

5.3 使用其他分隔符或行终止符打印

问题

你想使用 `print()` 函数输出数据，但是想改变默认的分隔符或者行尾符。

解决方案

可以使用在 `print()` 函数中使用 `sep` 和 `end` 关键字参数，以你想要的方式输出。比如：

```
>>> print('ACME', 50, 91.5)
ACME 50 91.5
>>> print('ACME', 50, 91.5, sep=',')
ACME,50,91.5
>>> print('ACME', 50, 91.5, sep=',', end='!!\n')
ACME,50,91.5!!
>>>
```

使用 `end` 参数也可以在输出中禁止换行。比如：

```
>>> for i in range(5):
...     print(i)
...
0
1
2
3
4
>>> for i in range(5):
...     print(i, end=' ')
...
0 1 2 3 4 >>>
```

讨论

当你想使用非空格分隔符来输出数据的时候，给 `print()` 函数传递一个 `sep` 参数是最简单的方案。有时候你会看到一些程序员会使用 `str.join()` 来完成同样的事情。比如：

```
>>> print(','.join(('ACME', '50', '91.5')))
ACME,50,91.5
>>>
```

`str.join()` 的问题在于它仅仅适用于字符串。这意味着你通常需要执行另外一些转换才能让它正常工作。比如：

```
>>> row = ('ACME', 50, 91.5)
>>> print(','.join(row))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: sequence item 1: expected str instance, int found
>>> print(','.join(str(x) for x in row))
ACME,50,91.5
>>>
```

当然也可以不用 `str.join()` 来完成，只需要像下面这样写：

你当然可以不用那么麻烦，只需要像下面这样写：

```
>>> print(*row, sep=',')  
ACME,50,91.5  
>>>
```