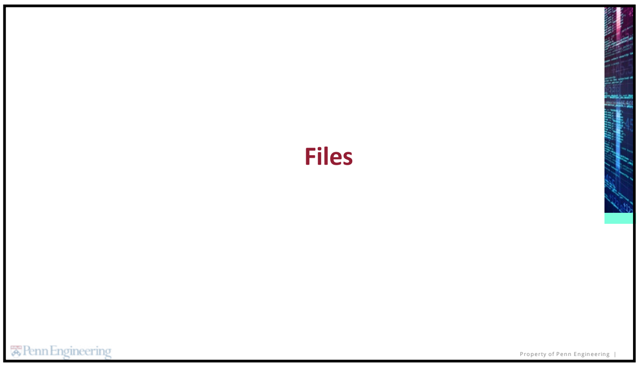
Files

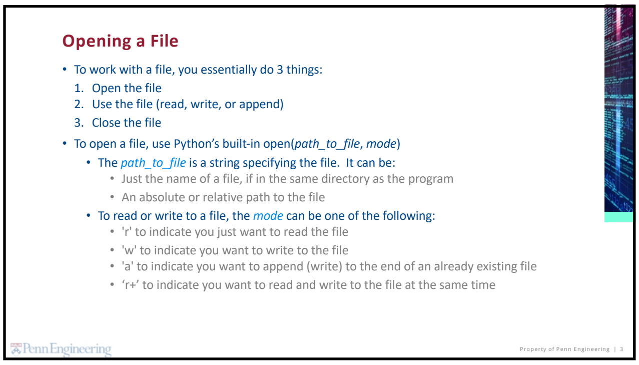
Brandon Krakowsky

1



Files

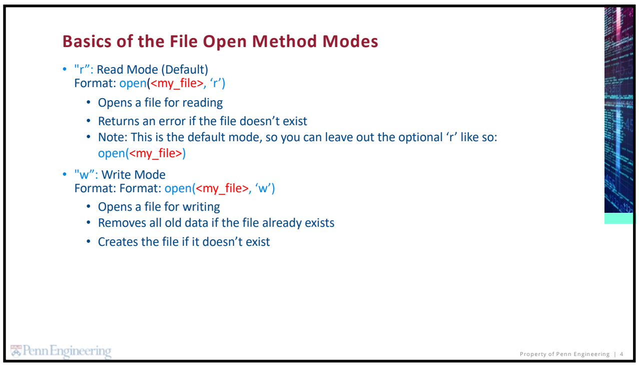
2



Opening a File

- To work with a file, you essentially do 3 things:
  1. Open the file
  2. Use the file (read, write, or append)
  3. Close the file
- To open a file, use Python's built-in open(*path_to_file*, *mode*)
  - The *path_to_file* is a string specifying the file. It can be:
    - Just the name of a file, if in the same directory as the program
    - An absolute or relative path to the file
  - To read or write to a file, the *mode* can be one of the following:
    - 'r' to indicate you just want to read the file
    - 'w' to indicate you want to write to the file
    - 'a' to indicate you want to append (write) to the end of an already existing file
    - 'r+' to indicate you want to read and write to the file at the same time

3



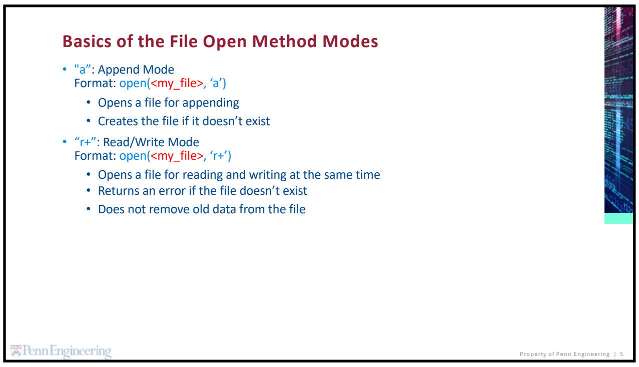
Basics of the File Open Method Modes

- "r": Read Mode (Default)
  Format: open(<my_file>, 'r')
  - Opens a file for reading
  - Returns an error if the file doesn't exist
  - Note: This is the default mode, so you can leave out the optional 'r' like so:
    open(<my_file>)
- "w": Write Mode
  Format: Format: open(<my_file>, 'w')
  - Opens a file for writing
  - Removes all old data if the file already exists
  - Creates the file if it doesn't exist

PennEngineering

Property of Penn Engineering | 4

4



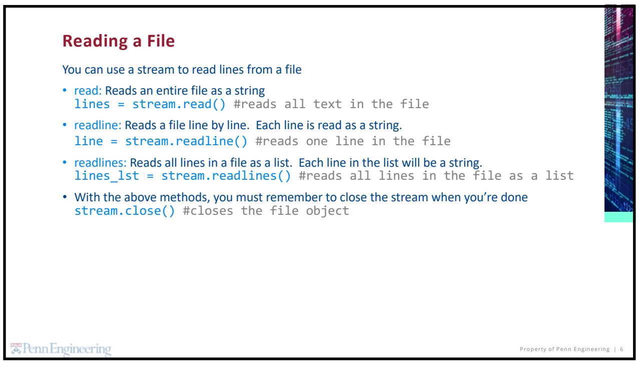
Basics of the File Open Method Modes

- "a": Append Mode
  Format: open(<my_file>, 'a')
  - Opens a file for appending
  - Creates the file if it doesn't exist
- "r+": Read/Write Mode
  Format: open(<my_file>, 'r+')
  - Opens a file for reading and writing at the same time
  - Returns an error if the file doesn't exist
  - Does not remove old data from the file

PennEngineering

Property of Penn Engineering | 5

5



Reading a File

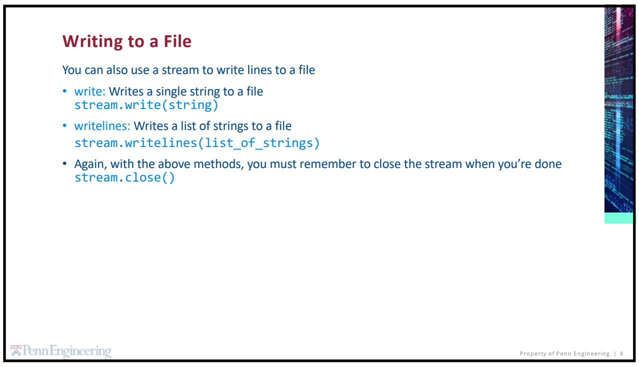
You can use a stream to read lines from a file

- read: Reads an entire file as a string
  ```
  lines = stream.read() #reads all text in the file
  ```
- readline: Reads a file line by line. Each line is read as a string.
  ```
  line = stream.readline() #reads one line in the file
  ```
- readlines: Reads all lines in a file as a list. Each line in the list will be a string.
  ```
  lines_lst = stream.readlines() #reads all lines in the file as a list
  ```
- With the above methods, you must remember to close the stream when you're done
  ```
  stream.close() #closes the file object
  ```

PennEngineering

Property of Penn Engineering | 6

6



Writing to a File

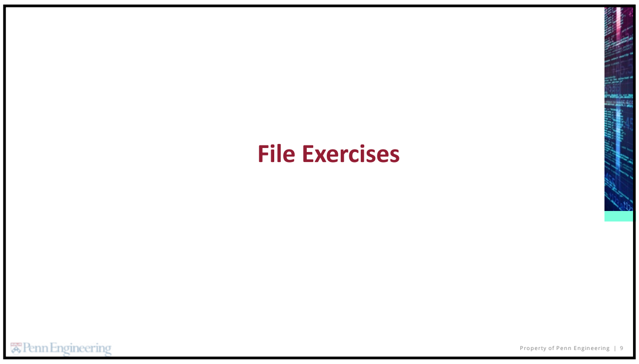
You can also use a stream to write lines to a file

- write: Writes a single string to a file
  `stream.write(string)`
- writelines: Writes a list of strings to a file
  `stream.writelines(list_of_strings)`
- Again, with the above methods, you must remember to close the stream when you're done
  `stream.close()`

Penn Engineering

Property of Penn Engineering | 8

7

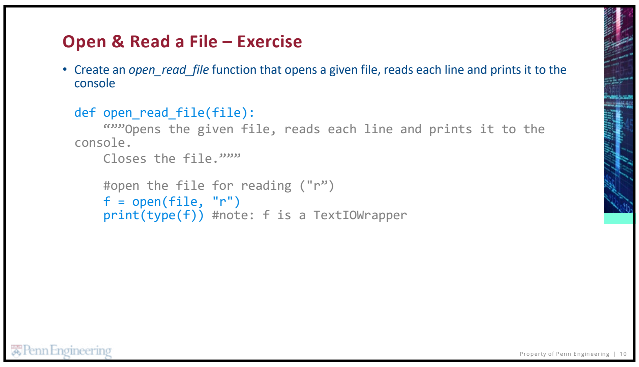


File Exercises

Penn Engineering

Property of Penn Engineering | 9

8



Open & Read a File – Exercise

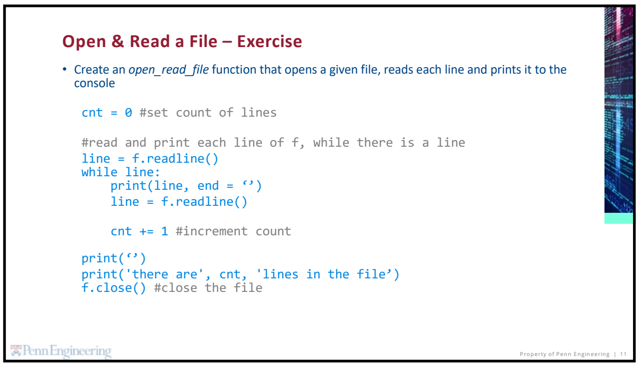
- Create an *open_read_file* function that opens a given file, reads each line and prints it to the console

```
def open_read_file(file):
    “”””Opens the given file, reads each line and prints it to the 
console.
    Closes the file.”””” 

    #open the file for reading ("r”)
    f = open(file, "r")
    print(type(f)) #note: f is a TextIOWrapper
```

PennEngineering

Property of Penn Engineering | 10

9



Open & Read a File – Exercise

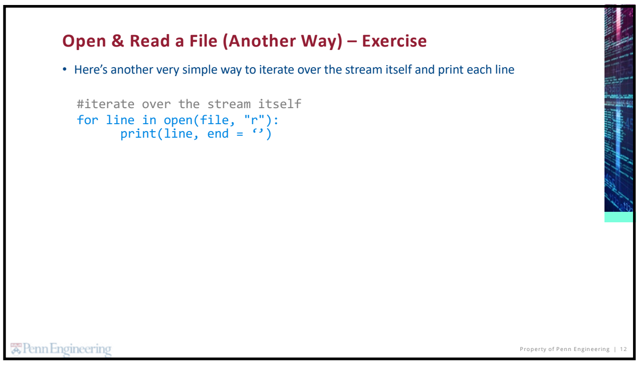
- Create an *open_read_file* function that opens a given file, reads each line and prints it to the console

```
cnt = 0 #set count of lines

#read and print each line of f, while there is a line
line = f.readline()
while line:
    print(line, end = ‘’)
    line = f.readline()

    cnt += 1 #increment count

print(‘’)
print('there are', cnt, 'lines in the file’)
f.close() #close the file
```

PennEngineering

Property of Penn Engineering | 11

10



Open & Read a File (Another Way) – Exercise

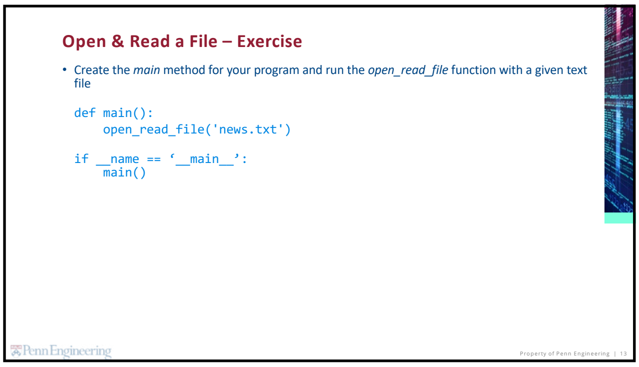
- Here’s another very simple way to iterate over the stream itself and print each line

```
#iterate over the stream itself
for line in open(file, "r"):
      print(line, end = ‘’)
```

PennEngineering

Property of Penn Engineering | 12

11



Open & Read a File – Exercise

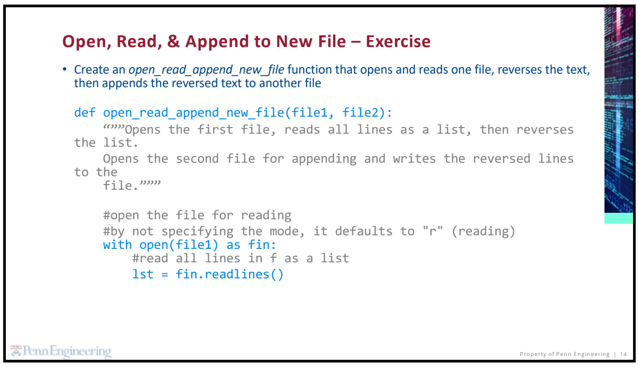
- Create the *main* method for your program and run the *open_read_file* function with a given text file

```
def main():
    open_read_file('news.txt')

if __name == ‘__main__’:
    main()
```

Penn Engineering

Property of Penn Engineering | 13

12



Open, Read, & Append to New File – Exercise

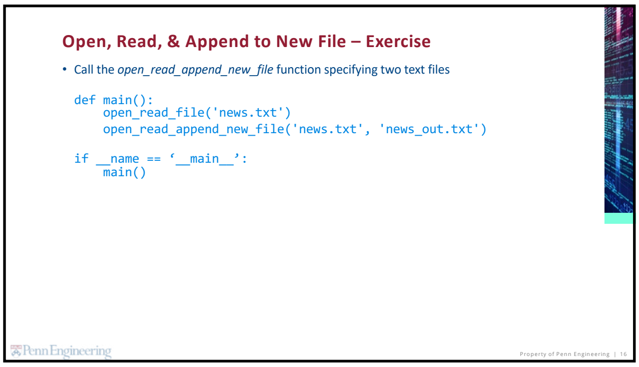
- Create an *open_read_append_new_file* function that opens and reads one file, reverses the text, then appends the reversed text to another file

```
def open_read_append_new_file(file1, file2):
    “””Opens the first file, reads all lines as a list, then reverses 
the list.
    Opens the second file for appending and writes the reversed lines 
to the
    file.”””

    #open the file for reading
    #by not specifying the mode, it defaults to "r" (reading)
    with open(file1) as fin:
        #read all lines in f as a list
        lst = fin.readlines()
```

Penn Engineering

Property of Penn Engineering | 14

13



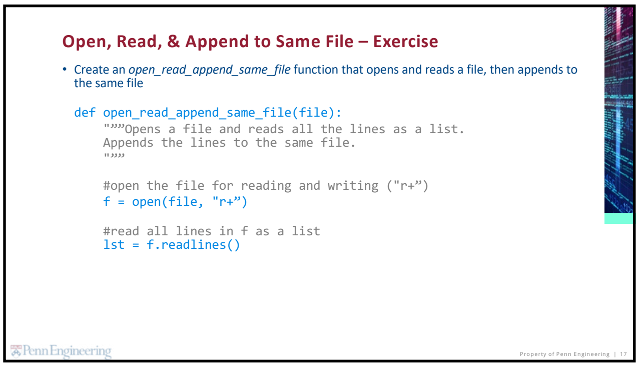
Open, Read, & Append to New File – Exercise

- Call the *open_read_append_new_file* function specifying two text files

```
def main():
    open_read_file('news.txt')
    open_read_append_new_file('news.txt', 'news_out.txt')

if __name__ == ‘__main__’:
    main()
```

Property of Penn Engineering | 16

14



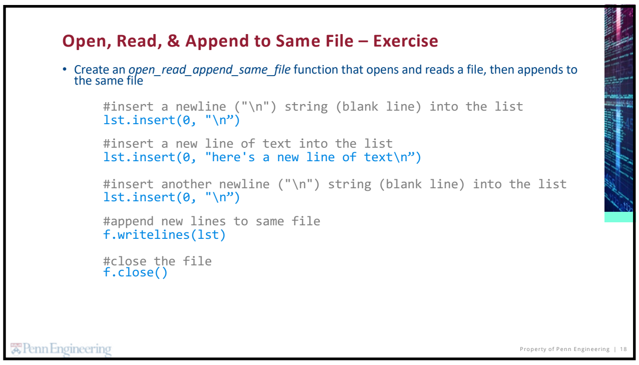
Open, Read, & Append to Same File – Exercise

- Create an *open_read_append_same_file* function that opens and reads a file, then appends to the same file

```
def open_read_append_same_file(file):
    "””Opens a file and reads all the lines as a list.
    Appends the lines to the same file.
    "””

    #open the file for reading and writing ("r+”)
    f = open(file, "r+”)

    #read all lines in f as a list
    lst = f.readlines()
```

Property of Penn Engineering | 17

15



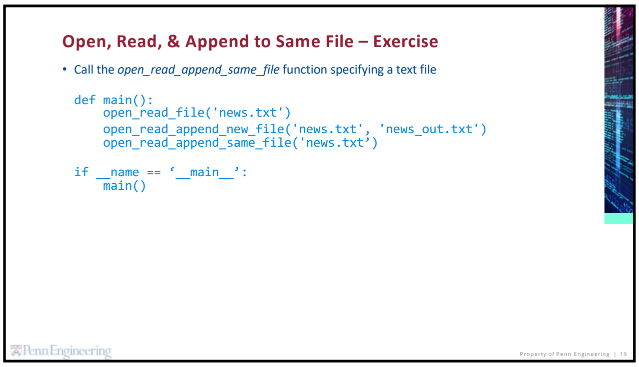
Open, Read, & Append to Same File – Exercise

- Create an *open_read_append_same_file* function that opens and reads a file, then appends to the same file

```
    #insert a newline ("\n") string (blank line) into the list
    lst.insert(0, "\n”)

    #insert a new line of text into the list
    lst.insert(0, "here's a new line of text\n”)

    #insert another newline ("\n") string (blank line) into the list
    lst.insert(0, "\n”)

    #append new lines to same file
    f.writelines(lst)

    #close the file
    f.close()
```

Property of Penn Engineering | 18

16



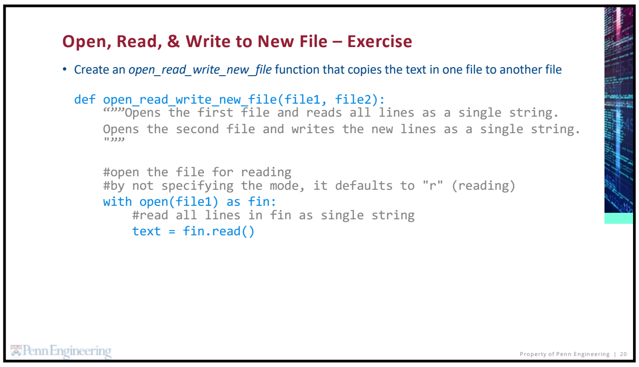
Open, Read, & Append to Same File – Exercise

- Call the *open_read_append_same_file* function specifying a text file

```
def main():
    open_read_file('news.txt')
    open_read_append_new_file('news.txt', 'news_out.txt')
    open_read_append_same_file('news.txt’)

if __name__ == ‘__main__’:
    main()
```

Property of Penn Engineering | 19

17



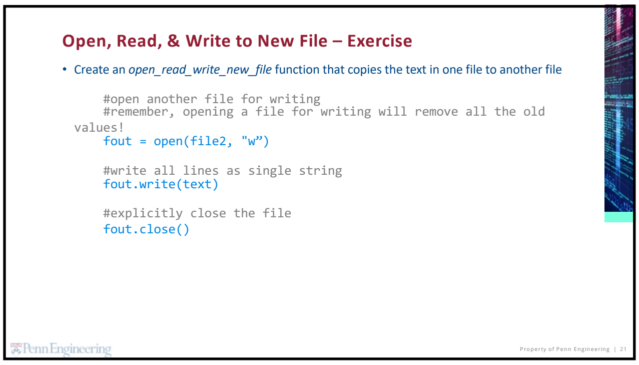
Open, Read, & Write to New File – Exercise

- Create an *open_read_write_new_file* function that copies the text in one file to another file

```
def open_read_write_new_file(file1, file2):
    “””Opens the first file and reads all lines as a single string.
    Opens the second file and writes the new lines as a single string.
    "””

    #open the file for reading
    #by not specifying the mode, it defaults to "r" (reading)
    with open(file1) as fin:
        #read all lines in fin as single string
        text = fin.read()
```

Property of Penn Engineering | 20

18



Open, Read, & Write to New File – Exercise

- Create an *open_read_write_new_file* function that copies the text in one file to another file

```
    #open another file for writing
    #remember, opening a file for writing will remove all the old
values!
    fout = open(file2, "w”)

    #write all lines as single string
    fout.write(text)

    #explicitly close the file
    fout.close()
```

Property of Penn Engineering | 21

19