

Python Basics – Count and Sort

Logical Thinking of Informatics
Lab 5

List vs. Dictionary

Feature	List	Dictionary
Structure	Linear (ordered sequence)	Associative (Key-value pairs)
Syntax	[item1,item2,item3....]	{key1:value1, key2:value2, key3:value3...}
Initialization	myList=[]	myDic={}
Accessing	myList[index]	myDic[key]
Uniqueness	No	Keys must be unique
Adding Element	myList.append(item)	myDic[key]=value
Removing	myList.remove(item)	myDic.pop(key)
Updating	myList[index]=newValue	myDic[key]=newvalue
Lookup efficiency	O(n) (linear search)	O(1) on average

Why Dictionary?



Advantages of Dictionary

- Key-value pairs
- Unique keys
- Constant-time lookups
- Flexible keys
- Nested data structures

Counting a Long List

```
longInput="0,1,3,5,1,5,3,6,8,1,4,8,2,1,3,2,5,2,4"
```

```
def readStringToDict(inputStr, splitSymbol):  
    inList=inputStr.split(splitSymbol)  
    resultDict={}  
    for element in inList:  
        if (element in resultDict.keys()):  
            resultDict[element]=resultDict[element]+1  
        else:  
            resultDict[element]=1  
    return(resultDict)  
  
inputDic=readStringToDict(longInput, ",")  
print(inputDic)
```

```
{'0': 1, '1': 4, '3': 3, '5': 3, '6': 1, '8': 2, '4': 2, '2': 3}
```

Sorting



Utilize Sort Function

```
print(inputDic.items())
```

```
dict_items([('0', 1), ('1', 4), ('3', 3), ('5', 3), ('6', 1), ('8', 2), ('4', 2), ('2', 3)])
```

```
# pair = (alphabet, count)      pair[1] = count
```

```
def sortDic(inDic):  
    sortList=[]  
    sortList=sorted(inDic.items(),key=lambda pair:pair[1])  
    return (sortList)
```

Sorted

■ Sort items

```
sorted_count1=sortDic(inputDic)  
print(sorted_count1)
```

```
[('0', 1), ('6', 1), ('8', 2), ('4', 2), ('3', 3), ('5', 3), ('2', 3), ('1', 4)]
```

■ Get top 5

- top_5_numbers = sorted_count[:5]

```
top_5_numbers = sorted_count[:5]  
print(top_5_numbers)  
print(type(top_5_numbers))
```

```
[('0', 1), ('6', 1), ('8', 2), ('4', 2), ('3', 3)]  
<class 'list'>
```


What If?

- We change `lambda pair:pair[1]` to `lambda pair:pair[0]`?

```
def sortDic(inDic):  
    sortList=[]  
    sortList=sorted(inDic.items(),key=lambda pair:pair[0])  
    return (sortList)
```

- We add “, reverse=True” after “pair:pair[1]”

```
def sortDic(inDic):  
    sortList=[]  
    sortList=sorted(inDic.items(),key=lambda pair:pair[0],reverse=True)  
    return (sortList)
```

Utilize Sort Function

```
def sortDic(inDic):  
    sortList=[]  
    sortList=sorted(inDic.items(),key=lambda pair:pair[1],reverse=True)  
    return (sortList)
```

pair = (alphabet, count) pair[1] = count

Sort from the max to min

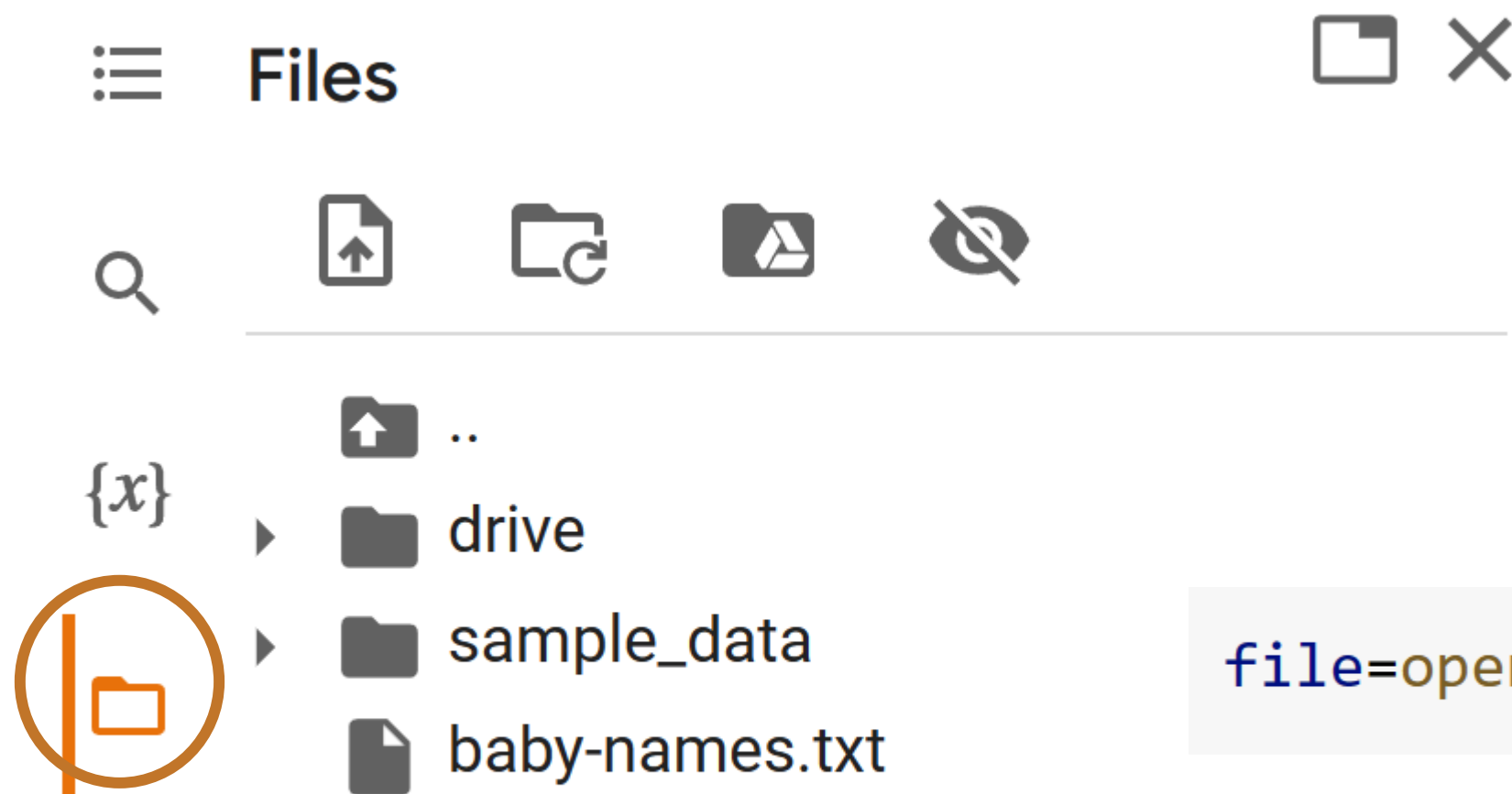
```
sorted_wordcount=sortDic(wordDic)  
print(sorted_wordcount)
```

```
[('', 12), ('of', 8), ('the', 5), ('in', 5), ('as', 5), ('science', 4),
```

Import File

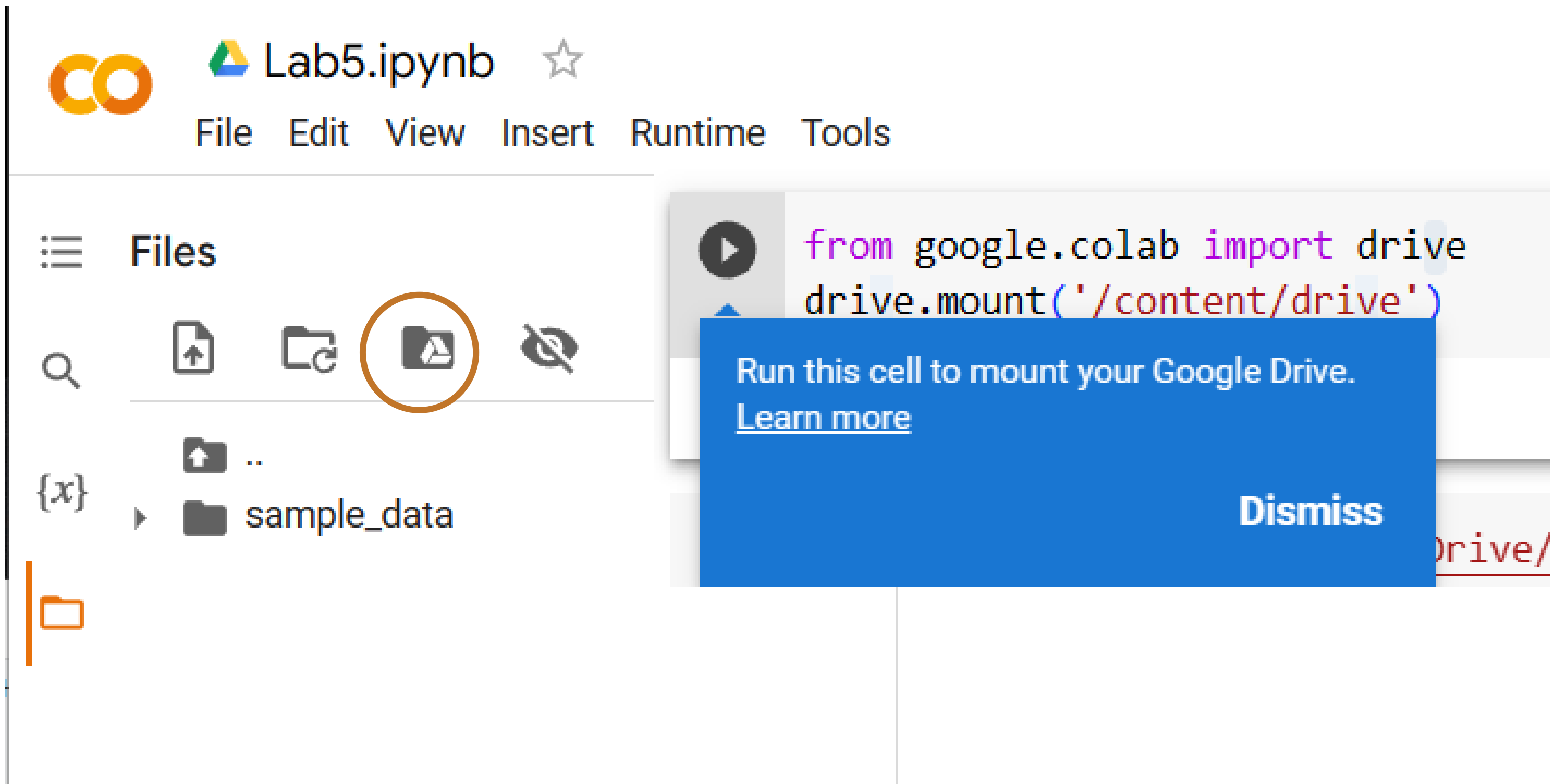


Import File into Colab



```
file=open('baby-names.txt',encoding='utf-8')
```

Mount Drive



The screenshot displays the Google Colab interface. At the top, the Colab logo is followed by the file name 'Lab5.ipynb' and a star icon. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', and 'Tools'. On the left, a 'Files' sidebar shows a search icon, upload, download, and mount icons (the mount icon is circled in orange), and a folder named 'sample_data'. The main area shows a code cell with the following Python code:

```
from google.colab import drive
drive.mount('/content/drive')
```

A blue tooltip box is overlaid on the code cell, containing the text: 'Run this cell to mount your Google Drive. [Learn more](#)'. A 'Dismiss' button is located at the bottom right of the tooltip.

Mount Drive

Perm

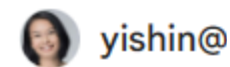
This not

Google

Google



Goog
desk
to yo
Acco



Cancel

Continue

English (United States)

Help

Privacy

Terms

English (United States)

Help

Privacy

Terms

Make sure you trust Google Drive for desktop

You may be sharing sensitive info with this site or app. Learn about how Google Drive for desktop will handle your data by reviewing its [terms of service](#) and [privacy policies](#). You can always see or remove access in your [Google Account](#).

[Learn about the risks](#)

Mounted at /content/drive

Files

{x}

..

drive

sample_data

Files

Download

Rename file

Delete file

Copy path

Refresh

Colab Notebooks

share_dataset

2024-Lab5.ipynb

Diffusion Models (1...

Diffusion Models.ip...

IBP-Summer Sessio...

Lab1.ipynb

Lab2.ipynb

Lab3.ipynb

Lab4.ipynb

Lab5-Summer.ipynb

Lab5.ipynb

Lab6.ipynb

Quick Sort.ipynb

baby-names.csv

baby-names.txt

Open

Option A

```
file=open('baby-names.txt',encoding='utf-8')
```

Option B

```
Line 1: def readFile(filename):  
Line 2:     inFile=open(filename, encoding='utf-8')  
Line 3:     resultArray=[]  
Line 4:     for line in inFile.readlines():  
Line 5:         line=line.strip()  
Line 6:         resultArray.append(line)  
Line 7:     return(resultArray)  
Line 8:  
Line 9: inputArray=readFile('baby-names.txt')
```


Name List

- Find the corresponding frequency of the input names, output a dictionary with the expected results

```
def readNameToDict(inputList):  
    resultDict={}  
    for element in inputList:  
        tempList=element.split("\t")  
        tempName=tempList[1]  
        if (tempName in resultDict.keys()):  
            resultDict[tempName]=resultDict[tempName]+float(tempList[2])  
        else:  
            resultDict[tempName]=float(tempList[2])  
    return(resultDict)
```

Sort Names

```
def sortDic(inDic):  
    sortList=[]  
    sortList=sorted(inDic.items(),key=lambda pair:pair[1],reverse=True)  
    return (sortList)
```

```
▶ returnList=sortDic(returnDic)  
print(returnList[:5])
```

```
↩ [ ('Emily', 0.156669), ('Madison', 0.12124099999999999), ('Hannah', 0.11588499999999999).
```

Word Cloud



Import

```
import pandas as pd  
import matplotlib.pyplot as plt  
from wordcloud import WordCloud
```

Plot

#create a WordCloud

```
wordcloud=WordCloud(width=800,height=400,background_color='white')
```

```
wordcloud.generate_from_frequencies(returnDic)
```

```
plt.figure(figsize=(10,5))
```

```
plt.imshow(wordcloud,interpolation='bilinear')
```

```
plt.axis('off')
```

```
plt.show()
```

Exercise

- Plot the wordCloud of an input string



Chinese Words

```
!wget -O TaipeiSansTCBeta-Regular.ttf
```

```
https://drive.google.com/uc?id=1eGAsTN1HBpJAkeVM57\_C7ccp7hbgSz3\_&export=download
```

```
from matplotlib.font_manager import fontManager
fontManager.addfont('TaipeiSansTCBeta-Regular.ttf')
mpl.rc('font', family='Taipei Sans TC Beta')
font_path = './TaipeiSansTCBeta-Regular.ttf'
wordcloud=WordCloud(font_path=font_path,max_font_size=40).generate(returnStr)
```