Lesson 27:

1. Quiz: Pick One

def pick\_one(value\_blooean, value\_one, value\_two):

return value\_one if value\_blooean else value\_two

1. Quiz: Triangular Numbers

def triangular(n):

summation = 0

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

summation += x

return summation

1. Quiz: Linear Time

Procedures one and two are linear time. No double loops.

1. Quiz: Remove Tags

def remove\_tags(input\_string):

# Tried to do this without Regular Expressions or .find() / .replace()

tagless\_string = ''

capture = True

for letter in input\_string:

if letter == '<':

capture = False

tagless\_string += ' '

if capture:

tagless\_string += letter

if letter == '>':

capture = True

return tagless\_string.split()

1. Quiz: Date Converter

def date\_converter(language, date):

date\_list = date.split('/')

day = date\_list[1]

month = language[int(date\_list[0])]

year = date\_list[2]

return day + ' ' + month + ' ' + year

1. Quiz: Termination

Not proc2 because n could be odd and never reach n == 0.

1. Quiz: Find and Replace

# I would have just used 4 inputs rather than a separate function.

def make\_converter(match, replacement):

return [match, replacement]

def apply\_converter(converter, string):

previous\_string = None

while previous\_string != string:

previous\_string = string

location\_of\_match = string.find(converter[0])

if location\_of\_match >= 0:

string = string[:location\_of\_match] + converter[1] + string[location\_of\_match + len(converter[0]):]

return string

1. Quiz: Longest Repetition

def longest\_repetition(input\_list):

if not input\_list:

return None

else:

chain = 0

element = input\_list[0]

max\_chain = 0

max\_element = input\_list[0]

for item in input\_list:

if element == item:

chain += 1

if chain > max\_chain:

max\_chain = chain

max\_element = element

else:

chain = 1

element = item

return max\_element

1. Quiz: Deep Reverse

def is\_list(p):

return isinstance(p, list)

def deep\_reverse(input\_list):

return\_list = list()

for item in input\_list:

if is\_list(item):

return\_list.append(deep\_reverse(item))

else:

return\_list.append(item)

return return\_list[::-1]