Collections

Counter

```
import collections

>>> cnt = collections.Counter()
>>> for word in ['red', 'blue', 'red',
'green', 'blue', 'blue']:
...     cnt[word] += 1
>>> cnt
Counter({'blue': 3, 'red': 2, 'green': 1})
```

Counter

```
>>> c = Counter()
# a new, empty counter
>>> c = Counter('gallahad')
# a new counter from an iterable
>>> c = Counter({'red': 4, 'blue': 2})
# a new counter from a mapping
>>> c = Counter(cats=4, dogs=8)
# a new counter from keyword args


>>> c = Counter(['eggs', 'ham'])
>>> c['bacon']
# count of a missing element is zero
0
```

Counter

```
elements()
>>> c = Counter(a=4, b=2, c=0, d=-2)
>>> list(c.elements())
['a', 'a', 'a', 'a', 'b', 'b']
most_common([n])
>>> Counter('abracadabra').most_common(3)
[('a', 5), ('r', 2), ('b', 2)]
subtract([iterable-or-mapping])
>>> c = Counter(a=4, b=2, c=0, d=-2)
>>> d = Counter(a=1, b=2, c=3, d=4)
>>> c.subtract(d)
>>> c
Counter({'a': 3, 'b': 0, 'c': -3, 'd': -6})
```

OrderedDict

collections.OrderedDict([items])
OrderedDict.popitem(last=True)

```
>>> # regular unsorted dictionary
>>> d = {'banana': 3, 'apple': 4, 'pear': 1,
'orange': 2}

>>> # dictionary sorted by key
>>> OrderedDict(sorted(d.items(), key=lambda
t: t[0]))
OrderedDict([('apple', 4), ('banana', 3),
('orange', 2), ('pear', 1)])
```

OrderedDict

```
>>> # dictionary sorted by value
>>> OrderedDict(sorted(d.items(), key=lambda
t: t[1]))
OrderedDict([('pear', 1), ('orange', 2),
('banana', 3), ('apple', 4)])

>>> # dictionary sorted by length of the key
string
>>> OrderedDict(sorted(d.items(), key=lambda
t: len(t[0])))
OrderedDict([('pear', 1), ('apple', 4),
('orange', 2), ('banana', 3)])
```