01：缩进规则

python是通过缩进来组织代码结构的。

1.1 代码组与代码块

缩进相同的一组语句构成一个代码块，也可叫做代码组。

首行以关键字开始，以冒号( : )结束，该行之后的一行或多行代码构成代码组。例如if、while、def和class这样的复合语句。

例:

def func(a, b):

c = a + b

print("c:", c)

return c

if a < b:

print('a:', a)

print('b:', b)

print('a<b')

while a < b:

a = a + 1

b = b - 1

print('a:', a)

print('b:', b)

class People(object):

name = '张三'

age = 20

def read(self):

print('{}正在读书'.format(self.name))

def sleep(self):

print('{}正在睡觉'.format(self.name))

上述四个例子，除了首行之外，首行下面的语句，称为代码块或代码组。简单理解，因为他们是放在一起的，所以叫做块，一般将首行及后面的代码组称为一个子句(clause)。

1.2 同行书写多条语句

Python 通常是一行写完一条语句，但当语句较长时，可以使用反斜杠(\)来实现多行语句。

例如：

print("Python 通常是一行写完一条语句，但当语句较长时，可以使用反斜杠(\)来实现多行语句。")

上面是一行写完一条语句的例子，也可以把一条语句分成多行来写，下面的例子与上面的例子效果完全一样

例:

print("Python 通常是一行写完一条语句，\

但当语句较长时，\

可以使用反斜杠(\)来实现多行语句。")

1.3 空行与缩进

1函数之间或类的方法之间用空行分隔，表示一段新的代码的开始。

例如：

def add(a, b):

return a + b

def sub(a, b):

return a - b

add函数和sub函数中间的空行表示是一段新的代码的开始。

2类和函数入口之间也用一行空行分隔，以突出函数入口的开始。

例:

class People(object):

name = "张三"

age = 20

def read(self):

print('{}正在读书'.format(self.name))

3空行与代码缩进不同，空行并不是Python语法的一部分。书写时不插入空行，Python解释器运行也不会出错。

4空行的作用在于分隔两段不同功能或含义的代码。空行也是程序代码的一部分。

5 Python使用缩进来表示代码块，不需要使用大括号{}。缩进的空格数是可变的，但是同一个代码块的语句必须包含相同的缩进空格数。