列表与二维列表

Python Reference

Lists

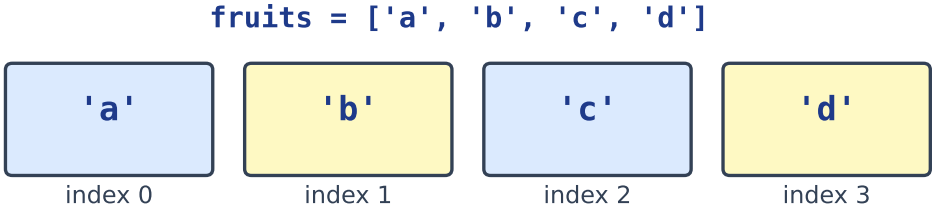
列表 (list) 在 `[]` 里按顺序保存许多值。每个元素 (item) 都有一个索引 (从 0 开始)。

```
scores = [88, 71, 95]
print(scores[0])      # 88
print(len(scores))    # 3
scores.append(60)     # add to the end
print(scores)         # [88, 71, 95, 60]
```

- 用索引改变某个元素:`scores[1] = 100`。
- 列表可以变长变短; 字符串不行。

天天要用的列表工具:

工具	作用
`a.append(x)`	在末尾加上 `x`
`a.insert(i, x)`	在位置 `i` 处插入 `x`
`a.remove(x)`	删除第一个 `x`
`a.pop()` / `a.pop(i)`	删除并返回最后一个元素 / 第 `i` 个
`a.sort()`	原地排序
`sorted(a)`	返回一个**新**的已排序列表
`x in a`	`x` 在列表里吗?
`len(a)`、`sum(a)`、`max(a)`、`min(a)`	大小和快捷计算



Exam tip: first item is index 0; last is len(list)-1; negative indices count from the end

List indices start at 0

Traversing a list

遍历 (traverse) 列表就是逐个访问每个元素。`for` 循环可以做到, 不需要索引。

```
scores = [88, 71, 95]
total = 0
for s in scores:
    total = total + s
print(total)          # 254
```

- 当你同时需要索引时, 用 `enumerate`。

```
for i, name in enumerate(["a", "b"]):
    print(i, name)    # 0 a  /  1 b
```

2-D lists (grids)

二维列表 (2-D list) 是" 列表的列表"— 一个由行和列组成的网格 (grid)。用两个索引:`grid[row][col]`。

```
grid = [[1, 2, 3],
        [4, 5, 6]]
print(grid[0][2])     # 3
print(grid[1][0])     # 4
```

- 用嵌套循环 (nested loop) 访问每一个格子。

```
grid = [[1, 2], [3, 4]]
for row in grid:
    for value in row:
        print(value, end=" ")
print()               # 1 2 3 4
```

List comprehensions

列表推导式 (list comprehension) 用一行就能构建一个新列表:`[expression for item in sequence]`。

```
squares = [x * x for x in range(5)]
print(squares)        # [0, 1, 4, 9, 16]
```

- 加上 `if` 只保留部分元素。

```
evens = [n for n in range(10) if n % 2 == 0]
print(evens)          # [0, 2, 4, 6, 8]
```

Tuples & sets

元组 (tuple) 是圆括号里的固定序列。创建之后不能修改 — 用它保存天生属于一起的值, 并把它解包 (unpack) 到多个名字里。

```
point = (3, 4)
x, y = point            # unpacking
print(x, y)             # 3 4
```

需要一次交回两个结果的函数, 就返回一个元组:

```
def min_max(nums):
    return min(nums), max(nums)

lo, hi = min_max([5, 2, 9])
print(lo, hi)           # 2 9
```

集合 (set) 每个值只存一次, 而且没有顺序。它最适合去除重复, 以及做快速的成员测试 (membership test)。

```
votes = ["red", "blue", "red", "green", "red"]
colours = set(votes)
print(len(colours))         # 3  (duplicates removed)
print("blue" in colours)    # True
```

常见错误

- `b = a` 共享同一个列表。`a.copy()` 或 `a[:]` 创建新的外层列表, 但嵌套的可变项仍被共享 (浅拷贝)。
- 最后一个元素是 `a[-1]`;`a[len(a)]` 会越界。
- `append` 只加**一个**元素; 要拼接另一个列表用 `extend` 或 `+`。
- 用 `[[0]*3]*3` 建网格会得到**同一**行的三个引用。要用循环逐行创建。
- 只有一个元素的元组要带逗号:`(5,)`, 不是 `(5)`。
- 集合没有顺序、没有重复, 所以不能用 `s[0]` 来索引。