Discussion 8:

Intro to Python Warm-Up 1.

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

ANSWER: print is use for simple display and if we use reasult futher return is use.

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

ANSWER:l use vs code. In vs code simply after write a code press Control + F5

1. What is the difference between running python3, python3<filename> , and python3 –i <filename>? What do each of them do?

ANSWER:

* python3: starts up a blank Python session
* python3 <filename>: runs the code in the provided file; immediately returns to the terminal
* python3 -i <filename>: runs your python script, opening an interactive session

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

ANSWER:working concept is same but the only difference is in python while loop run if condition is ture but in snap its runs if condition is false.

Learning a Not-So-Foreign Language

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

ANSWER:

1. foo == 5
2. foo = 5
3. foo += 5
4. foo = 'foo'
5. len('word')
6. 'word'[2]
7. 'hello ' + 'world'
8. 'word'[1:]
9. [1, 2, 3]
10. for item in [1, 2, 3]:

print(item)

1. while len(lst) != 0:

lst.pop()

1. [1, 2, 3][0]
2. [1, 2, 3][-1]
3. len([1, 2, 3])
4. [1, 2, 3].append(4)
5. [1, 2, 3].pop()\*
6. [1, 2, 3].pop(0)\*
7. def count\_to(n):

for i in range (1, n+1):

print(i)

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

ANSWER:

1. def is\_long(word):

if len(word) > 5:

return True

else: return False

1. 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

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!

ANSWER:

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:])

1. 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.

ANSWER:

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:

ANSWER:

def floor\_divide(big\_num, small\_num):

if small\_num = 0: #should be ==

return You cannot divide by zero! #should be in quotes, and indented

current\_num = small\_num

num times = 0 #should be num\_times

while current\_num <= big\_num #missing :

current\_num += small\_num

num times += 1 #should be num\_times

report num times #should be return, not report