Lists

Lists

Lists are used to store a collection of objects but are more flexible than tuples. You can create lists using the `list` function with another iterable object or square brackets `[]`:

[2]:

```
list1 = list((1, 2, 3))
print('list1', list1)

list2 = [4, 8, 9]
print('list2', list2)
```

```
list1 [1, 2, 3]
list2 [4, 8, 9]
```

You can access elements of the list by indexing and slicing it:

[6]:

```
letters = ['a', 'b', 'c', 'd', 'e']
print('Letters:', letters)
print('First character:', letters[0])
print('Second character:', letters[1])
print('Last character:', letters[-1])
print('Every second character:', letters[::2])
```

```
Letters: ['a', 'b', 'c', 'd', 'e']
First character: a
Second character: b
Last character: e
Every second character: ['a', 'c', 'e']
```

Unlike tuples you can alter the elements of a list after instancing it:

[5]:

```
letters = ['a', 'b', 'c', 'd', 'e']
print(letters)

print('Changing the third character')

letters[2] = 'z'
print(letters)
```

```
['a', 'b', 'c', 'd', 'e']
Changing the third character
['a', 'b', 'z', 'd', 'e']
```

You can also assign new values to slices:

[7]:

```python
letters = ['a', 'b', 'c', 'd', 'e']
print(letters)

print('Changing the first three characters')
letters[:3] = ['x', 'y', 'z']
print(letters)
```

```
['a', 'b', 'c', 'd', 'e']
Changing the first three characters
['x', 'y', 'z', 'd', 'e']
```

Concatenating Lists

The `+` operator acts on lists in a similar way to strings, concatenating the two lists:

[8]:

```python
list1 = [1, 2, 3]
list2 = ['a', 'b', 'c']

print(list1 + list2)
```

```
[1, 2, 3, 'a', 'b', 'c']
```

`list.append()`

You can add elements to the end of the list using the `.append()` method:

[15]:

```python
letters = ['a', 'b', 'c', 'd', 'e']
print(letters)

print('Appending an additional letter')

letters.append('f')
print(letters)
```

```
['a', 'b', 'c', 'd', 'e']
Appending an additional letter
['a', 'b', 'c', 'd', 'e', 'f']
```

`list.insert()`

If you want to insert an element into a specific place in the list you can use the `.insert()` method. This takes the index and the object you want to add as the arguments:

[16]:

```python
numbers = [1, 2, 4, 5, 6]
print(numbers)

print('Inserting number 3 at index 2')

numbers.insert(2, 3)
print(numbers)
```

```
[1, 2, 4, 5, 6]
Inserting number 3 at index 2
[1, 2, 3, 4, 5, 6]
```

`lists.remove()`

If you want to remove the first instance of an element of a list with a specific value you can use the `.remove()` method:

[23]:

```python
numbers = [1, 2, 1, 3, 4]
print(numbers)

print('Removing first 1 from numbers')

numbers.remove(1)
print(numbers)
```

```
[1, 2, 1, 3, 4]
Removing first 1 from numbers
[2, 1, 3, 4]
```

`list.pop()`

If you want to retrieve and remove an element at a particular index you can use the `.remove()` method, which takes the index of the element you want to retrieve as the argument:

[22]:

```python
numbers = [1, 2, 3, 4, 5]
print(numbers)

print('Retrieving number at index 2:', numbers.pop(2))

print(numbers)
```

```
[1, 2, 3, 4, 5]
Retrieving number at index 2: 3
[1, 2, 4, 5]
```