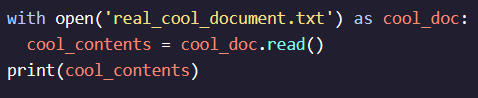
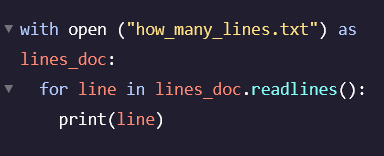
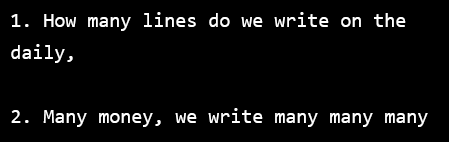
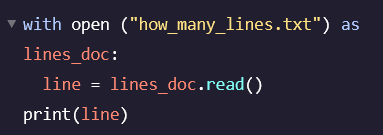
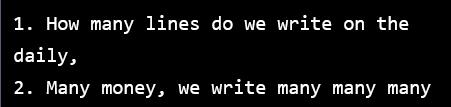
**Files:**

- Named locations on a computers disk that permanently store information for future use (stored on non-volatile memory as apposed to RAM)  
**- File Handling** – the process that a language uses to interact with a computers file system  
1). Open/Create the file  
2). Perform operations on the file (read, write, delete, etc..)  
3). Close the file to free up resources used  


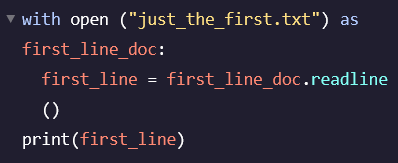
- *with* command is a “context manager” that automatically handles closing the file for us at the end of use  
- Can use *pass* command to make Python skip over EOF error

**Iterating Through Lines:**

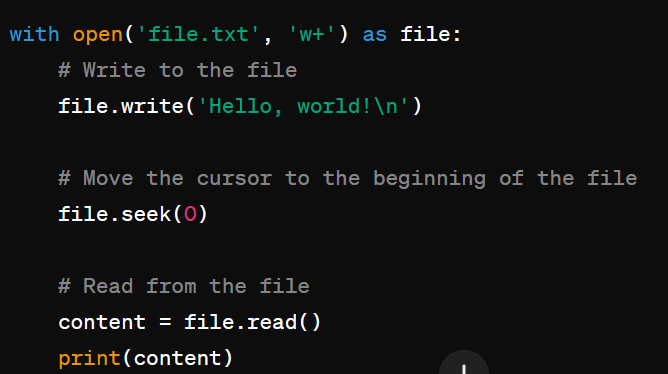
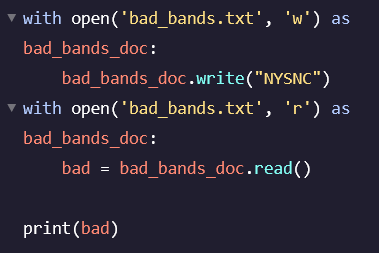
- Can use *.readlines()* to read a text file line by line and return as a list of strings  
- Memory intensive if file is large  
 

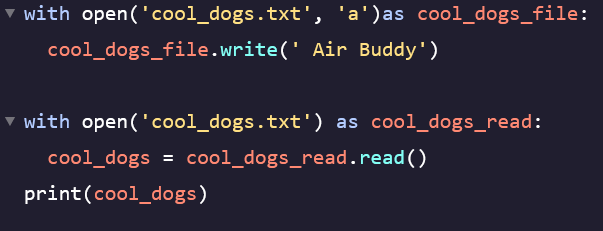
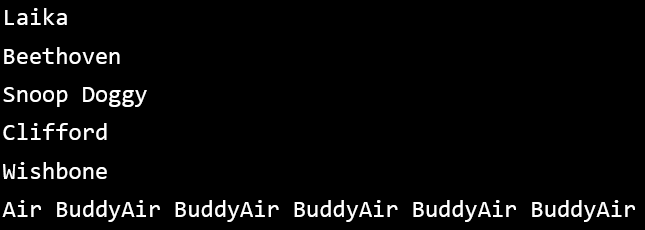
**Reading a Line:**

- Use *.readline()* to read only one line from the file each time it’s called  
- Reads until encounters end of line and then returns “ “   
 

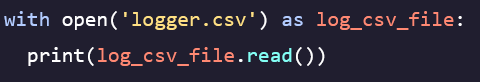
**Writing a Line:**

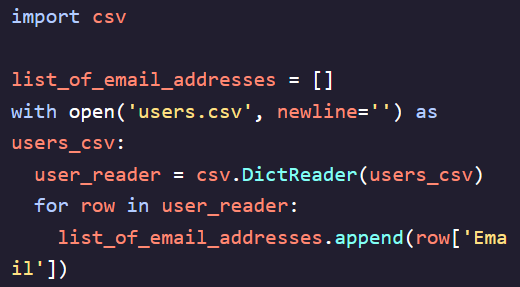
- Can also add text into a file once it’s open using the *.write()* function  
- Default mode when opening file is read (‘r’) so need to pass (‘w’) after text file name to indicate to Python it should be opened in write mode  
- The new file will overwrite your old file  
- Can’t read from a file opened in write mode! Need to reopen in read mode  
- **To read/write a file all at once use ‘w+’, if file doesn’t exist it will be created**  


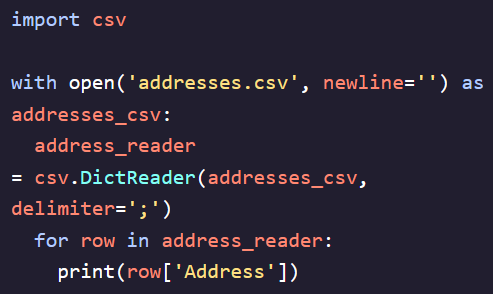
**Appending to a File:**

- Allows you to add to an existing file without you needing to overwrite it  
- Instead of using ‘w’ to open file in write mode, use *‘a’* to open it in append mode  
- Use *.write()* function to write to it again, appends to end of text file  
- Can’t read from a file opened in write mode! Need to reopen in read mode  
- Appends as many times as you hit run  
 

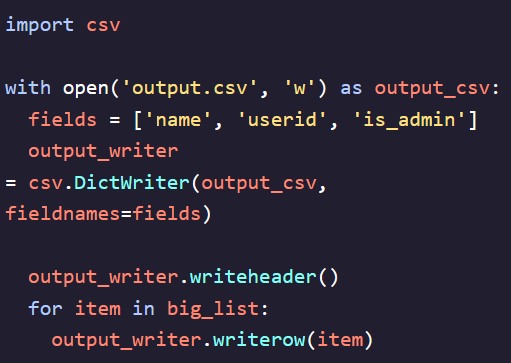
**Comma Separated Values (CSV):**

- Text files that impose a structure to their data (columns and rows)  
- Called and accessed just like regular files  


- Can access them in a more useable format by importing the csv module, and using the *DictReader()* function to convert the data into a dictionary  
 

- CSV is a catch all name for any file sorted using a *delimiter*. To open these other types of files we need to pass in the *delimiter ‘ ‘* command when using *DictReader()*

**Writing CSV’s:**

- Open using the *‘w’* function to write to a csv  
- Need to declare keys as *variable* and then associate that *variable* with *fieldnames* in .*DictWriter()   
­-* Call *.writeheader()* to collect headers from values put into *fieldnames* and make first row  
- Then we call *.writerow()* to write each item with the proper header name into a row

**Reading a JSON File:**

- “Java Script Object Notation” another popular file format for storing data. Initially based on Javascript but can now encompass multiple languages  
- Print statement can be in or out of loop  
 

**Writing a JSON File:**

- Useful in circumstances when you are using a Python library to serve web pages  
- Open file in write mode using ‘w’  
- Turn Python commands to JSON using the *.dump()* function  
- *.dump()* takes two arguments:  
1). The name of the new save location   
2). The name of the file you want to save  
