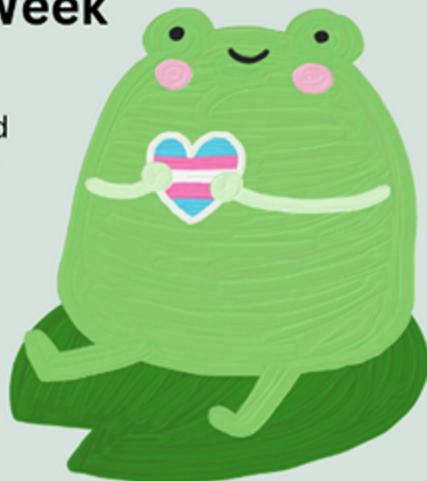


## LGBTQ+ Western proudly presents: Trans Week

These sessions are open for all, including students, faculty, staff, and community members centering our trans and nonbinary community at Western.



Follow us on Instagram to stay updated on events including partnerships with student organizations @lgbtqwestern



### Trans Week Events:

#### **Grab & Go Clothing Swap:**

M Nov 14th, 2-5 pm VU 723 | LGBTQ+ Western Office

#### **Trans Tea:** T Nov 15th, 4-5pm VU 462

**Trans Takeover Open Mic Night:** T Nov 15th, 7-9 pm, Viking Union Underground Coffee House, 3rd floor

#### **Black LGBTQ+ Thriving Collective Kickoff & Jeopardy:**

W Nov 16th, 6-8pm VU Multicultural Center Kitchen

#### **Info Table and Story Sharing:**

Th Nov 17th, 11-2 VU Lobby near Bookstore

#### **Boundaries Workshop with Counseling & Wellness:**

Th Nov 17th, 3-4:30pm VU Multicultural Center 735

#### **Trans Day of Remembrance Vigil:**

Fri Nov 18th, 10:30-12:30pm

VU Multicultural Center 735

#### **Co-Program with Roots: Adoption Month Celebration**

Fri Nov 18th VU MPR

10-3 (Presentation about LGBTQ+ Adoption at 1pm)



AA/EQ

For disability accommodations and event questions,  
please contact lgbtq@wwu.edu

## Other file I/O tips

Monday, February 28, 2022 12:32 AM

Want to skip lines that are comments?

```
if "#" in line:  
    continue
```

file locations

data.txt # same folder  
.. /data.txt # up a folder  
.. /myData /data.txt # up a folder, in  
 folder myData

/myData /data.txt # in this folder, in a  
 folder myData

Folders must exist already  
(you can make a folder but out of scope  
for this class)

input\_file.readline()  $\Rightarrow$  one line, including the  
newline

input\_file.readlines()  $\Rightarrow$  list of strings

Output-file = open("data.csv", "a")

a for append  $\rightarrow$  will just add to the end

"to" appends will just add to the end  
Many other tools available, you can look up  
and use whatever you like

# File seek tell read(3)

Wednesday, May 25, 2022 3:00 AM

```
1 infile = open("test.txt", "r")
2 print(infile.read(3))
3 print(infile.read(3))
4 infile.seek(0)
5 print(infile.read(3))
6
7 print(infile.tell())
8 infile.close()
9
10 with open("test.txt") as infile:
11     for line in infile:
12         print(line)
13
```

```
infile = open("test.txt", "r")
print(infile.read(3))
print(infile.read(3))
infile.seek(0)
print(infile.read(3))

print(infile.tell())
infile.close()
```

```
with open("test.txt") as infile:
    for line in infile:
        print(line)
```

## Write demo

Monday, May 23, 2022 1:38 PM

## Write Read Demo

20

```
1 # Author: Scott Wehrwein
2 # Date: 11/22/2019
3 # Demo reading from and writing to files.
4
5 def rev_words(in_filename, out_filename):
6     """ Write the contents of in_file to out_file, but with
7         each word written in reverse order. """
8     infile = open(in_filename, 'r')
9     outfile = open(out_filename, 'w')
10
11    # loop over lines of the file
12    for line in infile:
13        # newlines aren't remove automatically, so we do it here:
14        line = line.strip('\n')
15        line_words = line.split() # split the line on whitespace
16        for word in line_words: # for each word
17            # write the characters in reverse order:
18            for i in range(len(word)-1, -1, -1):
19                outfile.write(word[i])
20
21            outfile.write(" ") # put a space between each word
22            outfile.write('\n') # put a newline at the end of each line
23
24    # close both files
25    infile.close()
26    outfile.close()
27
28
29 rev_words("test.txt", "llorkcir.txt")
```

open a file for read  
another for write

→ by line  
→ remove Newline  
→ remove white space  
→ row by word  
→ reverse order  
→ write  
→ Space  
→ Newline

Be in this dir

At line 18 stop and let them try to complete this

# Read demo

Monday, May 23, 2022 1:39 PM

Using [https://wwu2-my.sharepoint.com/:t/g/personal/hardinc3\\_wwu\\_edu/EaUQs2HU8JNjnx\\_QlnlWa-8BPPBoc8LdUn56Jw6NdKcjN3Q?e=gzl2sn](https://wwu2-my.sharepoint.com/:t/g/personal/hardinc3_wwu_edu/EaUQs2HU8JNjnx_QlnlWa-8BPPBoc8LdUn56Jw6NdKcjN3Q?e=gzl2sn)

Read Demo 24.5

```
1 # Author: Scott Wehrwein
2 # Date: 11/22/2019
3 # Demo of various methods of file objects that
4 # are used to read strings from a file.
5
6
7 # To demo:
8 # read(size) - read the next size characters from the file
9 # read() - read the entire file into a string
10 # readline() - return a list of lines of the file
11 # seek(location) - move the cursor to location in the file
12 # tell() - return the current location of the file
13
14 # demo for iterating over lines of a file:
15 file_obj = open("test.txt", 'r')
16 for line in file_obj:
17     print(line)
18
19 file_obj.close()
20 # Notice: newline characters come with each line
21 #         use line.strip("\n") to remove
22
23 # To get the lines as they're iterated over above, but in
24 # list form, you can use readlines():
25 file_obj = open("test.txt", "r")
26 all_lines = file_obj.readlines()
27 file_obj.close()
28
29 # each element of all_lines is now a line of the file.
30 # as before, newlines are not removed.
31
32
33 # demo for seek, read, and tell. There's a cursor used by read.
34
35 # For the example's sake, suppose test_short.txt contains:
36 # Never gonna give you up
37 fobj = open("test_short.txt", "r")
38 print(fobj.read(3)) # Nev
39 print(fobj.read(4)) # er g
40 fobj.seek(2) # cursor is now at index 2, so 'v' is the next character to be read
41
42 # read with no argument prints the rest of the file,
43 # starting at the current cursor location:
44 print(fobj.read()) # ver gonna give you up
45
46 # tell returns the current cursor position
47 fobj.seek(10) # seek back to position 10
48 print(fobj.tell()) # 10
49
50 # extra - won't be on the exam, but is useful:
51 #with open("test.txt") as file_obj:
52 #     for line in file_obj:
53 #         print(line)
54
55 # file_obj is automatically closed when we exit the
56 # indented block of the with statement
```

iterate over lines → remember to close! after close errors if try to use

make list for iterating

There's a cursor used by read

Reading from a start index

Setting a index

No args = rest

Move cursor off file tell cursor location

a really nice for loop with built in close

## Csv file read

Monday, May 23, 2022 1:39 PM

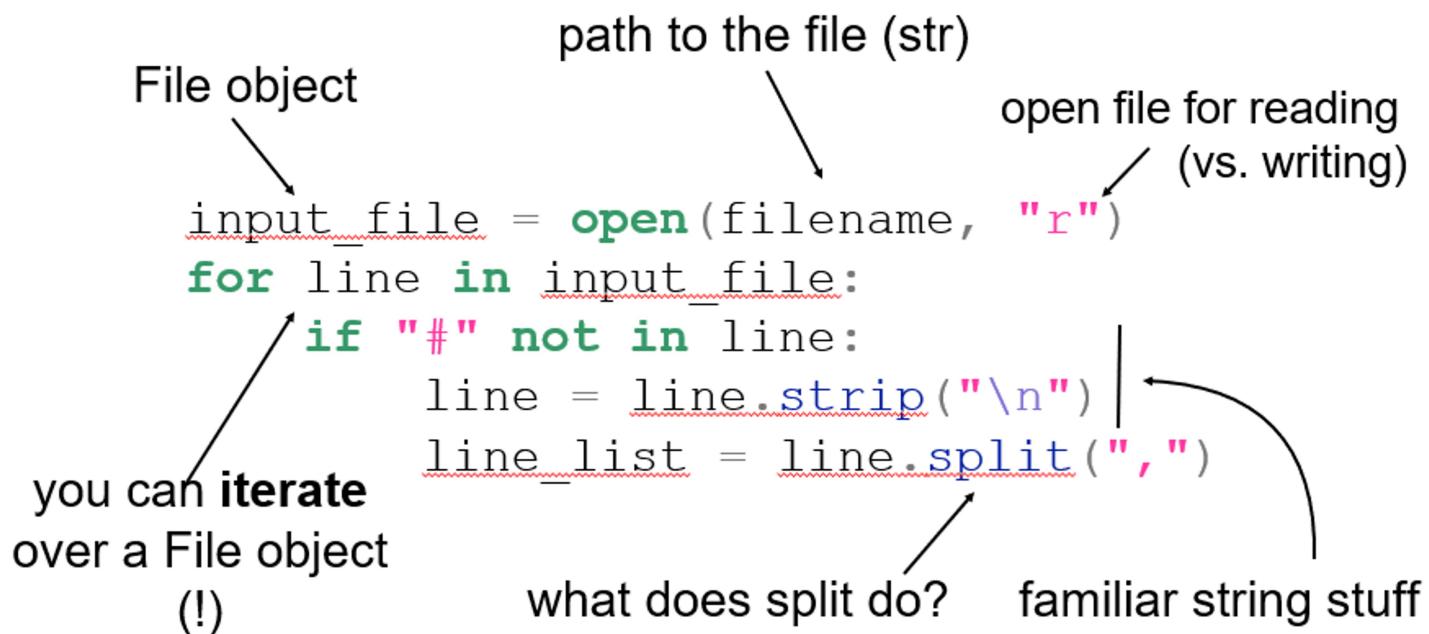
Suppose the file rick.txt contains:

```
Never gonna give you up
```

What is the output of the following code?

```
print(open("rick.txt", "r").read(5).split("e"))
```

- A. Nvr
- B. Never
- C. ["N", "e", "v", "e", "r"]
- D. ["N", "v", "r"]



# challenge

```
def z4 (d1, d2):
    a = d1
    d1 = {}
    d1 = d2
    d1[ "A" ] = 2
    return a

a = { "A": 4, "B": 6 }
b = { "A": 6, "B": 11 }
f = z4 (a, b)
print (a[ "A" ], b[ "A" ], f[ "A" ])
```