Clipart by: http://clipart-library.com/clipart/kiKB8kpLT.htm

L#16

eXternal Files Input/Output

Writing on External File

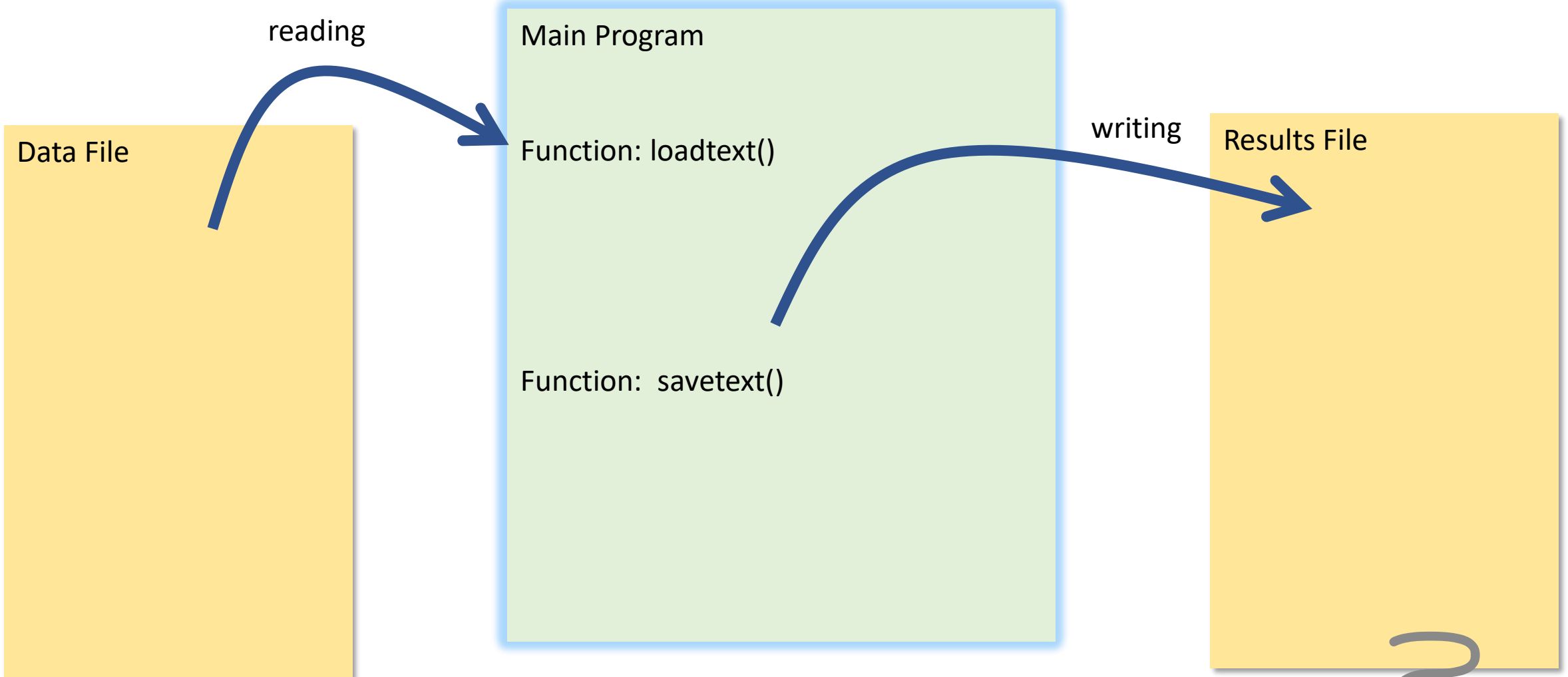
Reading From External Files

Reading: numpy.loadtext()

numpy.genfromtxt()

Writing: numpy.savetext()

Input/output from external files



Clipart by: http://clipart-library.com/clipart/kiKB8kpLT.htm

Create a data file to read from

- You need a data file with a bunch of numbers saved as plain text (*.txt).
- Excel, Spyder editor, DAS (data acquisition system), can produce such a file.
- Goal is to read large files but for this example, a “toy” file is enough.
- In the Syder editor open a new file and write the sample data and save it as **datos.txt** in the same folder that you plan to store your main program.

Data File: datos.txt

```
# x and y coordinates
1   1
2   4
3   9
4   16
5   25
```

Spyder Editor

Reading from an external file: loadtxt function

datain is an array of the same order of the data

Comments will be ignored, only numeric data will be read

datain = np.loadtxt(filename, comments='#')

filename = 'datos.txt'

filename='C:/Users/Marco/Documents/inge3016/datos.txt'

Reading from an external file

Data File: datos.txt

```
# x and y coordinates
1  1
2  4
3  9
4  16
5  25
```

```
import numpy as np


filename = 'datos.txt'
datain = np.loadtxt(filename, comments='#')
print("datain= \n",datain)
# datain is a 2D array
```

Output:

```
datain=
 [[ 1.  1.]
 [ 2.  4.]
 [ 3.  9.]
 [ 4. 16.]
 [ 5. 25.]]
```

IMPORTAT before running the code above you must have already created data file: datos.txt.

FILE: filei_o02.py

Some processing on the read data

```
import numpy as np

filename = 'datos.txt'
datain = np.loadtxt(filename, comments='#')
print("datain= \n",datain)   # datain is a 2D array

# PLAY WITH datain. EG split array datain into x and y arrays
x = datain[:,0];   print("x= \n", x)
y = datain[:,1];   print("y= \n", y)

# Modify a slice of datain:
datain[:,1] = np.log(y) # insert transformed y back into datain
print("new y=\n",y)
print("new datain=\n", datain)
```

Output:

```
datain=
 [[ 1.  1.]
 [ 2.  4.]
 [ 3.  9.]
 [ 4. 16.]
 [ 5. 25.]]
x=
 [1. 2. 3. 4. 5.]
y=
 [ 1.  4.  9. 16. 25.]
new y=
 [0.         1.38629436 2.19722458 2.77258872 3.21887582]
new datain=
 [[1.         0.        ]
 [2.         1.38629436]
 [3.         2.19722458]
 [4.         2.77258872]
 [5.         3.21887582]]
```

IMPORTAT before running the code below you must have created datos.txt.

FILE: filei_o02.py

QUIZ

- Write code to read raining data in file rain.txt. File is a 2-column data. First column x is the month and second column y is average rainfall in the month in inches. Plot the data

Quiz: Download data file to read from

- The data file for this exercise is in a separate file called rain.txt.
- Place the rain.txt in the same folder that you plan to store your main program, i.e., the current directory
- Proceed with your code to read the data, construct a plot and print the data and the plot.
- Submit code+output

Data File: rain.txt

```
1 4.6
2 3.2
3 3.1
4 4.8
5 7.2
6 5.9
7 7.1
8 7.7
9 7.4
10 6.7
11 7
12 5.9
```

Average rainfall in January: 4.6"

Source of data: https://www.weather-us.com/en/puerto-rico-usa/san-juan-climate#rainfall

Writing on external file: savetxt function

Numeric variable to be written in file

numpy.savetxt(**filename**, **results**, fmt='%10.5f',header=hdrtxt)

format

filename = 'results.dat'

hdrtxt=' x and y coordinates\n’

Writing on external file

```
# Assume the results of a program are stored  into an array called
# results and we want to write them into a file called results.dat:
import numpy
import numpy.random

results=numpy.random.random((4,3))  # assumed results

filename = 'results.dat'
hdrtxt=' x and y coordinates\n’            # write a title
hdrtxt+= ' more bla bla bla ‘  # write something more
numpy.savetxt(filename,results,fmt='%10.5f',header=hdrtxt)
```

‘’’ After the code had run, you won’t see results in the Console, they are stored in results.dat, thus you must open it. The file will be saved in the same folder where you save the program by default [alternatively you can add the filepath wherever you want the file to be stored]

‘’’

Results File: **results.dat**

results.dat

```
1 #  x and y coordinates
2 #  more bla bla bla
3    0.35006    0.11804    0.98894
4    0.60100    0.92283    0.19438
5    0.66734    0.76768    0.53063
6    0.19206    0.41713    0.27240
7
```

[file: filei_o02.py]

Appendix

Want filename at a different location, use the whole file path, EG

filename='C:/Users/Marco/Documents/inge3016/datos.txt'

To load a csv (comma separated value) file:

data = np.loadtxt(filename, delimiter=",")

To load a csv file with NumPy and skip the first three rows:

data = np.loadtxt(filename, delimiter=",", skiprows=3)

To load a csv file with NumPy, skip the first three rows and select some columns (of many), only:

data = np.loadtxt(filename, delimiter=",", skiprows=3, usecols=[0,1,2])

To importing data from tsv (tab separetd value) file, skip first row, specify delimiter as "\t":

data = np.loadtxt(tsv_file, delimiter="\t", skiprows=1, usecols=[0,1,2])

Reading different data type in a file

- A file of data from the source below, e.g., lluvia.txt, which contains columns with different data type, e.g., one column with strings and another with numbers.
- Place the lluvia.txt in the same folder that you plan to store your main program, i.e., the current directory

Data File: lluvia.txt

lluvia.txt*

```
1  January    4.6
2  February   3.2
3  March      3.1
4  April      4.8
5  May        7.2
6  June       5.9
7  July       7.1
8  August     7.7
9  September  7.4
10 October    6.7
11 November   7
12 December   5.9
```

Average rainfall in January: 4.6"

Source of data: https://www.weather-us.com/en/puerto-rico-usa/san-juan-climate#rainfall

Reading diff data type: genfromtxt function

arr is an array of the same order as the data

string

arr=np.genfromtxt(fname='lluvia.txt',dtype='str')

filename

print(arr)

```
[['January' '4.6']
 ['February' '3.2']
 ['March' '3.1']
 ['April' '4.8']
 ['May' '7.2']
 ['June' '5.9']
 ['July' '7.1']
 ['August' '7.7']
 ['September' '7.4']
 ['October' '6.7']
 ['November' '7']
 ['December' '5.9']]
```

FILE: filei_o03.py

Reading different data type: Split the sequences

```
[['January' '4.6']
 ['February' '3.2']
 ['March' '3.1']
 ['April' '4.8']
 ['May' '7.2']
 ['June' '5.9']
 ['July' '7.1']
 ['August' '7.7']
 ['September' '7.4']
 ['October' '6.7']
 ['November' '7']
 ['December' '5.9']]
```

```
months=arr[:,0]
print(months)
```

```
['January' 'February' 'March' 'April' 'May' 'June' 'July'
'August' 'September' 'October' 'November' 'December']
```

```
rain=np.array(arr[:,1],dtype='float')
print(rain)

rain=arr[:,1].astype(np.float)
print(rain)
```

Two alternatives

```
[4.6 3.2 3.1 4.8 7.2 5.9 7.1 7.7 7.4 6.7 7.  5.9]
```

FILE: filei_o03.py

Code	Output

import numpy as np
 	
# Import data from rain.txt 	
arr=np.genfromtxt(fname='lluvia.txt',dtype='str')
print(arr)
 	
# Split the sequences 	
 	
# months in array of strings: 	
months=arr[:,0]
print(months)
 	
# rain precipitation as float array: 	
rain=np.array(arr[:,1],dtype='float')
print(rain)
 	
# rain precipitation as float array: 	
rain=arr[:,1].astype(np.float)
print(rain)
	[['January' '4.6']
	['February' '3.2']
	['March' '3.1']
	['April' '4.8']
	['May' '7.2']
	['June' '5.9']
	['July' '7.1']
	['August' '7.7']
	['September' '7.4']
	['October' '6.7']
	['November' '7']
	['December' '5.9']]
	
	['January' 'February' 'March' 'April' 'May' 'June'
	'July' 'August' 'September' 'October' 'November'
	'December']
	
	[4.6 3.2 3.1 4.8 7.2 5.9 7.1 7.7 7.4 6.7 7. 5.9]
	
	[4.6 3.2 3.1 4.8 7.2 5.9 7.1 7.7 7.4 6.7 7. 5.9]

References

(1) https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.loadtxt.html

(2) https://cmdlinetips.com/2018/01/how-to-read-a-numerical-data-file-in-python-with-numpy/

(3) RWFilesPython.pdf

(4) https://www.guru99.com/reading-and-writing-files-in-python.html

(5) https://docs.scipy.org/doc/numpy/user/basics.io.genfromtxt.html

(6) https://www.earthdatascience.org/courses/earth-analytics-bootcamp/numpy-arrays/import-text-files-numpy-arrays/