\_\_future\_\_模块：Python 3.x引入了一些与Python 2不兼容的关键字和特性，在Python 2中，可以通过内置的\_\_future\_\_模块导入这些新内容

|  |  |
| --- | --- |
| Python2.x | Python3.x |
| Print是语句，不需要加入 () 进行调用。Print默认不换行输出。 | Print是函数调用，需要加入 () 进行调用。 Print默认换行输出，除非加入end=’’, 即是说最后加入的是空白字符，而不是默认回车。 |
| /除法默认为整除，向下取整。需要除出小数的时候，则需要在分母或者分子加入小数点。 | /除法默认为正常除法，将会得出小数。不需要特意加入小数点。整除需要使用// |
| Python2字符有两种类型，str和Unicode.默认建立为str类型，并且采用ascii编码 | Python3字符有两种类型，bytes和str.默认建立为str类型，并且采用Unicode编码 |
| range(x)函数返回一个x长度数组，xrange(5) 返回一个5长度的迭代器。（在循环迭代时，都可以循环，区别是range()直接生成一个长度为x的数组，而xrange()时使用时候一个个创建，节省了瞬时的内存用量） | Xrange() 函数取消，只剩下range()函数。不过range()函数的实现是python2的xrange()函数。所有Python3中range()函数返回的是迭代器，要再使用list()变为数组。 |
| input()获取用户输入的时候，保存为数字  raw\_input获取用户输入的时候，保存为字符串 | 只有一个函数为input(),保存为字符串 |
| Python2中对整数，有int和long类型。且int类型最大值不能超过sys.maxint，这是数值与平台相关。可以在数字的末尾附上L来定义长整形 | Python3里面只有一个int类型，长度与Python2中的long类型范围一样 |
| Python2支持<>作为! =同义词 | Python3不等于只能为! = |
| Python2可以以has\_key()函数判断键值是否存在与字典key当中 | Python3只能使用x in dict.keys()来判断 |
| Python2中的httplib, cookie, cookielib在python3中合成为http.client, http.cookie, http.cookiejar中 | |
| 包内的相对导入  包是由一组相关联的模块共同组成的单个实体。  在python２的时候，为了实现同一个包内模块的相互引用，你会使用import foo或者from foo import Bar。Python2解释器会先在当前目录里搜索foo.py，然后再去python搜索路径(sys.path)搜索。  在python3里这个过程有一点不同。Python3不会首先在当前路径搜索，它会直接在Python的搜索路径里寻找。如果想要包里的一个模块导入包的另一个模块，需要显式的提供两个模块的相对路径。  所以需要from .foo import Bar | |
| Filter函数以及map函数返回一个列表 | Filter函数以及map函数返回一个迭代器 |
| Python2捕获异常为：  try:  ….  Except ImportError,e:  pass | Python3捕获异常：  try:  ….  Except ImportError as e:  pass |
| Python2支持mysql-python | Python3不支持mysql-python  Python3使用mysqlclient |
| Python2可以使用commands包 | Python3去除commands包 |
| Python的编译文件直接生成为.pyc文件 | Python3会将编译文件集中放在\_\_pycache\_\_文件夹中 |
|  | Python3的telnetlib的read\_until, 以及read\_until方法返回的内容，write方法需要bytes对象。 |
| python2中有urllib、urllib2、urlparse，但在python3中这些全部都被整合到了urllib中。urllib和urllib2中的内容整合进了urllib.request模块中，urlparse整合进了urllib.parse中。python3中urllib中还包括response、error和robotparse这些子模块。 | |

🡪Python2和Python3的print都会自动换行