In [1]: `print(3*"Welcome to section!")`

Welcome to section!Welcome to section!Welcome to section!

In [58]: # Try these exercises out! Create the result strings below the prompt using the following strings. Try to do it in as few Statements as possible

str1 = 'go cs4!'  
str2 = 'puppies'  
str3 = 'babies'  
str4 = 'cit'

# bop - insert code below
bop = str3[0] + str1[1] + str2[0]
print(bop)

# apps
apps = str3[1] + str2[::3]
print(apps)

# sect - do this in two operations
sect = str3[-1:-3:-1] + str4[::2]
print(sect)

# guppies 4 bb - This is hard!
print(long_string)

bop  
apps  
sect  
guppies 4 bb

In [27]: # Let's trace through a program!

def simple_program(a_list, a=None, b=None):
   """Takes a slice of a_list (containing integers) defined from [a:b] and adds the f
It then adds that new value to a and b and returns the product of these values."

spliced_list = a_list[a:b]
print(spliced_list)
c=spliced_list[0]
new_element = a_list[0]+a_list[-1]
if a==None:
aprint a=0
a = a + new_element
b = b + new_element
final_product = a*b
return final_product

In [28]: #Let's do some examples
    print(simple_program([1,2,3,4,5,6,7,8], 1, 4))
[0, 1]
195

In [17]: print(simple_program([0,1,1,2,3,5,8,13], 2,5))
    #spliced_list=[1,2,3]
    #new_element=13
    #a=15
    #b=18
    #final_product=270
270

In [ ]: print(simple_program([0,1,1,2,3,5,8,13], 4,5))
In [ ]: print(simple_program([0,1,1,2,3,5,8,13], b=2))
In [ ]: print(simple_program([0,1,1,2,3,5,8,13], a=2))

What's one easy way to trace the program while it's running? Printing! Printing lets the user or the programmer see the values of the program. Lets try an example.

In [59]: def reduced_mass_function(a,b):
    """
    Returns the reduced mass of two objects. a and b are integers.
    """
    #Calculate the numerator first
    numerator= a*b
    print(numerator)
    full_value = numerator/(a+b)
    print(full_value)
    return full_value

def reduced_weight_diatomic(compound_name, mass1, mass2):
    reduced_mass = reduced_mass_function(mass1, mass2)
    print('The reduced mass of', compound_name, 'is', reduced_mass)
In [60]: reduced_weight_diatomic('hydrogen', 1, 1)

1
0.5
The reduced mass of hydrogen is 0.5

In [61]: # Let's write some basic functions now!

def reverse(string):
    ""
    Returns the string backwards. For instance, "Hello" becomes "olleH". Try writing this in one line!
    ""
    return string[::-1]

reverse("Hello")

Out[61]: 'olleH'

In [62]: def repeat(string, i):
    ""
    Given a string and a number, repeat the string that number of times. Print out your answer.
    ""
    return string*i

repeat("CS4", 4)

Out[62]: 'CS4CS4CS4CS4'

In [63]: def halfReverse(string):
    ""
    Take the first half of the string and reverse it. Then, return the reversed half and the normal second half as one.
    Try using the reverse function we wrote earlier.
    Example: "Hello" -> "eHllo"
    ""
    half_index = len(string)//2
    half_string = string[half_index-1::-1]
    return half_string + string[half_index:]

halfReverse("Hello")

Out[63]: 'eHllo'