List Examples

In this lesson, we're going to learn about **lists** and **dictionaries**.

Suppose we have three different sets of variables:

```
customer1 = "John Doe"
pickupLocation1 = "350 5th Ave"
package1 = customer1 + pickupLocation1

customer2 = "Jane Doe"
pickupLocation2 = "100 7th Ave"
package2 = customer2 + pickupLocation2

customer3 = "Joe Daniels"
pickupLocation3 = "11 1st Ave"
package3 = customer3 + pickupLocation3
```

Now let's create a list of these customer:

```
customerList = [customer1, customer2, customer3]
print(customerList[1])
```

List is opened and closed with `[` and `]`; inside you add list's elements.

Lists are indexed from 0, so by saying `customerList[1]` you reference the **second** element, not the first one.

Print Statement

To append something to an existing list, use `append`:

```
customer4 = "Jason, Bourne"
pickupLocation4 = "333 Lexington Ave"
package4 = customer4 + pickupLocation4

customerList.append(customer4)
```

Let's check length of the list now:

```
print(len(customerList))
```

Later you can remove element from the list:

```
customerList.remove(customer4)
print(len(customerList))
```

What if I want to reassign a customer?

```
customerList[0] = "John Doe II"
```

```
print(customerList[0])
print(customer1)
```

You can see than `customer1` retains its original value so we're just changing the entry inside of the list, but not the actual variable that was created.

Working with Dictionary

Now I'm going to create an address dictionary:

```
addressList = {customer1:pickupLocation1, customer2:pickupLocation2, customer3:pickupLocation3}
print(len(addressList))
print(addressList[customer2])
```

Dictionaries are opened and closed with `{ }` brackets. In dictionaries you have a key and a value delimited with `:`. Elements inside dictionaries are separated with `,`, just like in lists.

In this example I use customers as keys and pickup locations as values. Later I can reference any value by its key saying `addressList[customer2]`.