Introducing Lists

What Is a List?

- A *list* is a collection of items in a particular order.
- You can put anything you want into a list, and the items in your list don’t have to be related in any particular way.
- Because a list usually contains more than one element, it’s a good idea to make the name of your list plural, such as `letters`, `digits`, or `names`.

- In Python, square brackets ([ ]) indicate a list, and individual elements in the list are separated by commas.

```
bicycles = ['trek', 'cannondale', 'redline', 'specialized']

print(bicycles)
~
~
~
~
```

```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_03$ python
bicycles.py
['trek', 'cannondale', 'redline', 'specialized']
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_03$
```

Accessing Elements in a List

```
bicycles = ['trek', 'cannondale', 'redline', 'specialized']
print(bicycles[0].title())
~
~
~
```

```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_03$ python
bicycles_2.py
Trek
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_03$
```

Index Positions Start at 0, Not 1

```
>>> bicycles = ['trek', 'cannondale', 'redline', 'specialized']
>>> print(bicycles[1])
cannondale
>>> print(bicycles[3])
specialized
>>>
```

- Python has a special syntax for accessing the last element in a list.

```
>>> bicycles = ['trek', 'cannondale', 'redline', 'specialized']
>>> print(bicycles[-1])
specialized
>>> print(bicycles[-3])
cannondale
>>>
```

Using Individual Values from a List

- You can use individual values from a list just as you would any other variable.
- For example, you can use f-strings to create a message based on a value from a list.

```
bicycles = ['trek', 'cannondale', 'redline', 'specialized']
message = f"My first bicycle was a {bicycles[0].title()}."

print(message)
~
~
~
~
```

```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_03$ python
bicycles_1.py
My first bicycle was a Trek.
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_03$
```

Changing, Adding, and Removing Elements

Modifying Elements in a List

```
>>> motorcycles = ['honda', 'yamaha', 'suzuki']
>>> print(motorcycles)
['honda', 'yamaha', 'suzuki']
>>> motorcycles[0] = 'ducati'
>>> print(motorcycles)
['ducati', 'yamaha', 'suzuki']
>>>
```

Adding Elements to a List

- Appending Elements to the End of a List

```
>>> motorcycles = ['honda', 'yamaha', 'suzuki']
>>> print(motorcycles)
['honda', 'yamaha', 'suzuki']
>>> motorcycles.append('ducati')
>>> print(motorcycles)
['honda', 'yamaha', 'suzuki', 'ducati']
>>>
```

```
>>> motorcycles = []
>>> motorcycles.append('honda')
>>> motorcycles.append('yamaha')
>>> motorcycles.append('suzuki')
>>> print(motorcycles)
['honda', 'yamaha', 'suzuki']
>>>
```

- Inserting Elements into a List

```
>>> motorcycles = ['honda', 'yamaha', 'suzuki']
>>> motorcycles.insert(0, 'ducati')
>>> print(motorcycles)
['ducati', 'honda', 'yamaha', 'suzuki']
>>>
```

Removing Elements from a List

- Removing an Item Using the `del` Statement

```
>>> motorcycles = ['honda', 'yamaha', 'suzuki']
>>> print(motorcycles)
['honda', 'yamaha', 'suzuki']
>>> del motorcycles[0]
>>> print(motorcycles)
['yamaha', 'suzuki']
>>>
```

```
>>> motorcycles = ['honda', 'yamaha', 'suzuki']
>>> print(motorcycles)
['honda', 'yamaha', 'suzuki']
>>> del motorcycles[1]
>>> print(motorcycles)
['honda', 'suzuki']
>>>
```

- Removing an Item Using the `pop()` Method
  - The `pop()` method removes the last item in a list, but it lets you work with that item after removing it.

```
>>> motorcycles = ['honda', 'yamaha', 'suzuki']
>>> print(motorcycles)
['honda', 'yamaha', 'suzuki']
>>> popped_motorcycle = motorcycles.pop()
>>> print(motorcycles)
['honda', 'yamaha']
>>> print(popped_motorcycle)
suzuki
>>>
```

- Popping Items from any Position in a List
  - You can use `pop()` to remove an item from any position in a list by including the index of the item you want to remove in parentheses.

```
>>> motorcycles = ['honda', 'yamaha', 'suzuki']
>>> first_owned = motorcycles.pop(0)
>>> print(f"The first motorcycle I owned was a {first_owned.title()}.")
The first motorcycle I owned was a Honda.
>>>
```

- If you’re unsure whether to use the `del` statement or the `pop()` method, here’s a simple way to decide: when you want to delete an item from a list and not use that item in any way, use the `del` statement; if you want to use an item as you remove it, use the `pop()` method.

- Removing an Item by Value
  - Sometimes you won't know the position of the value you want to remove from a list.
  - If you only know the value of the item you want to remove, you can use the `remove()` method.

```
>>> motorcycles = ['honda', 'yamaha', 'suzuki', 'ducati']
>>> print(motorcycles)
['honda', 'yamaha', 'suzuki', 'ducati']
>>> motorcycles.remove('ducati')
>>> print(motorcycles)
['honda', 'yamaha', 'suzuki']
>>>
```

```
motorcycles = ['honda', 'yamaha', 'suzuki', 'ducati']
print(motorcycles)

too_expensive = 'ducati'
motorcycles.remove(too_expensive)
print(motorcycles)
print(f"\nA {too_expensive.title()} is too expensive for me.")
~
~
~
```

```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_03$ python
motorcycles.py
['honda', 'yamaha', 'suzuki', 'ducati']
['honda', 'yamaha', 'suzuki']

A Ducati is too expensive for me.
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_03$
```

- The `remove()` method deletes only the first occurrence of the value you specify.
- If there's a possibility the value appears more than once in the list, you'll need to use a loop to make sure all occurrences of the value are removed.

Organizing a List

Sorting a List Permanently with the `sort()` Method

```
>>> cars = ['bmw', 'audi', 'toyota', 'subaru']
>>> cars.sort()
>>> print(cars)
['audi', 'bmw', 'subaru', 'toyota']
>>>
```

```
>>> cars = ['bmw', 'audi', 'toyota', 'subaru']
>>> cars.sort(reverse=True)
>>> print(cars)
['toyota', 'subaru', 'bmw', 'audi']
>>>
```

Sorting a List Temporarily with the `sorted()` Function

```
cars = ['bmw', 'audi', 'toyota', 'subaru']

print("Here is the original list:")
print(cars)

print("\nHere is the sorted list:")
print(sorted(cars))

print("\nHere is the original list again:")
print(cars)
~
~
~
~
```

```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_03$ python
cars_1.py
Here is the original list:
['bmw', 'audi', 'toyota', 'subaru']

Here is the sorted list:
['audi', 'bmw', 'subaru', 'toyota']

Here is the original list again:
['bmw', 'audi', 'toyota', 'subaru']
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_03$
```

- The `sorted()` function can also accept a `reverse=True` argument if you want to display a list in reverse alphabetical order.

Printing a List in Reverse Order

```
>>> cars = ['bmw', 'audi', 'toyota', 'subaru']
>>> print(cars)
['bmw', 'audi', 'toyota', 'subaru']
>>> cars.reverse()
>>> print(cars)
['subaru', 'toyota', 'audi', 'bmw']
>>>
```

Finding the Length of a List

```
>>> cars = ['bmw', 'audi', 'toyota', 'subaru']
>>> len(cars)
4
>>>
```

Avoiding Index Errors When Working with Lists

```
>>> motorcycles = ['honda', 'yamaha', 'suzuki']
>>> print(motorcycles[3])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
>>>
```

- Keep in mind that whenever you want to access the last item in a list you use the index `-1`.

```
>>> motorcycles = ['honda', 'yamaha', 'suzuki']
>>> print(motorcycles[-1])
suzuki
>>>
```

- The only time this approach will cause an error is when you request the last item from an empty list:

```
>>> print(motorcycles[-1])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
>>>
```