Lists and Tuples Operations Cheat Sheet

Lists and Tuples Operations Cheat Sheet

Lists and tuples are both sequences, so they share a number of sequence operations. But, because lists are mutable, there are also a number of methods specific just to lists. This cheat sheet gives you a run down of the common operations first, and the list-specific operations second.

Common sequence operations

* len(sequence) Returns the length of the sequence
* for element in sequence Iterates over each element in the sequence
* if element in sequence Checks whether the element is part of the sequence
* sequence[i] Accesses the element at index i of the sequence, starting at zero
* sequence[i:j] Accesses a slice starting at index i, ending at index j-1. If i is omitted, it's 0 by default. If j is omitted, it's len(sequence) by default.
* for index, element in enumerate(sequence) Iterates over both the indexes and the elements in the sequence at the same time

Check out the [official documentation for sequence operations](https://docs.python.org/3/library/stdtypes.html#sequence-types-list-tuple-range).

List-specific operations and methods

* list[i] = x Replaces the element at index i with x
* list.append(x) Inserts x at the end of the list
* list.insert(i, x) Inserts x at index i
* list.pop(i) Returns the element a index i, also removing it from the list. If i is omitted, the last element is returned and removed.
* list.remove(x) Removes the first occurrence of x in the list
* list.sort() Sorts the items in the list
* list.reverse() Reverses the order of items of the list
* list.clear() Removes all the items of the list
* list.copy() Creates a copy of the list
* list.extend(other\_list) Appends all the elements of other\_list at the end of list

Most of these methods come from the fact that lists are mutable sequences. For more info, see the [official documentation for mutable sequences](https://docs.python.org/3/library/stdtypes.html#mutable-sequence-types) and the [list specific documentation](https://docs.python.org/3/library/stdtypes.html#lists).

List comprehension

* [expression for variable in sequence] Creates a new list based on the given sequence. Each element is the result of the given expression.
* [expression for variable in sequence if condition] Creates a new list based on the given sequence. Each element is the result of the given expression; elements only get added if the condition is true.