*Python Loops Study Sheet Last Update 24 Aug ‘23*

basic for loop syntax: for {variable} in {iterable}:

basic while loop syntax: while {bool condition}:

stop loop execution: break (breaks out of innermost loop)

what does *continue* do: skips current iteration of loop and ‘restarts’ next iteration of loop

else in loops: runs after final iteration of for loop

or when condition becomes false for while loop

(does not run if loop is broken)

emulate a do-while loop: while True:

#loop contains some condition #that will cause a break

alternatively, simply run first iteration of fcn/operations before the while loop

use range function to:

count from 0 to 9: range(10)

count from 1 to 7: range(1, 8)

count from 2 to 6 (inclusive) by 2: range(2, 7, 2)

count from 3 to -3 backwards: range(3, -4, -1)

range input restrictions: must be integers; can’t count by decimal vals

what does this code create?

my\_list = [‘alpha’, ‘bravo’, ‘charlie’]

enumerate(my\_list, 5)

creates an iterator object, creating the following tuples

counts starting from 5

(5, ‘alpha’) (6, ‘bravo’) (7, ‘charlie’) (conv to list to use)