

选择焦虑的解决方案

圣则IT技能学院
编程中级课





目录

CONTENTS

①

温故知新

②

嵌套if语句

③

逻辑运算符

④

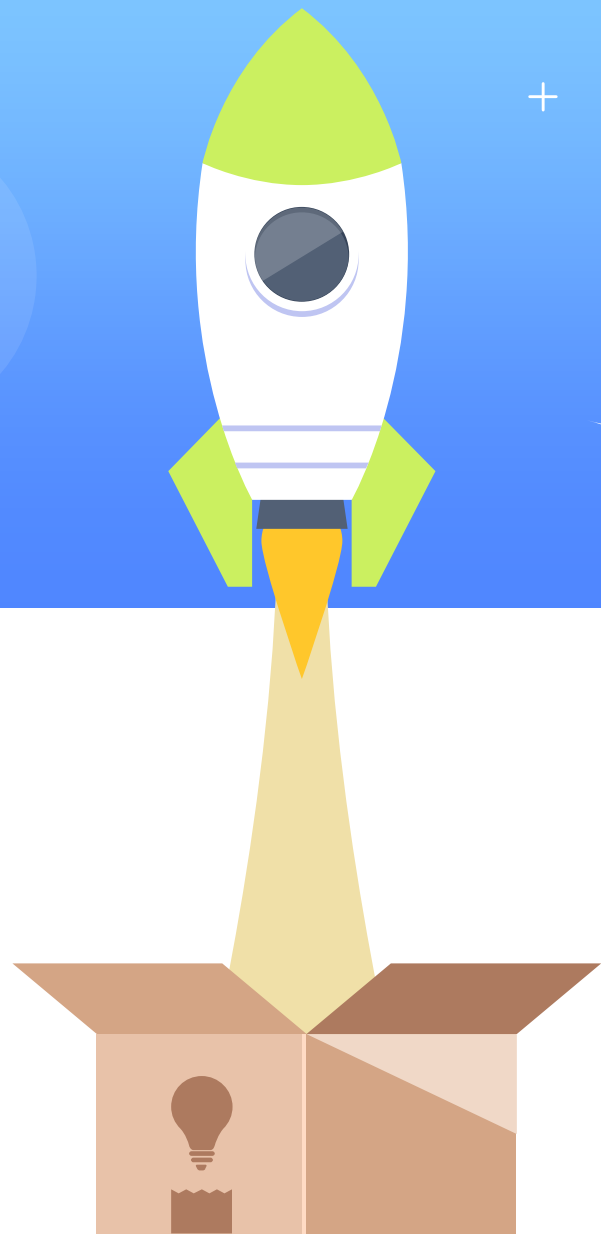
布置作业

⑤

参考资料

/01

温故知新



字符串的运算和方法



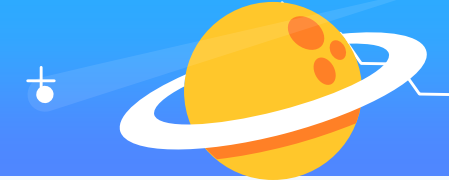
运算：

- 重复*, 取长度len(), 截取单个[]或多个字符[:], 成员in
- 内建方法
- 检查：是否满足某个条件 (True/False)
 - isupper(), islower(), isalpha(), isdigit()
- 转换：生成新的字符串
 - upper(), lower()
- 搜索：得到相关的信息
 - find(), count()

方法vs函数

- 字符串方法调用格式：**字符串.方法名(参数)**

If-else 语句格式



if 条件 :

语句1

条件是 True

else:

语句2

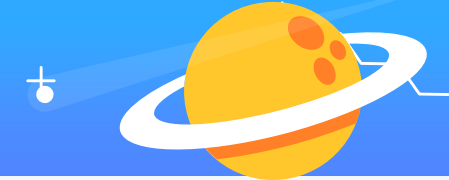
条件是 False

```
a = input("下雨了吗(y/n)?")  
if a == "n":  
    print("出去玩!")  
else:  
    print("在家学习!")
```

```
a = input("下雨了吗(y/n)?")  
if a == "y":  
    print("在家学习!")  
else:  
    print("出去玩!")
```

两种写法

If-elif-else 语句格式



```
if 条件1 :
```

```
    语句1
```

```
elif 条件2 :
```

```
    语句2
```

```
else :
```

```
    默认语句
```

```
if 条件1 :
```

```
    语句1
```

```
elif 条件2 :
```

```
    语句2
```

```
elif 条件3 :
```

```
    语句3
```

```
else :
```

```
    默认语句
```

3个分支

只有一个分支会运行

举例

```
x = input("x: ")
y = input("y: ")
x = float(x)
y = float(y)
if x > y:
    print("x 比 y 大")
elif x < y:
    print("y 比 x 大")
else:
    print("两者相等")
```

第一分支

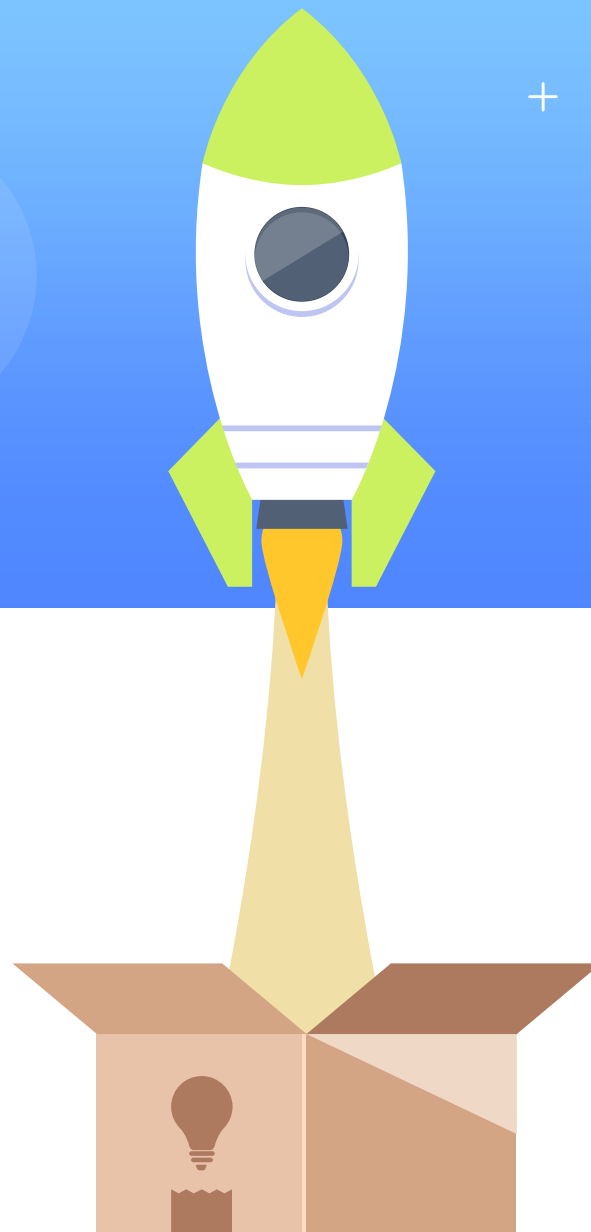
第二分支

第三分支

只有一个分支会运行!

/02

嵌套if语句



场景：世界杯爱好者



- 如果到了周末,
 - 如果凌晨有比赛, 那就凌晨4点起床.
 - 否则, 睡懒觉!
- 否则, 7点起床

```
w = input("明天是周末吗(y/n)?")
g = input("凌晨有比赛吗(y/n)?")
if w == "y":
    if g == "y":
        print("4点起床! ")
    else:
        print("睡懒觉! ")
else:
    print("7点起床! ")
```

```
w = input("明天是周末吗(y/n)?")
g = input("凌晨有比赛吗(y/n)?")
if w == "y":
    if g == "y":
        print("4点起床! ")
    else:
        print("睡懒觉! ")
else:
    print("7点起床! ")
```

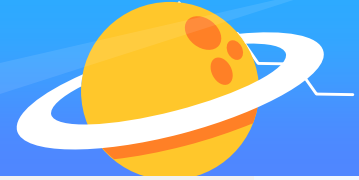
明天是否是周末	凌晨是否有比赛	显示结果
y	y	4点起床!
y	n	睡懒觉!
n	-	7点起床!

一个if语句
包含着
另一个if语句

```
w = input("明天是周末吗(y/n)?")
g = input("凌晨有比赛吗(y/n)?")
if w == "y":
    if g == "y":
        print("4点起床! ")
    else:
        print("睡懒觉! ")
else:
    print("7点起床! ")
```

嵌套if语句例子1
嵌套发生在if分支

世界杯爱好者 写法2



```
w = input("明天是周末吗(y/n)?")
g = input("凌晨有比赛吗(y/n)?")
if w == "n":
    print("7点起床! ")
else:
    if g == "y":
        print("4点起床! ")
    else:
        print("睡懒觉! ")
```

一个if语句
包含着
另一个if语句

嵌套if语句例子2
嵌套发生在else分支

```
w = input("明天是周末吗(y/n)?")
g = input("凌晨有比赛吗(y/n)?")
if w == "n":
    print("7点起床!")
```

第一分支 (if)

```
else:
```

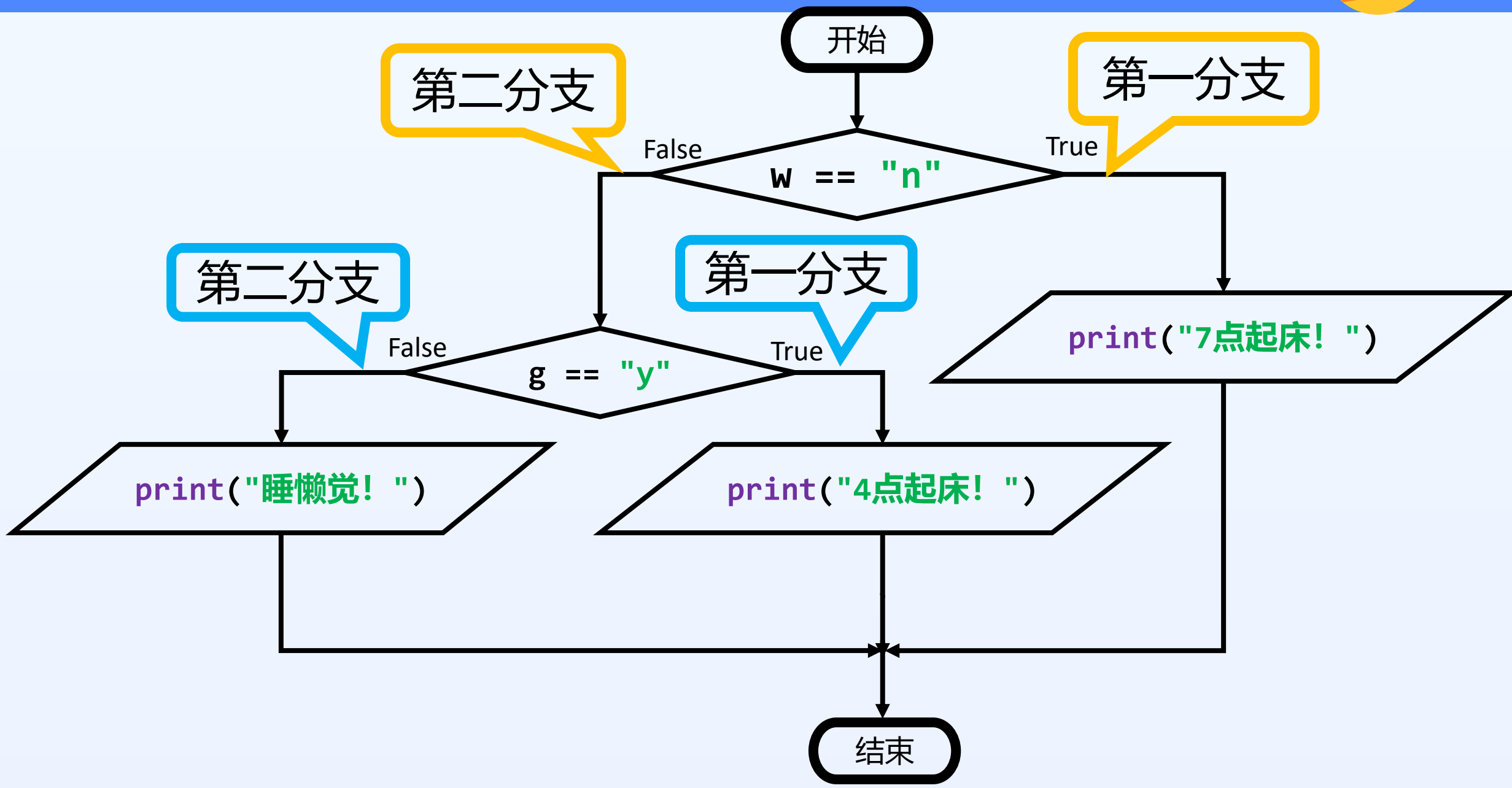
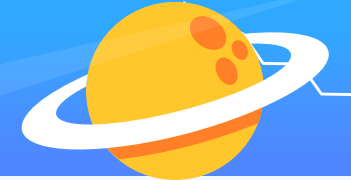
```
    if g == "y":
        print("4点起床!")
    else:
        print("睡懒觉!")
```

第一分支 (if)

第二分支 (else)

第二分支 (else)

流程图表示



世界杯爱好者 写法3



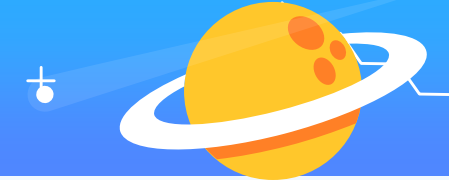
```
w = input("明天是周末吗(y/n)?")
g = input("凌晨有比赛吗(y/n)?")
if w == "n":
    print("7点起床! ")
elif g == "y":
    print("4点起床! ")
else:
    print("睡懒觉! ")
```

if语句链条

可能不是最自然的写法

```
x = input("x: ")
y = input("y: ")
x = float(x)
y = float(y)
if x > y:
    print("x 比 y 大")
elif x < y:
    print("y 比 x 大")
else:
    print("两者相等")
```


我们学过的数据类型及运算



- 数字（整数，浮点数）
- 字符串
- 布尔值（True 和 False）

数字运算符

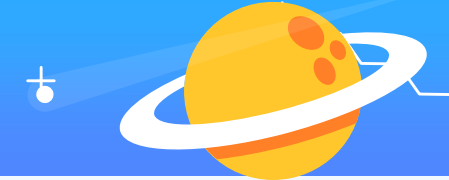
- +, -, *, /

字符串运算符

- +, *

布尔值的运算呢？

and 运算符



```
>>> False and False
False
>>> False and True
False
>>> True and False
False
>>> True and True
True
```

a	b	a and b
False	False	False
False	True	False
True	False	False
True	True	True

运算符	意思
and	如果左右两边的布尔值 都是 True，结果就是True。否认，结果是False

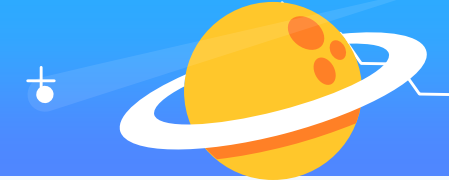
```
>>> age = 4
>>> h = 1.1
>>> (age > 5) and (h > 1.2)
False
>>> age = 6
>>> h = 1.1
>>> (age > 5) and (h > 1.2)
False
>>> age = 4
>>> h = 1.5
>>> (age > 5) and (h > 1.2)
False
>>> age = 6
>>> h = 1.5
>>> (age > 5) and (h > 1.2)
True
```

场景: 你刚上了一辆公交车。如果你的年龄是5岁以上 **并且** 你的身高超过1.2米, 你就要付车费。

问题: 你需要付车费吗?

```
age = int(input("输入年龄: "))
h = float(input("输入身高: "))
if (age > 5) and (h > 1.2) :
    print("请付车费! ")
else:
    print("免费! ")
```

or 运算符



```
>>> False or False
False
>>> False or True
True
>>> True or False
True
>>> True or True
True
```

a	b	a or b
False	False	False
False	True	True
True	False	True
True	True	True

运算符	意思
or	如果左右两边的 任何一个 布尔值是True，结果就是True。否认，结果是False

```
>>> age = 4
>>> h = 1.1
>>> (age > 5) or (h > 1.2)
False
>>> age = 4
>>> h = 1.5
>>> (age > 5) or (h > 1.2)
True
>>> age = 6
>>> h = 1.1
>>> (age > 5) or (h > 1.2)
True
>>> age = 6
>>> h = 1.5
>>> (age > 5) or (h > 1.2)
True
```

场景: 你刚上了一辆公交车。如果你的年龄是5岁以上 **或者** 你的身高超过1.2米, 你就要付车费。

问题: 你需要付车费吗?

```
age = int(input("输入年龄: "))
h = float(input("输入身高: "))
if (age > 5) or (h > 1.2) :
    print("请付车费! ")
else:
    print("免费! ")
```

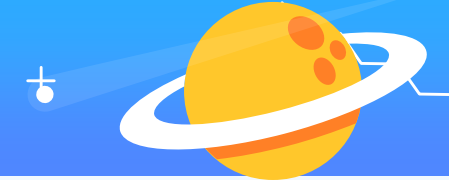
世界杯爱好者 写法4



```
w = input("明天是周末吗(y/n)?")
g = input("凌晨有比赛吗(y/n)?")
if w == "y":
    if g == "y":
        print("4点起床! ")
    else:
        print("睡懒觉! ")
else:
    print("7点起床! ")
```

```
w = input("明天是周末吗(y/n)?")
g = input("凌晨有比赛吗(y/n)?")
if w == "y" and g == "y":
    print("4点起床! ")
elif w == "y" and g != "y":
    print("睡懒觉! ")
else:
    print("7点起床! ")
```

not 运算符



只有一个操作数

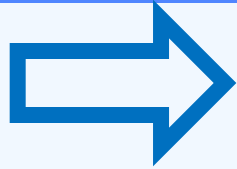
```
>>> not False
True
>>> not True
False
```

a	not a
False	True
True	False

运算符在前面,,
跟负号类似

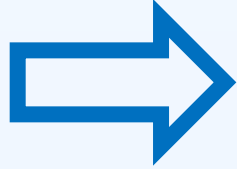
运算符	意思
not	反转结果, 如果结果为 true, 则返回 False

下雨吗? 是!



出去玩吗? 不

下雨吗? 不!



出去玩吗? 是

```
>>> raining = True
>>> not raining
False
>>> raining = False
>>> not raining
True
```

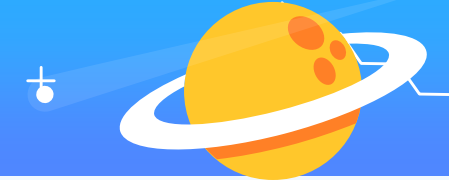
场景:

如果下雨, 我就不出去玩。如果不下雨, 我就出去玩。

问题:

我可以出去玩吗?

运算先后顺序



跟数学运算一致（从高到低）

运算符	意思
()	括号
**	求幂
*, /, //, %	乘除, 余数
+, -	加减

运算符	意思
()	括号
**	求幂
*, /, //, %	乘除, 余数
+, -	加减
<, <=, >, >=, !=, ==	比较
not x	
and	
or	

先后顺序一样的运算符号：从左到右

推荐写法



只需要记住乘除在加减以前

- 用括号() 来保证代码按照自己想要的顺序运算

原版: `not 5 > 2 or 3 * 4 > 10`

优化版: `(not (5 > 2)) or ((3 * 4) > 10)`

- 把复杂的表达式分解为几部分

原版: `5 > 6 or 7 > 6 and not 4 > 5`

优化版:

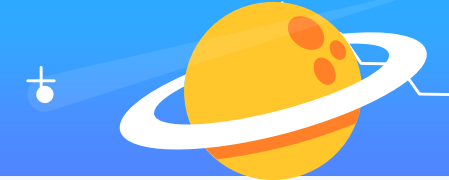
`b1 = 5 > 6`

`b2 = 7 > 6 and not 4 > 5`

`b1 or b2`

目的: 代码必须容易理解和容易发现错误

举例 +

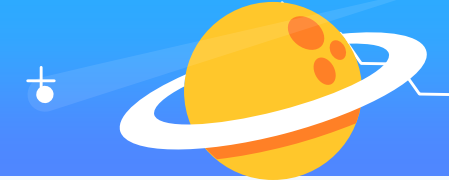


```
>>> score = 70
>>> (score > 50) and (score < 60)
False
>>> 60 > score > 50
False
>>> (score > 60) and (score <= 70)
True
>>> 60 < score <= 70
True
>>> grade = "A"
>>> (score > 80) and (grade == "B")
False
>>> (score <= 80) and (grade == "B")
False
>>> (score > 80) and (grade != "B")
False
>>> (score <= 80) and (grade != "B")
True
```

一样

一样

举例 +



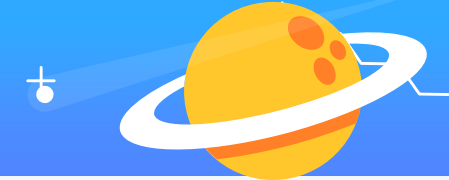
```
>>> score = 70
>>> grade = "A"
>>> (score > 80) or (grade == "B")
False
>>> (score <= 80) or (grade == "B")
True
>>> (score > 80) or (grade != "B")
True
>>> (score <= 80) or (grade != "B")
True
>>> x = 123
>>> (x % 2 == 1) or (x < 100)
True
>>> (x % 2 == 0) or (x < 100)
False
```

x 除以 2 的余数

举例 +



```
>>> score = 70
>>> score > 50
True
>>> not (score > 50)
False
>>> score < 60
False
>>> not (score < 60)
True
>>> grade = "C"
>>> grade == "B"
False
>>> not (grade == "B")
True
>>> not (grade != "B")
False
```



- 
- 嵌套if语句

https://www.w3school.com.cn/python/python_conditions.asp

- 逻辑运算符

https://www.w3school.com.cn/python/python_operators.asp

- 随机生成数字

<https://www.runoob.com/python/func-number-random.html>