

目标

掌握常用运算符的作用

运算符的分类

- 算数运算符
- 赋值运算符
- 复合赋值运算符
- 比较运算符
- 逻辑运算符

1. 算数运算符

运算符	描述	实例
+	加	1 + 1 输出结果为 2
-	减	1-1 输出结果为 0
*	乘	2 * 2 输出结果为 4
/	除	10 / 2 输出结果为 5
//	整除	9 // 4 输出结果为2
%	取余	9 % 4 输出结果为 1
**	指数	2 ** 4 输出结果为 16，即 2 * 2 * 2 * 2
()	小括号	小括号用来提高运算优先级，即 (1 + 2) * 3 输出结果为 9

注意：

- 混合运算优先级顺序：() 高于 ** 高于 * / // % 高于 + -

2. 赋值运算符

运算符	描述	实例
=	赋值	将 = 右侧的结果赋值给等号左侧的变量

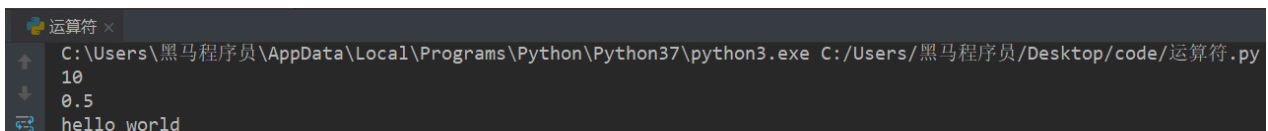
- 单个变量赋值

```
1 num = 1
2 print(num)
```

- 多个变量赋值

```
1 num1, float1, str1 = 10, 0.5, 'hello world'
2 print(num1)
3 print(float1)
4 print(str1)
```

结果如下：

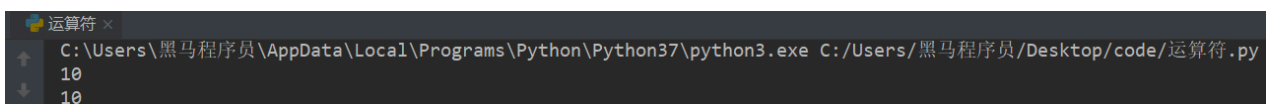


```
运算符 ×
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe C:/Users/黑马程序员/Desktop/code/运算符.py
10
0.5
hello world
```

- 多变量赋相同值

```
1 a = b = 10
2 print(a)
3 print(b)
```

结果如下：



```
运算符 ×
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe C:/Users/黑马程序员/Desktop/code/运算符.py
10
10
```

3. 复合赋值运算符

运算符	描述	实例
+=	加法赋值运算符	<code>c += a</code> 等价于 <code>c = c + a</code>
-=	减法赋值运算符	<code>c -= a</code> 等价于 <code>c = c - a</code>
*=	乘法赋值运算符	<code>c *= a</code> 等价于 <code>c = c * a</code>
/=	除法赋值运算符	<code>c /= a</code> 等价于 <code>c = c / a</code>
//=	整除赋值运算符	<code>c //= a</code> 等价于 <code>c = c // a</code>
%=	取余赋值运算符	<code>c %= a</code> 等价于 <code>c = c % a</code>
**=	幂赋值运算符	<code>c **= a</code> 等价于 <code>c = c ** a</code>

```
1 a = 100
2 a += 1
3 # 输出101 a = a + 1, 最终a = 100 + 1
4 print(a)
```

```

5
6 b = 2
7 b *= 3
8 # 输出6 b = b * 3,最终b = 2 * 3
9 print(b)
10
11 c = 10
12 c += 1 + 2
13 # 输出13, 先算运算符右侧1 + 2 = 3, c += 3 , 推导出c = 10 + 3
14 print(c)

```

4. 比较运算符

比较运算符也叫关系运算符，通常用来判断。

运算符	描述	实例
==	判断相等。如果两个操作数的结果相等，则条件结果为真(True)，否则条件结果为假(False)	如a=3,b=3, 则 (a == b) 为 True
!=	不等于。如果两个操作数的结果不相等，则条件为真(True)，否则条件结果为假(False)	如a=3,b=3, 则 (a == b) 为 True 如a=1,b=3, 则(a != b) 为 True
>	运算符左侧操作数结果是否大于右侧操作数结果，如果大于，则条件为真，否则为假	如a=7,b=3, 则(a > b) 为 True
<	运算符左侧操作数结果是否小于右侧操作数结果，如果小于，则条件为真，否则为假	如a=7,b=3, 则(a < b) 为 False
>=	运算符左侧操作数结果是否大于等于右侧操作数结果，如果大于，则条件为真，否则为假	如a=7,b=3, 则(a < b) 为 False 如a=3,b=3, 则(a >= b) 为 True
<=	运算符左侧操作数结果是否小于等于右侧操作数结果，如果小于，则条件为真，否则为假	如a=3,b=3, 则(a <= b) 为 True

```

1 a = 7
2 b = 5
3 print(a == b) # False
4 print(a != b) # True
5 print(a < b)  # False
6 print(a > b)  # True
7 print(a <= b) # False
8 print(a >= b) # True

```

5. 逻辑运算符

运算符	逻辑表达式	描述	实例
and	x and y	布尔"与": 如果 x 为 False, x and y 返回 False, 否则它返回 y 的值。	True and False, 返回 False。
or	x or y	布尔"或": 如果 x 是 True, 它返回 True, 否则它返回 y 的值。	False or True, 返回 True。
not	not x	布尔"非": 如果 x 为 True, 返回 False 。如果 x 为 False, 它返回 True。	not True 返回 False, not False 返回 True

```
1 a = 1
2 b = 2
3 c = 3
4 print((a < b) and (b < c)) # True
5 print((a > b) and (b < c)) # False
6 print((a > b) or (b < c))  # True
7 print(not (a > b))         # True
```

5.1 拓展

数字之间的逻辑运算

```
1 a = 0
2 b = 1
3 c = 2
4
5 # and运算符, 只要有一个值为0, 则结果为0, 否则结果为最后一个非0数字
6 print(a and b) # 0
7 print(b and a) # 0
8 print(a and c) # 0
9 print(c and a) # 0
10 print(b and c) # 2
11 print(c and b) # 1
12
13 # or运算符, 只有所有值为0结果才为0, 否则结果为第一个非0数字
14 print(a or b) # 1
15 print(a or c) # 2
16 print(b or c) # 1
```

总结

- 算数运算的优先级
 - 混合运算优先级顺序: `()` 高于 `**` 高于 `*` `/` `//` `%` 高于 `+` `-`
- 赋值运算符
 - `=`
- 复合赋值运算符
 - `+=`
 - `-=`
 - 优先级
 1. 先算复合赋值运算符右侧的表达式
 2. 再算复合赋值运算的算数运算
 3. 最后算赋值运算
- 比较运算符
 - 判断相等: `==`
 - 大于等于: `>=`
 - 小于等于: `<=`
 - 不等于: `!=`
- 逻辑运算符
 - 与: `and`
 - 或: `or`
 - 非: `not`