Line.strip()

去除前后空白(括号可以去除指定的东西，比如0)

Line.lstrip()

Line.rstrip()

Find()找不到返回-1

Rfind()从右向左找

读取文件

With open as file:

For line in file:

If nor line.isspace()://判断不能为空

Print(line.strip())

先split，变成list

返回list后索引

Excel：

From collection import defaultdict

With open( ) as file:

Counter=defaultdict(list)

For line in file:

If line.isspace():

Continue

If line.startwith(‘D’):

continue

A,b,c=line.split(‘.’)

Y,n,d=a.split(‘-‘)

If int(y) == year”

Counter[]

递归

找出stop case/base case

阶乘

F

Numpy

import numpy as np

# 创建一维数组

arr1d = np.array([1, 2, 3, 4, 5])

# 创建二维数组

arr2d = np.array([[1, 2, 3], [4, 5, 6]])

# 创建全零数组

zeros\_arr = np.zeros((3, 4))

# 创建全一数组

ones\_arr = np.ones((2, 3))

# 创建等间隔数组

range\_arr = np.arange(0, 10, 2)

# 创建均匀分布的数组

uniform\_arr = np.linspace(0, 1, 5)

# 数组形状

print(arr2d.shape)

# 数组维度

print(arr2d.ndim)

# 数组元素类型

print(arr2d.dtype)

# 数组转置

transposed\_arr = arr2d.T

# 数组索引和切片

print(arr2d[1, 2]) # 获取第二行第三列的元素

print(arr2d[:, 1]) # 获取所有行的第二列

# 数组运算

arr\_sum = arr1d + 2 # 所有元素加2

arr\_product = arr1d \* 3 # 所有元素乘以3

1. zip 函数：

zip 函数将传递给它的多个可迭代对象中的对应元素打包成元组，返回一个生成器。它会在最短的可迭代对象耗尽时停止。

list1 = [1, 2, 3]

list2 = ['a', 'b', 'c']

result = zip(list1, list2)# result 是一个生成器，可以通过 list 转换为列表

result\_list = list(result)

# 输出：[(1, 'a'), (2, 'b'), (3, 'c')]

print(result\_list)

2. itertools.zip\_longest 函数：

itertools.zip\_longest 函数也执行类似的任务，但是在最长的可迭代对象耗尽之前，它会一直生成值。如果某个可迭代对象比其他的要短，缺失的值将被填充。

from itertools import zip\_longest

list1 = [1, 2, 3]

list2 = ['a', 'b']

result = zip\_longest(list1, list2, fillvalue='N/A')# result 是一个生成器，可以通过 list 转换为列表

result\_list = list(result)

# 输出：[(1, 'a'), (2, 'b'), (3, 'N/A')]

print(result\_list)

使用 for 循环遍历 result：

list1 = [1, 2, 3]

list2 = ['a', 'b', 'c']

result = zip(list1, list2)

for item in result:

print(item)

输出：

(1, 'a')

(2, 'b')

(3, 'c')

将 result 转换为列表：

list1 = [1, 2, 3]

list2 = ['a', 'b', 'c']

result = zip(list1, list2)

result\_list = list(result)print(result\_list)

python的文件处理

打开文件：

使用 open() 函数可以打开一个文件。它接受文件路径和打开模式作为参数。

# 打开一个文件用于读取

file = open("example.txt", "r")

# 打开一个文件用于写入

file = open("example.txt", "w")

# 打开一个文件用于追加

file = open("example.txt", "a")

# 打开一个二进制文件

file = open("example.jpg", "rb")

读取文件：

使用文件对象的 read() 方法可以读取文件的内容。可以指定读取的字节数，如果不指定，将读取整个文件。

# 读取整个文件

content = file.read()

# 读取前 n 个字节

content = file.read(100)

逐行读取：

使用 readline() 方法逐行读取文件内容。

# 逐行读取文件

line = file.readline()

while line:

print(line)

line = file.readline()

读取所有行：

使用 readlines() 方法可以读取文件中的所有行，并将其返回为一个列表。

# 读取所有行

lines = file.readlines()

for line in lines:

print(line)

写入文件：

使用文件对象的 write() 方法可以向文件中写入内容。

# 写入内容

file.write("Hello, World!\n")

file.write("Another line.")

关闭文件：

使用文件对象的 close() 方法来关闭文件。关闭文件是一个好的习惯，因为它释放了系统资源。

# 关闭文件

file.close()

使用 with 语句：

为了确保文件被正确关闭，可以使用 with 语句，它会在退出代码块后自动关闭文件。

with open("example.txt", "r") as file:

content = file.read()

print(content)# 文件自动关闭

"r+" 模式：打开文件用于读取和写入，文件必须存在，否则会引发 FileNotFoundError。

defaultdict 是 Python 标准库中 collections 模块提供的一个类，它是 dict 的一个子类。defaultdict 允许指定一个默认的值（默认工厂函数），当访问字典中不存在的键时，会使用默认值进行初始化。这在处理某些场景下非常方便，特别是在构建字典时不需要手动检查键是否存在。

以下是 defaultdict 的基本用法：

from collections import defaultdict

# 创建一个 defaultdict，指定默认值为 int 类型的 0

my\_dict = defaultdict(int)

# 访问字典中不存在的键，将使用默认值 0 进行初始化

my\_dict['a'] += 1

my\_dict['b'] += 2

print(my\_dict) # 输出：{'a': 1, 'b': 2}

在上面的例子中，defaultdict(int) 创建了一个字典，其中默认值为整数类型的 0。当对字典中不存在的键进行访问并修改时，不需要手动检查键是否存在，而是直接使用默认值进行初始化。

defaultdict 的初始化默认值可以是任何可调用对象，例如，int、float、list、lambda 函数等。

以下是一个使用 defaultdict 处理列表元素的示例：

from collections import defaultdict

# 创建一个 defaultdict，指定默认值为列表

my\_dict = defaultdict(list)

# 向字典中的列表添加元素

my\_dict['fruits'].append('apple')

my\_dict['fruits'].append('banana')

my\_dict['vegetables'].append('carrot')

print(my\_dict)# 输出：{'fruits': ['apple', 'banana'], 'vegetables': ['carrot']}

在这个例子中，defaultdict(list) 创建了一个字典，其中默认值是一个空列表。这样，当访问字典中不存在的键时，会自动将默认值初始化为一个空列表，并在该列表上进行操作。

使用内置类型作为默认值：

from collections import defaultdict

# 创建一个 defaultdict，指定默认值为整数 0

my\_dict = defaultdict(int)

# 访问字典中不存在的键，使用默认值 0 进行初始化

my\_dict['a'] += 1

my\_dict['b'] += 2

print(my\_dict) # 输出：{'a': 1, 'b': 2}

# 创建一个 defaultdict，指定默认值为数字 1 my\_dict = defaultdict(lambda: 1)

读入数量和单价，输出总价

with open("/Users/yuan/Desktop/2.txt","r+") as file:  
 value=file.readline().strip().split(" ")  
 number=file.readline().strip().split(" ")  
 results=list(zip(value,number))  
 line=""  
 for result in results:  
 product=int(result[0])\*int(result[1])  
 line=line+str(product)+" "  
 file.write("\n")  
 file.write(line)

读入文件，转置后输出

import numpy as np  
list=[]  
file=open("/Users/yuan/Desktop/1.txt","r+")  
line=file.readline()  
while line:  
 line=line.strip()  
 list.append(line.split(" "))  
 line=file.readline()  
result=np.array(list).T  
r=result.shape[0]  
c=result.shape[1]  
file.write("\n")  
for i in range(r):  
 line=""  
 for j in range(c):  
 line=line+result[i,j]+" "  
 line+="\n"  
 file.write(line)  
file.close()