Lists

Part I: Review -- Data types

We have learned about two **data types** so far in class. What are they?

1. ____

2. ____

In total there are 5 main data types in Python and we are about to learn about the third – **Lists**!

Part II: What is a list?

A list is a data type that can store multiple values. Think of it like a list you might make such as a grocery list or a back-to-school shopping list. Here's the **syntax** for writing a list in Python:

[“apples”, “oranges”, “bananas”, “grapes”]

Name two things you notice about the list syntax:

1. ____

2. ____

Just like our other datatypes, a list can be assigned to a **variable**. What would be a good variable name for the list example above?

______________________ = [“apples”, “oranges”, “bananas”, “grapes”]

So, above you're seeing a list of strings. But a list can hold other data types as well.

A **list** can consist of **strings** and **integers**:

[1, “hello”, 4555, “what is up with this list?”, “this list is so crazy!”, 12.4]

A list can even hold lists! Here is a list that contains the three data types we know and love!

[[1, 2, 3, 4], 4.5, “wow, lists are amazing!”, [“I”, “love”, “this”, “list”], 10000]

And just like with strings, we can use this index value to find specific list items.

```
shopping_list = ["eggs", "milk", "broccoli", "cheese", "pasta"]
```

Position	0	1	2	3	4
String characters	eggs	milk	broccoli	cheese	pasta

`len(shopping_list)` **guess:______________ output: ________________**

Indexing a list is the same as indexing a string in python. Try it!

`print(shopping_list[2])` **guess:______________ output: ________________**
`print(shopping_list[0])` **guess:______________ output: ________________**
`print(shopping_list[-1])` **guess:______________ output: ________________**
`print(shopping_list[4])` **guess:______________ output: ________________**

You can even index an item in a list!
`print(shopping_list[4][1])` **guess:______________ output: ________________**

What is the index for broccoli in shopping_list? ________________________

We can concatenate lists the same way we concatenate strings.

```
new_list = shopping_list + ["twix", "ice cream"]
print(new_list)
```

1. Assign variables greeting and name to the strings below. Then match the code with the correct output by drawing a line.

CODE	**OUTPUT**
`print(shopping_list[:3])`	5
`print(shopping_list[-1][2])`	['cheese', 'pasta']
`print(shopping_list[1]`	ERROR
`print(shopping_list[3:5])`	8
`print(shopping_list[2:])`	['milk', 'broccoli', 'cheese']
`print(shopping_list[-2])`	['eggs', 'milk', 'broccoli']
`print(shopping_list[5][3])`	'milk'
`len(shopping_list[2])`	‘s’
`len(shopping_list)`	‘cheese’

Here is a new list:

```
crazy_list = [[1, 2, 3, 4], 4.5, “wow, lists are amazing!”, [“I”,
“love”, “this”, “list”], 10000]
```

`crazy_list[0]` = ____

`crazy_list[2]` = ____

`crazy_list[-1]` = ____

`crazy_list[3]` = ____

Okay, buckle up, because now things are going to get really crazy!
What if I told you that, not only can you index a string, AND a list, but also an indexable item on a list? Here's what it looks like:

```
amazing_list = [[1, 2, 3, 4], "wow, lists are amazing!", ["I", "love",
"this", "list"], "bye!"]
```

If:

```
amazing_list[2] = ["I", "love", "this", "list"]
```

What do you think the following code returns?

`amazing_list[2][1]` = ____

WOWZA! Let's practice some more:

`amazing_list[0][0]` = ____

`amazing_list[1][6]` = ____

`amazing_list[-1][-1]` = ____

`amazing_list[3][1]` = ____

2. Create a program that prints the following:
    a. the first, third and last item.
    b. the second character of the second item in the list
    c. the last character of the fourth item in the list.
    d. Add two more items to your list and print it!

```
myList= [ #..... make a list of 5 items of your choice!
print(myList[#..... try it!
```