Note: This is a new POGIL! Please let Iris know if you see any issues!

Name:_____ **Partners:** _____

Python Activity 24: Reading from Files

Files are persistent data, we can access it between sessions and from other sources!

Learning Objectives
Students will be able to:
Content:
- Explain how to open a text file for reading line-by-line using **for-each loops**.
- Explain the purpose of the **open()…as** block**.**

Process:
- Write code that opens, reads from, and closes a file

Prior Knowledge
- Python concepts: for-each loops, conditionals, if name, iteration over strings, newline

Critical Thinking Questions:

FYI: In Python, you can access data from a text file as well as from the keyboard. You can create a text file in any text editor.

1. Read through the following code and observe its sports.txt file on the left and output on the right.

sports.txt

```
soccer
cricket
hockey
tennis
volleyball
```

Python Program

```
def process_sports():
      with open("sports.txt") as sp_file:
            for line in sp_file:
                  print(line)

## Call the function when run as a script! ##
if __name__ == "__main__":
      process_sports()
```

Output

```
soccer

cricket

hockey

tennis

volleyball
```

a. What might the program do? _____

_____

b. In the first line of the function, what might the first **parameter value** (a string) for the **built-in function** `open()` represent?

_____

c. There are several levels of **indentation** in this code example. What might the indentation after the `with open()` line indicate?

_____

FYI: When we open a file, we must also close it in order to prevent memory leaks. In python, when we use a `with..as` block, the file is **implicitly closed** when we leave the indentation of the block.

d. What character is likely forming the empty line between each sport in the output?

e. What do you think the command `len('\n')` will output? _____
It actually outputs 1. Why might that be? _____

f. Write a function, `trim(line)` that takes a string argument, `line`, and returns a new string of the line with the trailing newline character (`'\n'`), if any, removed.

```
def trim(line):
```

_____

_____

_____

_____

_____

_____

_____

f. Rewrite the line of code in our Python Program example above to use your `trim()` function so each sport is printed without a newline at the end.

_____

4. Sometimes, we want to manipulate data in **comma-separated values (CSV) files**, like the **dogs.csv** file below. (If running this outside of class, ensure that the python file, text file, & your Terminal are all located in the same directory).

Python Program

```
def process_dogs():
      with open("dogs.csv") as dog_file:
            for dog_name in dog_file:
                  print(trim(dog_name))

## Call the function when run as a script! ##
if __name__ == "__main__":
      process_dogs()
```

dogs.csv file

```
Albrecht,Linus
Albrecht,Sally L.
Doret,Jerry
Freund,Wally
Howley,Pixel
Jannen,Artie
Keith,Velma
Shaw,Fiona
Shaw,Maisie
```

a. What might be the output of this program?

b. The first iteration through the for-each loop, "`Albrecht,Linus`" is stored in the variable `dog_name`. Write some code that will turn `dog_name` into two variables: `first_name` that stores the dog's first name and `last_name` that stores the dog's last name:

c. Where would be place code from (b) in the `process_dogs()` function, if we want to display the dog name in the style of "Sally L. Albrecht"?

d. Do we have to write any special code to handle the middle initial edge case?

e. Write a function, `split(line, delim)` that takes string arguments, `line` and `delim`, and returns a list of strings containing each item of `line` split based upon the character `delim`.

```
def split(line, delim):
    """
    >>> split('A, quick, brown,fox', ',')
    ['A', ' quick', ' brown', 'fox']
    """
```

Application Questions: Use the Python Interpreter to check your work

1. Create a text file that contains 10 numbers between 50 and 100. Write a program that reads the numbers from the file and totals the numbers. The program should print all the numbers and display the total when all the numbers have been added together. (Warning! The input from the file will be considered a string. Be sure to convert the input to **int** or **float** – just as you do when numbers are entered from the keyboard.)

2. a. Rewrite the following program to allow the user to enter the name of the file. Use the input to open the file.

```
def process_sports():
     with open("sports.txt",) as sp_file:
          for sport in sp_file:
               print(strip(sport))

##### Call the function! #####
if __name__ == "__main__":
     process_sports()
```

b. Rewrite the function above to read *only the first 10 lines of the file*:

```
def process10sports():
```

```
##### Call the function! #####
if __name__ == "__main__":
     process10sports()
```