Discussion 11: Python I

Learning a Not-So-Foreign Language

1. Translate the following expressions from Snap! to Python:


```
foo == 5
```


```
foo = 5
```

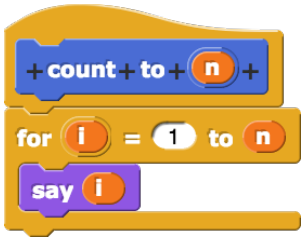

```
foo += 5
```


```
foo = 'foo'
```


```
len('word')
```


```
'word'[2]
```


```
'hello ' + 'world'
```



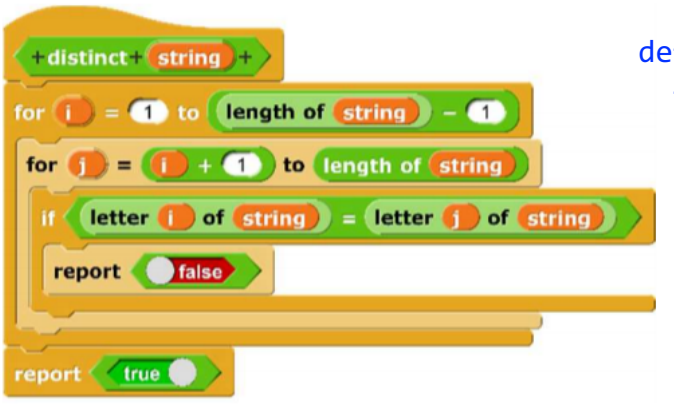

```
def count_to(n):
    for i in range(1, n+1):
        print(i)
```

2. Translate the following blocks of code from Snap! to Python line by line:

a.


```
def is_long(word):
    if len(word) > 5:
        return True
    else:
        return False
```

b.




```
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
```

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

`print` is like `say` in Snap: it displays a value, but nothing else. `return` is like `report` in Snap: it ascribes a value to a function call.

More Practice

1. Define the function Fizzbuzz 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 % 3 == 0:
               print(‘fizz’)
          elif i % 5 == 0:
               print(‘buzz’)
          else:
               print(i)
```

Notes: Pay attention to the order of these if statements. The function won’t necessarily work if you change the order (try playing around with this to see what happens!). Additionally, this code won’t work properly if we change it as it is to have “if” instead of “elif.”

2. Write a function that counts the number of times a given letter appears in a given string. Try writing this iteratively and recursively! Finish one way? Try the other way!

```
def find_num_letters(letter, str):
    num_letters = 0
    for ltr in str:
        if ltr == letter:
            num_letters += 1
    return num_letters
```

```
def find_num_letters(letter, str):
  if len(str) == 0:
    return 0
  if str[0] == letter:
    return 1 + find_num_letters(letter, str[1:])
  else:
    return find_num_letters(letter, str[1:])
```

Errors Galore

We wrote a function called `floor divide`, which divides a number, `big_num`, by another number, `small_num`, and then reports the answer rounded down to the nearest whole number. Unfortunately, there are a lot of syntax errors! Identify and fix all of the bugs in the code below.

```
def floor_divide(big_num, small_num):
    if small_num = 0: # should be ==
        return You cannot divide by zero! # should be in quotes
    current_num = small_num
    num_times = 0
```

```
while current_num <= big_num:
    current_num = current_num + small_num
    num_times = num_times + 1
report num_times # should be return not report
```