Python Tuples Cheat Sheet

Python Tuples

A tuple is an immutable ordered sequence of objects of any type.

Introduction to tuples

```
# Creating tuples
t0 = ()                         # empty tuple
t1 = tuple()                    # empty tuple
t = (1.2)                       # this isn't a tuple, it's a float!
type(t)                         # => float
t2 = (1.2,)                     # creating a tuple with a single element (comma is mandatory)
t3 = tuple('abc')               # creating a tuple from an iterable (string)
t4 = tuple([1, 3.2, 'abc'])     # creating a tuple from an iterable (list)
t5 = (1, 3.2, 'abc')

# Tuples are indexed like strings and lists
t5[0]       # => 1
t5[2]       # => 'abc'
t5[-1]      # => 'abc'
#t5[10]    # => IndexError: tuple index out of range

# Tuples are  immutable objects. Can't be changed.
# t5[0] = 4    # => TypeError: 'tuple' object does not support item assignment

# Tuples are sliced like strings and lists.
# The start is included and the stop is excluded
print(t5)    # => (1, 3.2, 'abc')
t5[0:2]      # => (1, 3.2)
t5[:2]       # => (1, 3.2)
t5[::]       # => (1, 3.2, 'abc')
t5[::2]        # => (1, 'abc') -> in steps of 2
```

```
t5[-1:0:-1]    # => ('abc', 3.2)
```

Iterating over a tuple

```
movies = ('The Wizard of Oz', 'The Legend', 'Casablanca')
for movie in movies:
    print(f'We are watching {movie}')
```

Tuple membership

in and not in operators test tuple membership

```
'The Legend' in movies      # => True
'The Legend' not in movies  # => False
```

Tuple Methods

```
dir(tuple)    # returns a list will all tuple methods

my_tuple = (1, 2.2, 'abc', 1)
len(my_tuple)    # => 4

# tuple.index() returns the index of an item
my_tuple.index(1)      # => 0 -> the index of the first element with value 1
my_tuple.index(10)    # => ValueError: tuple.index(x): x not in tuple

# tuple.count() returns the no. of occurrences of the item in tuple
my_tuple.count(1)    # => 2

# Sorting tuples
# tuple.sort() and sorted(tuple)
nums = (6, -1, 55, 2.3)
sorted(nums)    # => (-1, 2.3, 6, 55) -> returns a new sorted list
max(nums)       # => 55
min(nums)       # => -1
sum(nums)       # => 62.3
```