

# 推导式

- 目标
  - 列表推导式
  - 字典推导式
  - 集合推导式

## 1、列表推导式

作用：用一个表达式创建一个有规律的列表或控制一个有规律列表。

列表推导式又叫列表生成式。

### 1、1 快速体验

需求：创建一个0-10的列表。

- while循环实现

```
# 1. 准备一个空列表
list1 = []

# 2. 书写循环，依次追加数字到空列表list1中
i = 0
while i < 10:
    list1.append(i)
    i += 1

print(list1)
```

- for循环实现

```
list1 = []
for i in range(10):
    list1.append(i)

print(list1)
```

- 列表推导式实现

```
list1 = [i for i in range(10)]
print(list1)
```

### 1、2 带if的列表推导式

需求：创建0-10的偶数列表

- 方法一：range()步长实现

```
list1 = [i for i in range(0, 10, 2)]
print(list1)
```

- 方法二: if实现

```
list1 = [i for i in range(10) if i % 2 == 0]
print(list1)
```

## 2、字典推导式

思考: 如果有如下两个列表:

```
list1 = ['name', 'age', 'gender']
list2 = ['Tom', 20, 'man']
```

如何快速合并为一个字典?

答: 字典推导式

字典推导式作用: 快速合并列表为字典或提取字典中目标数据。

### 2、1 快速体验

1. 创建一个字典: 字典key是1-5数字, value是这个数字的2次方。

```
dict1 = {i: i**2 for i in range(1, 5)}
print(dict1) # {1: 1, 2: 4, 3: 9, 4: 16}
```

2. 将两个列表合并为一个字典

```
list1 = ['name', 'age', 'gender']
list2 = ['Tom', 20, 'man']

dict1 = {list1[i]: list2[i] for i in range(len(list1))}
print(dict1)
```

## 3、集合推导式

需求: 创建一个集合, 数据为下方列表的2次方。

```
list1 = [1, 1, 2]
```

代码如下:

```
list1 = [1, 1, 2]
set1 = {i ** 2 for i in list1}
print(set1) # {1, 4}
```

注意：集合有数据去重功能。

## 4、总结

---

- 推导式的作用：简化代码
- 推导式写法

# 列表推导式

```
[xx for xx in range()]
```

# 字典推导式

```
{xx1: xx2 for ... in ...}
```

# 集合推导式

```
{xx for xx in ...}
```