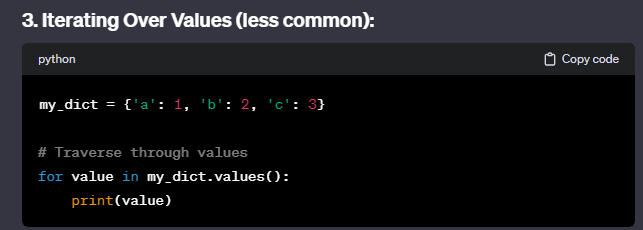
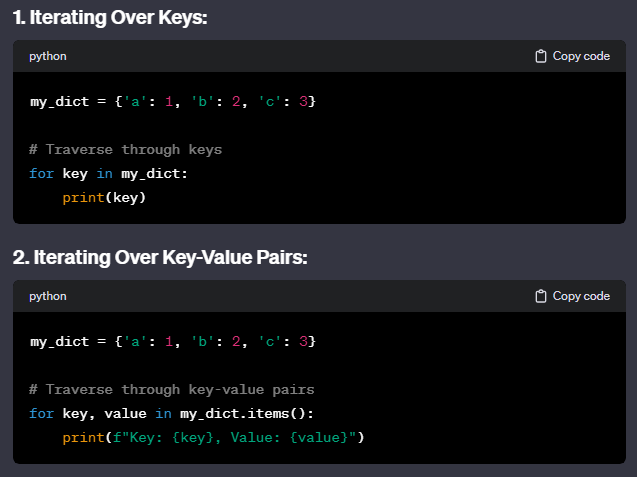
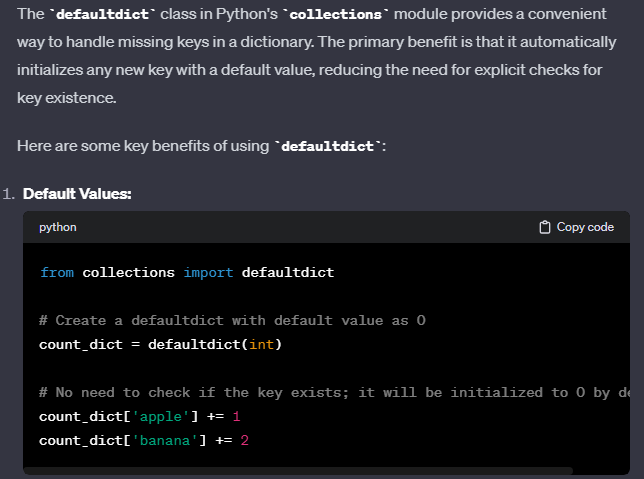
1. [Dictionary](#Dictionary)
2. [List](#List)
3. [Sorting](#Sorting)

Dictionary

Traversal



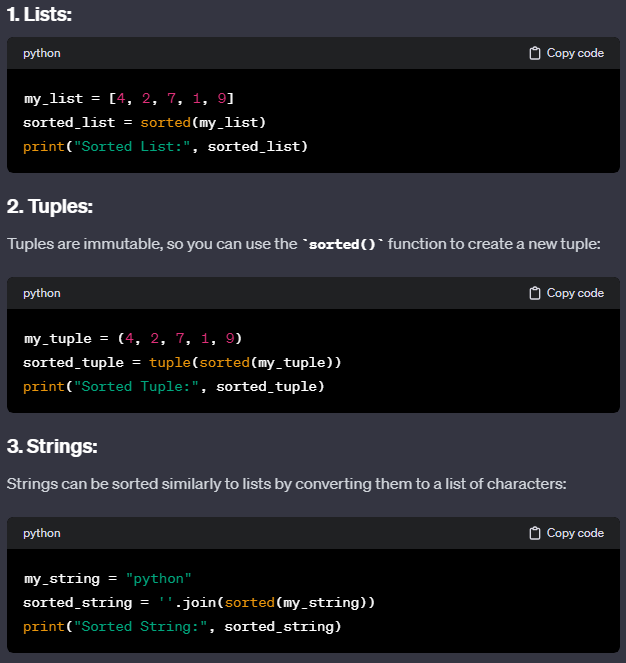
defaultdict(int)

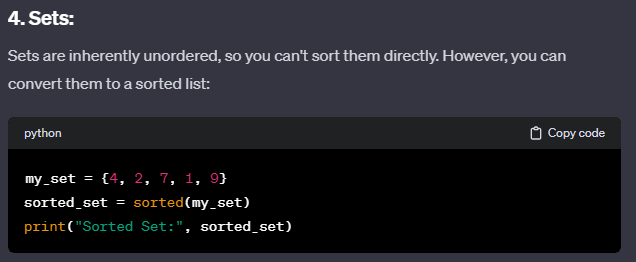


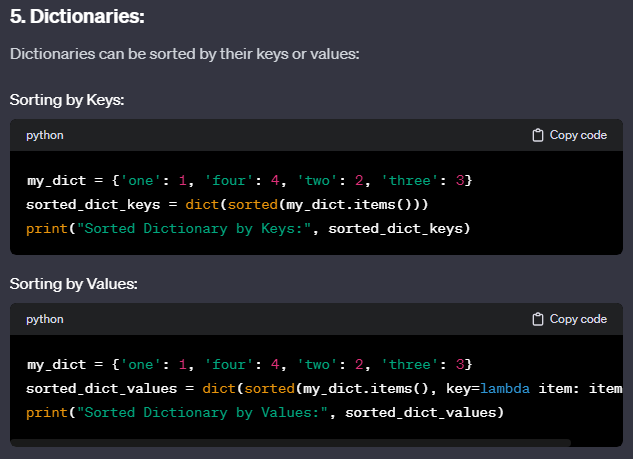
List

Sorting

sorted() will always make a copy and return a list. Need to remember to convert it back to the right data type accordingly







my\_dict = {'one': 1, 'four': 4, 'two': 2, 'three': 3}

sorted\_dict\_values = dict(sorted(my\_dict.items(), key=lambda item: item[1]))

print("Sorted Dictionary by Values:", sorted\_dict\_values)

Note:

my\_dict = {'a': 1, 'b': 4, 'c': 2, 'd': 3}

**# 1. Sorting Dictionary Keys**

sorted\_keys = sorted(my\_dict)

# Output: ['a', 'b', 'c', 'd']

**# 2. Sorting Dictionary Keys in Reverse Order**

sorted\_keys\_reverse = sorted(my\_dict, reverse=True)

# Output: ['d', 'c', 'b', 'a']

**# 3. Sorting Dictionary Items by Keys**

sorted\_items\_by\_keys = sorted(my\_dict.items())

# Output: [('a', 1), ('b', 4), ('c', 2), ('d', 3)]

**# 4. Sorting Dictionary Items by Values**

sorted\_items\_by\_values = sorted(my\_dict.items(), key=lambda x: x[1])

# Output: [('a', 1), ('c', 2), ('d', 3), ('b', 4)]

**# 5. Sorting Dictionary Items by Values in Reverse Order**

sorted\_items\_by\_values\_reverse = sorted(my\_dict.items(), key=lambda x: x[1], reverse=True)

# Output: [('b', 4), ('d', 3), ('c', 2), ('a', 1)]

**# 6. Sorting Dictionary Items with Custom Key Function**

sorted\_items\_custom\_key = sorted(my\_dict.items(), key=lambda x: len(x[0]))

# Output: [('a', 1), ('b', 4), ('c', 2), ('d', 3)]

**# 7. Sorting Dictionary Items with operator.itemgetter (Old Python Compatible)**

import operator

sorted\_items\_itemgetter = sorted(my\_dict.items(), key=operator.itemgetter(1))

# Output: [('a', 1), ('c', 2), ('d', 3), ('b', 4)]

Basically for key=lambda x: x[1], x here represents a tuple because my\_dict.items() returns a tuple of (key, val)