Python 程序设计基础

Python Programming

内置异常类

- BaseException 类是所有内置异常类的父类，其他所有的内置异常类都直接或间接地继承自 BaseException 类。
  - Exception 是除了 SystemExit 、 KeyboardInterrupt 和 GeneratorExit 之外的其他所有内置异常类的父类。此外，所有用户自定义异常类都应该派生自 Exception 类。
  - ArithmeticError 类是各种算术运算错误引发的内置异常类（如 ZeroDivisionError 、 OverflowError 、 FloatingPointError ）的父类。
  - LookupError 类是集合、字典或序列的键或下标无效时引发的内置异常类（如 KeyError 、 IndexError ）的父类。

内置异常类

- 常用内置异常类如下：
  - AssertionError ，断言语句（ assert ）失败抛出此异常。
  - AttributeError ，访问无效的对象属性时抛出此异常。

```
>>> set1 = {"air", "fire", "earth", "water"}
>>> set1.sort()
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    set1.sort()
AttributeError: 'set' object has no attribute 'sort'
```

  - EOFError，从文件中读取数据到文件末尾时抛出此异常。
  - IndexError，索引超出序列范围时抛出此异常。

```
>>> lst = [1, 2, 3, 4]
>>> lst[4]
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    lst[4]
IndexError: list index out of range
```

内置异常类

- **KeyError ，在集合、字典中找不到键时抛出此异常。**

```
>>> set1 = {111, 222, 333}
>>> set1.remove(444)
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    set1.remove(444)
KeyError: 444
```

```
>>> b = a * 12
Traceback (most recent call last):
  File "<pyshell#13>", line 1, in <module>
    b = a * 12
NameError: name 'a' is not defined
```

- **始化）时抛出此异常。**

内置异常类

- **OverflowError ，浮点数算术运算的结果太大时抛出此异常。**

```
>>> a = 88.5
>>> b = 74353634
>>> a ** b
Traceback (most recent call last):
  File "<pyshell#17>", line 1, in <module>
    a ** b
OverflowError: (34, 'Result too large')
```

```
>>> print "Hello, World!"
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("Hello, World!")?
```

- **SyntaxError ，遇到语法错误时抛出此异常。**

```
>>> num = eval(123)
Traceback (most recent call last):
  File "<pyshell#21>", line 1, in <module>
    num = eval(123)
TypeError: eval() arg 1 must be a string, bytes or code object
```

内置异常类

- **ValueError ，遇到无效的值时抛出此异常。**

```
>>> x = int("3.4")
Traceback (most recent call last):
  File "<pyshell#23>", line 1, in <module>
    x = int("3.4")
ValueError: invalid literal for int() with base 10: '3.4'
```

```
>>> a = 12
>>> b = 0
>>> a % b
Traceback (most recent call last):
  File "<pyshell#27>", line 1, in <module>
    a % b
ZeroDivisionError: integer division or modulo by zero
```

- **数为 0 时抛出此异常。**