Carrie Lindeman

Uptown Func You Up

Functions are where it is at. They are basically little minions who do tasks for you so that you don't have to **#blessed** . Think about it like when you convince your little sibling to do your chores for you. You don't care how they do it, you just care that it gets done. Functions have a very simple workflow of input, calculation, and output. For example, the input for the function *makeBrotherVacuum()* might be a pack of 

and the output is an only moderately cleaner living room floor.

Functions are here to make your life easier. If while you are writing your code you find yourself hitting **crtl-C** and **crtl-V** more than a few times then it is time to bring the func.

In this lesson we will cover what you need to know in order to use functions and how to actually create them yourself. Specifically we are going to work in the  Python language, not to be confused with the **parseltongue** language. **Yer a coder, Harry!**

What we are going to learn:

1. **Calling Functions** - Okay, so how do I make these function things happen?
2. **Passing Parameters** - These function need input? In where? In what?
3. **Catching Returns** - "If you fall I will catch you I'll be waiting, Time after time!" ♪
4. **Defining Functions** - Excuse me, I thought the point of functions was that I did LESS work! Now I have to write my own?

☎ Calling Functions

Lucky for you, calling functions is easier than calling to make your own dentist appointment. In fact, you've called a function before, like, a lot. So you know that magical spell that makes things print?  Well that is a function! You know what the input is and what to expect when you type it but **Oprah** knows you have no idea how it works.

```
print("Hey, I just met you")
print("And this is craaaaazy")
print("But here's my number")
print("Calling Functions")
```

Where you call a function is very important. The thing a function is supposed to do isn't going to happen until the line where the function is called. You know you've spied a function call when it ends with an open and close parenthesis. These parenthesis aren't always like your gym teacher's heart, empty.

Passing Parameters

You've actually already done this too. You're surprisingly good at this. Whenever you type something between the parentheses of a function call you are passing it a parameter. You can pass a function a literal or a variable. For example, the function *fridayNight()* takes two parameters. The code below passes a String variable, *tv*, and the String literal "chill". This code is totally valid (assuming they are watching something good, like OitNB).

```
tv = "Netflix"

fridayNight(tv, "chill")
```

Carrie Lindeman

Uptown Func You Up

Think of parameters like pockets on a pair of particularly *fashionable* cargo shorts. There is a set number of pockets and each pocket has a purpose. You have to pass the right number of items so that the function works.

Catching Returns

So far all of the functions we have looked at have *done* something, but what about functions that *return* something. Using what we know about assignment, we can assign a returned value to a variable.

Yes, catching this return value might be the only exercise  you get all day, and yes, you do deserve a varsity letter.

In the code below you will see a function that takes one parameter, a number representing your age. Depending on your age this function either returns the String "Yes" or "No". If we didn't catch these values then they would just go spinning into the dark void of computer's memory. But we are better than that. We assigned the returned value to the variable answer.

```
def canRentCar(age):
    if(age >= 21):
        return "Yes"
    else:
        return "No"

myAge = 17
answer = canRentCar(myAge)

print("I am " + str(myAge) + " years old. Can I rent a car? " + answer)
```

```
=============
I am 17 years old. Can I rent a car? No
>>>
```

Copy this program and getting it working. Remember that indentation matters.

Defining Functions

Easier than defining the relationship.

So far you are pretty pumped about functions but it is time to get real. If functions are going to be helpful at all, we have to be able to make our own. The syntax starts with a definition statement.

```
def example():
```

After the key term "def" you name your function. Try to come up with more meaningful function names than you come up with for your Pokémon on Pokémon Go  . In the parenthesis put in comma separated variables for your parameters. These variables are used within the function for your calculations. Whatever values get passed in the parameters when this function is called will be assigned to these variables. Easy as Py-thon… sry.

```
def example(val1, val2):
```

Carrie Lindeman

Uptown Func You Up

Write the meat of your program, whatever calculations you want done, (or brother vacuuming) make it happen!

```
val3 = val1 + val2
```

If you want this function to return a value you make a return statement.

```
return val3
```

You did it! You made a function! If when you run this function you get 5, then you even made a function right!

```
def example(val1, val2):
    val3 = val1 + val2
    return val3

print(example(2,3))
```

Let's up our game.

Code's the Reason for the Teardrops on my Keyboard 

Whether you love her, you hate her, or you are Kanye West, you have to admit that Taylor Swift is an extremely tall pop icon. And honestly, who can't get behind Taylor's alter ego, T-Swizzle. We are going to write a program that takes user input of a first name and a last name and prints the T-Swizzle version of the name.

Follow these instructions to make the T-Swizzle function.

1. Define a function with 2 parameters.
    a. Parameter1: first name
    b. Parameter2: last name
2. Write the calculations.
    a. Create a new string that takes the first letter of the first and last name and add "izzle" to the end.
    b. Example: Bob Dylan becomes "B-Dizzle"
3. Write a return statement.
    a. Return the "izzle"-ed
4. Call the function
    a. Call and catch the return of your function
    b. Print the caught value
5. Consider yourself to have swag

** If you are feeling fancy try and take user input for the first and last name!

Carrie Lindeman

Uptown Func You Up

Assignment Solution for Python 2.7

```
'''
TSwizzle - User creates a new Taylor Swift inspired name.
Date: 9/02/2016
Author: Carrie Lindeman
'''

#input: string first name, string last name
#calculation: creates new TSwizzle name
#output: string new name

def TSwizzle(fname, lname):
    name = fname[0] + "-" + lname[0] + "izzle"
    return name

firstName = raw_input("First name: ")
lastName = raw_input("Last name: ")

swizzleName = TSwizzle(firstName, lastName)

print(swizzleName)
```