Name:______________________________ **Partner:** ________________________

Python Activity 11: Lists of Lists

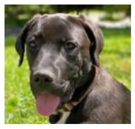
Lots of data requires a table or a matrix…

Learning Objectives
Students will be able to:
Content:
- Define a **nested list** or **list of lists**
- Identify **empty lists** and **empty strings**

Process:
- Write code to construct and add elements to lists of lists
- Write code to access elements at a given index
- Write code to iterate over lists of lists

Prior Knowledge
- Python concepts: lists, types, len(), string literals

Critical Thinking Questions:



1. Examine the sample code defining a list below.

Sample Code

```
dog2owner =
[["pixel","iris"],["chels","lida"],["artie","bill"]]
```

a. What element is at `dog2owner[0]`? ______________________________

What is this element's data type (circle one): string list int bool

b. Within `dog2owner[0]`, what is stored at index **1**? ____________________

What is this element's data type (circle one): string list int bool

c. We can access this same string value using only list indexing with `dog2owner[0][1]`.

How might we access the name of Iris' dog using list indexing? __________________

d. What element is in `dog2owner[2]`? __________________________________

Within `dog2owner[2]`, what is stored at index 0? _________________________

How would we write this with list indexing? ______________________________

e. Write a line of code to access and print the name of Lida's dog via list indexing:

____

f. When working with nested lists (such as in the Sample Code above), what does the *first* list index refer to? ____

What does the *second* list index refer to? _________________________________

2. Examine the two lists below:

```
alst = [ ["cat", "frog"],      | blst = ["cat", "frog",
         ["puma", "toad"],     |         "puma", "toad",
         ["lion", "newt"] ]    |         "lion", "newt"]
```

a. What element is stored at `alst[1][0]`? ________________________________

What is this element's data type (circle one): string list int bool

b. What element is stored at `blst[1][0]`? ________________________________

What is this element's data type (circle one): string list int bool

c. What kind of list is `alst`? *A list of* ________________________________

What kind of list is `blst`? *A list of* ________________________________

How do you know?

____

FYI: ***Lists*** are a sequence of elements and these elements can be of *any* data type, including more lists! While python does not require us to specify a variable's data type, we often assume the list has elements of a particular type. ***Lists of lists*** and ***lists of strings*** are easy to mix up as both lists and strings are sequences!

3. Examine the sample code below, it has a ***logic error***:

```
pet2age = [ ["pixel", "dog", 4], ["dizzy", "cat", 10] ]
pet2age = pet2age + ["moone", "demon", 2]
print(pet2age)
```

And its output:

```
[['pixel', 'dog', 4], ['dizzy', 'cat', 10], 'moone', 'demon', 2]
```

a. What kind of object is `pet2age` (circle one): string list int bool

b. What kind of objects are stored in `pet2age`: string list int bool

c. What kind of object did the programmer *try* to add in the second line of code? ________

What kind of object did the programmer *actually* add? ______________

d. What line of code should the programmer have written to ensure the new element added was of the same type as the rest of the elements in `pet2age`?

____

4. Examine the sample code below:

```
pet2age = [ ["pixel", 4], ["dizzy", 10], ["moone", 1] ]

cats_first = [ pet2age[-1], pet2age[1], pet2age[0]]
```

a. In the first line of code, what is stored in `pet2age[-1]`:

____

In the first line of code, what is stored in `pet2age[1]`:

____

In the first line of code, what is stored in `pet2age[0]`:

____

b. What might this sample code do?

____

c. We can achieve a similar output in `cats_first` by *reversing* our list of lists. How might we do that? ______________________________

5. Examine the three interactive python sessions below:

```
>>> new_str = ""    |  >>> new_lst = []    |  >>> lst_lst = [[]]
>>> len(new_str)    |  >>> len(new_lst)    |  >>> len(lst_lst)
0                   |  0                   |  1
```

a. Why might `len(new_str)` return 0? __________________
Why might `len(new_lst)` return 0? __________________
Why might `len(lst_lst)` return 1? __________________

b. Which of the above variables would we describe as an ***empty list***? __________________
Which of the above variables would we describe as an ***empty string***? __________________
Which of the above variables is not empty? __________________

c. What might the code `len("  ")` return ? __________________
Why?

____

d. What might the code `len([""])` return ? __________________
Why?

____

Application Questions: Use the Python Interpreter to check your work

1. Write a function, `switcheroo`, that take a list of lists, `lol`, as a parameter, and returns a new list of lists that has swapped the first and last items of each element of the list of lists.

____

____

____

____