7.2 只接受关键字参数的函数¶

问题¶

你希望函数的某些参数强制使用关键字参数传递

解决方案¶

将强制关键字参数放到某个*参数或者单个*后面就能达到这种效果。比如：

```
def recv(maxsize, *, block):
    'Receives a message'
    pass

recv(1024, True) # TypeError
recv(1024, block=True) # Ok
```

利用这种技术，我们还能在接受任意多个位置参数的函数中指定关键字参数。比如：

```
def minimum(*values, clip=None):
    m = min(values)
    if clip is not None:
        m = clip if clip > m else m
    return m

minimum(1, 5, 2, -5, 10) # Returns -5
minimum(1, 5, 2, -5, 10, clip=0) # Returns 0
```

讨论¶

很多情况下，使用强制关键字参数会比使用位置参数表意更加清晰，程序也更加具有可读性。 例如，考虑下如下一个函数调用：

```
msg = recv(1024, False)
```

如果调用者对recv函数并不是很熟悉，那他肯定不明白那个False参数到底来干嘛用的。 但是，如果代码变成下面这样子的话就清楚多了：

```
msg = recv(1024, block=False)
```

另外，使用强制关键字参数也会比使用**kwargs参数更好，因为在使用函数help的时候输出也会更容易理解：

```
>>> help(recv)
Help on function recv in module __main__:
recv(maxsize, *, block)
    Receives a message
```

强制关键字参数在一些更高级场合同样也很有用。 例如，它们可以被用来在使用*args和**kwargs参数作为输入的函数中插入参数，9.11小节有一个这样的例子。