Python basics: Lists

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Enclosed in square brackets

```
my_list = [1, 2, 3, 4, 5]
my_list
```

```
[1, 2, 3, 4, 5]
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Different data types are allowed

```
my_list = [1, "two", 3, 4, True]
my_list
```

```
[1, 'two', 3, 4, True]
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

A list can contain other lists

```
my_list = [1, "two", [3, 4, True]]
my_list
```

```
[1, 'two', [3, 4, True]]
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

And more complex data types

```
df = pd.DataFrame({"A": [1, 2, 3],
                   "B": [4, 5, 6]})

my_list = [1, "two", [3, df.columns, True]]
my_list
```

```
[1, 'two', [3, Index(['A', 'B'], dtype='object'), True]]
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

You can index a list

```
my_list = [1, 2, [3, 4, 5]]
print(f'Item at index 0 is {my_list[0]}')
print(f'Item at index 2 is {my_list[2]}')
print(f'Item at index 0 of the list at index 2 is {my_list[2][0]}’)

Item at index 0 is 1
Item at index 2 is [3, 4, 5]
Item at index 0 of the list at index 2 is 3
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

And slice with list[start:end:step]

```
my_list = [1, 2, [3, 4, 5], 6, 7]
print(f'From index 1 to the end: {my_list[3:]}')
```

```
From index 1 to the end: [6, 7]
```

```
my_list = [1, 2, [3, 4, 5], 6, 7]
print(f'Every other item: {my_list[::2]}’)
```

```
Every other item: [1, [3, 4, 5], 7]
```

```
my_list = [1, 2, [3, 4, 5], 6, 7]
print(f'Reversed: {my_list[::-1]}’)
```

```
Reversed: [7, 6, [3, 4, 5], 2, 1]
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

We can remove items by index using list.pop()

```
my_list = [1, 2, [3, 4, 5]]
popped_item = my_list.pop(1)
print(f'Popped item is {popped_item}')
print(f'List after popping is {my_list}')
```

```
Popped item is 2
List after popping is [1, [3, 4, 5]]
```

If we don’t pass an index to .pop() it will remove and return the last item in the list

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Insert items using list.insert

```
my_list = [1, 2, [3, 4, 5]]
my_list.insert(2, "hi")
my_list
```

```
[1, 2, 'hi', [3, 4, 5]]
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Extend a list with another list using list.extend

```
my_list = [1, 2, [3, 4, 5]]
my_list.extend(["hi", "there"])
my_list
```

```
[1, 2, [3, 4, 5], 'hi', 'there']
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Or using the + operator

```
my_list = [1, 2, [3, 4, 5]]
my_list = my_list + ["hi", "there"]
my_list
```

```
[1, 2, [3, 4, 5], 'hi', 'there']
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

And much more!

List Method	Explanation
.append(x)	Appends element x to the end of the list.
.extend(iterable)	Appends elements from the iterable to the list.
.insert(i, x)	Inserts element x at position i in the list.
.remove(x)	Removes the first occurrence of element x from the list.
.pop([i])	Removes and returns the item at index i. If i is not specified, removes and returns the last item.
.clear()	Removes all items from the list.
.index(x)	Returns the index of the first occurrence of element x.
.count(x)	Returns the number of times element x appears in the list.
.sort(key=None, reverse=False)	Sorts the list in ascending order (or descending order if reverse=True).
.reverse()	Reverses the order of elements in the list.
.copy()	Returns a shallow copy of the list.
len(list)	Returns the number of items in the list.

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata