Rex Jones II

Intro To Python List

List Introduction

In this session, I will discuss working with a List Data Type. It's a data type that contains multiple values. Therefore, we can put a lot of data in a list. The purpose is to manage the sequential data in a structure so we can keep track of the data. If you are interested in more content, feel free to read the transcript on GitHub or download the source code. Also, like this video and subscribe to my channel. Plus follow me on Twitter, connect with me on LinkedIn and Facebook.

Create List

In the IDE, to create a list, we start with an opening and closing bracket [ ]. Each value such as 5, 8, 2, 1, 7, 8 in the list is called an item or element that is separated by a comma. Since there are multiple items, normally we assign the values to a name that ends with an 's' like numbers =. We can create a collection of items that include numbers, True/False values that are Boolean, and letters = like ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']. If the values are related, it can also be a combination of numbers and letters. Python also allows us to create a list within a list like Booleans = values that contains values within the square brackets [[], [], []] that are separated by a comma [[0, 'False'], [1, 'True'], [-1, 'Error']].

```
numbers = [5, 8, 2, 1, 7, 8]
letters = ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
booleans = [[0, 'False'], [1,'True'], [-1, 'Error']]
```

Find The Length and Count The Items

To print(), we pass in the name of the list such as numbers? print(letters) and print(booleans).

```
print(numbers)
print(letters)
print(booleans)
```

When I run, we see each value in the console for numbers, letters, and booleans.

```
[5, 8, 2, 1, 7, 8]
['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
[[0, 'False'], [1, 'True'], [-1, 'Error']]
```

We have a function that count the number of items in a list or how many times an item appear in a list and that is the len() function which means length. So if you want to see how many items are in a list. We place len before each list name: len(numbers), len(letters), and len(booleans).

```
print(len(numbers))
print(len(letters))
print(len(booleans))
```

Run and the console returns 6, 7, 3.

```
numbers = [5, 8, 2, 1, 7, 8]
letters = ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
booleans = [[0, 'False'], [1,'True'], [-1, 'Error']]
```

```
lists
C:\Users\RexJo\PycharmProjects\pythonProject\venv\Scripts
 C:/Users/RexJo/PycharmProjects/pythonProject/lists.py
6
7
3
```

Notice, how the list within a list shows 3 and not 6. That's because each list is considered 1 value. However, the len() function is different from the count() method. The count method returns how many times a particular item is in a list. For example, if I erase these print statements and type print(numbers.count(5)) then run.

```
numbers = [5, 8, 2, 1, 7, 8]
letters = ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
booleans = [[0, 'False'], [1,'True'], [-1, 'Error']]

print(numbers.count(5))
```

We see 1 in the console.

```
1
```

Therefore, 1 in the console shows how many times 5 appears in the list. Now, if I change 5 to 8.

```
numbers = [5, 8, 2, 1, 7, 8]
letters = ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
booleans = [[0, 'False'], [1,'True'], [-1, 'Error']]

print(numbers.count(8))
```

We see 8 shows up twice, so when I run. The console shows 2.

```
2
```

Access A Specific Element

To access each item in a list, we write print(numbers[]) then pass in the location. Each item in the list has an index position that starts with 0. Therefore, the 1st item which is 5 represents index 0. Inside the square brackets, when I write 0 and run. The console returns 5 because it is the 1st item which is index 0.

```
numbers = [5, 8, 2, 1, 7, 8]
letters = ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
booleans = [[0, 'False'], [1,'True'], [-1, 'Error']]

print(numbers[0])
```

```
5
```

I'm going to pass in another value. We can also count backwards to count the last index using a negative number. However, the index starts at 1. Let's use a different list by changing numbers to letters then pass in negative 1 (-1) then run. The console shows James because it is the last item in the letters list.

```
numbers = [5, 8, 2, 1, 7, 8]
letters = ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
booleans = [[0, 'False'], [1,'True'], [-1, 'Error']]

print(letters[-1])
```

```
James
```

Accessing a particular item is different for a list within a list. For example, if I replace letters with booleans then run. We see [-1, 'Error'] in the console because it's also the last item.

```
numbers = [5, 8, 2, 1, 7, 8]
letters = ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
booleans = [[0, 'False'], [1,'True'], [-1, 'Error']]

print(booleans[-1])
```

```
[-1, 'Error']
```

Replace -1 with 0, run, and the console shows [0, ‘False’].

```
numbers = [5, 8, 2, 1, 7, 8]
letters = ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
booleans = [[0, 'False'], [1,'True'], [-1, 'Error']]

print(booleans[0])
```

```
[0, 'False']
```

To access an individual item for a list within a list, we need 2 sets of square brackets [] []. Let’s add a comment #. We know 0 and False is index position 0. So, I will write 0,0 for 0 and 0,1 for False. Index position 1 is 1 and True. Therefore, I’ll write 1,0 for 1 and 1,1 for True. It’s the same for the last item position 2. 2,0 is -1 and 2,1 is Error. So, if I want to print True the I add [1][1] in the square brackets. Run and the console shows True which is 1, 1.

```
numbers = [5, 8, 2, 1, 7, 8]
letters = ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
booleans = [[0, 'False'], [1,'True'], [-1, 'Error']]
#            0,0    0,1    1,0  1,1     2,0     2,1

print(booleans[1][1])
```

```
True
```

Index Errors & Type Errors

Let's look at a couple of errors for the numbers list. We see there are 6 items so if I write print(numbers(6)) and run. The console shows IndexError that says list index out of range.

```
numbers = [5, 8, 2, 1, 7, 8]
letters = ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
booleans = [[0, 'False'], [1,'True'], [-1, 'Error']]
#           0,0    0,1    1,0  1,1     2,0     2,1

print(numbers[6])
```

```
    print(numbers[6])
IndexError: list index out of range
```

It's out of range because the index starts at 0 and the last index stops at 5. Another error is the TypeError but let me explain the IndexError. Python searches this list and realize there is no index 6. That's why it throws an IndexError because it's out of range for 6. If I pass in 5 then it will be all good for the number 8.

```
numbers = [5, 8, 2, 1, 7, 8]
letters = ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
booleans = [[0, 'False'], [1,'True'], [-1, 'Error']]
#           0,0    0,1    1,0  1,1     2,0     2,1

print(numbers[5])
```

```
8
```

Python only allow integer values for the index position and not float values. Therefore, if I change 5 to 1.0 and run. The console shows TypeError and does not print the 2nd value.

```
numbers = [5, 8, 2, 1, 7, 8]
letters = ['A', 'a', 'X', 'b', 'Jane', 'John', 'James']
booleans = [[0, 'False'], [1,'True'], [-1, 'Error']]
#           0,0    0,1    1,0  1,1     2,0     2,1

print(numbers[1.0])
```

```
    print(numbers[1.0])
TypeError: list indices must be integers or slices
```

The error says list indices must be integers or slices. Next, I will show you how to slice a list. Also, demo how to add items, remove items, and sort items in a list.

Contact Info

- ✔ Email Rex.Jones@Test4Success.org
- ✔ YouTube https://www.youtube.com/c/RexJonesII/videos
- ✔ Facebook https://facebook.com/JonesRexII
- ✔ Twitter https://twitter.com/RexJonesII
- ✔ GitHub https://github.com/RexJonesII/Free-Videos
- ✔ LinkedIn https://www.linkedin.com/in/rexjones34/