Cheatography

Python3 data structures Cheat Sheet

by desmovalvo via cheatography.com/56139/cs/14893/

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(myListOrTuple)`

Get element in position x of list/tuple?

`myListOrTuple[x]` -- if not found, throws `IndexError`

Is element "x" in list/tuple?

`"x" in myListOrTuple`

Index of element "X" of list/tuple?

`myListOrTuple.index("x")` -- If not found, throws a `ValueError` exception

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

`myListOrTuple.count("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.remove("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.insert(x, "value")`
Tuples: tuples are immutable!

Append "x" to a list/tuple?

Lists: `myList.append("x")`
Tuples: tuples are immutable!

Convert a list/tuple to tuple/list

List to Tuple: `tuple(myList)`
Tuple to List: `list(myTuple)`

Slicing list/tuple

`myListOrTuple[ind1:ind2: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 simmetric difference.

Initialize an empty set

`mySet = set()`

Initialize a not empty set

`mySet = set(element1, element2...)` -- 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.add("x")`

Remove element "x" from a set

Method 1: `mySet.remove("x")` -- If "x not present, raises a `KeyErorr`
Method 2: `mySet.discard("x")` -- Removes the element, if present

Remove every element from the set

`mySet.clear()`

Check if "x" is in the set

`"x" in mySet`

Union of two sets

Method 1: `mySet1.union(mySet2)`
Method 2: `mySet1 | mySet2`

Intersection of two sets

Method 1: `mySet1.intersect(mySe )`
Method 2: `mySet1 & mySet2`

Difference of two sets

Method 1: `mySet1.difference( myS 2)`
Method 2: `mySet1 - mySet2`

Simmetric difference of two sets

Method 1: `mySet1.symmetric_difference(mySet2)`
Method 2: `mySet1 ^ mySet2`

Size of the set

`len(mySet)`

By **desmovalvo**

cheatography.com/desmovalvo/

Published 2nd March, 2018.
Last updated 2nd March, 2018.

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.cl ear()
```