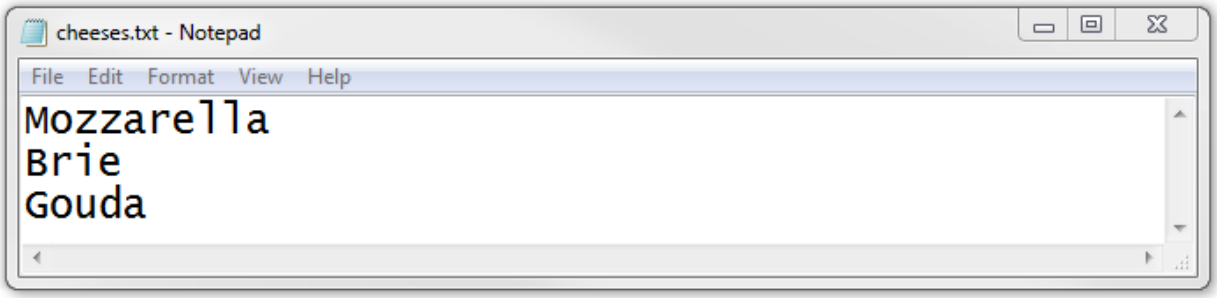
Files I/O

File Name Specification

- assume file with name:
  D:\bu\python\python_cs521\files\cheeses.txt



- '\' has special meaning
  (1) use escape characters

```
>>> f_name='D:\\bu\\python\\files\\cheeses.txt'
```

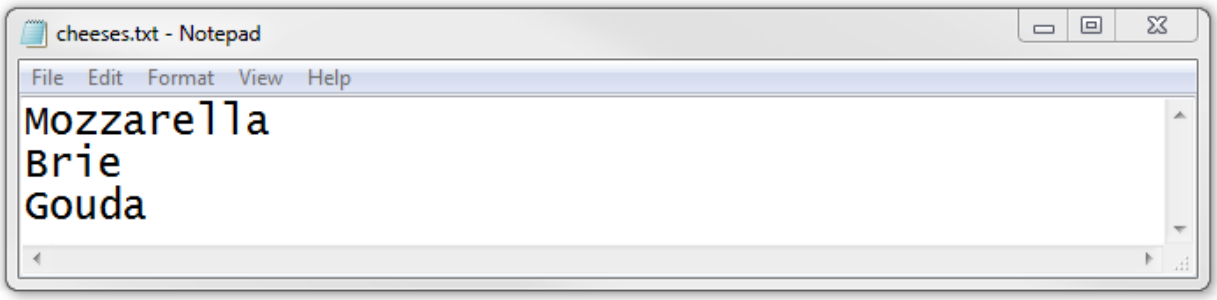
(2) use Python 'os' module

```
>>> import os
>>> f_name=os.path.join('D:/',bu, 'python',
                        'files', 'cheeses.txt')
>>> f_name
'D:/bu\\python\\files\\cheeses.txt'
```

File Access Modes

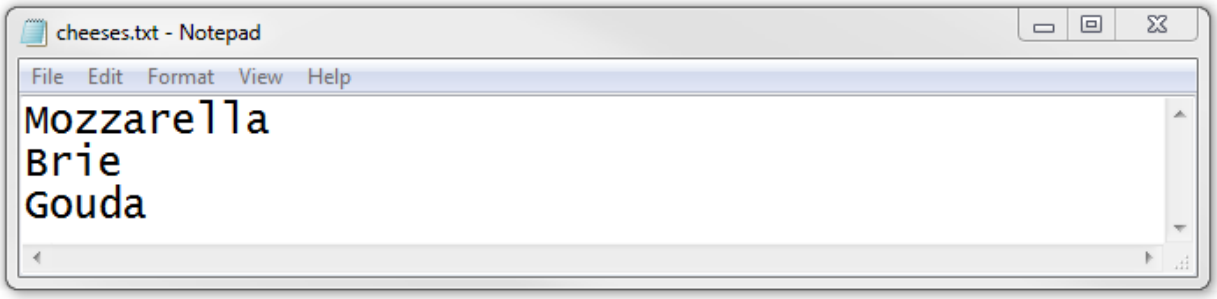
Mode	Meaning
‘a’	append to a file
‘a+’	append to and read from a file
‘r’	read
‘r+’	read from and write to a file
‘w’	write to a file
‘w+’	write to and read from a file

Reading Whole Text File




```
>>> f_name='D:\\bu\\python\\files\\cheeses.txt'
>>> text_file = open(file_name, "r")
>>> result = text_file.read()
>>> text_file.close()
>>> print(result)
Mozzarella
Brie
Gouda
>>> type(result)
<type 'str'>
```

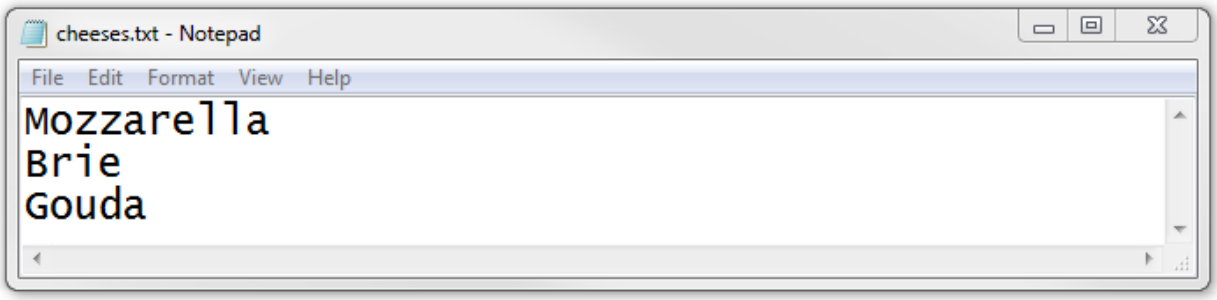
Read Characters From File




```
>>> f_name = 'D:\\bu\\python\\files\\cheeses.txt'
>>> text_file = open(f_name, "r")
>>> print(text_file.read(13))
Mozzarella
Br
>>> print(text_file.read(18))
ie
Gouda
>>> text_file.close()
```

- read next *n* characters

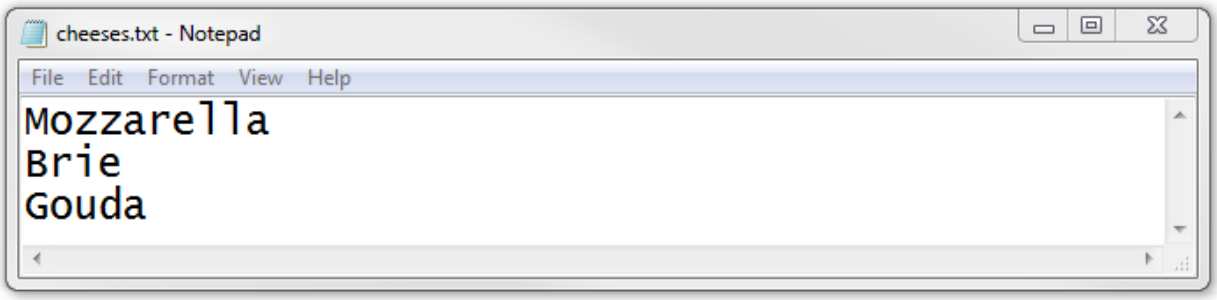
Read Characters From Line




```
>>> f_name = 'D:\\bu\\python\\files\\cheeses.txt'
>>> text_file = open(f_name, "r")
>>> print(text_file.readline(13))
Mozzarella

>>> print(text_file.readline(18))
Brie

>>> text_file.close()
```

- read next *n* characters till newline

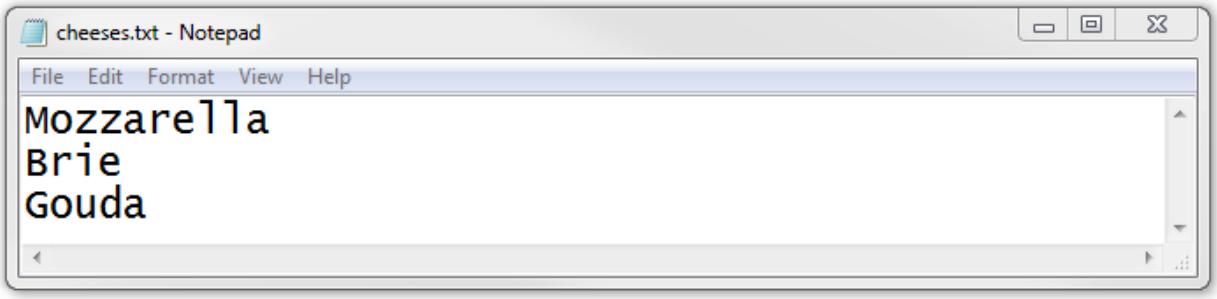
Read Lines From Line



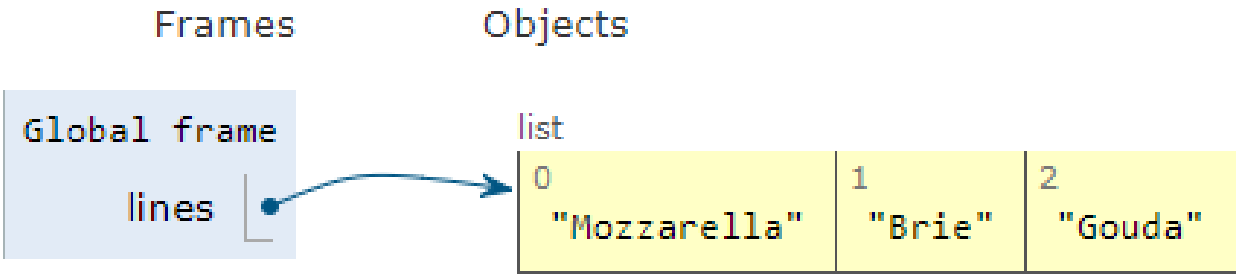

```
>>> f_name = 'D:\\bu\\python\\files\\cheeses.txt'
>>> text_file = open(f_name, "r")
>>> print(text_file.readline())
Mozzarella
>>> print(text_file.readline())
Brie
>>> text_file.close()
```

- read one line at a time

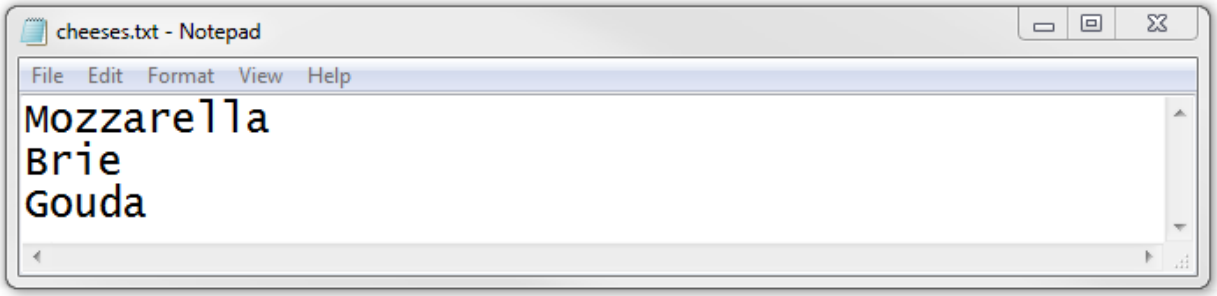
Read Lines Into List



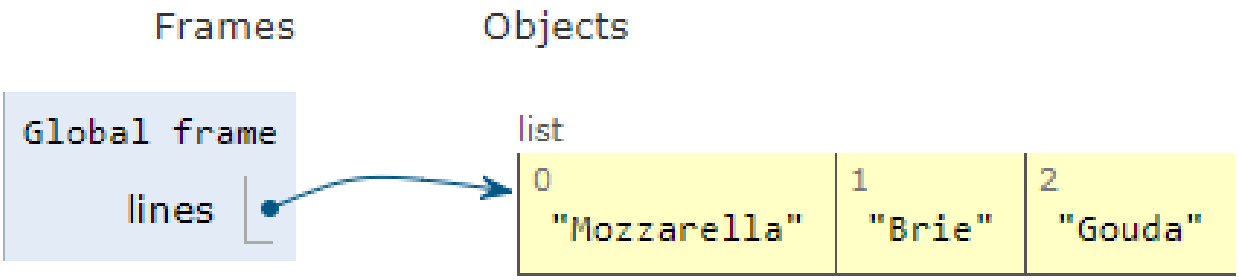

```
>>> f_name = 'D:\\bu\\python\\files\\cheeses.txt'
>>> text_file = open(f_name, "r")
>>> lines = text_file.readlines()
>>> text_file.close()
```



Using Content Manager to Read Lines Into List

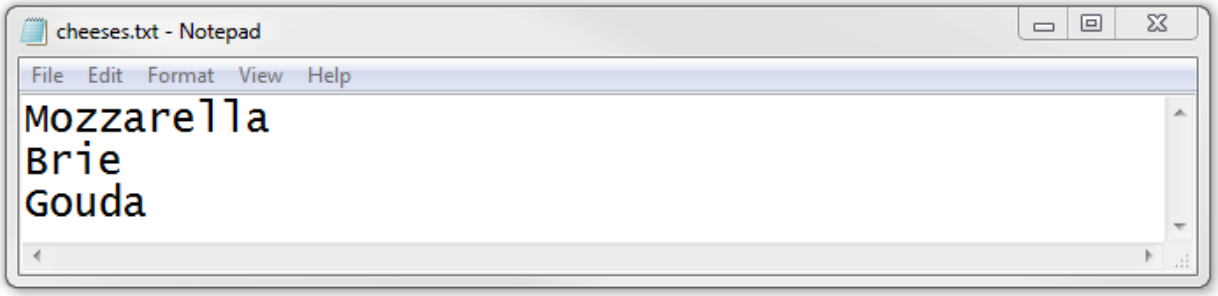



```
>>> f_name = 'D:\\bu\\python\\files\\cheeses.txt'
>>> with open(f_name, "r") as text_file:
>>>     lines = text_file.readlines()
>>> # file is automatically closed
```

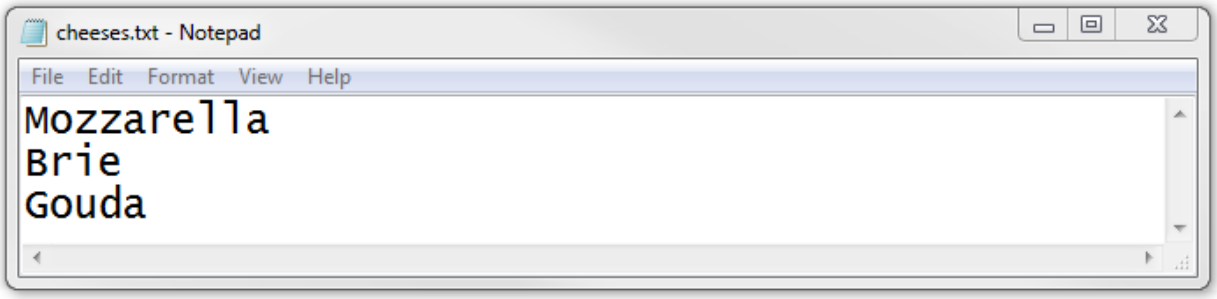


Write Strings to File

```
>>> f_name = 'D:\\bu\\python\\files\\cheeses.txt'
>>> text_file = open(f_name, "w")
>>> text_file.write('Mozzarella\n')
>>> text_file.write('Brie\n')
>>> text_file.write('Gouda\n')
>>> text_file.close()
```



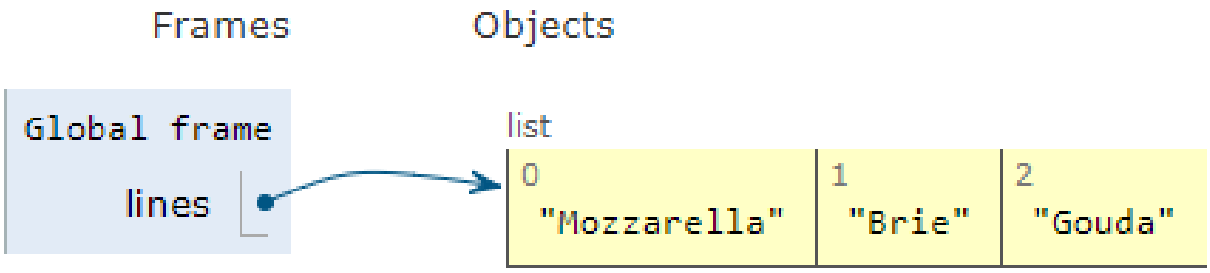
Looping Line by Line



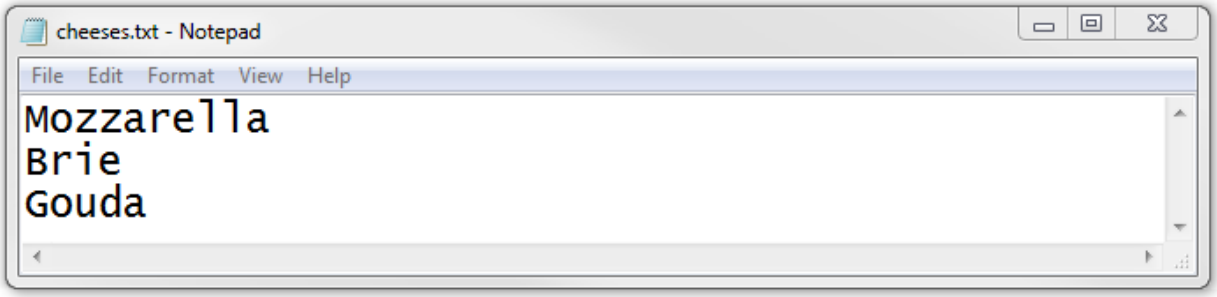

```
>>> f_name = 'D:\\bu\\python\\files\\cheeses.txt'
>>> text_file = open(f_name, "r")
>>> for line in text_file:
...         print lines
Mozzarella

Brie

Gouda
>>> text_file.close()
```

Write List of Strings to File

```
>>> f_name = 'D:\\bu\\python\\files\\cheeses.txt'
>>> lines = ['Mozzarella', 'Brie', 'Gouda']
```



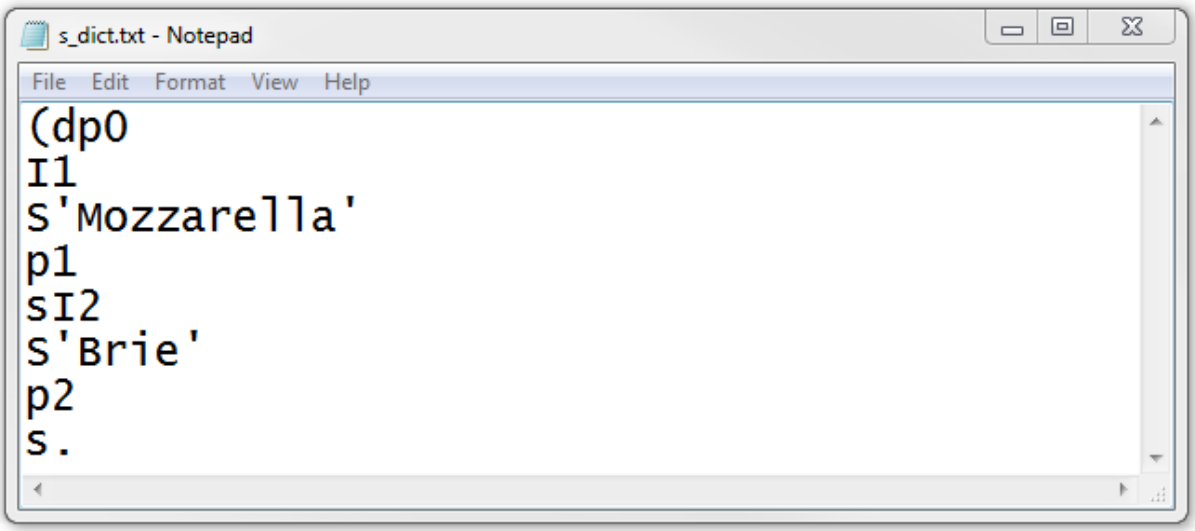

```
>>> text_file = open(f_name, "w")
>>> text_file.writelines(lines)
>>> text_file.close()
```



Save Objects to a File

- Python has a module to save objects

```
>>> import pickle
>>> f_name = D:\\bu\\python\\files\\s_dict.txt'
>>> x = {1:'Mozzarella', 2:'Brie'}
>>> pickle.dump(x, f_name)
```




```
>>> y = pickle.load(f_name)
```

- y is equal to x (but not identical)

Review Problems

Interview Problem

- check the file existence and file type

Interview Problem

- display the contents of text file in reverse order

Interview Problem

- name the file-related modules in Python?

Interview Problem

- describe the shortest way to open a text file and display its contents?

Interview Problem

- explain all the file processing modes supported by Python?

Interview Problem

- how to delete a file?

Interview Problem

- explain the use of *with* statement?

Interview Problem

- print length of each line in file “data.txt” (exclude whitespaces at end of lines)