virtualenv：

$ pip3 install virtualenv

第一步，创建目录：

Mac:~ michael$ mkdir myproject

Mac:~ michael$ cd myproject/

Mac:myproject michael$

第二步，创建一个独立的Python运行环境，命名为venv：

Mac:myproject michael$ virtualenv --no-site-packages venv #不带任何第三方包

Mac:myproject michael$ source venv/bin/activate

进入venv环境

(venv)Mac:myproject michael$

退出当前的venv环境，使用deactivate命令：

(venv)Mac:myproject michael$ deactivate

Mac:myproject michael$

select操作时加update锁

sqlalchemy

If you are using the ORM, try the with\_for\_update function:

foo = session.query(Foo).filter(Foo.id==1234).with\_for\_update().one()

# this row is now locked

foo.name = 'bar'

session.add(foo)

session.commit()

# this row is now unlocked

Python supports closures: functions that refer to variables from the scope in which they were defined.

When you reference a variable in an expression, the Python interpreter will traverse the

scope to resolve the reference in this order:

1. The current function’s scope

2. Any enclosing scopes (like other containing functions)

3. The scope of the module that contains the code (also called the global scope)

4. The built-in scope (that contains functions like len and str)

If none of these places have a defined variable with the referenced name, then a

NameError exception is raised.

Assigning a value to a variable works differently. If the variable is already defined in the current scope, then it will just take on the new value. If the variable doesn’t exist in the current scope, then Python treats the assignment as a variable definition. The scope of the newly defined variable is the function that contains the assignment.

PIP install 时 报需要SSL模块

在你的“C:\Users\你的用户名\”目录下创建“pip”目录，“pip”目录下创建“pip.ini”文件（注意：以UTF-8 无BOM格式编码）；

“pip.ini”文件内容：

[global]

index-url=http://mirrors.aliyun.com/pypi/simple/

[install]

trusted-host=mirrors.aliyun.com

国内镜像

http://pypi.douban.com/simple/ 豆瓣

http://mirrors.aliyun.com/pypi/simple/ 阿里

http://pypi.hustunique.com/simple/ 华中理工大学

http://pypi.sdutlinux.org/simple/ 山东理工大学

http://pypi.mirrors.ustc.edu.cn/simple/ 中国科学技术大学

https://pypi.tuna.tsinghua.edu.cn/simple 清华

以名字访问元组内容：namedtuple

from collections import namedtuple

Student = namedtuple('student', ['name', 'age', 'sex', 'email'])

s = Student('Jim', 18, 'male', 'jim@gmail.com')

s.name #Jim

isinstance(s,tuple) #True

统计字频：Counter

from collections import Counter

有序字典：OrderedDict

from collections import OrderedDict

双端循环队列：deque

from collections import deque

python对象与文件的保存和加载：pickle

pickle.dump(obj, open('file','w'))

obj = pickle.load(open('file'))

可迭代对象(iterable)、迭代器(iterator):

iter(obj) 产生 iterator #obj is iterable, obj has \_\_iter\_\_ or \_\_getitem\_\_

from collections import Iterable, Iterator

实现可迭代对象：

class MyIterable(Iterable):

def \_\_init\_\_(self,...):

def \_\_iter\_\_(self,...): return MyIterator(self,...)

实现迭代器：

class MyIterator(Iterator):

def \_\_init\_\_(self,...):

def next(self,...):

for语句迭代多个可迭代对象

并行：zip

串行：itertools.chain

拆分含有多种分隔符的字符串：

1、连续使用split()方法

2、使用re.split(r'[...]',str)

变量作用域：

Python 不要求声明变量，但是假定在函数定义体中赋值的变量是局部变量，如果在函数中赋值时想让解释器把赋值的变量当成全局变量，要使用 global 声明该赋值的变量。

闭包：

闭包是一种函数，它会保留定义函数时存在的自由变量的绑定，这样调用函数时，虽然定义作用域不可用了，但是仍能使用那些绑定。注意，只有嵌套在其他函数中的函数才可能需要处理不在全局作用域中的外部变量。

avg.\_\_code\_\_.co\_varnames #avg是函数对象，co\_varnames 局部变量名称

avg.\_\_code\_\_.co\_freevars #co\_freevars 自由变量名称

avg.\_\_closure\_\_中的各个元素对应于avg.\_\_code\_\_.co\_freevars 中的一个名称。这些元素是 cell 对象，有个 cell\_contents 属性，保存着真正的值。avg.\_\_closure\_\_[0].cell\_contents

但是对数字、字符串、元组等不可变类型来说，只能读取，不能更新。如果尝试重新绑定，例如 count = count + 1，其实会隐式创建局部变量 count。为保持变量为自由变量，Python 3 引入了 nonlocal 声明。它的作用是把变量标记为自由变量，即使在函数中为变量赋予新值了，也会变成自由变量。如果为 nonlocal 声明的变量赋予新值，闭包中保存的绑定会更新。

例如：def make\_averager():

count = 0

total = 0

series = []

def averager(new\_value):

nonlocal count, total

count += 1

total += new\_value

series.append(new\_value)

return total / count

return averager

可变类型不需要设置成nonlocal

装饰器：

1. 何时执行装饰器：它们在被装饰的函数定义之后立即运行。这通常是在导入时（即 Python 加载模块时），函数装饰器在导入模块时立即执行，而被装饰的函数只在明确调用时运行。

注册装饰器：很多 Python Web 框架使用这样的装饰器把函数添加到某种中央注册处，例如把 URL 模式映射到生成 HTTP 响应的函数上的注册处。这种注册装饰器可能会也可能不会修改被装饰的函数。

不过，多数装饰器会修改被装饰的函数。通常，它们会定义一个内部函

数，然后将其返回，替换被装饰的函数。使用内部函数的代码几乎都要

靠闭包才能正确运作。