MVA Microsoft Virtual Academy

How to read from a file

read, readline, reader

Microsoft

Susan Ibach | Technical Evangelist
Christopher Harrison | Content Developer

Writing something down to remember it is only helpful if you can read it when you need it later!

- Reading a shopping list at the grocery store so you know what to buy
- Checking the number of guests on a guest list so you can see if you have enough food
- Looking up a phone number so you can call someone

In programs we often have to read information that was saved in files

- When you start your e-book reader, it looks up what page you were on when you last shut down
- When you start up your game, it looks up what treasures you had already collected so you can pick up where you left off
- There all also thousands of interesting OpenData files out there you can read to find out cool information you can use in your programs. For example: you can find out what fauna they have at the museum in Tasmania (You would be amazed at the data files you can find on the internet!)

MVA Microsoft Virtual Academy

Text files

Microsoft

How do we read a file with code?

- Use the **open** function

```
myFile = open(fileName, accessMode)
```

- Look familiar? Yes, it's the same method we use to write to a file
- So how does the program know whether to read or write?
- The access mode

Access mode	Action
r	Read the file
w	Write to the file
a	Append to the existing file content
b	Open a binary file

How do you read the file contents?

- Use the **read** method

```
fileContent= myFile.read()
```

- The read method will return the entire contents of the file into the specified string variable

If you prefer you can read one line at a time

- Use the **readline** method

```
fileContent= myFile.readline()
```

- The readline method will return one line from the file

Microsoft
Virtual
Academy

DEMO

Read a file

MVA Microsoft Virtual Academy

CSV files

Microsoft

If you are reading a CSV file, there is a **csv** library that will help you!

To access the features in the **csv** library you must import it

```
import csv
```

Now you can use the **reader** function to return all the rows from the file into a list

The reader function will take an open csv file and return each row from the file into a list

```
dataFromFile = csv.reader(myCSVfile)
```

If your file is not using a comma to separate the values, you can tell the reader function what character is used as a delimiter

```
dataFromFile = csv.reader(myCSVFile, delimiter=",")
```

Now we can open and read a csv file

```
fileName = "GuestList.txt“
accessMode = "r"
with open(fileName, accessMode) as myCSVFile:
    #Read the file contents
    dataFromFile = csv.reader(myCSVFile)
```

Why do we have a ‘with’ and ‘:’ ?

```
with open(fileName, accessMode) as myCSVFile:
```

- Programs should always open a file, and close it when they are done
- If they don’t sometimes the code crashes when you try to re-open a file that wasn’t closed last time you ran your code
- The ‘**with**’ ‘**:**’ syntax is used for certain methods to make sure clean up code such as close file runs even if there is an error.

Once we have all the rows from the csv files returned, how do we access the individual rows?

- Use a for loop to loop through the values in the list
- Each row will be one value

```
with open(fileName, accessMode) as myCSVFile:
    #Read the file contents
    dataFromFile = csv.reader(myCSVFile)
    #For loop that will run once per row
    for row in dataFromFile :
        print(row)
```

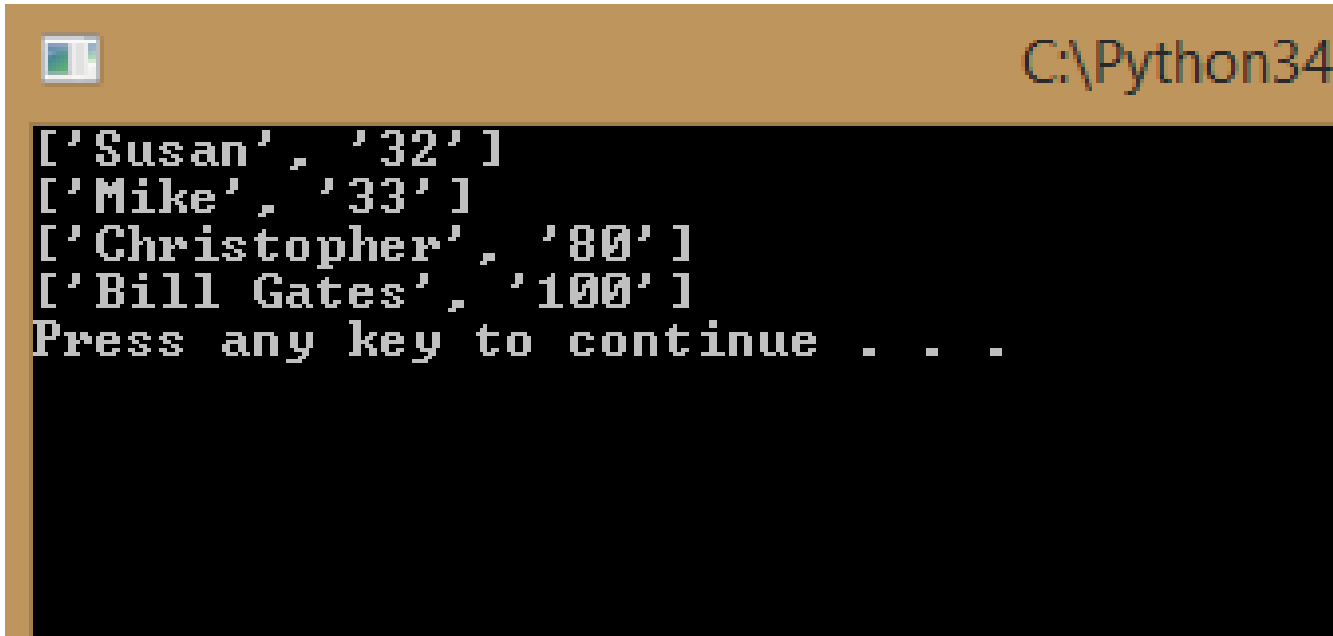
Put it all together and it looks something like this

```
fileName = "GuestList.txt"
accessMode = "r"

with open(fileName, accessMode) as myCSVFile:

      #Read the file contents
      dataFromFile = csv.reader(myCSVFile)

      #For loop that will run
      #once per row
      for row in dataFromFile :
          print(row)
```



DEMO

Read a CSV file

What if I want to access an individual value from a row and not just print the whole row?

- The row returned in the loop is actually a list of the words in that row

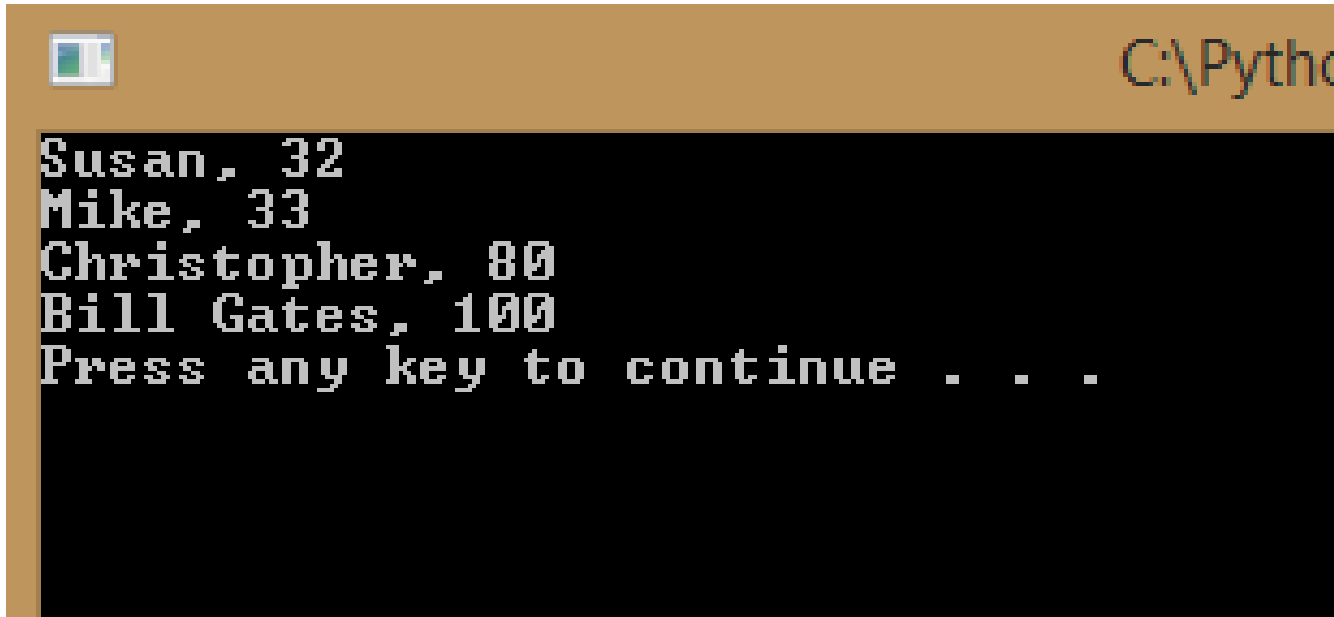
```
for row in dataFromFile :
    print(row)
    for value in row :
        print(value + "\n")
```

- So you can just create a nested loop to loop through the words in the row

But I don't like those square brackets and quotes it added to the rows!

- You can use the join function to format the output
  - SeparatorToDisplay.join(myList)

```
for row in dataFromFile :
        print (', '.join(row))
```

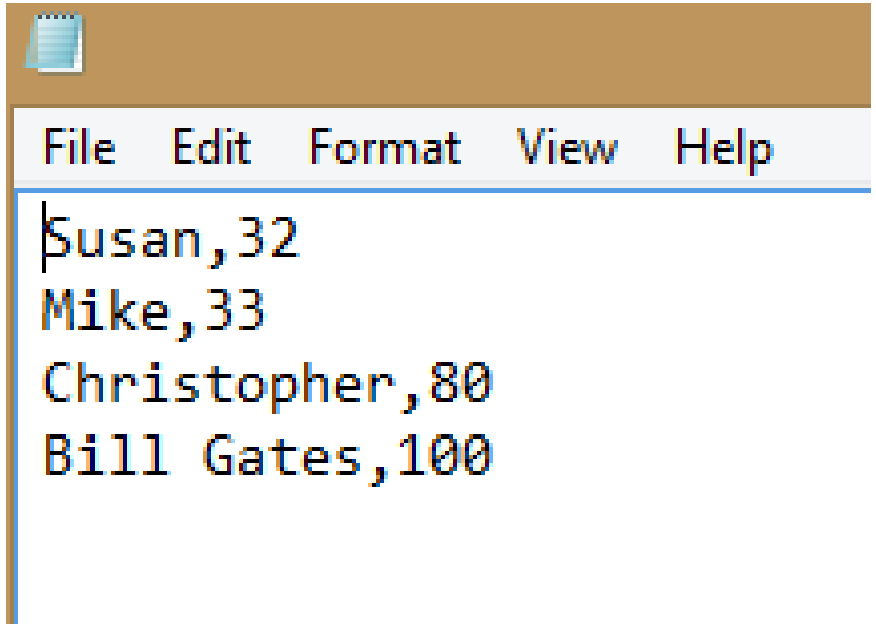


DEMO

Reading individual values from a CSV file

Your challenge

- Write a program that will print the names and ages of the guests in the guest list file you created in the last module
- If you didn't do the last challenge, you can just create a file to read using Notepad that contains names and ages



Congratulations

- You can now write a program that can receive or retrieve information from a file!

Microsoft

©2013 Microsoft Corporation. All rights reserved. Microsoft, Windows, Office, Azure, System Center, Dynamics and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.