

Lists & 2-D lists

Python Reference

Lists

A list 列表 holds many values in order, inside []. Each item 元素 has an index (from 0).

```
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]
```

- Change an item by its index: `scores[1] = 100`.
- A list can grow and shrink; a string cannot.

Traversing a list

To traverse 遍历 a list is to visit each item. A for loop does this with no index needed.

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

- Use `enumerate` when you also need the index.

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

2-D lists (grids)

A 2-D list 二维列表 is a list of lists —a grid 网格 of rows and columns. Use two indexes: `grid[row][col]`.

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

- A nested loop 嵌套循环 visits every cell.

```
grid = [[1, 2], [3, 4]]
for row in grid:
    for value in row:
```

```
    print(value, end=" ")
print()          # 1 2 3 4
```

List comprehensions

A list comprehension 列表推导式 builds a new list in one line: [expression for item in sequence].

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

- Add if to keep only some items.

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