

# 赋值运算符

- 基本赋值运算符

运算符	描述	实例
=	赋值运算符	把 = 号右边的结果 赋给 左边的变量，如 num = 1 + 2 * 3，结果num的值为7

```
# 单个变量赋值
>>> num = 10
>>> num
10

# 同时为多个变量赋值(使用等号连接)
>>> a = b = 4
>>> a
4
>>> b
4
>>>

# 多个变量赋值(使用逗号分隔)
>>> num1, f1, str1 = 100, 3.14, "hello"
>>> num1
100
>>> f1
3.14
>>> str1
"hello"

# 语法错误, 值的个数超过变量的个数, 赋值失败
>>> num1,num2=(1,2,3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: too many values to unpack (expected 2)

# 语法错误, 变量的个数超过值得个数, 赋值失败
>>> num1,num2,num3,num4=(1,2,3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: not enough values to unpack (expected 4, got 3)
```

- 复合赋值运算符

运算符	描述	实例
<code>+=</code>	加法赋值运算符	<code>c += a</code> 等效于 <code>c = c + a</code>
<code>-=</code>	减法赋值运算符	<code>c -= a</code> 等效于 <code>c = c - a</code>
<code>*=</code>	乘法赋值运算符	<code>c *= a</code> 等效于 <code>c = c * a</code>
<code>/=</code>	除法赋值运算符	<code>c /= a</code> 等效于 <code>c = c / a</code>
<code>//=</code>	取整除赋值运算符	<code>c //= a</code> 等效于 <code>c = c // a</code>
<code>%=</code>	取模赋值运算符	<code>c %= a</code> 等效于 <code>c = c % a</code>
<code>**=</code>	幂赋值运算符	<code>c **= a</code> 等效于 <code>c = c ** a</code>

# 示例: +=

```
>>> a = 100
>>> a += 1 # 相当于执行 a = a + 1
>>> a
101
```

# 示例: \*=

```
>>> a = 100
>>> a *= 2 # 相当于执行 a = a * 2
>>> a
200
```

# 示例: \*=, 运算时, 符号右侧的表达式先计算出结果, 再与左边变量的值运算

```
>>> a = 100
>>> a *= 1 + 2 # 相当于执行 a = a * (1+2)
>>> a
300
```

注意: 赋值运算符是从右往左运算, 将等号右边的值赋值给等号左边, 所以, **等号的左边一定不能是常量或者表达式。**