Filip Konior sprawozdanie nr6

# Write  
# Using sys for command-line tools.  
  
  
#add.py  
# import sys  
def adding(\*y):  
 sum = 0.0  
 for x in y:  
 try:  
 sum+=float(x)  
 except Exception:  
 continue  
 return sum  
  
  
print(adding(4,1))  
print(adding(8,6,7,5,3,0,9))  
print(adding(17, 38, "Hey wassup", "hello"))  
  
  
# if len(sys.argv) <= 1:  
# print(" Usage: python3 add.py < nums >\n Add some numbers together")  
# else:  
# print(adding(\*sys.argv))  
  
#  
print()  
  
#Extracting data with re  
import re  
import collections  
import string  
import itertools  
  
#Regex Crossword Checker  
def regex\_crossword\_check(horizontal\_patterns, vertical\_patterns, candidate, alphabet=string.ascii\_uppercase):  
 # Check horizontal clues  
 for pattern, horiz\_line in zip(horizontal\_patterns, candidate):  
 line = ''.join(horiz\_line)  
 if re.match(pattern, line) is None:  
 return False  
  
 # Check vertical clues  
 for pattern, vert\_line in zip(vertical\_patterns, zip(\*candidate)):  
 line = ''.join(vert\_line)  
 if re.match(pattern, line) is None:  
 return False  
  
 return True  
  
def test\_regex\_crossword\_check():  
 horiz = [r'HE|LL|O+', r'[PLEASE]+']  
 vert = [r'[^SPEAK]+', r'EP|IP|EF']  
 candidate = [  
 ['H', 'E'],  
 ['L', 'P']  
 ]  
 print(regex\_crossword\_check(horiz, vert, candidate))  
  
 horiz = [r'(Y|F)(.)\2[DAF]\1', r'(U|O|I)\*T[FRO]+', r'[KANE]\*[GIN]\*']  
 vert = [r'(FI|A)+', r'(YE|OT)K', r'(.)[IF]+', r'[NODE]+', r'(FY|F|RG)+']  
 candidate = [  
 ['F', 'O', 'O', 'D', 'F'],  
 ['I', 'T', 'F', 'O', 'R'],  
 ['A', 'K', 'I', 'N', 'G']  
 ]  
  
  
  
  
  
  
test\_regex\_crossword\_check()  
  
#  
print()  
#Working with itertools  
  
def tabulate(f):  
 return map(f, itertools.count())  
  
sqgen = tabulate(lambda x: x \*\* 2)  
print(next(sqgen)) # => 0  
print(next(sqgen)) # => 1  
print(next(sqgen)) # => 2  
print(next(sqgen)) # => 4  
print(next(sqgen)) # => 9