# OrderedDict

To control the order of items in a dictionary, you can use an **OrderedDict** from the **collections** module. It exactly preserves the original insertion order of data when iterating. For example:

from collections import OrderedDict

d = OrderedDict()

d['foo'] = 1

d['bar'] = 2

d['spam'] = 3

d['grok'] = 4

# Outputs "foo 1", "bar 2", "spam 3", "grok 4"

for key in d:

print(key, d[key])

An **OrderedDict** can be particularly useful when you want to build a mapping that you may want to later serialize or encode into a different format. For example, if you want to precisely control the order of fields appearing in a JSON encoding, first building the data in an **OrderedDict** will do the trick:

>>> import json

>>> json.dumps(d)

'{"foo": 1, "bar": 2, "spam": 3, "grok": 4}'

>>>

Be aware that the size of an OrderedDict is more than twice as large as a normal dictionary due to the extra linked list that’s created.