Python collections模块之deque详解

    deque()

        append()

        appendleft()

        extend()

        extendleft()

        pop()

        popleft()

        count()

        insert(index,obj)

        rotate(n)

        clear()

        remove()

        maxlen

    collections模块 ==> Python标准库，数据结构常用的模块；collections包含了一些特殊的容器，针对Python内置的容器，例如list、dict、set和tuple，提供了另一种选择。

collections模块常用类型有：

    双向队列（deque）

    类似于list的容器，可以快速的在队列头部和尾部添加、删除元素

    请点击deque

    计数器（Counter）

    dict的子类，计算可hash的对象

    请点击Counter

    默认字典（defaultdict）

    dict的子类，可以调用提供默认值的函数

    请点击defaultdict

    有序字典（OrderedDict）

    dict的子类，可以记住元素的添加顺序

    可命名元组（namedtuple）

    可以创建包含名称的tuple

deque()

deque是栈和队列的一种广义实现，deque是"double-end queue"的简称；deque支持线程安全、有效内存地以近似O(1)的性能在deque的两端插入和删除元素，尽管list也支持相似的操作，但是它主要在固定长度操作上的优化，从而在pop(0)和insert(0,v)（会改变数据的位置和大小）上有O(n)的时间复杂度。

常用方法：

append()

从右端添加元素（与list同）

st = "abcd"

list1 = [0, 1, 2, 3]

dst = deque(st)

dlist1 = deque(list1)

dst.append(4)

dlist1.append("k")

print(dst)

print(dlist1)

#结果：

#deque(['a', 'b', 'c', 'd', 4])

#deque([0, 1, 2, 3, 'k'])

    1

    2

    3

    4

    5

    6

    7

    8

    9

    10

    11

appendleft()

从左端添加元素

st = "abcd"

list1 = [0, 1, 2, 3]

dst = deque(st)

dlist1 = deque(list1)

dst.appendleft(4)

dlist1.appendleft("k")

print(dst)

print(dlist1)

#结果：

#deque([4, 'a', 'b', 'c', 'd'])

#deque(['k', 0, 1, 2, 3])

    1

    2

    3

    4

    5

    6

    7

    8

    9

    10

    11

extend()

从右端逐个添加可迭代对象（与list同）

Python中的可迭代对象有：列表、元组、字典、字符串

from collections import deque

ex = (1, "h", 3)

st = "abcd"

list1 = [0, 1, 2, 3]

dst = deque(st)

dlist1 = deque(list1)

dst.extend(ex)

dlist1.extend(ex)

print(dst)

print(dlist1)

#结果：

#deque(['a', 'b', 'c', 'd', 1, 'h', 3])

#deque([0, 1, 2, 3, 1, 'h', 3])

    1

    2

    3

    4

    5

    6

    7

    8

    9

    10

    11

    12

    13

extendleft()

从左端逐个添加可迭代对象

Python中的可迭代对象有：列表、元组、字典、字符串

from collections import deque

ex = [("a", 1), 3]

st = "abcd"

list1 = [0, 1, 2, 3]

dst = deque(st)

dlist1 = deque(list1)

dst.extend(ex)

dlist1.extend(ex)

print(dst)

print(dlist1)

#结果：

#deque(['a', 'b', 'c', 'd', ('a', 1), 3])

#deque([0, 1, 2, 3, ('a', 1), 3])

    1

    2

    3

    4

    5

    6

    7

    8

    9

    10

    11

    12

    13

    14

pop()

移除列表中的一个元素（默认最右端的一个元素），并且返回该元素的值（与list同），如果没有元素，将会报出IndexError

from collections import deque

st = "abcd"

list1 = [0, 1, 2, 3]

dst = deque(st)

dlist1 = deque(list1)

p = dst.pop()

p1 = dlist1.pop()

print(p)

print(p1)

print(dst)

print(dlist1)

#结果:

#d

#3

#deque(['a', 'b', 'c'])

#deque([0, 1, 2])

    1

    2

    3

    4

    5

    6

    7

    8

    9

    10

    11

    12

    13

    14

    15

    16

popleft()

移除列表中的一个元素（默认最左端的一个元素），并且返回该元素的值，如果没有元素，将会报出IndexError

from collections import deque

st = "abcd"

list1 = [0, 1, 2, 3]

dst = deque(st)

dlist1 = deque(list1)

p = dst.popleft()

p1 = dlist1.popleft()

print(p)

print(p1)

print(dst)

print(dlist1)

#结果:

#a

#0

#deque(['b', 'c', 'd'])

#deque([1, 2, 3])

    1

    2

    3

    4

    5

    6

    7

    8

    9

    10

    11

    12

    13

    14

    15

    16

count()

统计队列中的元素个数（与list同）

from collections import deque

st = "abbcd"

dst = deque(st)

p = dst.count("b")

print(dst)

print(p)

#结果:

#deque(['a', 'b', 'b', 'c', 'd'])

#2

    1

    2

    3

    4

    5

    6

    7

    8

    9

    10

insert(index,obj)

在指定位置插入元素（与list同）

from collections import deque

st = "abbcd"

dst = deque(st)

dst.insert(0, "chl")

print(dst)

#结果:

#deque(['chl', 'a', 'b', 'b', 'c', 'd'])

    1

    2

    3

    4

    5

    6

    7

    8

rotate(n)

rotate(n)， 从右侧反转n步，如果n为负数，则从左侧反转。

d.rotate(1) 等于 d.appendleft(d.pop())

from collections import deque

st = "abbcd"

dst = deque(st)

dst.rotate(1)

print(dst)

#结果:

#deque(['d', 'a', 'b', 'b', 'c'])

    1

    2

    3

    4

    5

    6

    7

    8

clear()

将deque中的元素全部删除，最后长度为0

from collections import deque

st = "abbcd"

dst = deque(st)

dst.clear()

print(dst)

#结果:

#deque([])

    1

    2

    3

    4

    5

    6

    7

    8

remove()

移除第一次出现的元素，如果没有找到，报出ValueError

from collections import deque

st = "abbcd"

dst = deque(st)

dst.remove("a")

print(dst)

dst.remove("f")

#结果:

#deque(['b', 'b', 'c', 'd'])

#ValueError: deque.remove(x): x not in deque

    1

    2

    3

    4

    5

    6

    7

    8

    9

    10

maxlen

只读的属性，deque限定的最大长度，如果无，就返回None。

当限制长度的deque增加超过限制数的项时, 另一边的项会自动删除。

from collections import deque

dst = deque(maxlen=2)

dst.append(1)

dst.append(2)

print(dst)

dst.append(3)

print(dst)

print(dst.maxlen)

#结果:

#deque([1, 2], maxlen=2)

#deque([2, 3], maxlen=2)

#2

    1

    2

    3

    4

    5

    6

    7

    8

    9

    10

    11

    12

    13

    此外,deque还支持迭代、序列化、len(d)、reversed(d)、copy.copy(d)、copy.deepcopy(d)，通过in操作符进行成员测试和下标索引。