

python 基础 05：缩进和选择

1、缩进

Python 最具特色的是用缩进来标明成块的代码。我下面以 if 选择结构来举例。if 后面跟随条件，如果条件成立，则执行归属于 if 的一个代码块。

先看 C 语言的表达方式（注意，这是 C，不是 Python!）

```
if (i > 0) {    x = 1;    y = 2; }
```

如果 $i > 0$ 的话，我们将进行括号中所包括的两个赋值操作。括号中包含的就是块操作，它隶属于 if。

在 Python 中，同样的目的，这段话是这样的

```
if i > 0:    x = 1    y = 2
```

在 Python 中，去掉了 $i > 0$ 周围的括号，去除了每个语句句尾的分号，表示块的花括号也消失了。

多出来了 if ...之后的:(冒号), 还有就是 $x = 1$ 和 $y = 2$ 前面有四个空格的缩进。通过缩进，Python 识别出这两个语句是隶属于 if。

Python 这样设计的理由纯粹是为了程序好看。

2、if 语句

写一个完整的程序，命名为 ifDemo.py。这个程序用于实现 if 结构。

```
i = 1    x = 1if i > 0:    x = x+1print x
```

```
$python ifDemo.py # 运行
```

程序运行到 if 的时候，条件为 True，因此执行 $x = x + 1$ 。

print x 语句没有缩进，那么就是 if 之外。

如果将第一句改成 `i = -1`，那么 if 遇到假值 (False), `x = x+1` 隶属于 if, 这一句跳过。print x 没有缩进，是 if 之外，不跳过，继续执行。

这种以四个空格的缩进来表示隶属关系的书写方式，以后还会看到。强制缩进增强了程序的可读性。

复杂一些的 if 选择：

```
i = 1
```

```
if i > 0:
```

```
    print 'positive i'    i = i + 1elif i == 0:
```

```
    print 'i is 0'        i = i * 10else:
```

```
    print 'negative i'    i = i - 1
```

```
    print 'new i:',i
```

这里有三个块，分别属于 if, elif, else 引领。

Python 检测条件，如果发现 if 的条件为假，那么跳过后面紧跟的块，检测下一个 elif 的条件；如果还是假，那么执行 else 块。

通过上面的结构将程序分出三个分支。程序根据条件，只执行三个分支中的一个。

整个 if 可以放在另一个 if 语句中，也就是 if 结构的嵌套使用：

```
i = 5if i > 1:    print 'i bigger than 1'    print 'good'    if i > 2:        print 'i bigger than 2'        print 'even better'
```

if i > 2 后面的块相对于该 if 缩进了四个空格，以表明其隶属于该 if，而不是外层的 if。

3、总结

if 语句之后的冒号

以四个空格的缩进来表示隶属关系, Python 中不能随意缩进

```
if <条件 1>:
```

```
    statement
```

```
elif <条件 2>:
```

```
    statement
```

```
elif <条件 3> :
```

```
    statement
```

```
else:
```

```
    statement
```