

Introduction to Python

Learning a Not-So-Foreign Language

Translate the following expressions.

	<code>foo == 5</code>	
	<code>foo = 5</code>	
	<code>foo += 5</code>	<code>for i in range(1, 11):</code>
	<code>len("word")</code>	<code>print(i)</code>
	<code>"word"[2]</code>	
	<code>"h"+"i"</code>	

```
def distinct(string):  
    for i in range(0, len(string) - 1) :  
        for j in range(i + 1, len(string)) :  
            if string[i] == string[j]:  
                return False  
    return True
```

What is the difference between `print` and `return`?

Print is similar to "say" in Snap: it just displays a value. Return on the other hand, is like report: it attributes a value to a function call.

Monty Python's Practice Circus

1. Fill in the fizzbuzz function so that it does the following:

- Print out the numbers 1 through 100
- If the number is divisible by 3, print "fizz".
- If it is divisible by 5, print "buzz".
- If it is divisible by 15, print "fizzbuzz".

```
def fizzbuzz():  
    for i in range(1, 101):  
        if i % 15 == 0:  
            print("fizzbuzz")  
        elif i % 5 == 0:  
            print("buzz")  
        elif i % 3 == 0:  
            print("fizz")  
        else:  
            print(i)
```

```
Alternate solution:  
def fizzbuzz():  
    s = ""  
    for i in range(1, 101):  
        if i % 3 == 0:  
            s += "fizz"  
        if i % 5 == 0:  
            s += "buzz"  
        if s == "":  
            print(i)  
        else:  
            print(s)
```

2. Write a function that will count the number of times a letter appears in a string. For example, if the string was "tinny", and we were going to find the number of times the letter "n" appears in the string, our function will return 2. If we tried to find the number of times "d" appeared in the string, our function would return 0.

Try writing this iteratively and recursively. Finish one way? Try it the other way!

<pre>def find_num_of_letters (str, letter): #iterative count = 0 for item in str: if item == letter: count += 1 return count</pre>	<pre>def find_num_of_letters (str, letter): #recursive if len(str) == 0: return 0 elif str[0] == letter: return 1 + find_num_of_letters(str[1:], letter) else: return find_num_of_letters(str[1:], letter)</pre>
--	--

Bugs? What Bugs?

We decide to write a function called `floor_divide` which will report the number of times a smaller number can fit into a bigger number. We know our algorithm is right but we notice there are a lot of Python syntax bugs in our code. Identify and fix them!

```
def floor_divide(big_num, small_num):  
    if small_num = 0: if small_num == 0: String must be in  
        return You cannot divide by zero! quotes:  
    current_num = small_num return "You cannot  
    num_times = 0 divide by zero!"  
    while current_num <= big_num needs colon after big_num  
        current_num = current_num + small_num  
        num_times = num_times + 1  
    return num_times
```

Extra for Experts: Falling Factorial

Write a function `falling`, which is a "falling" factorial that takes two arguments, `n` and `k`, and returns the product of `k` consecutive numbers, starting from `n` and working downwards. For example, `falling(10, 3)` will return 720 ($10 * 9 * 8$).

```
def falling(n, k):  
    if k == 0:  
        return 1  
    else:  
        return n * falling(n - 1, k - 1)
```