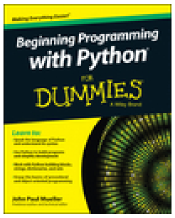
dummies
A Wiley Brand

HOW TO MODIFY LISTS IN PYTHON



RELATED BOOK

Beginning Programming with Python For Dummies

By **John Paul Mueller**

You can modify the content of a list as needed with Python. Modifying a list means to change a particular entry, add a new entry, or remove an existing entry. To perform these tasks, you must sometimes read an entry. The concept of modification is found within the acronym CRUD, which stands for Create, Read, Update, and Delete. Here are the list functions associated with CRUD:

`append()` : Adds a new entry to the end of the list.

`clear()` : Removes all entries from the list.

`copy()` : Creates a copy of the current list and places it in a new list.

`extend()` : Adds items from an existing list and into the current list.

`insert()` : Adds a new entry to the position specified in the list.

`pop()` : Removes an entry from the end of the list.

`remove()` : Removes an entry from the specified position in the list.

The following steps show how to perform modification tasks with lists.

1 **Open a Python Shell window.**

You see the familiar Python prompt.

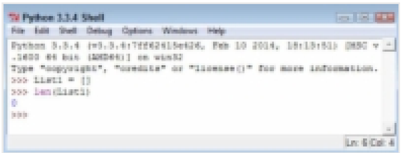
2 Type List1 = [] and press Enter.

Python creates a list named `List1` for you.

Notice that the square brackets are empty. `List1` doesn't contain any entries. You can create empty lists that you fill with information later. In fact, this is precisely how many lists start because you usually don't know what information they will contain until the user interacts with the list.

3 Type len(List1) and press Enter.

The `len()` function outputs 0. When creating an application, you can check for an empty list using the `len()` function. If a list is empty, you can't perform tasks such as removing elements from it because there is nothing to remove.



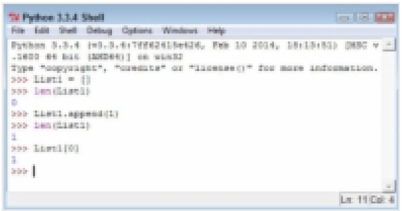
```
Python 3.3.4 (v3.3.4:7ff62415e426, Feb 10 2014, 18:13:51) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> List1 = []
>>> len(List1)
0
>>>
```

4 Type List1.append(1) and press Enter.

Check for empty lists as needed in your application.

5 Type len(List1) and press Enter.

The `len()` function now reports a length of 1.



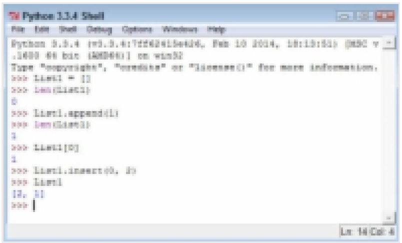
```
Python 3.3.4 (v3.3.4:7ff62415e426, Feb 10 2014, 18:13:51) [MSC v
.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> List1 = []
>>> len(List1)
0
>>> List1.append(1)
>>> len(List1)
1
>>> List1[0]
1
>>> |
```

6 Type List1[0] and press Enter.

You see the value stored in element 0 of List1.

7 Type List1.insert(0, 2) and press Enter.

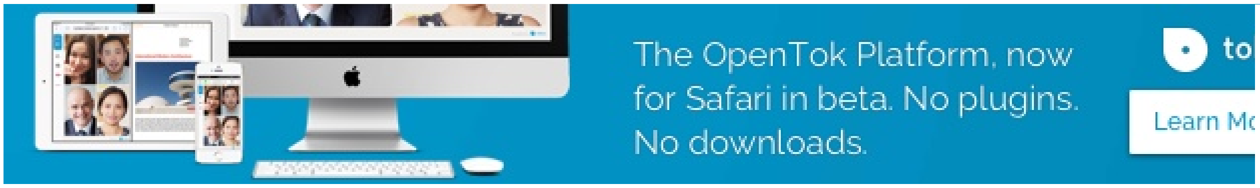
The `insert()` function requires two arguments. The first argument is the index of the insertion, which is element 0 in this case. The second argument is the object you want inserted at that point, which is 2 in this case.



```
Python 3.3.4 (v3.3.4:7ff62415e426, Feb 10 2014, 18:13:51) [MSC v
.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> List1 = []
>>> len(List1)
0
>>> List1.append(1)
>>> len(List1)
1
>>> List1[0]
1
>>> List1.insert(0, 2)
>>> List1
[2, 1]
>>> |
```

8 Type List1 and press Enter.

Python has added another element to List1. However, using the `insert()` function lets you add the new element before the first element.

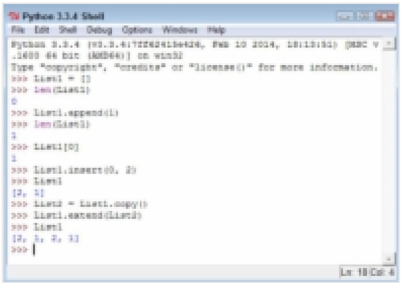


9 Type List2 = List1.copy() and press Enter.

The new list, `List2`, is a precise copy of `List1`. Copying is often used to create a temporary version of an existing list so that a user can make temporary modifications to it rather than to the original list. When the user is done, the application can either delete the temporary list or copy it to the original list.

10 Type List1.extend(List2) and press Enter.

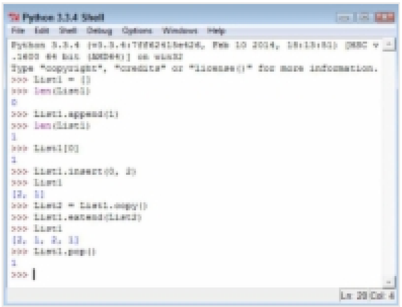
Python copies all the elements in `List2` to the end of `List1`. Extending is commonly used to consolidate two lists.



```
Python 3.3.4 (v3.3.4:7ff62415e426, Feb 10 2014, 18:13:51) [MSC v
.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> List1 = []
>>> len(List1)
0
>>> List1.append(1)
>>> len(List1)
1
>>> List1[0]
1
>>> List1.insert(0, 2)
>>> List1
[2, 1]
>>> List2 = List1.copy()
>>> List1.extend(List2)
>>> List1
[2, 1, 2, 1]
>>> 
```

11 Type List1 and press Enter.

You see that the copy and extend processes have worked. `List1` now contains the values 2, 1, 2, and 1.



```
Python 3.3.4 (v3.3.4:7ff62415e426, Feb 10 2014, 18:13:51) [MSC v
.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> List1 = []
>>> len(List1)
0
>>> List1.append(1)
>>> len(List1)
1
>>> List1[0]
1
>>> List1.insert(0, 2)
>>> List1
[2, 1]
>>> List2 = List1.copy()
>>> List1.extend(List2)
>>> List1
[2, 1, 2, 1]
>>> List1.pop()
1
>>> 
```

12 Type List1.pop() and press Enter.

Python displays a value of 1. The 1 was stored at the end of the list, and `pop()` always removes values from the end.

13 Type List1.remove(1) and press Enter.

This time, Python removes the item at element 1. Unlike the `pop()` function, the `remove()` function doesn't display the value of the item it removed.

14 Type List1.clear() and press Enter.

Using `clear()` means that the list shouldn't contain any elements now.

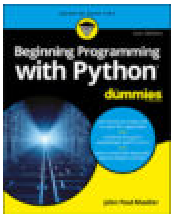
15 Type len(List1) and press Enter.

You see that the output is 0. `List1` is definitely empty. At this point, you've tried all the modification methods that Python provides for lists. Work with `List1` some more using these various functions until you feel comfortable making changes to the list.

16 Close the Python Shell window.

Congratulations on a job well done!

HOW TO GET ADDITIONAL PYTHON LIBRARIES



RELATED BOOK

Beginning Programming with Python For Dummies, 2nd Edition

By **John Paul Mueller**

The **Pythonware site** doesn't look all that interesting until you start clicking the links. It provides you with access to a number of third-party libraries that help you perform additional tasks using Python. Although all the links provide you with useful resources, the "Downloads (`downloads.effbot.org`)" link is the one you should look at first. This download site provides you with access to

- **aggdraw:** A library that helps you create anti-aliased drawings.
- **celementtree:** An add-on to the elementtree library that makes working with XML data more efficient and faster.
- **console:** An interface for Windows that makes it possible to create better console applications.
- **effbot:** A collection of useful add-ons and utilities, including the EffNews RSS news reader.
- **elementsoap:** A library that helps you create Simple Object Access Protocol (SOAP) connections to web services providers.
- **elementtidy:** An add-on to the elementtree library that helps you create nicer-looking and more functional XML tree displays than the standard ones in Python.
- **elementtree:** A library that helps you interact with XML data more efficiently than standard Python allows.
- **exemaker:** A utility that creates an executable program from your Python script so that you can execute the script just as you would any other application on your machine.
- **ftpparse:** A library for working with FTP sites.
- **grabscreen:** A library for performing screen captures.

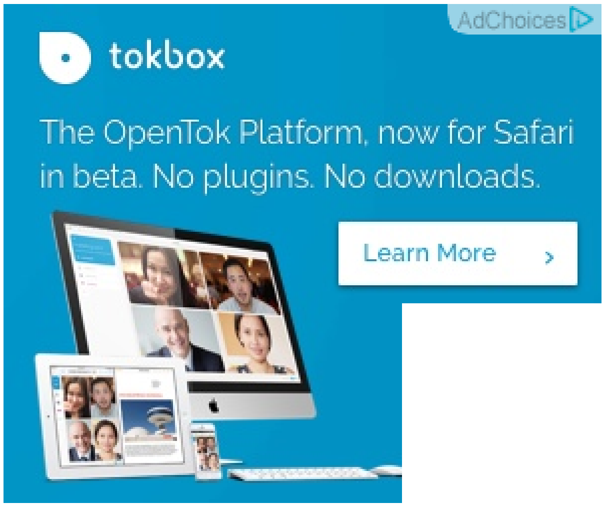
- **imaging:** Provides the source distribution to the Python Imaging Library (PIL) that lets you add image-processing capabilities to the Python interpreter. Having the source lets you customize PIL to meet specific needs.

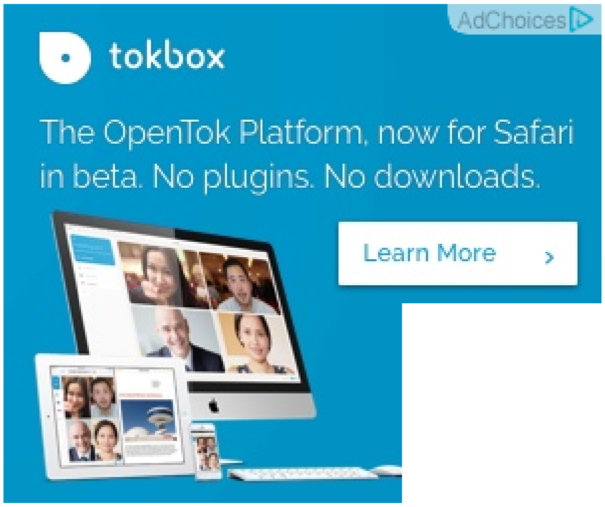
- **pil:** Binary installers for PIL, which make obtaining a good installation for your system easier. (There are other PIL-based libraries as well, such as pilfont — a library for adding enhanced font functionality to a PIL-based application.)

- **pythondoc:** A utility for creating documentation from the comments in your Python code that works much like JavaDoc.

- **squeeze:** A utility for converting your Python application contained in multiple files into a one- or two-file distribution that will execute as normal with the Python interpreter.

- **tkinter3000:** A widget-building library for Python that includes a number of subproducts. *Widgets* are essentially bits of code that create controls, such as buttons, to use in GUI applications. There are a number of add-ons for the tkinter3000 library, such as wckgraph, which helps you add graphing support to an application.


AdChoices
tokbox
The OpenTok Platform, now for Safari in beta. No plugins. No downloads.
Learn More ›


AdChoices
tokbox
The OpenTok Platform, now for Safari in beta. No plugins. No downloads.
Learn More ›