Lists

Lists

Lists allow you to use one identifier to group similar items together

coworkers = ["Sarah", "Matt", "Sophie"]

price = [32.23, 12.25, 56.38, 77.55, 39.0]

Creating a List

```
# Create an empty list
myList = []

# Create a list with stuff in it
coworkers = ["Sarah", "Matt", "Sophie"]
```

Indices

The index indicates which value in the array is being accessed.

coworkers = ["Sarah", "Matt", "Sophie"]

Indices

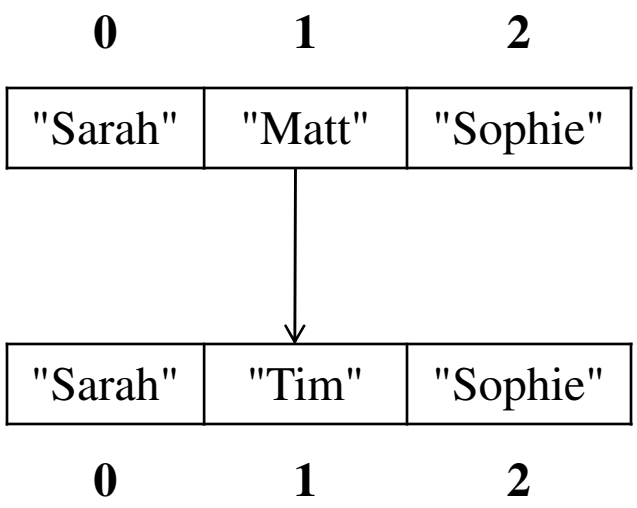
0	1	2
"Sarah"	"Matt"	"Sophie"

Indices

Indices allow you to get or change data in a list

```
coworkers = ["Sarah",
             "Matt",
             "Sophie"]

coworkers[1] = "Tim"
```



Getting data from a List

```
coworkers = ["Sarah", "Matt", "Sophie"]

# Print the items in the list
print coworkers[0]
print coworkers[2]
```

Output

Sarah
Sophie

0	1	2
"Sarah"	"Matt"	"Sophie"

Getting data from a List

```
myList = [2, 1, 31, 7, 5, 12, 8]

# Print the items in the list
print myList[1 + 2]
print myList[9 - 3]
print myList[11 / 2]
```

Output

```
7
8
12
```

0	1	2	3	4	5	6
2	1	31	7	5	12	8

open lists.py

Lists

frequently used methods

Name	Use
list.append(x)	Adds x to the end of *list*
list.insert(i, x)	Inserts x at position i in *list*
list.remove(x)	Remove the first item of the *list* whose value is x
list.pop(i])	Remove the item at position i in the list, and return it
len(*list*)	Gets the length of *list*

Adding data to the end of a list

The append method allows you to add data to the end of a list

```
coworkers = ["Sarah", "Matt", "Sophie"]

# Add a name to coworkers
coworkers.append("Larry")
print coworkers
```

Output

```
['Sarah', 'Matt', 'Sophie', 'Larry']
```

Adding data in the middle of a list

The insert method allows you to add data to the middle of a list

```
coworkers = ["Sarah", "Matt", "Sophie"]

# Add a name to coworkers at index 1
coworkers.insert(1, "Larry")
print coworkers
```

Output

```
['Sarah', 'Larry', 'Matt', 'Sophie']
```

Removing data from a List

The remove method allows you to remove data from a list

```
coworkers = ["Sarah", "Matt", "Sophie"]

# Remove a name from coworkers
coworkers.remove("Sarah")
print coworkers
```

Output

```
['Matt', 'Sophie']
```

Removing data from a list with an index

The pop method allows you to remove data from a specific index in the list

```
coworkers = ["Sarah", "Matt", "Sophie"]

# Remove a name from coworkers
coworkers.pop(1)
print coworkers
```

Output

['Sarah', 'Sophie']

Length of a list

The len method returns the length of a list

```
coworkers = ["Sarah", "Matt", "Sophie"]

# Print length of coworkers
print len(coworkers)
```

Output

3

open listmethods.py

Tuples

Tuples are lists that cannot be changed

```
myTuple = (255, 0, 89)

print myTuple
```

Output

(255, 0, 89)

Tuples

The data is accessed with indices

```
myTuple = (255, 0, 89)

print myTuple[2]
```

Output

89

Tuples

Since tuples cannot be changed, they have no methods

```
myTuple = (255, 0, 89)

myTuple.append(35)
```

```
AttributeError: 'tuple' object has
no attribute 'append'
```

open tuples.py

Using loops to add items to a list

```
myList = []

for num in range(5):
    myList.append(num)

print myList
```

Output

[0, 1, 2, 3, 4]

Using a for each loop to go through all items in a list

num will represent each number in myList.

```
myList = [56,65,98,2,25]

for num in myList:
    print num
```

Output

```
56
65
98
2
25
```

Using a for each loop to go through all items in a list

```
myList = [21,16,12,27,36]
total = 0

for num in myList:
    total += num
print total
```

Output

112

open listloops.py

Start work on List Labs