

## midtermCheatSheet

```
#midterm cheat sheet
#Eileen Connor
#10/10/2018

# gets a number and save to variable n
n = eval( input('Enter a number: ') )

s = "Hello, how, are you!?"
>>> s.replace(",","").replace("?", "")

s = "Hello, how, !.;;...are you!?"
>>> for punct in ".,!?;:":
    s = s.replace(punct, "")
    return

# 2d search
def contains( target, lst ):

    for row in lst:
        for item in row:
            # look for target
            if item.lower()==target.lower():
                #print( item )
                return True
    return False

####
def findTreasure( tmap ):

    locations = []

    lines = tmap.split() # list of strs
    # visit all the locations on the map
    for i in range(len(lines)):
        # iterate over the str lines[i]
        for j in range(len(lines[i])):
            if lines[i][j]=='X': #treasure
                #print( i,j, lines[i][j] )
                locations.append( (i,j) )
    return locations

#####
def acronym( phrase ):

    acro = ""
    for word in phrase.split():
        #print( word[0],end='' )
        acro += word[0]
```

```
return acro.upper()
# print( acro.upper() )
```

```
#####
def factorial(n):
```

```
    product = 1
    for i in range(2,n+1):
        #print(i)
        product *= i
    return product
```

```
#all upper case letters are less than all lower case letters
```

```
#slicing
    s[start:stop] - substring of s
>>> 'apple' < 'pear'
True
>>> 'Pear' < 'apple'
True
>>> 'Z' > 'a'
False
```

```
t = 'railroad'
>>> t[0:4]
'rail'
```

```
#lists:
append(item) - add item to the end
pop() - removes and returns last item
count(item)
index(item) - index of first occurrence
remove(item) - removes (one copy)
reverse()
sort()
```

```
#####
print( arguments, sep=' ', end = '\n' )
    by default sep = ' '
        end = '\n'
```

```
#####
open(filename,mode='r')
```

```
modes
    'r'ead - the default
```

## midtermCheatSheet

```
'w'rite  
'a'ppend
```

### file methods

```
read() - reads contents as a single str  
readlines() - read contents as a list  
           of strings, one per line
```

```
write(s) - OVERWRITE writes a single string s  
           to a file, less friendly than print  
           write() argument must be str, not int
```

```
>>> outfile.write( str(5) )
```

```
#####
```

```
def numChars(filename):  
    return len( open(filename).read() )
```

```
#####
```

```
eval( open('numbers.csv').readlines()[1].split(',')[3])  
21
```

```
#####
```

```
def getCell(filename,row,col):  
    # the -1's are to translate from excel row,col numbering  
    # to python indices  
    return eval( open(filename).readlines()[row-1].split(',')[col-1])
```

```
#####
```

```
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
```

```
#####
```

```
import math  
def collision(x1,y1,r1,x2,y2,r2):  
    distance= math.sqrt((x2-x1)**2+(y2-y1)**2)  
    if distance > r1+r2:  
        return False  
    else:  
        return True
```

```
#####
```

```
def vowelCount(phrase):  
    #countLst indexed as a,e,i,o,u  
    countLst=[0,0,0,0,0]
```

## midtermCheatSheet

```
#vowels=""
for char in phrase:
    if char.lower() == "a":
        countLst[0]+=1
    elif char.lower()=="e":
        countLst[1]+=1
    elif char.lower()=="i":
        countLst[2]+=1
    elif char.lower()=="o":
        countLst[3]+=1
    elif char.lower()=="u":
        countLst[4]+=1
print("a, e, i, o, and u appear, respectively,
      " + str(countLst).strip('[]') + " times.")
```

```
#####
def crypto(someFile):
    file=open(someFile)
    print(file.read().replace('secret','xxxxxx'))
    file.close()
```

```
#####
def exclamation(someStr):
    newLst=""

    for vowel in someStr:
        if vowel in "aeiouAEIOU":
            for i in range(4):
                newLst+=vowel
        else:
            newLst+=vowel
    return newLst+'!'
```

```
#####
def geometric(intLst):
    ratios=[]
    for i in range(len(intLst)-1):
        ratios.append(intLst[i+1]/intLst[i])
    for i in range(len(ratios)-1):
        if ratios[i] != ratios[i+1]:
            return False
    return True
```

```
#####
```

```
def prime(num):
    count=0
```

```
for i in range(num+1):
    if i>=1:
        if num%i==0:
            count+=1
if count>2:
    return False
return True
```