Python分支语句

广州协和学校信息技术科

2024年10月

关系运算符和bool类型

- 六种关系运算符用于比较大小

相等	==
不等	!=
大于	>
小于	<
大于等于	>=
小于等于	<=

- 比较的结果是bool类型，成立则为True,反之为False
- bool类型数据只有两种取值，True或False

关系运算符和bool类型

```
print(3 < 5)                #>>True
print(4 != 7)               #>>True

a = 4
print(2 < a < 6 < 8)        #>>True
print(2 < a == 4 < 6)       #>>True
print(2 < a > 5)            #>>False
```

```
b = a < 6
print(b)                    #>>True
print( b == 1)              #>>True
print( b == 2)              #>>False

b = a > 6
print( b == 0)              #>>True

a = True
print(a == 1)               #>>True
```

逻辑运算符和逻辑表达式

- 逻辑运算符用于表达式的逻辑操作，有 and or not 三种，操作的结果是True或False
- 与: exp1 and exp2

当且仅当exp1和exp2的值都为True（或相当于True）时，结果为True（或相当于True)

```
n = 4
n > 4 and n < 5                    # false
n >= 2 and n < 5 and n%2 == 0      # true
print(5 and False)                 #>>False
print(4 and True)                  #>>True
```

逻辑运算符和逻辑表达式

- 或 : exp1 or exp2

  当且仅当exp1和exp2的值都为False（或相当于False）时，结果为False（或相当于False）

  ```
  n = 4
  n > 4 or n < 5       #True
  n <= 2 or n > 5      #False
  ```

逻辑运算符和逻辑表达式

非 : not exp

exp值为True(或相当于True)时, 结果为False(或相当于False)

exp值为False(或相当于False)时, 结果为True(或相当于True)

```
not 4 < 5                    #False
not 5                        #False
not 0                        #True
not "abc"                    #False
not ""                       #True
not 4 < 5 and 4 > 6          #False
not []                       #True
not [1]                      #False
```

逻辑运算符的优先级

- not > and > or

```
print(3 < 4 or 4 > 5 and 1 > 2 )                    #>>True
print((3 < 4 or 4 > 5) and 1 > 2 )                  #>>False
not 4 < 5 and 4 > 6 即 (not 4 < 5) and (4 > 6)
```

条件分支语句

- 有时，并非所有的程序语句都要被顺序执行到，会希望满足某种条件就执行这部分语句，满足另一条件就执行另一部分语句。这就需要“条件分支语句”。

```
if 表达式1:
    语句组1
elif 表达式2:
    语句组2
...# 可以有多个elif
elif 表达式n:
    语句组n
else:
    语句组n+1
```

依次计算表达式1、表达式2…只要碰到一个表达式i为真，则执行语句组i（前面为假的表达式对应的语句组不会被执行）,且后面的表达式不再计算，后面的语句组也都不会被执行。

若所有表达式都为假，则执行语句组n+1

注意，缩进的前一行末尾有 ' :'

条件分支语句

可以没有 elif，也可以没有 else，也可以都没有。

```
if 表达式1:
    语句组1
else:
    语句组2
```

```
if 表达式1:
    语句组1
elif 表达式2:
    语句组2
```

```
if 表达式1:
    语句组1
```

if 语句示例

例题: 写一个判断整数奇偶性的程序，要求输入一个整数，如果是奇数，就输出“It's odd.”，如果是偶数，就输出“It's even.”。

```
a = int(input())
if a % 2 == 1:
        print("It's odd.")
else:
        print("It's even.")
```

if 语句嵌套

- 在一条if语句的某个分支(语句组）里，还可以再写if语句。

```
a = int(input())
if a > 0:
        if a % 2:
                print("good")
        else:
                print("bad")
```

输入: 4　　输出：bad

输入: 3　　输出：good

输入: -1　　无输出

if 语句嵌套

- 在一条if语句的某个分支(语句组）里，还可以再写if语句。

```
a = int(input())
if a > 0:
        if a % 2:
                print("good")
else:
        print("bad")
```

输入: 4　　无输出

输入: 3　　输出：good

输入: -1　　bad