

第二章 第二节 Python 运算符

Python 运算符

运算符用于对变量和值执行操作。

Python 在以下组中划分运算符：

算术运算符

赋值运算符

比较运算符

逻辑运算符

身份运算符

成员运算符

位运算符

第二章 第二节 Python 运算符

Python 算术运算符

算术运算符与数值一起使用来执行常见的数学运算：

运算符	名称	实例	
+	加	<code>x + y</code>	
-	减	<code>x - y</code>	
*	乘	<code>x * y</code>	
/	除	<code>x / y</code>	
%	取模	<code>x % y</code>	
**	幂	<code>x ** y</code>	
//	地板除（取整除）	<code>x // y</code>	

第二章 第二节 Python 运算符

Python 赋值运算符

赋值运算符用于为变量赋值：

运算符	实例	等同于	
=	x = 5	x = 5	
+=	x += 3	x = x + 3	
-=	x -= 3	x = x - 3	
*=	x *= 3	x = x * 3	
/=	x /= 3	x = x / 3	
%=	x %= 3	x = x % 3	
//=	x //= 3	x = x // 3	
**=	x **= 3	x = x ** 3	
&=	x &= 3	x = x & 3	
=	x = 3	x = x 3	
^=	x ^= 3	x = x ^ 3	
>>=	x >>= 3	x = x >> 3	
<<=	x <<= 3	x = x << 3	

第二章 第二节 Python 运算符

Python 比较运算符

比较运算符用于比较两个值：

运算符	名称	实例	
==	等于	x == y	
!=	不等于	x != y	
>	大于	x > y	
<	小于	x < y	
>=	大于或等于	x >= y	
<=	小于或等于	x <= y	

第二章 第二节 Python 运算符

Python 比较运算符

比较运算符用于比较两个值：

运算符	名称	实例	
==	等于	x == y	
!=	不等于	x != y	
>	大于	x > y	
<	小于	x < y	
>=	大于或等于	x >= y	
<=	小于或等于	x <= y	

第二章 第二节 Python 运算符

Python 逻辑运算符

逻辑运算符用于组合条件语句：

运算符	描述	实例	
and	如果两个语句都为真，则返回 True。	<code>x > 3 and x < 10</code>	
or	如果其中一个语句为真，则返回 True。	<code>x > 3 or x < 4</code>	
not	反转结果，如果结果为 true，则返回 False	<code>not(x > 3 and x < 10)</code>	

Python 身份运算符

身份运算符用于比较对象，不是比较它们是否相等，但如果它们实际上是同一个对象，则具有相同的内存位置：

运算符	描述	实例	
is	如果两个变量是同一个对象，则返回 true。	<code>x is y</code>	
is not	如果两个变量不是同一个对象，则返回 true。	<code>x is not y</code>	

第二章 第二节 Python 运算符

Python 逻辑运算符

逻辑运算符用于组合条件语句：

运算符	描述	实例	
and	如果两个语句都为真，则返回 True。	<code>x > 3 and x < 10</code>	
or	如果其中一个语句为真，则返回 True。	<code>x > 3 or x < 4</code>	
not	反转结果，如果结果为 true，则返回 False	<code>not(x > 3 and x < 10)</code>	

Python 身份运算符

身份运算符用于比较对象，不是比较它们是否相等，但如果它们实际上是同一个对象，则具有相同的内存位置：

运算符	描述	实例	
is	如果两个变量是同一个对象，则返回 true。	<code>x is y</code>	
is not	如果两个变量不是同一个对象，则返回 true。	<code>x is not y</code>	

第二章 第二节 Python 运算符

Python 成员运算符

成员资格运算符用于测试序列是否在对象中出现：

运算符	描述	实例	
in	如果对象中存在具有指定值的序列，则返回 True。	x in y	
not in	如果对象中不存在具有指定值的序列，则返回 True。	x not in y	

Python 位运算符

位运算符用于比较（二进制）数字：

运算符	描述	实例
&	AND	如果两个位均为 1，则将每个位设为 1。
	OR	如果两位中的一位为 1，则将每个位设为 1。
^	XOR	如果两个位中只有一位为 1，则将每个位设为 1。
~	NOT	反转所有位。
<<	Zero fill left shift	通过从右侧推入零来向左移动，推掉最左边的位。
>>	Signed right shift	通过从左侧推入最左边的位的副本向右移动，推掉最右边的位。