Csc401 midterm notes4

4.24

def cheer(mascot):

mascot=mascot.upper()

mc = mascot.capitalize()

print('How do you spell winner?')

print('I know, I know!')

for i in list(mascot):

print(i, end=' ')

print("!\nAnd that's how you spell winner!")

print('Go '+mc+'!')

-----------------------------------------------------------------------------

4.25

def vowelCount(strinput):

strinput = strinput.casefold()

vcheck = 'aeiou'

for v in strinput:

if v in vcheck:

ac=strinput.count('a')

ec=strinput.count('e')

ic=strinput.count('i')

oc = strinput.count('o')

uc=strinput.count('u')

return ac,ec,ic,oc,uc

>>> vowelCount('yes')

(0, 1, 0, 0, 0)

>>> vowelCount('oohawehagsrhf')

(2, 1, 0, 2, 0)

------------------------------------------------------------------------------------------------------

#4.26

def crypto(fILe):

msg = open('crypto.txt', 'r')

lines = msg.read()

newmsg = lines.replace('secret', 'xxxxxx')

print(newmsg)

msg.close()

#4.28

def links(fIle):

fILe = open(fIle, 'r')

char = fILe.read()

lct=char.count('</a>')#counts the ending anchor tags for hyperlinks

#print(lct)

return lct

fILe.close()

Write a function doubleVowel() that, given a word,

returns either True or False indicating whether

the word contains two adjacent vowels.

A vowel is one of the letters: a,e,i,o,u.

>>> doubleVowel('banana')

False

>>> doubleVowel('snail')

True

>>> doubleVowel('Eel')

True

>>> doubleVowel('aArdvark')

True

def doubleVowel(w):

v = 'aeiouAEIOU'

for i in range( len(w)-1):

if w[i]in v and w[i+1] in v:

return True

return False

split method examples:

'You killed my father. Prepare to die.'.split()

['You', 'killed', 'my', 'father.', 'Prepare', 'to', 'die.']

'quoth the raven, nevermore.'.split(',')

['quoth the raven', ' nevermore.']

index:

>>> [23,36,77,88,9,26.6,9999,25][3]

88

#5.39--needs work

def exclamation(wd):

n\_wd=wd

t=''

for i in wd:#goes through the list

if i in 'aeiouAEIOU':#look for the vowel

if wd.count(i)>4:

t= t+i

#wd=wd.replace(t,nv)

#print('loop '+i)

else:

nv=i\*4

wd=wd.replace(i,nv)

return wd+'!'

iteration review

standard) iteration in python uses iterators, performs a "sequential assignment"

for item in lst:

for char in string:

for i in range():

def oddEven(numlst):

for num in numlst:

if num%2==0:#even

print('even')

else:

print('odd')

#why no use return

#because return terminates the function, hence all loops

#so you cant return multiple times

''' below is no good:

def oddEven(numlst):

for num in numlst:##return makes things stop HERE, so an if/else with these don't work

if num%2==0:#even

return('even')

else:

return('odd')

'''

#MAKE SURE A RETURN IN A LOOP WORKS IN CONTEXT

#be caeful about returns in loops

Accumulator pattern:

Accumulator: computes an accumulated value/statistic/container through iteration. Each iteration computes a partial/running value for the portion of data already seen.

ex:

sum, product,count, max, min

return multiple item, list, str

four step process:-------------------------------------------------------------

0) set up iteration, visit all items of iterest, write "print"version fisrt

1) intitilaize running value before loop

2)accumulate in the loop

3)return after the loop

'''

def total(numlst):

#step 1-intialize

runningTotal = 0

for num in numlst:

#step 2-accumulate

runningTotal += num

# is equivalent to the following

#runningTotal = runningTotal+num

#print(num, runningTotal)-------didnt need this, just helps to see where you are

#3-return after loop

return runningTotal

aside: <op>=

<var> <op>= <expr>

same as (short hand for)

<var> = <var> <op> <expr>

works with most to all operators

ex:

make this work:

>>>>countNegs([2,-55,99,-23,44,2])

2

count:

'''

def countNegs(numlst):

#1

count=0

#0

for num in numlst:

if num<0:

#print(num)

#2

count+=1 #don't print, count!

#3

return count

list review

operators:

+ ,- concatenate

[i]-index, returns item

[start:stop: step] slice, returns sublist

In, not in

Functions:

Len

Max,min

Sum

Methods

Append(item)-adds item to end

Pop()-removes and returns last item

Pop(index)

Count(item)

Remove(item)

Index(item)-returns location

Sort()

Reverse()

Make acronym code:

>>> def makeAcro(wc):

wc=wc.split()#splits by space value making it split by word

for i in range(len(wc)):

print(wc[i][0],end=' ')

>>> makeAcro('Federal Bureau Investigation')

F B I

>>> makeAcro('Central Intelligence Agency')

C I A