

Lists and Tuples

What are lists and tuples?

Ordered sequence of values indexed by integer numbers. Tuples are immutable.

How to initialize an empty list/tuple?

Lists: `myList = []`
Tuples: `myTuple = ()`

Size of list/tuple?

`len(my List or Tuple)`

Get element in position x of list/tuple?

`myList OrT uple[x]` -- if not found, throws `IndexError`

Is element "x" in list/tuple?

`" x" in myList OrTuple`

Index of element "X" of list/tuple?

`myList OrT upl e.i nde x("x ")`
 -- If not found, throws a `ValueError` exception

Number of occurrences of "x" in list/tuple?

`myList OrT upl e.c ount("x ")`

Update an item of a list/tuple?

Lists: `myList[x] = " x"`
Tuples: tuples are immutable!

Remove element in position x of list/tuple?

Lists: `del myList[x]`
Tuples: tuples are immutable!

Remove element "x" of a list/tuple?

Lists: `myList.re mov e("x ")`.
 Removes the first occurrence
Tuples: tuples are immutable!

Lists and Tuples (cont)

Concatenate two lists or two tuples?

Lists: `myList1 + myList2`
Tuples: `myTuple1 + myTuple2`

Concatenating a List and a Tuple will produce a `TypeError` exception

Insert element in position x of a list/tuple?

Lists: `myList.in sert(x, " val u e")`
Tuples: tuples are immutable!

Append "x" to a list/tuple?

Lists: `myList.ap pen d("x ")`
Tuples: tuples are immutable!

Convert a list/tuple to tuple/list

List to Tuple: `tuple(myList)`
Tuple to List: `list(m yTuple)`

Slicing list/tuple

`myList OrT upl e[i nd1 :in d2: step]` -- step is optional and may be negative

Sets

What is a set?

Unordered collection with **no duplicate** elements. Sets support mathematical operations like union, intersection, difference and symmetric difference.

Initialize an empty set

`mySet = set()`

Initialize a not empty set

`mySet = set(el ement1, elemen t2.. .)` -- Note: strings are split into their chars (duplicates are deleted). To add strings, initialize with a Tuple/List

Sets (cont)

Add element "x" to the set

`mySet.a dd ("x")`

Remove element "x" from a set

Method 1: `mySet.r em ove ("x")` -- If "x" not present, raises a `KeyError`
Method 2: `mySet.d is car d("x ")` -- Removes the element, if present

Remove every element from the set

`mySet.c lear()`

Check if "x" is in the set

`" x" in mySet`

Union of two sets

Method 1: `mySet1.un ion (my Set2)`
Method 2: `mySet1 | mySet2`

Intersection of two sets

Method 1: `mySet1.in ter sec t(m ySe)`
Method 2: `mySet1 & mySet2`

Difference of two sets

Method 1: `mySet1.di ffe ren ce(myS 2)`
Method 2: `mySet1 - mySet2`

Symmetric difference of two sets

Method 1: `mySet1.sy mme tri c_d iff re nce (my Set2)`
Method 2: `mySet1 ^ mySet2`

Size of the set

`len(mySet)`



By desmovalvo

cheatography.com/desmovalvo/

Published 2nd March, 2018.

Last updated 2nd March, 2018.

Page 1 of 2.

Sponsored by [ApolloPad.com](https://apollopod.com)

Everyone has a novel in them. Finish

Yours!

<https://apollopod.com>

Dictionaries

What is a dictionary?

Unordered set of key:value pairs .
Members are indexed by keys
(immutable objects)

Initialize an empty Dict

```
myDict = {}
```

Add an element with key "k" to the Dict

```
myDict ["k"] = value
```

Update the element with key "k"

```
myDict ["k"] = newValue
```

Get element with key "k"

```
myDict ["k"] -- If the key is not  
present, a KeyError is raised
```

Check if the dictionary has key "k"

```
" k" in myDict
```

Get the list of keys

```
myDict.keys()
```

Get the size of the dictionary

```
len(my Dict)
```

Delete element with key "k" from the dictionary

```
del myDict ["k"]
```

Delete all the elements in the dictionary

```
myDict.clear()
```



By **desmovalvo**

cheatography.com/desmovalvo/

Published 2nd March, 2018.

Last updated 2nd March, 2018.

Page 2 of 2.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish
Yours!

<https://apollopad.com>