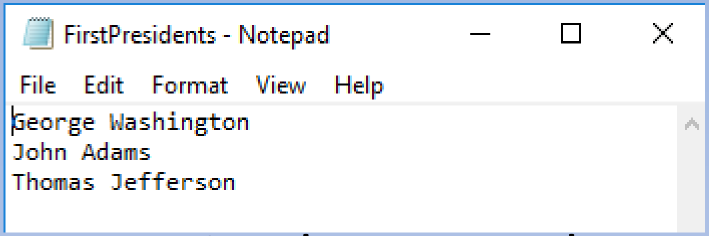
Reading Text Files

- File Contents:



```
George Washington
John Adams
Thomas Jefferson
```

- MUST Establish connection between the program and the file
  - File is said to be opened for input (notion of a ***file pointer***)

```
infile = open(fileName, 'r')
```

- First method:
  - Places entire contents of file (including the newline characters) into single string.

```
strVar = infile.read()
```

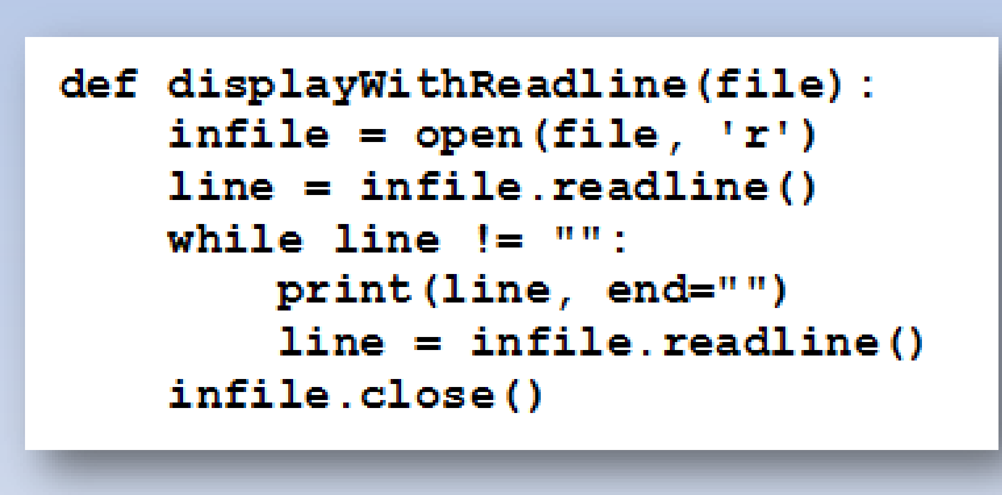
  - Not the preferred method especially for large files
- Second method:

```
strVar = infile.readline()
```

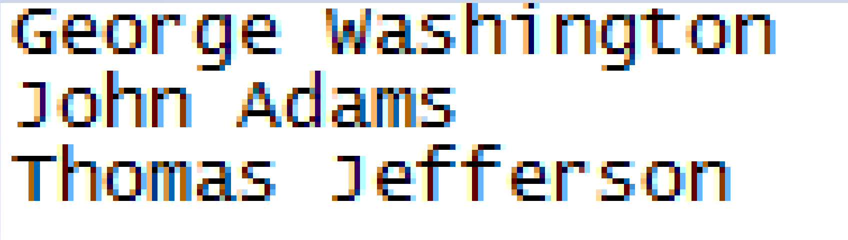
  - Current line is assigned to *strVar* and pointer advances to end of that line
  - Method returns the empty string after all lines have been read

Reading Text Files

- Example (*using second method*) function to open file, read one line at a time, print it and close the file at the end

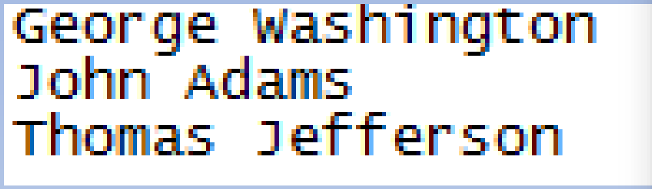


```
def displayWithReadline(file):
    infile = open(file, 'r')
    line = infile.readline()
    while line != "":
        print(line, end="")
        line = infile.readline()
    infile.close()
```

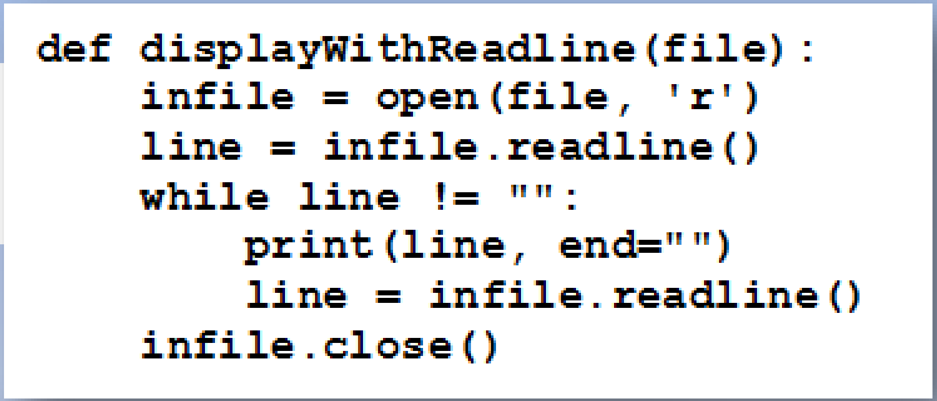


```
George Washington
John Adams
Thomas Jefferson
```

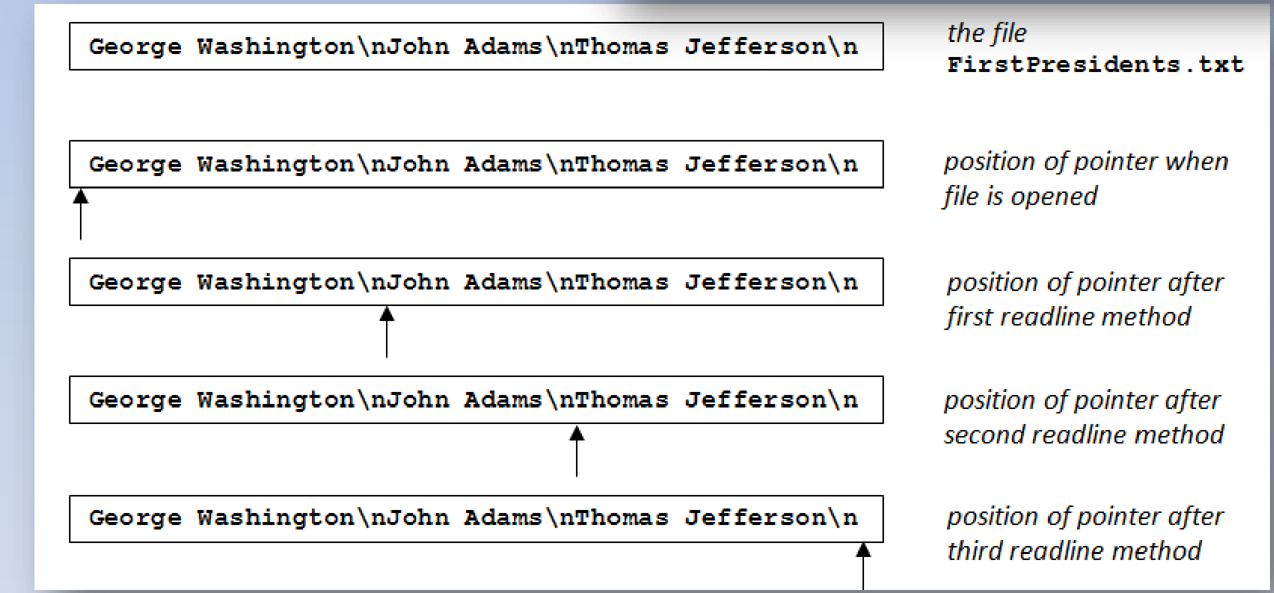
Reading Text Files



```
George Washington
John Adams
Thomas Jefferson
```



```
def displayWithReadline(file):
    infile = open(file, 'r')
    line = infile.readline()
    while line != "":
        print(line, end="")
        line = infile.readline()
    infile.close()
```



```
George Washington\nJohn Adams\nThomas Jefferson\n
```

the file FirstPresidents.txt

```
George Washington\nJohn Adams\nThomas Jefferson\n
```

position of pointer when file is opened

```
George Washington\nJohn Adams\nThomas Jefferson\n
```

position of pointer after first readline method

```
George Washington\nJohn Adams\nThomas Jefferson\n
```

position of pointer after second readline method

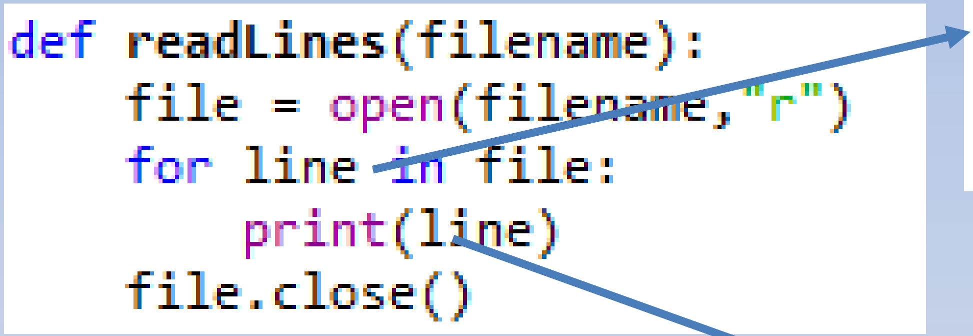
```
George Washington\nJohn Adams\nThomas Jefferson\n
```

position of pointer after third readline method

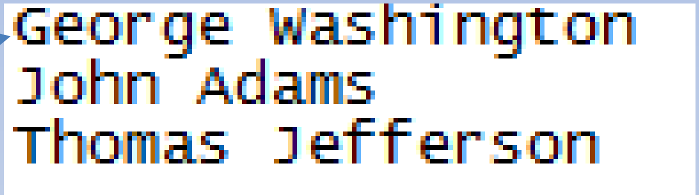
Pointer positions during the execution of the *displayWithReadline* function

Reading Text Files – For Loop

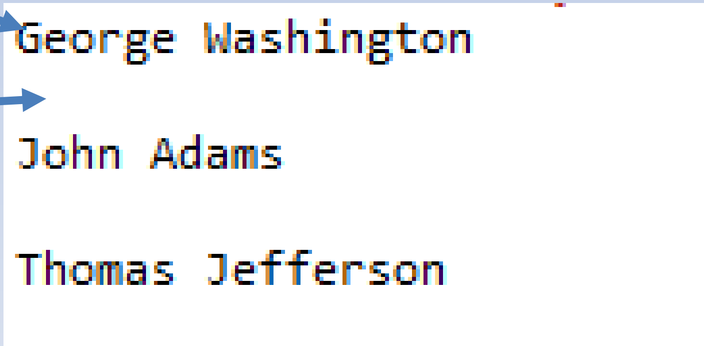
- Third method - Use For Loop with File object as Iterator



```
def readLines(filename):
    file = open(filename,"r")
    for line in file:
        print(line)
    file.close()
```



```
George Washington
John Adams
Thomas Jefferson
```

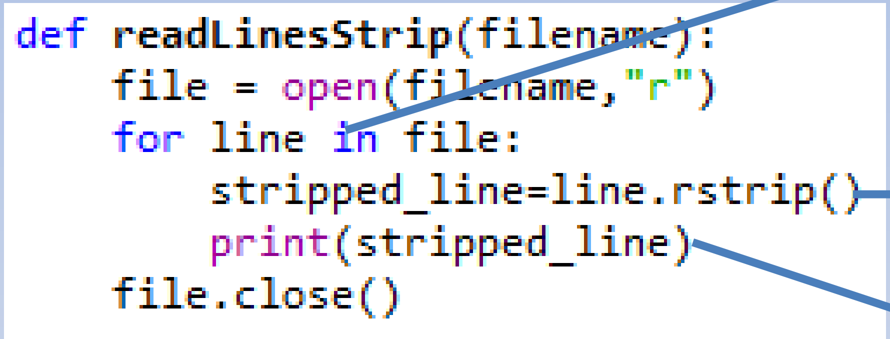


```
George Washington

John Adams

Thomas Jefferson
```

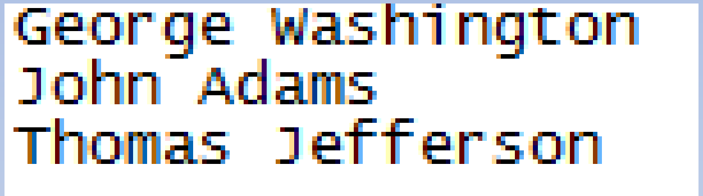
Why is additional new line printed?

Reading Text Files – For Loop

- Use For Loop with File object as Iterator, strip out ‘white space at the end of each line’

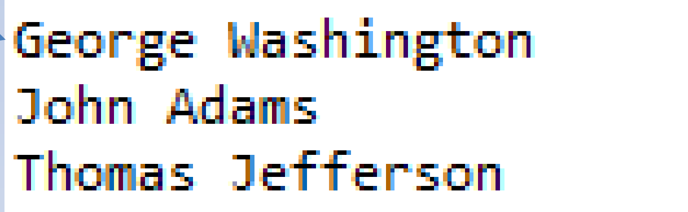


```
def readLinesStrip(filename):
    file = open(filename,"r")
    for line in file:
        stripped_line=line.rstrip()
        print(stripped_line)
    file.close()
```



```
George Washington
John Adams
Thomas Jefferson
```

Strip out the newline character

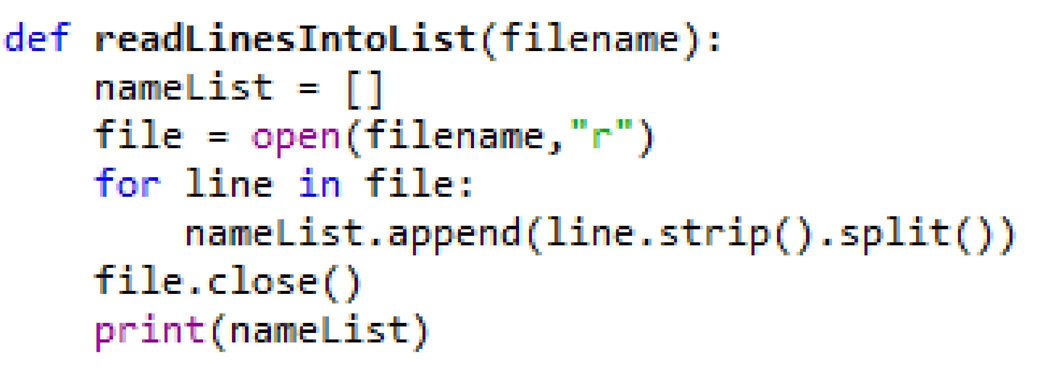


```
George Washington
John Adams
Thomas Jefferson
```

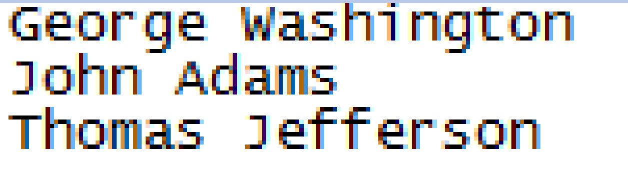
- rstrip() removes trailing characters
- Specific trailing characters are removed by passing them as argument values to rstrip()

Read and save into a list item

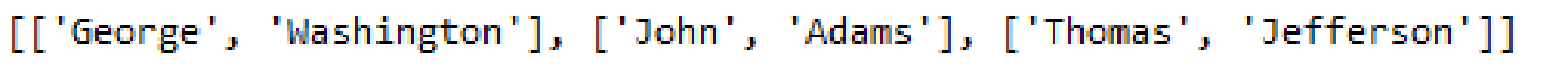
- Use For Loop with File object as Iterator, strip out ‘white space at the end of each line’, append string into a list item



```python
def readLinesIntoList(filename):
    nameList = []
    file = open(filename,"r")
    for line in file:
        nameList.append(line.strip().split())
    file.close()
    print(nameList)
```



```
George Washington
John Adams
Thomas Jefferson
```

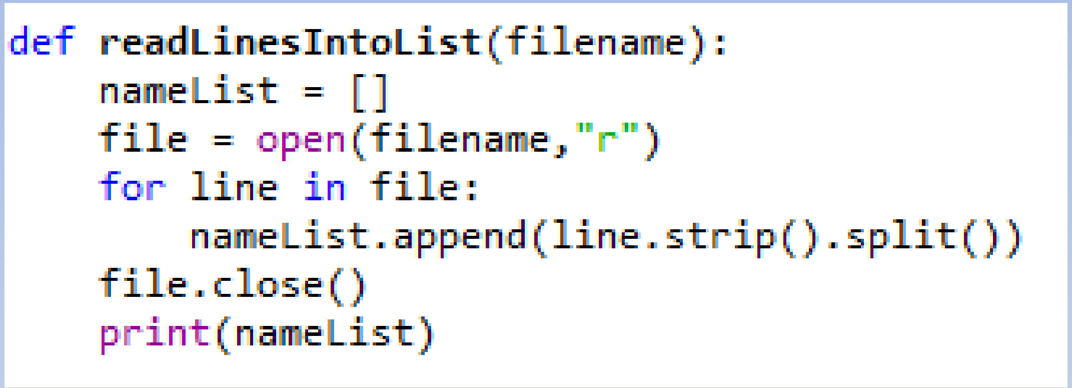


```
[['George', 'Washington'], ['John', 'Adams'], ['Thomas', 'Jefferson']]
```

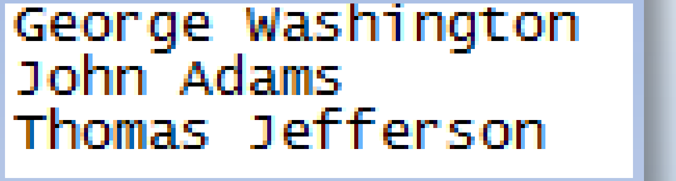
- strip() removes leading and trailing characters

Read and save into a list item

- Breaking it down one statement at a time!

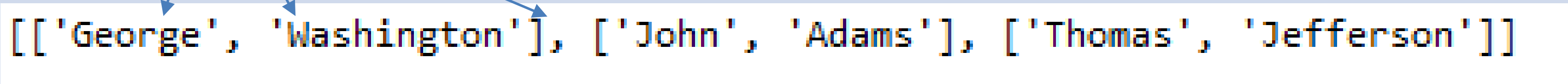


```
def readLinesIntoList(filename):
    nameList = []
    file = open(filename,"r")
    for line in file:
        nameList.append(line.strip().split())
    file.close()
    print(nameList)
```



```
George Washington
John Adams
Thomas Jefferson
```

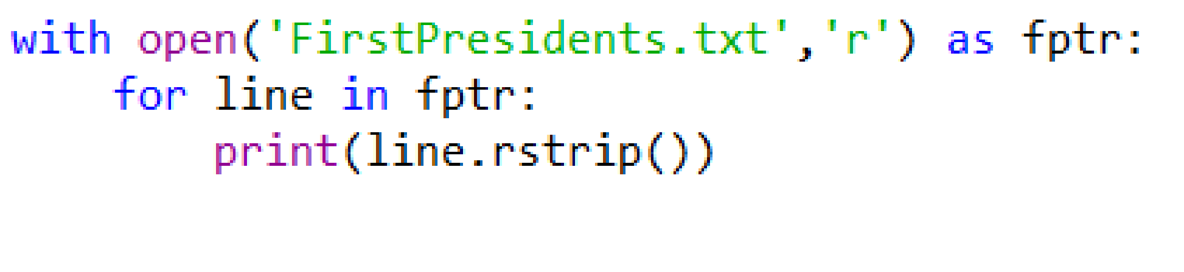
line = line.strip()   #strips out leading and trailing white space
name = line.split() #creates a list called name as [‘George’, ‘Washington’]
nameList.append(name) #appends the list object into another list



```
[['George', 'Washington'], ['John', 'Adams'], ['Thomas', 'Jefferson']]
```

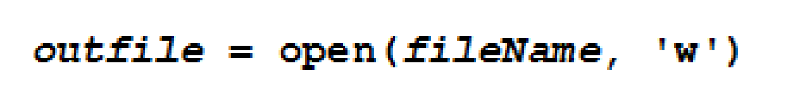
Using “with” clause

- The with clause allows us to open and work with a file within a block
- It automatically closes the file at the end of the block!



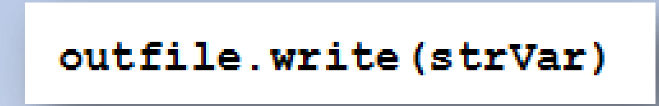
```
with open('FirstPresidents.txt','r') as fptr:
    for line in fptr:
        print(line.rstrip())
```

Creating Text Files



```
outfile = open(fileName, 'w')
```

- Creates a new text file with the specified name when ‘w’ mode is used
  - Said to be opened for writing



```
outfile.write(strVar)
```

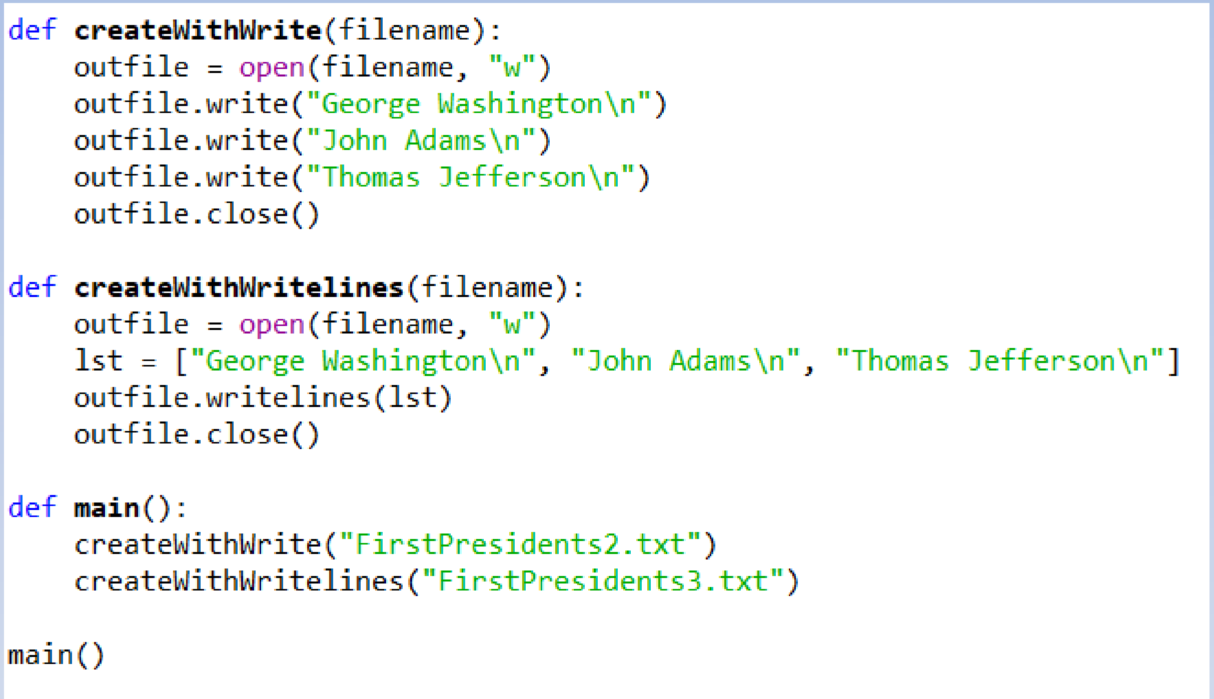
- Adds the value of *strVar* as a line of the file
- File must be closed to guarantee that all data has been physically transferred to the disk
- write() is a simple way to write one string at a time into the file

Creating Text Files

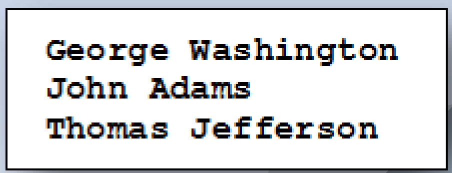
- write() is a simple way to write one string at a time into the file
- writelines() can write an entire list onto the file – it’s your responsibility to make sure that the list items are exactly what you want to write (one list item will write one line into the outputfile)

Creating Text Files

- Example: Program shows two ways to create files identical to *FirstPresidents.txt*.



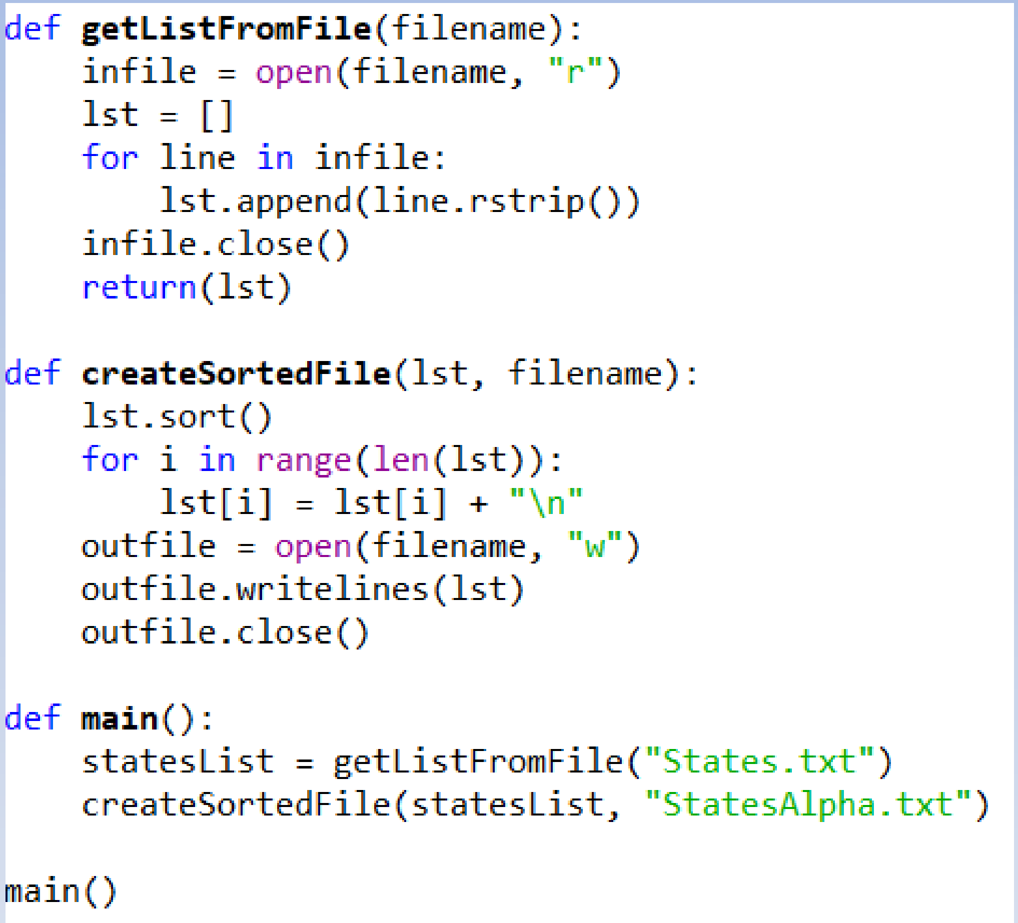
```python
def createWithWrite(filename):
    outfile = open(filename, "w")
    outfile.write("George Washington\n")
    outfile.write("John Adams\n")
    outfile.write("Thomas Jefferson\n")
    outfile.close()

def createWithWritelines(filename):
    outfile = open(filename, "w")
    lst = ["George Washington\n", "John Adams\n", "Thomas Jefferson\n"]
    outfile.writelines(lst)
    outfile.close()

def main():
    createWithWrite("FirstPresidents2.txt")
    createWithWritelines("FirstPresidents3.txt")

main()
```



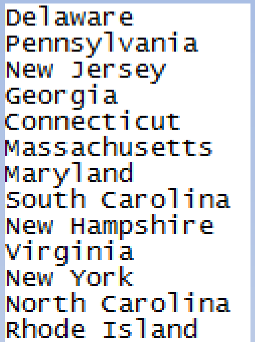
```
George Washington
John Adams
Thomas Jefferson
```

Creating Text Files

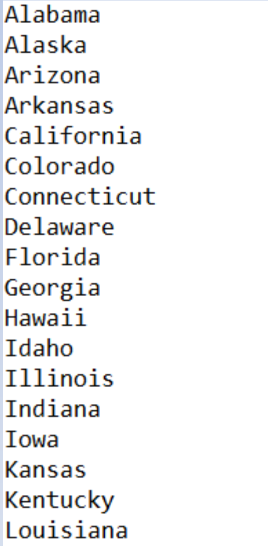
- Example: Program creates a text file named *StatesAlpha.txt* containing the states in alphabetical order.



```
def getListFromFile(filename):
    infile = open(filename, "r")
    lst = []
    for line in infile:
        lst.append(line.rstrip())
    infile.close()
    return(lst)

def createSortedFile(lst, filename):
    lst.sort()
    for i in range(len(lst)):
        lst[i] = lst[i] + "\n"
    outfile = open(filename, "w")
    outfile.writelines(lst)
    outfile.close()

def main():
    statesList = getListFromFile("States.txt")
    createSortedFile(statesList, "StatesAlpha.txt")

main()
```



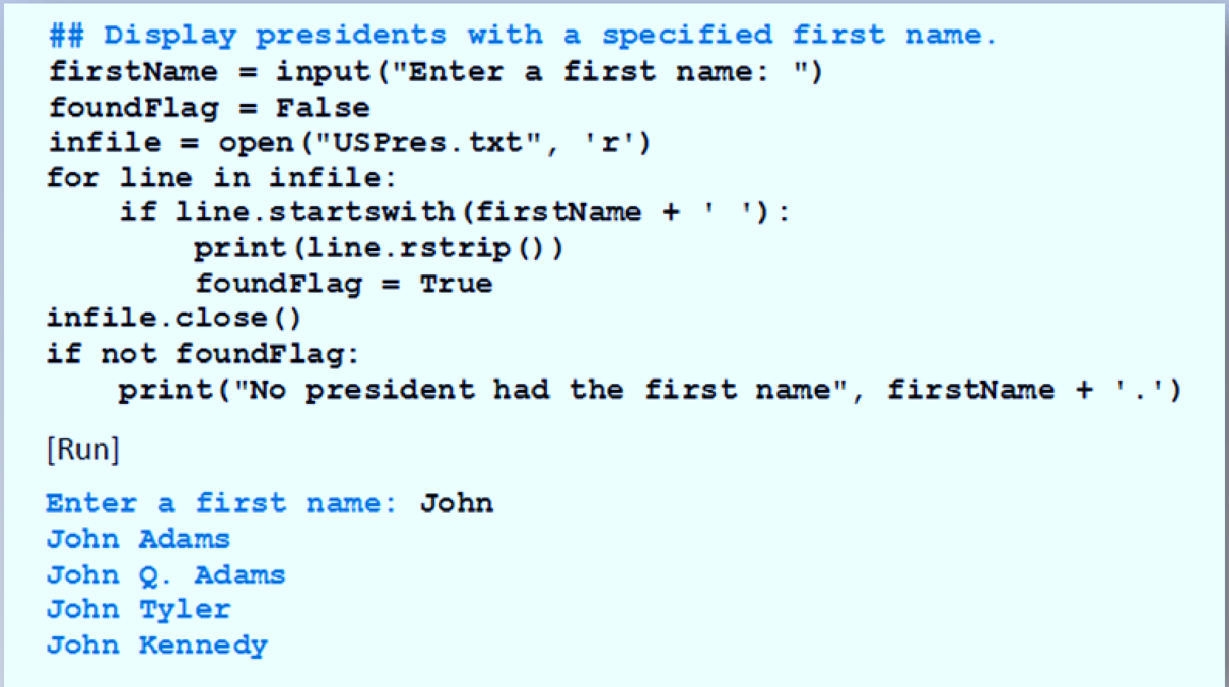
```
Delaware
Pennsylvania
New Jersey
Georgia
Connecticut
Massachusetts
Maryland
South Carolina
New Hampshire
Virginia
New York
North Carolina
Rhode Island
```



```
Alabama
Alaska
Arizona
Arkansas
California
Colorado
Connecticut
Delaware
Florida
Georgia
Hawaii
Idaho
Illinois
Indiana
Iowa
Kansas
Kentucky
Louisiana
```

Looping Through Lines of Text File

- Example: Program requests a first name and then displays names of U.S. presidents having that first name



```
## Display presidents with a specified first name.
firstName = input("Enter a first name: ")
foundFlag = False
infile = open("USPres.txt", 'r')
for line in infile:
    if line.startswith(firstName + ' '):
        print(line.rstrip())
        foundFlag = True
infile.close()
if not foundFlag:
    print("No president had the first name", firstName + '.')
```

[Run]

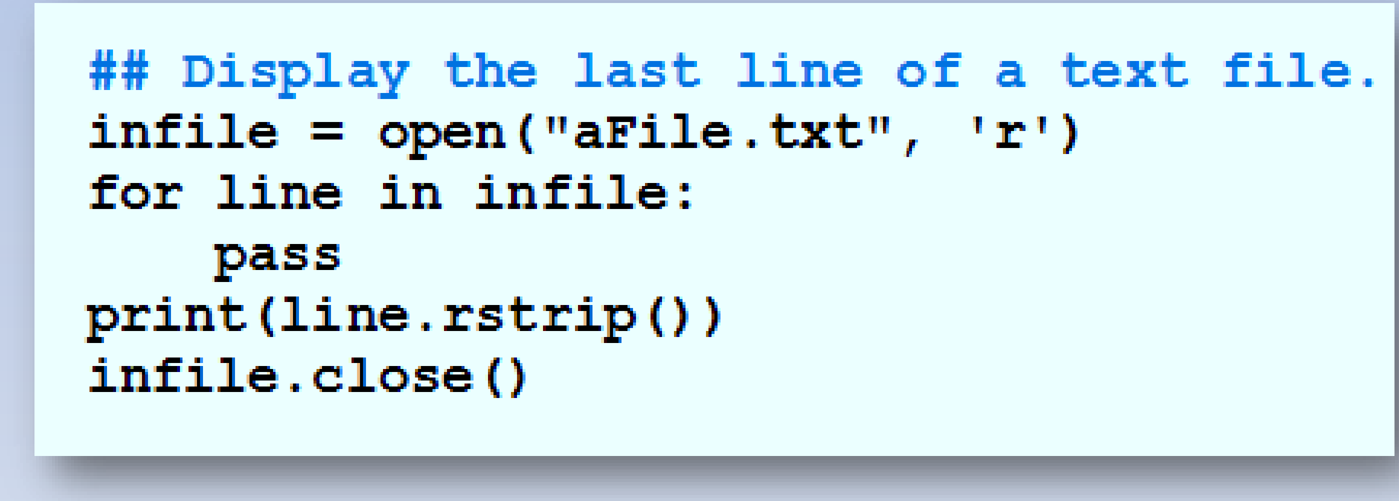
```
Enter a first name: John
John Adams
John Q. Adams
John Tyler
John Kennedy
```

The *pass* Statement

- There are times when you want your loop to cycle through a sequence and not do anything
  - The *pass* statement should be used.
- The *pass* statement is a do-nothing placeholder statement

The *pass* Statement

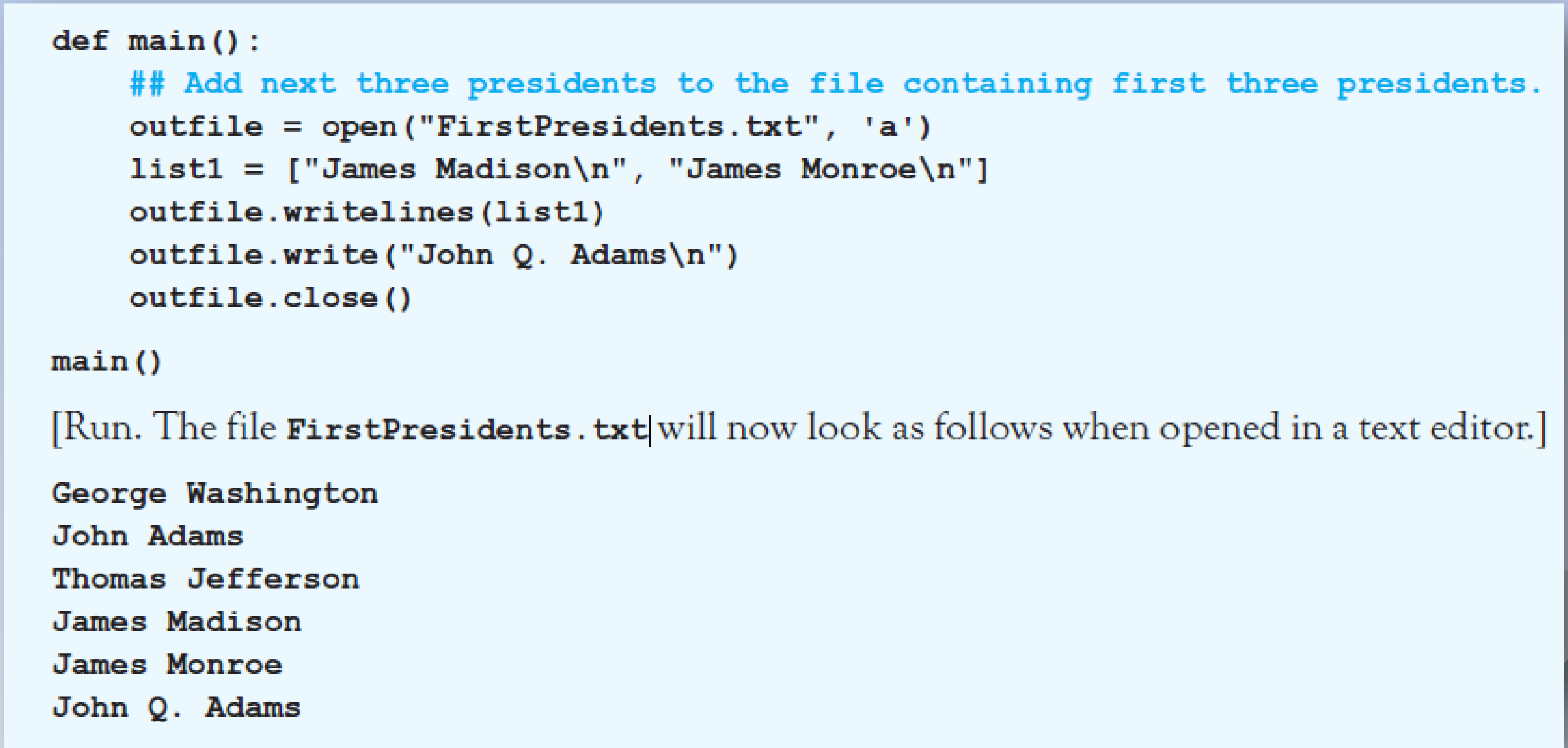
- Example: Program displays the last line of a file



```python
## Display the last line of a text file.
infile = open("aFile.txt", 'r')
for line in infile:
    pass
print(line.rstrip())
infile.close()
```

Adding Lines to an Existing Text File

- Example: Program adds 3 more names to file. Uses open statement with ‘a’ parameter for ***appending*** to the file



```
def main():
    ## Add next three presidents to the file containing first three presidents.
    outfile = open("FirstPresidents.txt", 'a')
    list1 = ["James Madison\n", "James Monroe\n"]
    outfile.writelines(list1)
    outfile.write("John Q. Adams\n")
    outfile.close()

main()
```

[Run. The file `FirstPresidents.txt` will now look as follows when opened in a text editor.]

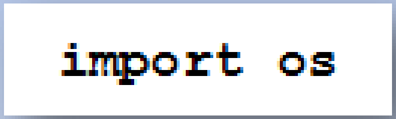
```
George Washington
John Adams
Thomas Jefferson
James Madison
James Monroe
John Q. Adams
```

Altering Items in a Text File

- Altering, inserting, or deleting a line of a text file
  - Cannot be made directly
- New file must be created
  - Read each item from the original file
  - Record it, with changes, into new file
  - Old file is then erased
  - New file is renamed with name of the original

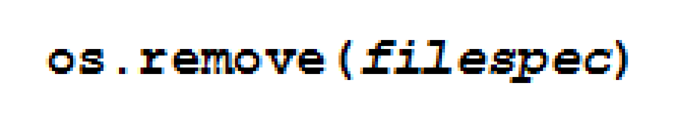
Altering Items in a Text File

- Functions needed for these tasks must be imported from standard library module *os*



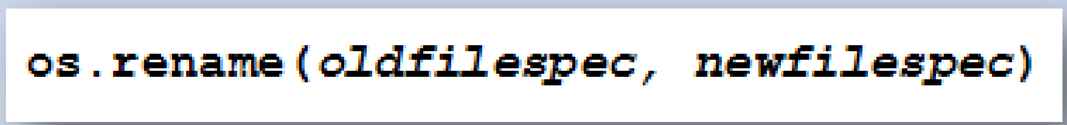
```
import os
```

- To remove a file



```
os.remove(filespec)
```

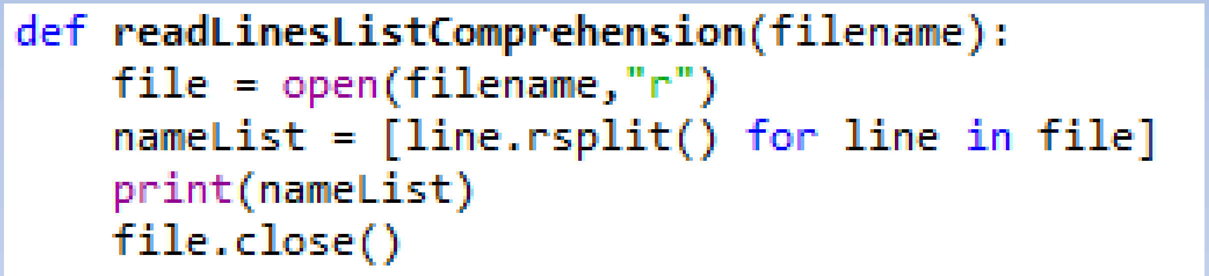
- To rename a file



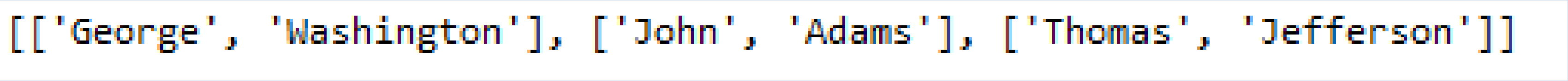
```
os.rename(oldfilespec, newfilespec)
```

Reading Text Files using List Comprehension

- Read lines of file in a loop and append it to a list at the same time!

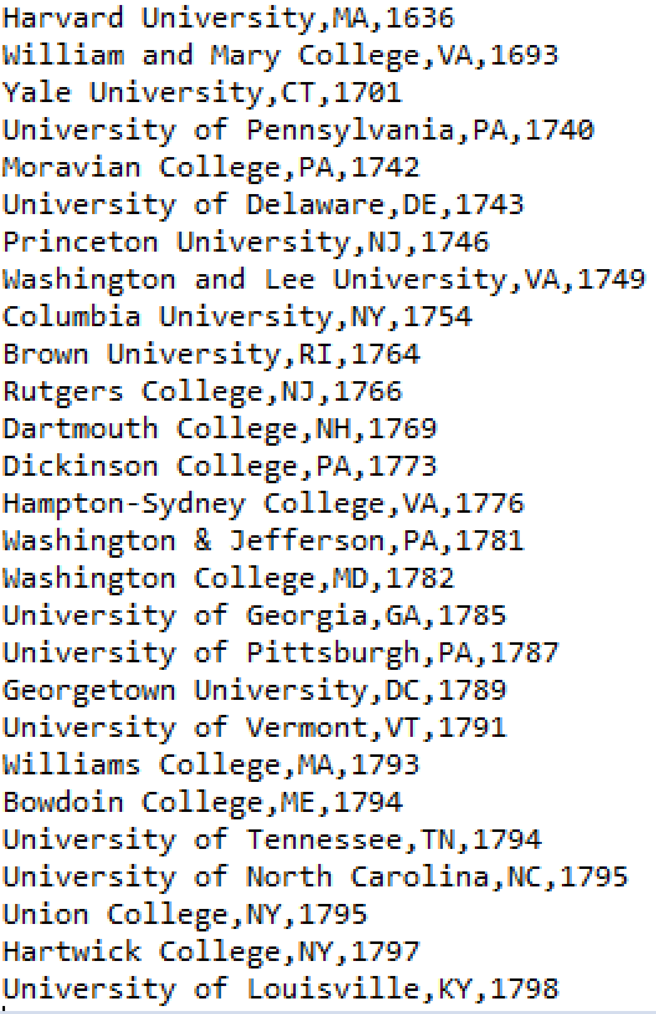


```
def readLinesListComprehension(filename):
    file = open(filename,"r")
    nameList = [line.rsplit() for line in file]
    print(nameList)
    file.close()
```

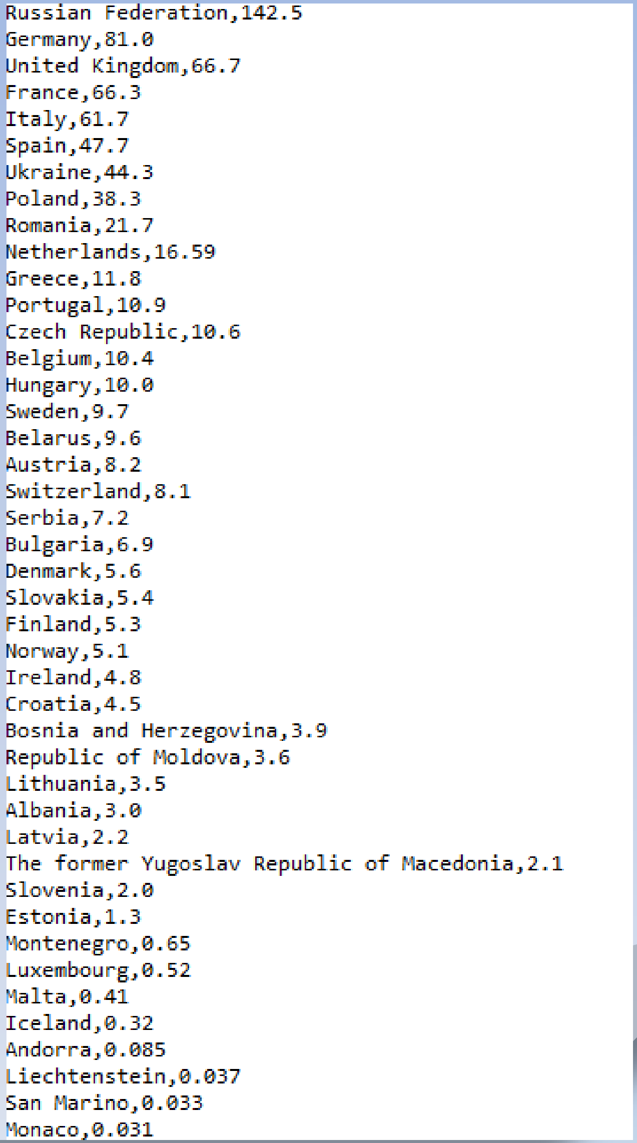


```
[['George', 'Washington'], ['John', 'Adams'], ['Thomas', 'Jefferson']]
```

Data Files containing “records”



```
Harvard University,MA,1636
William and Mary College,VA,1693
Yale University,CT,1701
University of Pennsylvania,PA,1740
Moravian College,PA,1742
University of Delaware,DE,1743
Princeton University,NJ,1746
Washington and Lee University,VA,1749
Columbia University,NY,1754
Brown University,RI,1764
Rutgers College,NJ,1766
Dartmouth College,NH,1769
Dickinson College,PA,1773
Hampton-Sydney College,VA,1776
Washington & Jefferson,PA,1781
Washington College,MD,1782
University of Georgia,GA,1785
University of Pittsburgh,PA,1787
Georgetown University,DC,1789
University of Vermont,VT,1791
Williams College,MA,1793
Bowdoin College,ME,1794
University of Tennessee,TN,1794
University of North Carolina,NC,1795
Union College,NY,1795
Hartwick College,NY,1797
University of Louisville,KY,1798
```



```
Russian Federation,142.5
Germany,81.0
United Kingdom,66.7
France,66.3
Italy,61.7
Spain,47.7
Ukraine,44.3
Poland,38.3
Romania,21.7
Netherlands,16.59
Greece,11.8
Portugal,10.9
Czech Republic,10.6
Belgium,10.4
Hungary,10.0
Sweden,9.7
Belarus,9.6
Austria,8.2
Switzerland,8.1
Serbia,7.2
Bulgaria,6.9
Denmark,5.6
Slovakia,5.4
Finland,5.3
Norway,5.1
Ireland,4.8
Croatia,4.5
Bosnia and Herzegovina,3.9
Republic of Moldova,3.6
Lithuania,3.5
Albania,3.0
Latvia,2.2
The former Yugoslav Republic of Macedonia,2.1
Slovenia,2.0
Estonia,1.3
Montenegro,0.65
Luxembourg,0.52
Malta,0.41
Iceland,0.32
Andorra,0.085
Liechtenstein,0.037
San Marino,0.033
Monaco,0.031
```