

在Python2中： $7 / 2 == 3$ ，即整数除以整数，得到的还是整数。这和C语言是一致的，是计算机的思维方式。

在Python3中： $7 / 2 == 3.5$ ，即整数除以整数，得到的是浮点数。这和人类的思维是一致的。

Python2.6和Python3.0中对除法操作的异同

(<https://blog.csdn.net/u014291497/article/details/50984626>)

X / Y 类型：

在Python2.6或者之前，这个操作对于整数运算会省去小数部分，而对于浮点数运算会保持小数部分；

在Python3.0中变成真除法（无论任何类型都会保持小数部分，即使整除也会表示为浮点数形式）。

$X // Y$ 类型：

Floor除法：在Python 2.2中新增的操作，在Python2.6和Python3.0中均能使用，这个操作不考虑操作对象的类型，总是省略小数部分，剩下最小的能整除的整数部分。

在Python 2.6中， $' / '$ 执行传统除法，如果操作数都是整数的话，执行截断的整数除法（即对于结果只保留整数部分），否则，执行浮点除法（保留余数）； $' //'$ 执行Floor除法，与Python3.0一样，对于整数执行截断除法，浮点数执行浮点除法。

在Python 3.0中， $' / '$ 总是执行真除法，不管操作数的类型，都会返回包含任何余数的浮点结果； $' //'$ 执行Floor除法，截除掉余数并且针对整数操作数返回一个整数，如果有任何一个操作数是浮点数，则返回一个浮点数。