

容器 Collections

容器(Collections)

Python附带一个模块，它包含许多容器数据类型，名字叫作collections。我们将讨论它的作用和用法。

我们将讨论的是：

- defaultdict
- counter
- deque
- namedtuple
- enum.Enum (包含在Python 3.4以上)

defaultdict

我个人使用defaultdict较多，与dict类型不同，你不需要检查key是否存在，所以我们能这样做：

```
from collections import defaultdict

colours = (
    ('Yasoob', 'Yellow'),
    ('Ali', 'Blue'),
    ('Arham', 'Green'),
    ('Ali', 'Black'),
    ('Yasoob', 'Red'),
    ('Ahmed', 'Silver'),
)

favourite_colours = defaultdict(list)

for name, colour in colours:
    favourite_colours[name].append(colour)

print(favourite_colours)
```

运行输出

```
# defaultdict(<type 'list'>,
#     {'Arham': ['Green'],
#      'Yasoob': ['Yellow', 'Red'],
#      'Ahmed': ['Silver'],
#      'Ali': ['Blue', 'Black']}
# )
```

另一种重要的是例子就是：当你在一个字典中对一个键进行嵌套赋值时，如果这个键不存

在，会触发`keyError`异常。`defaultdict`允许我们用一个聪明的方式绕过这个问题。首先我分享一个使用`dict`触发`KeyError`的例子，然后提供一个使用`defaultdict`的解决方案。

问题：

```
some_dict = {}
some_dict['colours']['favourite'] = "yellow"
```

```
## 异常输出：KeyError: 'colours'
```

解决方案：

```
import collections
tree = lambda: collections.defaultdict(tree)
some_dict = tree()
some_dict['colours']['favourite'] = "yellow"
```

```
## 运行正常
```

你可以用`json.dumps`打印出`some_dict`，例如：

```
import json
print(json.dumps(some_dict))

## 输出：{"colours": {"favourite": "yellow"}}
```

counter

`Counter`是一个计数器，它可以帮助我们针对某项数据进行计数。比如它可以用来计算每个人喜欢多少种颜色：

```
from collections import Counter

colours = (
    ('Yasoob', 'Yellow'),
    ('Ali', 'Blue'),
    ('Arham', 'Green'),
    ('Ali', 'Black'),
    ('Yasoob', 'Red'),
    ('Ahmed', 'Silver'),
)

favs = Counter(name for name, colour in colours)
print(favs)

## 输出：
## Counter({
##     'Yasoob': 2,
##     'Ali': 2,
##     'Arham': 1,
##     'Ahmed': 1
## })
```