Cheatography

Python Basics - Lists, Tuples and Dictionaries Cheat Sheet

by Mario (mariofreitas) via cheatography.com/42813/cs/13210/

Lists and Tuples Syntax

`L = [1, 2, 3, 4, 5]`	Lists are created with **[ ]**
`T = (10, 20, 30, 40, 50)`	Tuples are created with **( )**
`L[0]`	Returns 1st element of *L* (1)
`T[0]`	Returns 1st element of *T* (10)
`L[1:4]`	Returns 2nd to 4th element of *L* ([2, 3, 4])
`T[1:4]`	Returns 2nd to 4th element of *T* ((20, 30, 40))
`L[0:-1:2]`	Returns 1st to 2nd last element of *L* skipping one at a time ([1, 3])
`T[0:-1:2]`	Returns 1st to 2nd last element of *L* skipping one at a time ((10, 30))
`L[1] = 22`	Assigns *22* to 2nd element of *L* (L == [1, 22, 3, 4, 5])
`T[1] = 22`	ERROR: You can't assing anything to tuples
`L[0:2] = [11, 22]`	Assigns *11* and *22* to 1st and 2nd element of *L* respectively (L == [11, 22, 3, 4, 5])

Lists are mutable and Tuples are NOT mutable

Lists - Methods

```
a = ['a', 'b', 'c']
```

```
b = [1, 3, 2]
```

`a + b`	Returns *a* concatenated with *b* (['a', 'b', 'c', 1, 3, 2])
`'c' in a`	Returns *True* if *'c'* is in the list *a* and *False* otherwise (True)
`len(a)`	Returns the number of elements in *a* (3)
`a.append('d')`	Appends 'd' to the end of the list *a* (a == ['a', 'b', 'c', 'd'])
`a.extend(['d', 'e', 'f'])`	Appends every element of the iterable to the end of *a* (a == ['a', 'b', 'c', 'd', 'e', 'f'])
`a.insert(1, 'd')`	Inserts *'d'* to index *1* of *a* (a == ['a', 'd', 'b', 'c'])
`a.pop()`	Returns the last element of the list and deletes it from the list. ('c')

Lists - Methods (cont)

`a.pop(1)`	Returns 2nd element of *a* and removes it from the list ('b')
`a.remove('b')`	Removes first occurrence of *'b'* in *a* (a == ['a', 'c'])
`a.clear()`	Clears the list entirely (a == [])
`a.index('b')`	Returns the index of the first occurrence of *'b'* (1)
`a.count('b')`	Returns the number of occurrences of *'b'* in *a* (1)
`b.sort()`	Returns a sorted version of *b* ([1, 2, 3])
`a.reverse()`	Reverses the list *a* (['c', 'b', 'a'])
`a.copy()`	Returns a copy of *a*

The copy() method returns a list identical to the original, but with a different ID. It means that they are allocated in different places of memory.

Tuple - Methods

```
t1 = ('a', 'b', 'c')
```

```
t2 = (1, 2, 3)
```

`t1 + t2`	Returns a concatenated version of *t1* and *t2*
`2 in t2`	Returns *True* if *2* is in *t2* and *False* otherwise (True)
`len(t1)`	Returns the number of elements in *t1* (3)
`t2.count(2)`	Returns the number of occurrences of *2* in *t2* (1)
`t2.index(1)`	Returns the index of the 1st occurrence of *1* (0)

Lists - Loops 1

```
a = ['one' , 'two', 'three']
for i in a:
    print(i)
```

```
one
two
three
```

Lists - Loops 2

```
a = ['one' , 'two', 'three']
for i in range(len(a)):
    print(f"a[{i}] == {a[i]}")
```

```
a[0] == one
a[1] == two
a[2] == three
```