Python2.x与Python3.x的区别

- 性能
  - py3.x起始比py2.x效率低，但是py3.x有极大的优化空间，效率正在追赶
- 编码
  - py3.x原码文件默认使用utf-8编码，使得变量名更为广阔
    - python3

```
>>> 中国="CHI"
>>> 中国
'CHI'
>>>
```

- 语法
  - 去除了<>，改用!=
    - python2

```
>>> 1<>2
True
>>> 1!=2
True
>>>
```

    - python3

```
>>> 1<>2
  File "<stdin>", line 1
    1<>2
      ^
SyntaxError: invalid syntax
>>> 1!=2
True
>>>
```

  - 加入as和with关键字，还有True,False,None
  - 整型触发返回浮点数，整除请使用//
    - python2

```
>>> 5/3
1
>>> 5.0/3
1.6666666666666667
>>>
```

    - python3

```
6666667
```

  - 加入nonlocal语句
  - 去除print语句，加入print()函数

```
2.X: print "The answer is", 2*2
3.X: print("The answer is", 2*2)
2.X: print x,                     # 使用逗号结尾禁止换行
3.X: print(x, end=" ")            # 使用空格代替换行
2.X: print                        # 输出新行
3.X: print()                      # 输出新行
2.X: print >>sys.stderr, "fatal error"
3.X: print("fatal error", file=sys.stderr)
2.X: print (x, y)                 # 输出repr((x, y))
3.X: print((x, y))                # 不同于print(x, y)!
```

  - 去除了raw_input，加入input()函数
  - 新的super()，可以不再给super()传参数

```
class C(object):
    def __init__(self, a):
        print('C', a)
class D(C):
    def __init(self, a):
        super().__init__(a) # 无参数调用super()
```

  - 改变了顺序操作符的行为，例如x<y，当x和y类型不匹配时抛出TypeError而不是返回随即的 bool值
    - python2

```
>>> 2<"4"
True
>>>
```

    - python3

```
>>> 2<"4"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: '<' not supported between instances of 'int' and 'str'
>>>
```

  - 新式的8进制字变量
    - python2

```
>>> 0666
438
>>>
```

    - python3

```
>>> 0666
  File "<stdin>", line 1
    0666
       ^
SyntaxError: invalid token
>>> 0o666
438
>>>
```

- 字符串和字节串
  - python2
    - 字符串以 8-bit 字符串存储
  - python3
    - 字符串以 16-bit Unicode 字符串存储
    - 现在字符串只有str一种类型
- 数据类型
  - 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
  - 新增了bytes类型，对应于2.X版本的八位串

```
>>> b = b'china'
>>> b
b'china'
>>> type(b)
<class '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
- 面向对象
  - 引入抽象基类
- 异常
  - 所以异常都从 BaseException继承，并删除了StardardError
  - python2

```
try:
    ......
except Exception, e:
    ......
```

  - python3

```
try:
    ......
except Exception as e :
    ......
```

- 其他
  - xrange() 改名为range()，要想使用range()获得一个list，必须显式调用
    - python2

```
>>> list(xrange(5))
[0, 1, 2, 3, 4]
>>>
```

    - python3

```
>>> list(range(5))
[0, 1, 2, 3, 4]
>>>
```

  - file类被废弃
    - python2

```
>>> file
<type 'file'>
>>>
```

      - 打开文件
        - file(path)
        - open(path)
    - python3

```
>>> file
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'file' is not defined
>>>
```

      - 打开文件 open(path)