

Python编程惯例

“惯例”这个词指的是“习惯的做法，常规的办法，一贯的做法”，与这个词对应的英文单词叫“idiom”。由于Python跟其他很多编程语言在语法和使用上还是有比较显著的差别，因此作为一个Python开发者如果不能掌握这些惯例，就无法写出“Pythonic”的代码。下面我们总结了一些在Python开发中的惯用的代码。

1. 让代码既可以被导入又可以被执行。

```
1 | if __name__ == '__main__':
```

2. 用下面的方式判断逻辑“真”或“假”。

```
1 | if x:  
2 | if not x:
```

好的代码：

```
1 | name = 'jackfrued'  
2 | fruits = ['apple', 'orange', 'grape']  
3 | owners = {'1001': '骆昊', '1002': '王大锤'}  
4 | if name and fruits and owners:  
5 |     print('I love fruits!')
```

不好的代码：

```
1 | name = 'jackfrued'  
2 | fruits = ['apple', 'orange', 'grape']  
3 | owners = {'1001': '骆昊', '1002': '王大锤'}  
4 | if name != '' and len(fruits) > 0 and owners != {}:  
5 |     print('I love fruits!')
```

3. 善于使用in运算符。

```
1 | if x in items: # 包含  
2 | for x in items: # 迭代
```

好的代码：

```
1 | name = 'Hao LUO'  
2 | if 'L' in name:  
3 |     print('The name has an L in it.')
```

不好的代码：

```

1 name = 'Hao LUO'
2 if name.find('L') != -1:
3     print('This name has an L in it!')

```

4. 不使用临时变量交换两个值。

```

1 a, b = b, a

```

5. 用序列构建字符串。

好的代码：

```

1 chars = ['j', 'a', 'c', 'k', 'f', 'r', 'u', 'e', 'd']
2 name = ''.join(chars)
3 print(name) # jackfrued

```

不好的代码：

```

1 chars = ['j', 'a', 'c', 'k', 'f', 'r', 'u', 'e', 'd']
2 name = ''
3 for char in chars:
4     name += char
5 print(name) # jackfrued

```

6. EAFP优于LBYL。

EAFP - **E**asier to **A**sk **F**orgiveness than **P**ermission.

LBYL - **L**ook **B**efore **Y**ou **L**ean.

好的代码：

```

1 d = {'x': '5'}
2 try:
3     value = int(d['x'])
4     print(value)
5 except (KeyError, TypeError, ValueError):
6     value = None

```

不好的代码：

```

1 d = {'x': '5'}
2 if 'x' in d and isinstance(d['x'], str) \
3     and d['x'].isdigit():
4     value = int(d['x'])
5     print(value)
6 else:
7     value = None

```

7. 使用enumerate进行迭代。

好的代码：

```
1 fruits = ['orange', 'grape', 'pitaya', 'blueberry']
2 for index, fruit in enumerate(fruits):
3     print(index, ': ', fruit)
```

不好的代码：

```
1 fruits = ['orange', 'grape', 'pitaya', 'blueberry']
2 index = 0
3 for fruit in fruits:
4     print(index, ': ', fruit)
5     index += 1
```

8. 用生成式生成列表。

好的代码：

```
1 data = [7, 20, 3, 15, 11]
2 result = [num * 3 for num in data if num > 10]
3 print(result) # [60, 45, 33]
```

不好的代码：

```
1 data = [7, 20, 3, 15, 11]
2 result = []
3 for i in data:
4     if i > 10:
5         result.append(i * 3)
6 print(result) # [60, 45, 33]
```

9. 用zip组合键和值来创建字典。

好的代码：

```
1 keys = ['1001', '1002', '1003']
2 values = ['骆昊', '王大锤', '白元芳']
3 d = dict(zip(keys, values))
4 print(d)
```

不好的代码：

```
1 keys = ['1001', '1002', '1003']
2 values = ['骆昊', '王大锤', '白元芳']
3 d = {}
4 for i, key in enumerate(keys):
5     d[key] = values[i]
6 print(d)
```

说明：这篇文章的内容来自于网络，有兴趣的读者可以阅读[原文](#)。