Python time tzset()方法 Python IDE

Python2.x与3.x版本区别

Python的3.0版本，常被称为Python 3000，或简称Py3k。相对于Python的早期版本，这是一个较大的升级。

为了不带入过多的累赘，Python 3.0在设计的时候没有考虑向下相容。

许多针对早期Python版本设计的程式都无法在Python 3.0上正常执行。

为了照顾现有程式，Python 2.6作为一个过渡版本，基本使用了Python 2.x的语法和库，同时考虑了向Python 3.0的迁移，允许使用部分Python 3.0的语法与函数。

新的Python程式建议使用Python 3.0版本的语法。

除非执行环境无法安装Python 3.0或者程式本身使用了不支援Python 3.0的第三方库。目前不支援Python 3.0的第三方库有Twisted, py2exe, PIL等。

大多数第三方库都正在努力地相容Python 3.0版本。即使无法立即使用Python 3.0，也建议编写相容Python 3.0版本的程式，然后使用Python 2.6, Python 2.7来执行。

Python 3.0的变化主要在以下几个方面:

print 函数

print语句没有了，取而代之的是print()函数。 Python 2.6与Python 2.7部分地支持这种形式的print语法。在Python 2.6与Python 2.7里面，以下三种形式是等价的：

```
print "fish"
print ("fish") #注意print后面有个空格
print("fish") #print()不能带有任何其它参数
```

然而，Python 2.6实际已经支持新的print()语法：

```
from __future__ import print_function
print("fish", "panda", sep=', ')
```

Unicode

Python 2 有 ASCII str() 类型，unicode() 是单独的，不是 byte 类型。

现在， 在 Python 3，我们最终有了 Unicode (utf-8) 字符串，以及一个字节类：byte 和 bytearrays。

由于 Python3.X 源码文件默认使用utf-8编码，这就使得以下代码是合法的：

```
>>> 中国 = 'china'
>>>print(中国)
china
```

Python 2.x

```
>>> str = "我爱北京天安门"
>>> str
'\xe6\x88\x91\xe7\x88\xb1\xe5\x8c\x97\xe4\xba\xac\xe5\xa4\xa9\xe5\xae\x89\xe9\x97\xa8'
>>> str = u"我爱北京天安门"
>>> str
u'\u6211\u7231\u5317\u4eac\u5929\u5b89\u95e8'
```

Python 3.x

```
>>> str = "我爱北京天安门"
>>> str
'我爱北京天安门'
```

除法运算

Python中的除法较其它语言显得非常高端，有套很复杂的规则。Python中的除法有两个运算符，/和//

首先来说/除法:

在python 2.x中/除法就跟我们熟悉的大多数语言，比如Java啊C啊差不多，整数相除的结果是一个整数，把小数部分完全忽略掉，浮点数除法会保留小数点的部分得到一个浮点数的结果。

在python 3.x中/除法不再这么做了，对于整数之间的相除，结果也会是浮点数。

Python 2.x:

```
>>> 1 / 2
0
>>> 1.0 / 2.0
0.5
```

Python 3.x:

```
>>> 1/2
0.5
```

而对于//除法，这种除法叫做floor除法，会对除法的结果自动进行一个floor操作，在python 2.x和python 3.x中是一致的。

python 2.x:

```
>>> -1 // 2
-1
```

python 3.x:

```
>>> -1 // 2
-1
```

注意的是并不是舍弃小数部分，而是执行 floor 操作，如果要截取整数部分，那么需要使用 math 模块的 trunc 函数

python 3.x:

```
>>> import math
>>> math.trunc(1 / 2)
0
>>> math.trunc(-1 / 2)
0
```

异常

在 Python 3 中处理异常也轻微的改变了，在 Python 3 中我们现在使用 as 作为关键词。

捕获异常的语法由 **except exc, var** 改为 **except exc as var**。

使用语法except (exc1, exc2) as var可以同时捕获多种类别的异常。 Python 2.6已经支持这两种语法。

- 1. 在2.x时代，所有类型的对象都是可以被直接抛出的，在3.x时代，只有继承自BaseException的对象才可以被抛出。
- 2. 2.x raise语句使用逗号将抛出对象类型和参数分开，3.x取消了这种奇葩的写法，直接调用构造函数抛出对象即可。

在2.x时代，异常在代码中除了表示程序错误，还经常做一些普通控制结构应该做的事情，在3.x中可以看出，设计者让异常变的更加专一，只有在错误发生的情况才能去用异常捕获语句来处理。

---

xrange

在 Python 2 中 xrange() 创建迭代对象的用法是非常流行的。比如： for 循环或者是列表/集合/字典推导式。

这个表现十分像生成器（比如。"惰性求值"）。但是这个 xrange-iterable 是无穷的，意味着你可以无限遍历。

由于它的惰性求值，如果你不得仅仅不遍历它一次，xrange() 函数 比 range() 更快（比如 for 循环）。尽管如此，对比迭代一次，不建议你重复迭代多次，因为生成器每次都从头开始。

在 Python 3 中，range() 是像 xrange() 那样实现以至于一个专门的 xrange() 函数都不再存在（在 Python 3 中 xrange() 会抛出命名异常）。

```
import timeit

n = 10000
def test_range(n):
    return for i in range(n):
        pass

def test_xrange(n):
    for i in xrange(n):
        pass
```

Python 2

```
print 'Python', python_version()

print '\ntiming range()'
%timeit test_range(n)

print '\n\ntiming xrange()'
%timeit test_xrange(n)

Python 2.7.6

timing range()
1000 loops, best of 3: 433 µs per loop


timing xrange()
1000 loops, best of 3: 350 µs per loop
```

Python 3

```
print('Python', python_version())

print('\ntiming range()')
%timeit test_range(n)

Python 3.4.1

timing range()
1000 loops, best of 3: 520 µs per loop
```

```
print(xrange(10))
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-5-5d8f9b79ea70> in <module>()
----> 1 print(xrange(10))

NameError: name 'xrange' is not defined
```

八进制字面量表示

八进制数必须写成0o777，原来的形式0777不能用了；二进制必须写成0b111。

新增了一个bin()函数用于将一个整数转换成二进制字串。 Python 2.6已经支持这两种语法。

在Python 3.x中，表示八进制字面量的方式只有一种，就是0o1000。

python 2.x

```
>>> 0o1000
512
>>> 01000
512
```

python 3.x

```
>>> 01000
  File "<stdin>", line 1
    01000
        ^
SyntaxError: invalid token
>>> 0o1000
512
```

不等运算符

Python 2.x中不等于有两种写法 != 和 <>

Python 3.x中去掉了<>, 只有!=一种写法，还好，我从来没有使用<>的习惯

去掉了repr表达式``

Python 2.x 中反引号``相当于repr函数的作用

Python 3.x 中去掉了``这种写法，只允许使用repr函数，这样做的目的是为了使代码看上去更清晰么？不过我感觉用repr的机会很少，一般只在debug的时候才用，多数时候还是用str函数来用字符串描述对象。

```
def sendMail(from_: str, to: str, title: str, body: str) -> bool:
    pass
```

多个模块被改名（根据PEP8）

旧的名字	新的名字
_winreg	winreg
ConfigParser	configparser
copy_reg	copyreg
Queue	queue
SocketServer	socketserver

repr	reprlib

StringIO模块现在被合并到新的io模组内。 new, md5, gopherlib等模块被删除。 Python 2.6已经支援新的io模组。

httplib, BaseHTTPServer, CGIHTTPServer, SimpleHTTPServer, Cookie, cookielib被合并到http包内。

取消了exec语句，只剩下exec()函数。 Python 2.6已经支援exec()函数。

5.数据类型

1）Py3.X去除了long类型，现在只有一种整型——int，但它的行为就像2.X版本的long

2）新增了bytes类型，对应于2.X版本的八位串，定义一个bytes字面量的方法如下：

```
>>> b = b'china'
>>> type(b)
<type 'bytes'>
```

str对象和bytes对象可以使用.encode() (str -> bytes) or .decode() (bytes -> str)方法相互转化。

```
>>> s = b.decode()
>>> s
'china'
>>> b1 = s.encode()
>>> b1
b'china'
```

3）dict的.keys()、.items 和.values()方法返回迭代器，而之前的iterkeys()等函数都被废弃。同时去掉的还有 dict.has_key()，用in替代它吧 。

← Python time tzset()方法 Python IDE →

```
raw_input( "提示信息" )
```

改为:

```
input( "提示信息" )
```

在python2.x中raw_input()和input( )，两个函数都存在，其中区别为：

- raw_input()---将所有输入作为字符串看待，返回字符串类型
- input()-----只能接收"数字"的输入，在对待纯数字输入时具有自己的特性，它返回所输入的数字的类型（int, float ）

在python3.x中raw_input()和input( )进行了整合，去除了raw_input()，仅保留了input()函数，其接收任意任性输入，将所有输入默认为字符串处理，并返回字符串类型。

heisenbug601 1年前 (2017-11-04)

map、filter 和 reduce

这三个函数号称是函数式编程的代表。在 Python3.x 和 Python2.x 中也有了很大的差异。

首先我们先简单的在 Python2.x 的交互下输入 map 和 filter,看到它们两者的类型是 built-in function(内置函数):

```
>>> map
<built-in function map>
>>> filter
<built-in function filter>
>>>
```

它们输出的结果类型都是列表:

```
>>> map(lambda x:x *2, [1,2,3])
[2, 4, 6]
>>> filter(lambda x:x %2 ==0,range(10))
[0, 2, 4, 6, 8]
>>>
```

但是在Python 3.x中它们却不是这个样子了：

```
>>> map
<class 'map'>
>>> map(print,[1,2,3])
<map object at 0x10d8bd400>
>>> filter
<class 'filter'>
>>> filter(lambda x:x % 2 == 0, range(10))
<filter object at 0x10d8bd3c8>
>>>
```

首先它们从函数变成了类，其次，它们的返回结果也从当初的列表成了一个可迭代的对象, 我们尝试用 next 函数来进行手工迭代:

```
>>> f =filter(lambda x:x %2 ==0, range(10))
>>> next(f)
0
>>> next(f)
2
>>> next(f)
4
>>> next(f)
6
>>>
```

对于比较高端的 reduce 函数，它在 Python 3.x 中已经不属于 built-in 了，被挪到 functools 模块当中。

坤少 1年前 (2018-02-26)