

Discussion 10: Intro to Python

Warm-Up

1. What is the difference between `print` and `return` in Python?

`print` displays a value, like `say in Snap!`, and `return` gives the output of a function, like `return in Snap!`

2. Once you write Python code, how do you run it?

You first save it as a `.py` file. Then you open your terminal, use `cd` to enter the directory (folder) in which the file is saved, and type `python3 <filename>` or `python3 -i <filename>`.

3. What is the difference between running `python3`, `python3 <filename>`, and `python3 -i <filename>`? What do each of them do?

`python3`: Opens a Python interpreter in your terminal to write/run Python

`python3 <filename>`: Runs a Python file


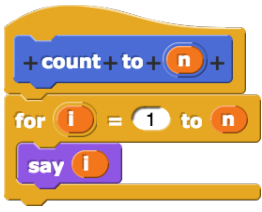








`python3 -i <filename>`: Runs a Python file and then keeps Python open

4. How are `while` loops in Python similar to `repeat until` loops in Snap? How do they differ?

Both `repeat` based on a condition. `while` loops in Python repeat until their condition is false, and `repeat until` loops repeat until their condition is true.

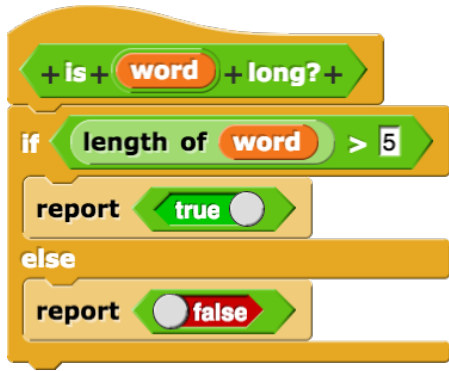
Learning a Not-So-Foreign Language

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

	<code>foo == 5</code>	 <pre>def count_to(n): for i in range(1, n + 1): print(i)</pre>
	<code>foo = 5</code>	
	<code>foo += 5</code>	
	<code>foo = "foo"</code>	
	<code>len("word")</code>	
	<code>"word"[2]</code>	
	<code>"hello " + "world"</code>	
	<code>"word"[1:]</code>	
	<code>var[:-1]</code>	

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

a.



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

b.



```
def slice(word):  
    return word[2:-2]
```

Let's Write Some Python

1. Write a function that counts the number of times a given letter appears in a given string. Try writing this both iteratively and recursively!

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

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

2. Define the function Fizzbuzz so that it does the following:

- Iterates through the numbers 1 – 100, and for each number:
 - Prints “fizz” if it is divisible by 3.
 - Prints “buzz” if it is divisible by 5.
 - Prints “fizzbuzz” (and *not* “fizz” or “buzz”) if it is divisible by 15.
 - Prints the number otherwise.

```
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)
```

Errors Galore

We wrote the function `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, it has a lot of syntax errors and doesn't run. Identify and fix the syntax errors in the code below:

```
def floor_divide(big_num, small_num):
    if small_num == 0:
        return "You cannot divide by zero!" should be indented
    current_num = small_num
    num_times = 0
    while current_num <= big_num:
        current_num += small_num
        num_times += 1
    return num_times
```

Changes:

1. Line 2 missing double equals
2. Line 3 should be indented, string missing quotes
3. Line 5, 8, 9: variable names can't have spaces, added underscore
4. Line 6: while condition missing colon after
5. Line 9: Should be return, not report