program. However, make sure you save your lab04.py program before you run the tests. It can't tell that you've got some unsaved changes in the editor.

```
def test_play_again_returns_True(self):
    user_input = ["Yes"]
    with patch('sys.stdout', new_callable=StringIO) as es:
        with patch('builtins.input', side_effect=user_input):
            play_again()

Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\binghamkl-pc\Box\Course Work\101L\Week 05\Solution\Slot_UnitTests.py
.....
-----
Ran 10 tests in 0.051s
OK
>>> |
```

play_again() function

The `play_again()` function will ask the user if they want to play again. It will return True if the user enters Y or YES regardless of the case. The function will return False if the user enters N or NO. If the user enters anything else, then it will warn the user and ask them to enter another value.

To test this function, run the program like normal and then stop the execution by hitting ctrl-C, or command C on the mac. In the shell, you can type the function as shown below to test the results

Examples

```
KeyboardInterrupt
>>> play_again()
Do you want to play again? ==> e

You must enter Y/YES/N/NO to continue. Please try again
Do you want to play again? ==> 3

You must enter Y/YES/N/NO to continue. Please try again
Do you want to play again? ==> yEs
True
>>> play_again()
Do you want to play again? ==> 9

You must enter Y/YES/N/NO to continue. Please try again
Do you want to play again? ==> n
False
>>>
```

get_wager(bank : int) -> int: function

This function should return an integer. It asks the user how many chips they want to wager. It must be greater than 0, and less than or equal to the bank. Otherwise, the user gets warned, and they are prompted until they enter a valid value.

get_slot_results() -> tuple: function

Returns 3 random reels at once. It returns them as a tuple. Notice you can return 3 integers in the same return separated by a comma. Notice in the main program, we can assign them to 3 variables when the function is called. Each reel is a random number between 1 and 10 inclusive.

get_matches(reela, reelb, reelc) -> int: function

This function returns 3 if all 3 reels match, 2 if there are 2 matching reels, and 0 otherwise.

get_bank() -> int: function

Ask the user how many chips they want to start with. It must be greater than 0 and less than 101. If not a valid entry then it will warn the user and keep asking until they enter valid input and return the valid input.

get_payout(wager, matches): function

Returns the payout based on the # of matches and the wager. It assumes that the bank did not automatically initially subtract the wager. If the matches is 3, then the payout is 10 times the wager and subtract the wager. If the matches is 2 then the payout is 3 times the wager, and subtract the wager. If the matches is 0, then the payout is the negative of the wager.

Finishing the program.

You will have to modify the main program in order to finish, but most of the work is done, and you can see how functions make the main program much smaller, more compact, easier to comprehend, and to debug.

Example

```
How many chips do you want to start with? ==> -1
Too low a value, you can only choose 1 - 100 chips
How many chips do you want to start with? ==> 500
Too high a value, you can only choose 1 - 100 chips
How many chips do you want to start with? ==> 10
How many chips do you want to wager? ==> -1
The wager amount must be greater than 0. Please enter again.
How many chips do you want to wager? ==> 12
The wager amount cannot be greater than how much you have. 10
How many chips do you want to wager? ==> 5
Your spin 4 2 9
You matched 0 reels
You won/lost -5
Current bank 5

How many chips do you want to wager? ==> 1
Your spin 9 6 6
You matched 2 reels
You won/lost 2
Current bank 7

How many chips do you want to wager? ==> 8
The wager amount cannot be greater than how much you have. 7
How many chips do you want to wager? ==> 7
Your spin 3 3 3
You matched 3 reels
You won/lost 63
Current bank 70

How many chips do you want to wager? ==> 70
Your spin 6 1 3
You matched 0 reels
You won/lost -70
Current bank 0

You lost all 10 in 4 spins
The most chips you had was 70
Do you want to play again? ==> e
```

```
You must enter Y/YES/N/NO to continue. Please try again  
Do you want to play again? ==> n  
>>>
```

Grading and Turning In

Turn in your program before the end of the lab. Only upload the Lab04.py file, as other files will be ignored.

Property of UMKC