

Cheat sheet: Container data types (#1)

Tuple (immutable, ordered)

`()`, `(1,)`, `(1, 3.0, "Hello")`

List (mutable, ordered)

`[]`, `[1,]`, `[1, 3.0, "Hello"]`

Tuple/List operations

`tpl[pos]`, `tpl[from:to:incr]`
`item in l` `l` contains item?
`l1 + l2` concatenate
`l1 * n` repeat
`len(l)` length

List manipulation

`l[pos] = val` Overwrite item
`l[i:j:k] = l2` Overwrite range
`l.append()` Append item
`l.extend()` Extend with list
`l += [...]` Extend with list
`l.sort()` Sort
`del l[ind]` delete ind or range

Dictionary (mutable, (un)ordered)

`{key1: value1, key2: value2}`
keys: unique, type immutable
Python \geq 3.7: dict ordered

Dictionary operations

`dct[key]` get value
`dct.get(key, default=None)`
 get with default value
`dct[key] = val` store
`del dct[key]` delete
`key in dct` check

Set (mutable, unordered)

`{item1, item2, item3}`

Set operations

`st.add()` add
`st.remove()` remove
`item in st` check

Cheat sheet: Container data types (#2)

Iterating over containers

```
for item in tpl:    Tuple items
for item in lst:    List items
for key in dct:    Dict keys
for value in dct.values():
    Dict values
for key, val in dct.items():
    Dict key/value pairs
for item in st:    Set items
    Order undetermined
```

Enumeration iterator

```
for ind, item in enumerate(lst):
```

List comprehension

```
lst = [expr
        for var in iterable
        if condition]
```

Dictionary comprehension

```
dct = {keyexpr: valuexpr
        for var in iterable
        if condition}
```

Comparing collections

```
==, !=
>, >=, <, <=    (if ordered)
```

In place operators

```
+=, -=, *=, /=, // =
```

Some string methods

```
str.split(sep)    split into list
sep.join(lst)     join from list
str.lower()        to lower case
str.upper()        to upper case
str.[l,r]strip()  strip limiting
                    [left, right]
                    whitespaces
```