

迭代器

[2.7旧版教程](#)

阅读: 189307

我们已经知道，可以直接作用于 `for` 循环的数据类型有以下几种：

一类是集合数据类型，如 `list`、`tuple`、`dict`、`set`、`str` 等；

一类是 `generator`，包括生成器和带 `yield` 的 `generator function`。

这些可以直接作用于 `for` 循环的对象统称为可迭代对象：`Iterable`。

可以使用 `isinstance()` 判断一个对象是否是 `Iterable` 对象：

```
>>> from collections import Iterable
>>> isinstance([], Iterable)
True
>>> isinstance({}, Iterable)
True
>>> isinstance('abc', Iterable)
True
>>> isinstance((x for x in range(10)), Iterable)
True
>>> isinstance(100, Iterable)
False
```

而生成器不但可以作用于 `for` 循环，还可以被 `next()` 函数不断调用并返回下一个值，直到最后抛出 `StopIteration` 错误表示无法继续返回下一个值了。

可以被 `next()` 函数调用并不断返回下一个值的对象称为迭代器：`Iterator`。

可以使用 `isinstance()` 判断一个对象是否是 `Iterator` 对象：

```
>>> from collections import Iterator
>>> isinstance((x for x in range(10)), Iterator)
True
>>> isinstance([], Iterator)
False
>>> isinstance({}, Iterator)
False
>>> isinstance('abc', Iterator)
False
>>> isinstance(100, Iterator)
False
```

生成器都是 `Iterator` 对象，但 `list`、`dict`、`str` 虽然是 `Iterable`，却不是 `Iterator`。

把 `list`、`dict`、`str` 等 `Iterable` 变成 `Iterator` 可以使用 `iter()` 函数：

```
>>> isinstance(iter([]), Iterator)
True
>>> isinstance(iter('abc'), Iterator)
True
```

这是因为Python的 `Iterator` 对象表示的是一个数据流，`Iterator`对象可以被 `next()` 函数调用并不断返回下一个数据，直到没有数据时抛出 `StopIteration` 错误。可以把这个数据流看做是一个有序序列，但我们却不能提前知道序列的长度，只能不断通过 `next()` 函数实现按需计算下一个数据，所以 `Iterator` 的计算是惰性的，只有在需要返回下一个数据时它才会计算。

`Iterator` 甚至可以表示一个无限大的数据流，例如全体自然数。而使用`list`是永远不可能存储全体自然数的。

小结

凡是可作用于 `for` 循环的对象都是 `Iterable` 类型；

凡是可作用于 `next()` 函数的对象都是 `Iterator` 类型，它们表示一个惰性计算的序列；

集合数据类型如 `list`、`dict`、`str` 等是 `Iterable` 但不是 `Iterator`，不过可以通过 `iter()` 函数获得一个 `Iterator` 对象。

Python的 `for` 循环本质上就是通过不断调用 `next()` 函数实现的，例如：

```
for x in [1, 2, 3, 4, 5]:
    pass
```

实际上完全等价于：

```
# 首先获得Iterator对象:
it = iter([1, 2, 3, 4, 5])
# 循环:
while True:
    try:
        # 获得下一个值:
        x = next(it)
    except StopIteration:
        # 遇到StopIteration就退出循环
        break
```

参考源码

[do_iter.py](#)

感觉本站内容不错，读后有收获？

¥ 我要小额赞助，鼓励作者写出更好的教程

还可以分享给朋友

分享 赶快成为第一个分享的人吧

珠峰NODLJS全栈开发

Angular React Gulp

Meteor Vue 微信开发

技术陪伴成长社区 珠峰培训

百万级python导师亲身指导

保你120天
变身python大牛

有时候，你需要的只是一句点拨

立即咨询

掘金

一个只有高手分享
的技术社区

立即加入

深度学习在线课程
通向无人驾驶的必经之路

< [生成器](#)

[函数式编程](#) >

评论

发表评论

Sign In to Make a Comment

廖雪峰的官方网站©2015
Powered by iTranswarp.js
由阿里云托管
广告合作



友情链接: 中华诗词 - 阿里云 - SICP - 4clojure