

day15: raise语句, 迭代器

笔记本: Python基础

创建时间: 2018/6/25 9:08

更新时间: 2018/7/2 19:38

作者: liuchang_0412@163.com

1.raise 语句: 生成一个错误, 让程序进入异常状态

语法:

raise 异常类型

raise 异常对象

示例: 见 raise.py

2.assert 语句: (断言语句)

语法:

assert 真值表达式, 错误数据 (通常是字符串)

作用:

当真值表达式返回错误时, 用错误数据创建一个AssertionError类型的错误, 抛出后进入异常状态 (等同于if 真值表达式==false: raise AssertionError("XXXX"))

示例:

见assert.py

3.迭代器 Iterator 和 生成器 generator

①迭代器 Iterator: 迭代器是指能用next(it)函数取值的对象 (实例)

说明:

用iter函数可返回一个可迭代对象的迭代器

迭代器是访问可迭代对象的一种方式

迭代器只能往前, 不能后退

示例:

```
L = [1,1,2,3,5]
```

```
it = iter(L)
```

```
next(it)  #1
```

```
next(it)  #1
```

```
next(it)  #2
```

```
next(it)  #3
```

```
next(it)  #5
```

```
next(it)  #StopIteration 异常
```

示例2:

见 iterator.py

作用:

iter(x)函数: 从一个对象x中返回一个迭代器, x必须是能提供一个迭代器的对象

next(it): 从迭代器it中获取下一个记录, 如果无法获取下一个记录, 则触发StopIteration异常

②生成器 generator: 生成器是能够提供迭代器的对象 (实例)

生成器函数：

含有yield语句的函数是生成器函数

yield语句：

yield 表达式

说明：

yield用于def函数中，目的是将此函数作为生成器函数使用

yield用于生成数据，供迭代器next(it)函数使用

示例：

见yield.py

说明：

生成器函数的调用将返回一个生成器对象，生成器对象是可迭代对象

在生成器函数内调用return会产生一个StopIteration异常

生成器函数与列表等可迭代对象相比可以降低内存，提高效率

range函数是一个生成器函数

#练习：

1.写一个生成器myodd(x)，传入一个终止值，可生成1,3,5,...,x奇数的一个生成器，用for语句打印这些奇数。

2.写一个生成器函数my_range(stop)来生成一系列整数，

要求：my_range功能与range功能完全相同，不允许调用range

4.生成器表达式：

语法：

(表达式 for 变量 in 可迭代对象 [if 真值表达式]) #if 语句可以省略

作用：

用推导式形式生成一个新的生成器（可迭代对象）

5.迭代工具函数：生成一个个性化的可迭代对象

zip(iter1 [,iter2 [...]])

返回一个zip对象，此对象用于生成一个元组，此元组的个数由最小的可迭代对象决定

示例：见zip.py

enumerate(iterable, [,start])

生成带索引的迭代器，返回的迭代器类型为索引-值对(index-value对)，默认索引从0开始，也可以用start指定。

示例：enumerate.py

#练习：

用生成器函数生成素数，给出起始值begin和终止值stop，生成器函数为：

def primes(begin, stop):

...

如果 [x for x in primes(10,20)]将得到列表：

6.文件操作:

文件是用于数据存储的单位, 通常用于长期存储数据

Linux/Unix下文件可分为普通文件和设备文件。

普通文件是由文件名和文件中的数据两部分组成的。

①文件的打开和关闭

文件需要在使用时, 先打开才能读写

在不需要读写文件时, 应及时关闭文件以释放系统资源

任何操作系统, 打开的文件数都有最大限制

②打开文件的函数:

`F=open("filename", mode = 'rt')` #用于打开文件, 返回此文件的操作对象,
如果打开失败则会触发错误, 抛出异常

模式字符串的含义:

'r' 以只读方式打开文件 (默认)

'w' 以只写方式打开文件, 删除原有文件的内容 (如果文件不存在, 则创建该文件并以只写方式打开文件)

'x' 创建一个新文件, 并以写模式打开文件, 如果文件存在则会产生
`FileExistsError`错误

'a' 以追加写方式打开一个文件, 如果有原文件, 则追加到原文件末尾

'b' 用二进制模式打开

't' 用文本文件模式打开 (默认)

'+' 为更新内容打开一个磁盘文件 (可读可写)

注: 缺省模式是 'rt'

③文件关闭的方法:

`F.close()` #关闭文件, 释放系统资源

④文本文件的常用方法:

`F.readline()` 读取一行数据, 如果到达文件尾则返回空行

`F.readlines(n)` 返回每行字符串的列表, n代表最大行数

`F.writelines(lines)` 写入字符串中的列表

`F.flush()` 把写入文件对象的缓存内容写入到磁盘

`F.write(x)` 将数据 (字符/字节) 写入到文件中

#练习:

1.将如下数据写入到文件data.txt中, 数据如下:

张某某 13845679876

李某某 13899999999

赵某某 13088888888

写程序读取数据,打印出姓名和电话号码。

2.写一个生成器函数, 用来生成斐波那契数列的前n个数。

```
def fibonacci(n):
```

```
....
```

例

```
for x in fibonacci(7):
```

```
    print (x, end= ' ') #1 1 2 3 5 8 13
```

3.写一个程序，输入任意多人的名字，电话号码，住址存入"info.txt"中，文件格式自定义，完成后查看文件格式是不是你想要的格式。

4.写一个程序，读入"info.txt"中的内容，以表格的形式打印这些内容。

5.为学生管理系统的程序添加保存学生信息和打开学生信息功能。