Learning in REPL

start your python repl
with either python or python3

```
py$ my_friends = [
py> 'Bridget Villalobos',
py> 'Atlas Knapp',
py> 'Bridget Villalobos',
py> ]
py$
py$ my_friends
```

What has changed in the repl next to py?
Why did it change?
What key on the keyboard was used to move to a new line?
What type of list is 'my_friends' ?

```
py$ inspect(my_friends)
                        <class 'list'>
 Built-in mutable sequence.

  ['Bridget Villalobos', 'Atlas Knapp', 'Bridget Villalobos']

 36 attribute(s) not shown. Run inspect(inspect) for options.
```

What can be done with the list?

in the repl type
'name.' and press the tab key on your keyboard.
You should get

```
py$ my_friends.
my_friends.append(   my_friends.extend(   my_friends.remove(
```

```
my_friends.clear()   my_friends.index(
my_friends.reverse()
my_friends.copy()    my_friends.insert(   my_friends.sort(
my_friends.count(    my_friends.pop(
```

How many of those are you already familiar with?
What is an index?
What is the count? How does it work?

Let's try append

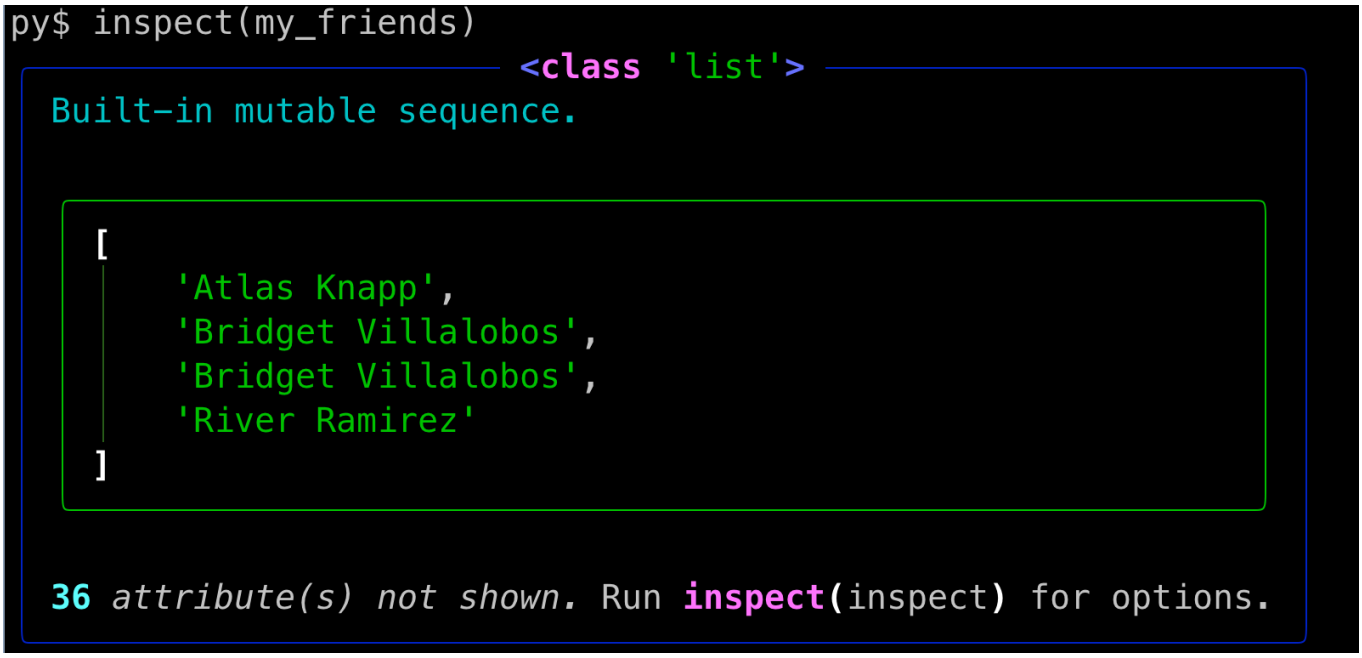
```
py$ my_friends.append('River Ramirez')
py$ my_friends
['Bridget Villalobos', 'Atlas Knapp', 'Bridget Villalobos',
'River Ramirez']
```

```
py$ my_friends.reverse()
py$ my_friends
['River Ramirez', 'Bridget Villalobos', 'Atlas Knapp',
'Bridget Villalobos']
```

```
py$ my_friends.sort()
py$ my_friends
['Atlas Knapp', 'Bridget Villalobos', 'Bridget Villalobos',
'River Ramirez']
```

using the rich api use inspect to explore 'my_friends'

What's changed between the one above and this one?



What if you had duplicate entires ?

```
py$ my_friends
[
    'Atlas Knapp',
    'Bridget Villalobos',
    'Bridget Villalobos',
    'River Ramirez',
    'Bridget Villalobos'
]
```

```
py$ no_dupes = list(set(my_friends))
py$ no_dupes
['Bridget Villalobos', 'Atlas Knapp', 'River Ramirez']
```

We're using the list and set modules
read them

```
help(list)
```

When you read about sets -you learn that set does the following: Build an unordered collection of unique elements.

Add a friend to a specific location in the set.

```
py$ my_friends.insert(1, 'Jordan Lee')
py$ my_friends
[
    'Atlas Knapp',
    'Jordan Lee',
    'Bridget Villalobos',
    'Bridget Villalobos',
    'River Ramirez',
    'Bridget Villalobos'
]
```

Why is 'Jordan Lee' in the second position? We told it '1'.
What would put Jordan in 'Atlas Knapp' position?

What index position is River Ramirez?

```
py$ my_friends.index('River Ramirez')
4
```

What about sorting and ordering can we do that?

What command was used for the first line of code below?

```
my_friends.____()
```

```
py$ my_friends
[
    'Atlas Knapp',
    'Bridget Villalobos',
    'Bridget Villalobos',
    'Bridget Villalobos',
    'Jordan Lee',
    'River Ramirez'
```

```
]
py$ my_friends.reverse()
py$ my_friends
[
    'River Ramirez',
    'Jordan Lee',
    'Bridget Villalobos',
    'Bridget Villalobos',
    'Bridget Villalobos',
    'Atlas Knapp'
]
```

What are your take aways?

What did you learn?

How would you use this?