## day1

### 认识Python\_1

Python 是一种面向对象（OOP）的解释型计算机程序设计语言。

### 认识Python\_2

优点：简单易学、免费开源、丰富的第三方库、可以移植、面向对象

缺点：运行速度慢、代码不能加密、强制缩进、GIL全局解释器锁

就业方向：Web后台开发、爬虫开发、数据分析、运维开发、机器学习、人工智能

### 第一个Python程序

### 注释

### Pycharm快速体验

### 变量和数据类型\_01

1. 变量的定义：变量名=数据
2. Python基本数据类型

数字（num）：int（有符号整数）、long（长整形）、float（浮点数）、complex（复数）、bool（布尔值True、False）

字符串（str）

字典（dict）

元组（Tuple）

列表（list）

### 变量和数据类型\_02

### 算数和比较运算

### 逻辑运算符

### 赋值运算符

### Python的输入与输出

## day2

### 复习和课程目标

### 流程控制结构

### 选择流程1

### 选择流程2

### 循环流程\_1

### 循环流程\_2

### 循环流程\_3

### 循环流程\_4

### 小结

### 第二天作业

### 复习

## day3-列表、字典、元组

### 字符串操作

### 列表操作

### 元组操作

### 字典操作

### 公用方法

## day4-函数

### 复习和目标

### 初识函数

### 函数参数

### 关键字可变参数

### 函数返回值

### 函数嵌套调用

### 函数作业

## day5

### 复习

### 全局变量和局部变量

当全局变量和局部变量出现重复，程序会优先执行函数内部定义的变量。

### 函数参数引用传值

### 匿名函数

Python中使用lambda关键字创造匿名函数，所谓匿名即这个函数没有名字不用def关键字创建标准的函数。

lambda 参数1,参数2,参数3：执行代码语句特点：  
（1）使用lambda关键字去创建函数  
（2）没有名字的函数  
（3）匿名函数冒号后面的表达式有且只有一个，注意：是表达式，而不是语句  
（4）匿名函数自带return，而这个return点结果就是表达式计算后的结果

### 递归函数

### 函数下小结

## day6

### 复习和学习目标

### 内置函数-数学运算

abs（）：绝对值

round（）：对浮点数取近似值

pow（）：求次方

divmod（）：求商和余数

max（）：求最大值

min（）：求最小值

sum（）：求和

eval（）：动态执行字符串表达式

### 内置函数-类型转换

int（）、float（）、str（）、ord（）、bool（）

chr（）：数字转字符

bin（）：转为二进制

hex（）：转十六进制

oct（）：

list（）：元组转列表

tuple（）：列表转元组

dict（）：创建字典

bytes（）：转为字节数组

### 序列操作\_1

all（）：用于判断给定的可迭代函数iterable中的所有元素是否都为True，所有元素都为True，结果才为True

any（）：用于判断给定的可迭代函数iterable中的所有元素是否都为False，有一个元素为True，返回True

sorted（）：对所有可迭代对象进行排序

sort（）：应用在list上的方法

reverse（）：反转、反向列表中的元素

range（）：创建一个整数列表，一般用在for循环中

zip（）：将可迭代的对象作为参数，将对象中对应的元素打包成一个个元组，然后返回由这些元组组成的列表

enumerate（）：用于一个可遍历的数据对象（如列表、元组或字符串）组合为一个索引序列，同时列出数据和数据下标，一般用在for循环当中

### 序列操作\_2

### 序列操作\_3

### 内置函数\_set

Set（集合）是Python中的一种数据类型，是一个无序且不重复的元素集合。

add（）：向集合添加元素

clear（）：清空集合中的元素

difference（）：两个集合的差集，a中存在，b中不存在

intersection（）：两个集合的交集，a、b中都存在

union（）：并集操作，a、b中包含的所有的元素

pop（）：移除数据，移除某个元素并获取那个参数，集合pop没有参数

discard（）：移除指定数据

update（）：更新

### 内置函数\_小结

### 内置函数作业

## day7-OOP

### 回顾和学习目标

### 面向对象介绍

### 类和对象

### 定义类和创建对象

### 实例方法和属性

实例方法：在类的内部，使用def关键字来定义，第一个参数默认是self【名字标识可以是其他名字，但是这个位置必须被占用】

类属性：在类的内部定义的变量

实例属性：在方法内部定义的【通过类似self.变量名】变量

### \_\_init\_\_

### 理解self

self和对象指向同一个内存地址，可以认为self就是对象的引用。

### 魔术方法\_1\_new

在python中，有一些内置好的特定的方法，方法名声\_\_xxx\_\_，在进行特定的操作时会自动被调用，这些方法称之为魔法方法。比如：

\_\_init\_\_()：初始化一个类，在创建实例对象为其赋值时使用

\_\_str\_\_()：在将对象转换成字符串 str (对象)测试的时候，打印对象的信息

\_\_new\_\_()：创建并返回一个实例对象，调用了一次，就会得到一个对象

\_\_class\_\_()：获得已知对象的类（对象.\_\_class\_\_）

\_\_del\_\_()：对象在程序运行结束后进行对象销毁的时候调用这个方法，来释放资源

### 魔术方法\_\_2

### 案例-双人对决(决战紫禁之巅)

### 小结及课后作业

## day8-OOP2

### 课程内容介绍

### 析构方法

1. 当一个对象被删除或者被销毁时，Python解释器也会默认调用一个方法，这个方法为\_\_del\_\_()方法，也称为析构方法。
2. 当整个程序脚本执行结束后，会自动调用析构方法\_\_del\_\_()
3. 当对象被手动销毁时也会自动调用\_\_del\_\_()方法
4. 析构函数一般用于资源回收，利用\_\_del\_\_()方法销毁对象内存等资源

### 单继承

### 多继承\_1

### 多继承\_2

### 重写父类方法

所谓重写，就是子类中，有一个和父类相同名字的方法，在子类中的方法会覆盖掉父类中同名的方法。

### 调用父类方法

### 多态

### 类属性和实例属性\_1

### 类属性和实例属性\_2

### 类方法和静态方法\_1

### 类方法和静态方法\_2

1. 类方法的第一个参数是类对象cls，通过cls引用类对象的属性和方法。
2. 实例方法的第一个参数是实例对象self，通过self引用的可能是类属性、也有可能是实例属性（这个需要具体分析），不过在存在相同名称的类属性和实例属性的情况下，实例属性优先级更高。
3. 静态方法中不需要额外定义参数，因此在静态方法中引用类属性的话，必须通过类对象来引用。

## day9-OOP3

### 复习和目标

### 私有化属性\_1

为了更好的保存属性安全，切不能随意修改，将属性定义为私有属性，添加一个可调用的方法去访问。

私有化的实例属性是不能在外部直接使用的。私有属性是不能被子类继承的。

### 私有化属性\_2

### 私有化方法

### Property属性函数

### \_\_new\_\_和单例模式

创建并返回一个对象，如果\_\_new\_\_只调用了一次，就会得到一个对象。

### 异常处理\_1

AssertError：当assert语句失败时引发

AttributeError：当属性分配或引用失败时引发

EOFError：当input函数达到文件结束条件时引发

FloatingPointError：当浮点运算失败时引发

GeneratorExit：调用生成器的close方法时引发

ImportError：找不到导入的模块时引发

KeyError：在字典中找不到键时引发

KeyboardInterrupt：当用户按下断键（ctrl+C或delete）时引发

MemoryError：当操作耗尽时引发

NameError：当局部或全局范围内找不到变量时引发

NotImplementedError：由抽象方法提出

OSError：当系统操作导致系统相关错误时引发

OverflowError：当算数运算的结果太大而无法表示时引发

ReferenceError：使用弱引用代理访问垃圾收集的引用时引发

RuntimeError：当错误不属于任何其它类别时引发

StopInteration：函数引发，以指示迭代器不再返回任何项

SynatxError：遇到语法错误时由解析起引发

IndentationError：缩进不正确时引发

TabError：当缩进由不一致的制表符和空格组成时引发

SystemError：当解释器检测到内部错误时引发

SystemExit：由sys.exit()函数引发

TypeError：将函数或操作应用在类型不正确的对象时引发

UnboundLocalError：当在函数或方法中引用局部变量，但没有将值绑定到该变量时引发

UnicodeError：当发生与unicode相关的编码或解码错误时引发

UnicodeEncodeError：当编码过程中发生与unicode相关的错误时引发

UnicodeDecodeError：当解码过程中发生与unicode相关的错误时引发

UnicdoeTranslateError：翻译过程总发生与unicode相关的错误时引发

ValueError：当函数得到类型正确但值不正确的参数时引发

ZaroDivisionError：当除法或模运算的第二个操作数为零时引发

### 异常处理\_2

### 异常处理\_3

### 动态绑定属性和方法

动态语言：运行时可以改变其结构的语言，例如新的函数、对象、甚至代码可以被引进，已有的函数可以被删除或是其他结构上的变化。如php、javascript、python都是动态语言，C、C#、java是静态语言。

### \_\_slots\_\_

### 课程小结

### 课后作业

## day10-Planedemo

### 复习和目标

1. 搭建界面和键盘检测
2. 添加背景音乐
3. 运行优化
4. 控制玩具飞机-面向过程
5. 控制玩具飞机-面向对象
6. 玩家飞机发射子弹
7. 显示敌机
8. 敌机发射子弹
9. 抽象基类

### 明确需求和准备工作

plane pro需求的描述：

1、存在四个对象：我方飞机、敌方飞机、我方子弹、敌方子弹

2、功能：

背景音乐的添加

我方飞机可以移动（根据按键来控制的）

地方飞机也可以移动（随机的自动移动）

双方飞机都可以发射子弹

3、步骤

创建一个窗口

创建一个我方飞机，根据方向键左右移动

给我方飞机添加发射子弹的功能（按下空格键去发送）

创建一个敌人飞机

敌人飞机可以自由的移动

敌人飞机可以自动的发射子弹

### 创建主体窗和键盘检测

### 添加背景音乐和玩家图片

### 创建飞机对象和键盘控制函数

### 玩家发射子弹

### 敌机创建和随机发射子弹

### 优化结构

## day11-文件读写

### 复习和概要

### 文件读写\_1

1. 文件操作的一般步骤

打开文件：open（‘文件名称’, ’打开模式’） open（‘test.txt’, ’w’）

1. 读/写文件
2. 保存文件
3. 关闭文件

### 文件读写\_2

with上下文管理：with语句，不管在处理文件过程中是否发生异常，都能保证with语句执行完毕后已经关闭打开的文件句柄

### 案例：备份文件

### 文件定位\_1

1. tell( )：文件定位，指的是当前文件指针读取到的位置，光标位置，在读写文件的过程中，如果想知道当前的位置，可以使用tell( )来获取。
2. seek( )：如果在操作文件的过程，需要定位到其它位置进行操作，用seek( )。

seek（offset，from）有两个参数，offset：偏移量，单位字节，负数是往回偏移，正数是往前偏移，from：位置，0表示文件开头，1表示当前位置，2表示文件末尾。

### 文件定位\_2

### 文件定位\_3

### 模块导入方式

### Python-os模块的使用

getcwd()：获取当前目录

chdir()：切换目录

os.path.join(path1[,path2[,...]])：路径拼接

### 模块制作

1. 模块的定义：在Python中，一个.py文件就是一个模块
2. 作用：模块可以使我们有逻辑的组织我们的Python代码；以库的形式去封装功能，便于调用者使用；可以定义类、函数、变量，也能包含可执行的代码
3. 注意：不同的模块可以定义相同的变量名，但是每个模块中的变量名的作用域只是在本模块中

### 模块的发布和安装

1. 将写好的包放到一个jiaoxue/目录下
2. 在jiaoxue/目录下创建一个setup.py文件，文件里写入代码

from distutils.core import setup  
  
'''  
name 模块名称  
version 版本号  
description 描述  
author 作者  
py\_modules 要发布的内容  
'''  
setup(name='modelTest', version='1.0', description='文件统计模块',  
 author='lv', py\_modules=['modelTest'])

1. 创建模块

python setup.py builds

1. 生成压缩包

python setup.py sdist

1. tree看下jiaoxue目录下的结构

**模块的安装**

1. 将上一节生成的压缩包复制到桌面解压，解压后会生成一个文件夹
2. 进入文件夹
3. 执行命令安装python setup.py install
4. 查看是否安装成功：在python安装目录下的site-packages目录下

### 回顾小结

## day12-Python垃圾回收机制

### 复习和学习目标

### 引用计数介绍

1. 引用计数机制概述

Garbage collection（GC）

Python采用的是引用计数机制为主，标记-清除和分代收集两种机制为辅的策略

1. 引用计数原型

Python里每一个东西都是对象，他们的核心就是一个结构体PyObject，Pyobject是每个对象必有的内容，其中ob\_refcnt就是做为引用计数。当一个对象有新的引用时，它的obrefcnt就会增加，当引用它的对象被删除时，它的ob\_refcnt就会减少，当引用计数为0时，该对象的生命就结束了。

### 信息标记分代收集

GC负责的主要任务：为新生成的对象分配内存、识别那些垃圾对象、从垃圾对象那里回收内存。

1. 标记-清除机制：顾名思义，首先标记对象，然后清除垃圾
2. 引用计数不能回收的案例分析（1）

如果一个数据结构引用了它自身，即如果这个数据结构是一个循环数据结构，那么某些引用计数值是肯定无法变为0的。

1. Python 零代链表

### Python-GC

1. 导致引用计数+1的情况

对象被创建、对象被引用、

对象作为参数传入到一个函数中、对象作为一个元素存储在容器中

1. 导致引用计数-1的情况

对象的别名被显式销毁、对象的别名被赋予新的对象

一个对象离开它的作用域，例如f函数执行完毕时，func函数中的局部变量（全局变量不会）

对象所在的容器被销毁，或从容器中删除对象

1. 内存泄露：申请了某些内存，但是忘记了释放，那么这就造成了内存的浪费，久而久之内存就不够用了
2. 手动调用gc回收垃圾：gc.collect( )
3. 有三种情况会触发垃圾回收

当gc模块的计数器达到阈值的时候，自动回收垃圾

调用gc.collect( )，手动回收垃圾

程序退出的时候，python解释器来回收垃圾

### 内存优化

小整数与大整数对象池

Python为了优化速度，使用了小整数对象池，避免为整数频繁申请和销毁内存空间。

Python对小整数的定义是[-5，256]这些整数对象是提前建立好的，不会被垃圾回收。

### Python PEP8说明

1. Python pep8 原则

Guido的主要见解之一是代码读取的次数比写入次数多得多。这里提供的准则旨在提高代码的可读性，并使其在各种Python代码中保持一致。

1. 代码布局

缩进：4个空格的缩进（编辑器都可以完成此功能），使用Tab，不能混合使用Tab和空格。

1. 缩进：每个缩进级别使用4个空格
2. 悬挂缩进不一定是4个空格
3. 行宽：每行最大长度79
4. 模块导入

### 命令行参数\_1

1. sys模块

Python可以使用sys模块中的sys.argv来获取命令行参数

1. argparse模块

argparse模块可以轻松编写用户友好的命令行界面，该程序定义了它需要的参数，argparse并将找出如何解析这些参数sys.argv，该argv模块还会自动生成帮助和用法消息，并在用户给出程序无效参数时发出错误。

(1)添加参数选项—add\_arugment

(2)

### 命令行参数\_2

## day13-正则表达式

### 复习和目标

学习目标：

（1）区分正则表达式和Python正则表达式的相同点和不同点

（2）使用re模块的相关方法处理数据

（3）区分贪婪模式和非贪婪模式，使用非贪婪模式解决特殊的数据处理问题

### Python正则快速体验

re模块，提供了Perl风格的正则表达式模式。

1. re.match：尝试从字符串的起始位置去匹配一个规则，匹配成功就返回match对象，否则返回None。可以使用group（）获取匹配成功的字符串。

语法：re.match（pattern，string，flags=0）

pattern：匹配的正则表达式

string：要匹配的字符串

flags：标志位，用于控制正则表达式的匹配方式，如是否区分大小写，多行匹配等

re.I：使匹配对大小写不敏感

re.L：做本地化识别匹配

re.M：多行匹配

re.S：使.匹配包括换行在内的所有字符

re.U：根据Unicode字符集解析字符，这个标志影响\w,\W,\b.\B

re.X：该标志通过给予你更灵活的格式以便你将正则表达式写得更易于理解

### 初始Re模块

### 正则表达式\_规则\_1

.（点）：匹配任意一个字符，除了换行符\n

[abc]：匹配abc中的任意一个字符

\d：匹配一个数字，即0-9

\D：匹配非数字，即不是数字

\s：匹配空白，即空格，tab键

\S：匹配非空白，除空格，tab键之类的

\w：匹配单词字符，即a-z、A-Z、0-9、\_

\W：匹配非单词字符

### 正则表达式\_规则\_2

### 正则匹配数量1

\*：匹配前一个字符出现0次或者无限次，即可有可无

+：匹配前一个字符出现1次或者无限次，即至少有一次

\?：匹配前一个字符出现1次或者0次，即要么有1次，要么没有

{m}：匹配前一个字符出现m次

{m,}：匹配前一个字符至少出现m次

{m,n}：匹配前一个字符出现从n到m次

### 正则匹配数量2

### 转义开头和结尾\_1

python字符串中\作为转义字符开头，比如\n表示换行，\t表示tab键，为了表示\本身，再加一个\，成为\\形式。在python中表示路径：

### 转义开头和结尾\_2

^：匹配字符串开头

$：匹配字符串结尾

### 分组匹配\_1

|：匹配左右任意一个表达式

(ab)：将括号中字符作为一个分组

\num：引用分组num匹配到的字符串

(?P)：分组起别名

(?P=name)：引用别名为name分组匹配到的字符串

### 分组匹配\_2

### Re中编译函数\_1

1、re.compile：compile将正则表达式模式编译成一个正则表达式对象

reg=re.compile(pattern) result=reg.match(string)

等价于result=re.match(pattern,string)

使用re.compile()和保存所产生的正则表达式对象重用效率更高

2、re.search：search在全文中匹配一次，匹配到就返回

语法：re.search(pattern,string,flags=0)

规则是在全文中只匹配一次，匹配到就返回

1. re.findall：匹配所有返回一个列表，这个方法使用频率较高

语法：findall(string[,pos[,endpos]])

1. re.sub()：将匹配到的数据进行替换

语法：sub(pattern,repl,string,count=0,flags=0)

1. re.split：根据匹配进行切割字符串，并返回一个列表

语法：split(pattern,string,maxsplit=0,flags=0)

### Re中编译函数\_2

### 贪婪模式和非贪婪模式

Python 里数量词默认是贪婪的，总是尝试匹配尽可能多的字符；非贪婪则相反，总是尝试匹配尽可能少的字符。

在 \* ？ + {m,n} 后面加上？，使贪婪变成非贪婪

### 小结完整版

### 课后作业

## day14-Linux系统安装与常用命令介绍

### 复习和学习目标

### Linux操作系统简介\_1

1、操作系统是管理和控制计算机硬件与软件资源的计算机程序，是直接运行在裸机上的最基本的系统软件。

2、操作系统类型

（1）桌面操作系统：主要用于个人计算机，Linux操作系统和Windows操作系统

（2）服务器操作系统：Linux安全、稳定、免费；Windows Server付费、安全性稍低

（3）嵌入式操作系统：一个自带了固定应用软件的巨大泛用程序

（4）移动设备操作系统：IOS苹果手机，Android小米、华为、三星

### Linux操作系统简介\_2

1. 虚拟机指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统。
2. 虚拟系统通过生成现有操作系统的全新虚拟镜像，具有与真实操作系统完全一样的功能

### Linux和虚拟机使用

1. Unix的发展

1965年左右贝尔实验室计划要建立一套MULTICS操作系统

1969年项目进展缓慢，资金短缺，贝尔实验室退出研究

1969年Ken Thomposon使用会变写出了Unix操作系统的原型

1970年，美国贝尔实验室的Kenthomposon，用B语言写了第一个UNIX操作系统

1971年，DennisM.Ritchie，加入Thomposon的开发项目，合作开发UNIX

1972年，DennisM.Ritchie，在B语言的基础上最终设计出一种新的语言，即C语言

1973年，C语言主体完成，Thomposon和Ritchie完全重写了Unix操作系统

1. Minix发展

由于通用电气的政策改变，UNIX源代码私有化，大学中不能再使用UNIX源代码，AndrewS.Tanenbaum教授自行开发了与UNIX兼容的操作系统，将其称为Minix。

1. Linux发展

1991年，林纳斯就读于赫尔辛基大学期间，对Unix产生浓厚兴趣，尝试在Minix上做一些开发工作，自己编写了磁盘驱动程序和文件系统，这些在后来称为Linux第一个内核的雏形。林纳斯利用GNU的Bash当做开发环境，gcc当做编译工具，编写了Linux内核，一开始Linux并不能兼容Unix。

虚拟机安装

下载地址：百度云盘：<https://pan.baidu.com/s/1kUCyM2r> 密码：zypl

### Ubuntu系统20.04安装

### 命令格式和帮助

### linux命令使用\_1

cd ..：返回上一级目录

ll：当前目录下所有的文件

pwd：查看当前目录的路径

cd /：到根目录

pwd --help：--help可以打印当前操作的作用

man man：可以查看所有命令

q：退出文档

su root：切换至root下

技巧：

（1）命令可以使用tab补全

（2）曾经使用过的命令可以用上下剪头来回切换

（3）忘了一些命令的选项参数，可以用--help去查看帮助

（4）输入了命令不想执行可以使用ctrl+c取消，或者用ctrl+u进行删除

### linux命令使用\_2

1. ls查看目录文件

-a 显示隐藏文件

-l 以列表形式显示

-h 以人性化的方式显示文件内容大小

-R 递归显示子目录

1. cd切换工作目录

绝对路径：指的是在输入路径时，最前面是/或者~，表示从根目录开始的具体目录位置

相对路径：指的是以当前目录开始，不以/或者~开头，表示不是以根目录或者家目录开始的目录

cd 路径：切换到指定目录

cd .：切换到当前目录

cd ..：切换到上一级目录

cd ~：切换到家目录

cd -：上一次工作目录互相切换

1. touch新建文件命令

touch [文件名称]

touch 文件名称 [文件名称] touch 新建文件夹，可以同时新建多个

1. medir新建目录

-p：递归创建

medir [目录名]

medir -p a/b/c：在b目录不存在时，可以用-p参数同时创建b目录

1. rmdir删除目录

-p：递归删除

rmdir [目录] 注意：rmdir只能删除空目录，非空目录无法删除

1. rm删除文件或目录

-r：递归的删除目录下的内容，删除文件夹时必须加此参数

-f：强制删除，忽略不存在的文件，无需提示

1. mv移动和重命名

-f：覆盖前不询问

-i：覆盖前询问

-n：不覆盖已经存在的文件

mv命令在同一个文件夹下移动文件，即重命名功能。

1. cp复制

选项说明：

-i：覆盖前提示

-：r若给出的源文件是目录文件，则cp将递归复制该目录下的所有子目录和文件，目标文件必须为一个

目录名 cp src des 如果是复制文件夹，则加上-r选项

1. cat查看文件内容、合并文件

-n：输出行编号

-s：不输出多行空行

-b：对非空行进行编号

cat [-nsb] filename ：查看文件的内容

cat filename1 filename2>filename：将filename1和filename2的文件合并写入到filename文件中

1. more查看文件内容（分页查看）

more filename：查看filename文件的内容

1. history查看历史命令

history -c：删除历史命令记录

## day15-Linux权限与远程管理

### 回顾和学习目标

### 用户和用户组\_1

（1）用户：要登录Linux必须要有一个用户，一台Linux系统下可以有多个用户，并且每个用户可以用不同的权限。在Linux中可以指定用户对不同的文件、目录拥有不同的权限

（2）用户组：Linux有一个组的概念，不同的用户分配到一个组，那么同组下的用户，都拥有这个组队权限。

（3）权限：Linux权限有三种：读r、写w、执行-

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 目录 | 拥有者权限user | | | 组权限group | | | 其它用户权限other | | |
| 文件权限示例 | - | r | w | - | r | w | - | r | - | - |
| 目录权限示例 | d | r | w | x | r | w | x | r | - | x |

（4）用户管理：创建用户、删除用户、修改用户账号属性、创建用户组、修改用户组属性

其中，创建用户、删除用户、修改其他用户密码 的终端命令都需要sudo执行。

创建用户：useradd [用户名]

-d：指定新账号的主目录

-g：指定用户的所属组

-G：指定用户附加组

-s：指定用户登录shell

-m：自动创建家目录

删除用户：userdel [用户名]

-r：删除用户的同时删除家目录

修改用户：usermod

-u：用户id

-g：所属组id

-a -G GID：不使用-a选项，会覆盖此前的附加组

-d -m：将家目录内容移至新位置

-s：该用户账号的新登录

-l：新的登录名称

用户切换：su - [用户名] 加-同时切换到用户的家目录，不加直接在当前目录切换到新用户

1. 用户组管理

创建用户组：groupadd

删除用户组：groupdel

修改组属性：groupmod

提示：

创建用户时，会默认创建一个与用户同名的组名

创建成功后可在/etc/password文件下查看

新创建的用户没有sudo权限,需要将用户加到adm,sudo这两个组中才拥有sudo权限

### 用户和用户组\_2

su root：切换至root下

whoami：可以查看当前用户

clear：清空终端页面信息

### 用户和用户组\_3

### 用户和用户组\_4

### 权限设定

1、权限管理

命令：chmod +/- rwx 文件名|目录名 + 增加权限，-取消权限

每个文件，都有三组不同的权限，第一组文件所有者，第二组是文件所属组，第三组是其他用户。

第一组：u 文件所有者 修改所有者权限：chmod u+/- r/w/x filename

第二组：g 文件所属组 修改所属组权限：chmod g+/- r/w/x filename

第三组：o 其他用户 修改所属组权限：chmod o+/- r/w/x filename

2、权限管理

chmod 755文件名|目录名 指定权限修改

chmod中第一个数字代表所有者权限，第二个数字代表所属组权限，第三个数字代表其他人权限。

|  |  |
| --- | --- |
| 权限 | 数字表示法 |
| r | 4 |
| w | 2 |
| x | 1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | 2 | 1 |  | 7 |  | rwx |
| 4 | 2 | 0 | 6 | rw- |
| 4 | 0 | 1 | 5 | r-x |
| 4 | 0 | 0 | 4 | r-- |
| 0 | 2 | 1 | 3 | -wx |
| 0 | 2 | 0 | 2 | -w- |
| 0 | 0 | 1 | 1 | --x |
| 0 | 0 | 0 | 0 | --- |

### 远程管理

1、ifconfig

查看系统ip地址

2、ping命令

检查网络是否正常通信，Linux下默认ping不会结束，使用ctrl+c强制结束

示例：ping www.baidu.com

ping -c：指定ping几个数据包结束

ping -i：指定发送数据包的间隔，单位是秒

ping -s：指定发送数据的大小，单位字节

ping -t：设置TTL的大小，TTL网络调数大小

3、ssh远程连接

SSH是一种网络协议，用于计算机之间的加密登录。Linux下默认开启sshd服务，只有开启sshd服务才能进行ssh连接。

可以使用service sshd status查看是否开启sshd服务。active(running)表示已经开启。

1. Windows下连接方式

借助第三方软件xshell或者putty来连接

ssh默认端口号：22

1. Linux下ssh命令连接方式

ssh -p端口 主机#-p可以指定端口。比如：ssh 192.168.136.113 -p 22

ssh一般使用默认的22，看到welcome表示已连接成功。

1. scp文件复制

scp就是secure copy，是一个Linux下用来进行远程拷贝文件的命令

示例1：把远程的e.txt文件拷贝到本地/home/python/Desktop下

scp [lvshl@192.168.254.131:/home/python/a/e.txt](mailto:python@192.1683254.131:/home/python/a/e.txt) /home/python/Desktop

示例2：将本地Desktop/requirement.txt文件夹拷贝到远程主机的家目录下

scp [Desktop/requirement.txt lvshl@192.168.254.131:/home/python](mailto:python@192.1683254.131:/home/python/a/e.txt)

示例3：scp -r可以复制文件夹，把当前目录下的demo文件夹复制到远程目录下的Desktop

scp -r demo user@remote:Desktop

示例4：将远程主机的a文件夹包括子文件，复制到本机的桌面

scp -r [lvshl@192.168.254.131:/home/python/a](mailto:lvshl@192.168.254.131:/home/python/a) Desktop

### 系统命令

1. data查看系统时间
2. 查看磁盘空间
3. df -TH查看磁盘分区，以及挂载情况
4. du -sh 目录名：查看目录大小
5. du -h 文件名：查看文件大小
6. 查看内核、操作系统、CPU信息
7. uname -a 查看内核/操作系统/CPU信息
8. uname -i 查看硬件平台
9. uname -m 查看CPU
10. uname -n 查看节点名称
11. uname -o 查看操作系统
12. uname -v 查看内核版本
13. uname -r v查看发行版本号
14. 查看进程

（1）top查看进程实时运行情况，即系统资源实时使用情况，退出top界面输入q

（2）ps查看系统所有进程的状态

命令：ps -ajx 一般使用ps命令带选项ajx一起使用

|  |  |
| --- | --- |
| 符号 | 状态 |
| D | 不可中断系统进程 |
| R | 运行中的进程 |
| S | 中断sleep状态 |
| T | 停止 |
| Z | 僵死，一般称之为僵尸进程 |

（3）结束进程

kill 参数 进程号

kill -9 进程号：强制结束进程

kill -15 进程号：结束进程，等级没有-9高

提示：在Linux中1号进程（init进程）是所有进程的祖先进程，是不能被结束的。

### 课后作业

## day16-Linux文件查找与编辑器使用

### 复习和目标

### Linux文件查找\_1

1. which命令

which命令用于查找并显示给定命令的绝对路径，环境变量Path中保存了查找命令时需要遍历的目录。which指令会在环境变量$Path设置的目录里查找符合条件的文件。也就是说，使用which命令，就可以看到某个命令是否存在，以及执行的到底是哪一个位置的命令，此命令回去搜索$Path环境变量中的目录路径，可以使用echo $PATH查看

1. whereis

与which功能相似，也可以查找到命令的绝对路径

与whereis不同，which会列出这个命令的别名记录，而whereis会显示出这个命令的帮助文档所在位置

1. locate

locate 搜索关键字

其执行权限为root

### Linux文件查找\_2

find命令：在一个目录（及子目录）中搜索文件，可以指定一些匹配条件，如按照文件名、文件类型、用户等条件查找文件。

语法：find 搜索路径 搜索选项 filename

选项说明：

-name： filename 查找名为filename的文件

-size： +/-大小，按照文件大小来查找，+大于，-小于

-user：username按文件所属查找

通过时间查找：

-ctime -atime -mtime（以天为单位）

-cmin -amin -nmin（以分钟为单位）

-type： 按文件的类型

-inum：根据i节点进行查找

-group：组名，按照所属的组查找

-a and 逻辑于 -o or 逻辑或

### Linux文件查找**\_**3

### Linux\_Grep

1. grep命令

grep命令是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。

grep “python” filename #在filename文件中查找python，并把匹配的行打印出来

grep “python” filename filename1 filename2# 再多个文件中查找python

grep -E “[a-c]” filename #加上-E选项可以使用正则表达式

### Linux-管道符和重定向

1、| 管道符的作用，将左边的输出当右边的输入

ps -ajx|grep ssh：将ps命令的输出结果当grep地输入过滤

1. 输出重定向：>，比如：ll<./text
2. 输出重定向：追加>>

### Linux-解压缩命令

1. 文件打包

tar -cvf 打包文件名.tar 被打包的文件/路径

-c 创建一个新的归档

-v 详细地列出处理的文件

-f 使用归档文件

1. 文件解包

tar -xvf 打包过的文件.tar

-x 将打包过的文件解包

提示：-f必须放在最后面

1. gzip命令

gzip一般跟tar一起使用，完成打包压缩

tar只负责打包并为左压缩，使用-z选项可以调用gzip压缩，完成压缩操作。

使用tar包压缩的文件名，一般命名成xxx.tar.gz区别其他文件。

1. 压缩文件

tar -zcvf py.tar.gz a.txt b.txt c.txt

1. 解压缩文件

tar -zxvf py.tar.gz

1. 解压缩到指定路径

tar -zxvy 打包文件名.tar.gz -C 目标路径

-C 指定解压到那里，解压的目录必须存在

### Linux-软件管理apt

软件安装：sudo apt install 软件包

软件更新：sudo apt upgrade [软件包]

软件卸载：sudo apt remove 软件包

### Linux-vim编辑器

vim三种模式：命令模式、插入模式、末行模式

1. 命令模式

使用vim打开文件的时候，就进入到命令模式。

vim filename：打开或新建文件，并将光标置于第一行首

vim+n filename：打开文件，并将光标置于第n行首

vim + filename：打开文件，并将光标置于最后一行首

vim +/pattern filename：打开文件，并将光标置于第一个与pattern匹配的串处

vim -r filename：在上次正用vi编辑时发生系统崩溃，恢复filename

vim filename ...filename：打开多个文件，依次进行编辑

命令模式下只能输入命令，不能进行编辑，只有进入输入模式才能做文件编辑。

1. 插入模式
2. 末行模式

在命令模式下按“：”即可进入末行模式

:q 退出

:q! 退出并不保存

:w 保存

:wq 退出并保存

:x 退出并保存

### Linux-小结

## day17-Mysql基本使用

### 学习目标

### 数据库概述

1. 数据库类型（按数据模型特点分）

网状型数据库：采用记录类型为节点的网状数据模型

层次型数据库：采用层次模型模拟现实世界中按层次组织起来的事物

关系型数据库：采用二维表结构组织和管理数据，并规定了表内和表间数据的依赖关系

1. Mysql数据库介绍

SQL查询语言：

DQL：数据查询语言，对数据进行查询，如select

DML：数据操作语言，对数据进行增加、修改、修改，如insert、update、delete

TPL：事务处理语言，对事务进行处理，包括begin transaction、commit、rollback

DCL：数据控制语言，进行授权与权限回收，如grant、revoke

DDL：数据定义语言，进行数据库、表达管理，如create、drop

CCL：指针控制语言，通过控制指针完成表的操作，如declare cursor

### 数据库的安装

### 数据库和数据表管理\_1

1、数据库管理

（1）连接数据库

mysql -u账号 -p密码 -h主机地址 -P端口;

mysql -uroot -pmysql;

（2）查看数据库版本

select version();

（3）显示当前时间

select now();

（4）查看所有数据库

show databases;

（5）创建数据库

create database 数据库名 charset=utf8;

（6）切换数据库

use 数据库名

（7）查看当前正在使用哪个数据库

select databse();

（8）删除数据库

drop database 数据库名;

2、数据表管理

（1）数据表设计包括ER图、表的主键、字段、数据类型、约束、表之间关系的设计

（2）E-R模式即实体-关系模型中主要用于定义数据的存储需求，该模型已经广泛用于数据库设计中。E-R模型由实体、属性和关系三个基本要素组成。

（3）主键

（4）实体间的关系与外键

（5）约束

约束是定义在表上的一种强制规则。当某个表定义约束后，对该表做的所有SQL操作都必须满足约束的规则要求，否则操作将失败。

约束类型：

NOT NULL：非空

UNIQUE：唯一性约束

PRIMARY KEY：外键约束

FOREIGN KEY：外键约束

CHECK：检查性约束，在列上指定一个必须满足的条件

1. 创建表
2. 查看当前数据库中的表

show tables;

1. 创建表

create table 表名（

id int unsigned auto\_increment primary key not null,

name varchar(10) not null,

is\_delete bit(1) not null default 0

）;

1. comment注释，在创建表达时候如果字段很多，可以给字段添加注释
2. 查看创建表达sql语句：
3. show create table 表名;
4. 修改表
5. 添加字段

alter table 表名 add 列名 类型;

eg：alter table student add birthday data;

1. 删除字段

alter table 表名 drop 字段名字;

eg：alter table student drop gender;

1. 修改字段

alter table 表名 modify 列名 类型及约束;

eg：alter table student modify hometown varchar(40) dafault null;

alter table 表名 change 原名 新名 类型及约束;

eg：alter table student change sex gender bit(1);

1. 删除表

drop table 表名;

eg：drop table student;

### 数据库和数据表管理\_2

### Navicat连接Mysql说明 ：mysql -u root -p密码

navicat for mysql 连接不上mysql服务器的原因排除：

（1）确保mysql的数据库服务正确启动

（2）输入的用户名和密码是否正确

（3）数据库是否开启远程连接（默认安装mysql数据完成后，只允许本地连接）

注意：对mysql配置文件更改完毕之后，一定要重启mysql服务

### 数据库操作\_1

1. 查询操作

select \* from 表名

1. 插入操作

insert into 表名 values(......)

insert into 表名 [字段1,字段2] values [值1,值2]

insert into 表名 values(......)，(......)，(......)

注意：主键用0占位。

1. 修改数据

update 表名 set 字段=xxx where 字段=xxx;

1. 删除数据

delete from 表名 where xxx=xxx;

### 数据库操作\_2

### 数据备份和恢复

**1、备份数据库**

（1）mysqldump -uroot -p 数据库名 > python.sql;

mysqldump -uroot -p python > python.sql;

eg：mysqldump -uroot -p stuDB > TestBackFile.sql

（2）图形界面

转储SQL文件即可进行数据库备份

**2、恢复数据库**

mysql -uroot -p 新数据库名 < python.sql

### 课后作业

## day18 Mysql数据查询

### 复习和学习目标

### 条件查询\_1

查询集只返回条件为True的内容

### 条件查询\_2

模糊查询：

like表示模糊查询

%表示任意多个字符

\_表示一个字符

rlike可以匹配正则

in 包含在里面的

如果需要匹配%本身，那么这需要使用%%

### 条件查询\_3

1、null值判断：

select \*from students where hometown is null;

2、排序

order by 字段【desc/asc】

desc：表示降序（从大到小）

asc：表示升序（从小到大，默认排序规则）

select \* from students order by id desc;

### 聚合函数

聚合函数主要是为了快速得到结果，经常使用的几个聚合函数，常见函数列表：

count：统计计数

max：计算最大值

min：计算最小值

数学函数：ceilling向上取整、round（X，D）四舍五入到最近的整数

sum：求和函数

avg：求平均数

round函数

时间函数

substr函数

### 分组和分页

1、分组查询：group by

eg：select gender,count(\*) from students group by gender;

2、as取别名

eg：select gender,count(\*) as ‘人数’ from students group by gender;

3、分组后条件筛选

分组后不能使用where做条件筛选，需要使用一个新的having函数

4、where与having的区别

where用在from之后的条件过滤

having用在分组之后的条件过滤，两个功能是一样的，只是作用的位置不一样

1. limit分页

eg：select \* from students limits start,count;

### 内连接

mysql三种连接查询

1. 内连接查询

查询的结果为两个表匹配到的数据，两个表都能匹配上的数据将返回给结果集。

select \* from students inner join class on students.class\_id=class.id;

使用内连接查询学生表和班级表

在连接查询的时候需要多次用到表名，如果表名过长可以使用as给表取名。

1. 右连接查询

查询结果为两个表匹配到的数据，右表特有的数据，对于左表中不存在的数据使用null填充。

select \* from students right join class on students.class\_id=class.id;

1. 左连接查询

查询结果为两个表匹配到的数据，左表特有的数据，对于右表中不存在的数据使用null填充。

select \* from students left join class on students.class\_id=class.id;

### 左右连接

### 子查询\_1

1. 子查询的概念

在一个select语句中嵌入了另一个select语句，嵌入的这个select语句就是子查询语句。子查询是辅助主查询的，充当数据源，或者充当条件。子查询是一条独立的语句，即使是单独拿出子查询也是可以正常执行的。

1. 子查询的四种类型
2. 标量子查询

子查询返回一行一列的数据，称之为标量子查询

eg：查询学生年龄小于平均年龄的学生信息：

子查询语句先查出平均年龄：select avg(age) from students;

select \* from students where age<(select avg(age) from students);

1. 列级子查询

子查询返回的是一行多列的数据，称之为列级子查询

eg：查询class表中已经安排学生的班级信息

select \* from class where id in(select class\_id from students);

1. 行级子查询

子查询返回的是一列多行

1. 表级子查询

子查询返回多行多列，称之为表级子查询

### 子查询\_2

sql语句中any、some、all的使用：都是用在子查询里面，用作比较运算

1. any表示任意一个，有一个满足了结果就为True

查询出在学生表中，1班级中，任意一个年龄大于2班级学生年龄的学生

select \* from students where age>any(select age from students where class\_id=2) and class\_id=1;

1. some和any是一样的效果
2. all全部满足了，结果才为True

查询出在学生表中，1班级中，年龄大于2班级所有学生年龄的学生

select \* from students where age>all(select age from students where class\_id=2) and class\_id=1;

### 子查询\_3

### 保存查询结果

1. 语句格式：insert into 表名(列1,列2) select ...
2. 合并查询：union all将两次查询的结果集合并到一起显示

union去重效果

### 小结

### 课后作业

## day19 Mysql高级

### 复习和目标

### 用户与权限管理

1、Mysql的账户管理包括登录和退出Mysql服务器、创建用户、删除用户、密码管理和权限管理等内容。通过账户管理，可以保证Mysql数据库的安全性。

Mysql中root账号拥有最高权限，包括删库，删表。所以在生产环境下一般不会使用root账号登录数据库的。Mysql中的用户信息保存在mysql库下单user表中。

2、创建用户并分配权限

老版本语法：grant 权限列表 on 数据库 to ‘用户名’@’访问主机’ identified by ’密码’;

新版本语法：创建用户：create user ‘用户名’@’ip地址’

授权操作：grant 权限 on 数据库 to ‘用户名’@’ip地址’

Mysql中权限有create、alter、drop、insert、update、delete、select等，如果要分配所有权限，直接使用：all privileges

1. 查看、回收权限

（1）查看用户权限

show grants for lv;

（2）回收权限

revoke select on \*.\* from ‘lv’@’%’

1. 修改密码
2. 普通用户修改自己的密码：在终端上修改不需要进到数据库

Mysqladmin -upython -p password 新密码

1. root账号修改普通用户的密码：

修改Mysql.user表

update mysql.user set authentication\_string=password(123)(新密码) where user=’python’(用户名)

刷新权限：flush privileges;

1. 删除用户

方法1：drop user ‘用户名’@’主机’;

方法2：进入到mysql这个库 delete from user where user=’用户名’;

### 事务\_1

1. 事务：也称工作单元，是由一个或多个sql语句所组成的操作序列，这些sql语句作为一个完整的工作单元，要么全部执行成功，要么全部执行失败。在数据库中，通过事务来保证数据的一致性。
2. 事务处理语言：Transcation Process Language，简称TPL，主要用来对组成事务的DML语句的操作结果进行确认或取消。确认也就是使DML操作生效，使用提交COMMIT命令实现；取消也就是使DML操作失效，使用回滚ROLLBACK命令实现。
3. 通过事务的使用，能防止数据库中出现数据不一致现象。
4. Mysql是支持事务的，跟使用的引擎相关。Mysql中支持多种引擎，默认使用Innodb引擎支持事务。

MyISAM：不支持事务，用于只读程序提高性能

Innodb：支持ACID事务，支持行级锁

Berkeley DB：支持事务

1. 事务的特性ACID

原子性：事务不可分割，组成事务的DML操作语句，要么全成功，要么全失败

一致性：一旦事务完成，不管是成功还是失败，整个系统处于数据一致的状态

隔离性：一个事务的执行不会被另一个事务干扰

持久性：也称为永久性，事务一旦提交，对数据的改变就是永久的，不可以再被回滚

1. 事务处理-手动提交事务

用begin、rollback、commit来实现

1. 事务处理-自动提交模式

Mysql默认是自动提交的，也就是你提交一个sql，就直接执行。可以通过set autocommit=0禁止自动提交，set autocommit=1开启自动提交，来实现事务的处理。

但要注意当用set autocommit=0的时候，以后所有的sql都将作为事务处理，直到用commit确认或rollback结束，注意当结束这个事务的同时也开启了新的事务。按第一种方法只将当前的作为第一个事务。

1. 事务处理-隐式处理

（1）隐式提交：当下列任意一种情况发生时，会发生隐式提交

执行一个DDL（CREATE、ALTER、DROP、TRUNCATE、RENAME）语句

执行一个DCL（GRANT、REVOKE）语句

（2）隐式回滚：当下列任意一种情况发生时，会发生隐式回滚

客户端强行退出

客户端连接到服务器端异常中断

系统崩溃

### 事务\_2

### 视图

对于复杂的查询，在多个地方被使用，如果需求发生了改变，需要更改sql语句，则需要在多个地方进行修改，维护起来非常麻烦。这种情况可以定义视图来解决，视图本质上是对查询语句的封装。视图实际上就是查询。保密诉求可以过滤敏感数据。

1. 创建视图：create view 视图名称 as select 语句;
2. 查看视图：查看表会将所有的视图也列出来

show tables;

show TABLE status;

1. 删除视图：drop view 视图名称;
2. 调用视图：select \* from v\_stu\_score;

### 索引

Mysql中常见的索引：

1、创建索引

（1）主键索引（PRIMARY KEY）

alter table 表名 add primary key（列名）;

（2）唯一索引（UNIQUE）

alter table 表名 unique（列名）;

1. 普通索引（INDEX）

alter table 表名 add index 索引名称（列名）;

1. 全文索引（FULLTREXT）

alter table 表名 add fulltext（列名）;

1. 组合索引

alter table 表名 add index 索引名（列1,列2,列3）;

2、查看索引

show index from 表名;

1. 删除索引

drop index 索引名 on 表名;

### 存储过程\_1

1. 存储过程，也翻译为存储程序，是一条或者多条sql语句的集合，可以视为批处理，但是其作用不仅仅局限于批处理。
2. 创建存储过程

语法：

delimiter //

create precedure 存储过程名称（参数列表）

begin

sql语句

end

//

delimiter;

delimiter用于设置sql语句分割符号，默认为分号。

在sql语句部分，编写的语句需要以分好结尾，此时回车会直接执行，所以需创建存储过程前需要指定其它符号作为分隔符，此处用//，也可以使用其它字符。

1. 查看创建的存储过程

（1）所有存储过程和函数，都存储在mysql数据库下proc表中

proc表中的字段说明：

name 表示名称

type表示类型，为存储过程、函数

body表示正文脚本

db表示属于的数据库

（2）查询刚才创建的存储过程

select name，type，body from mysql.proc where db=’python’;

（3）查询python表中的存储过程和函数

select name，type，body from mysql.proc where db=’python’;

1. 查看所有的存储过程

show precedure status;

1. 调用存储过程

call 存储过程名称();

1. 存储过程中的变量

要在存储过程中声明一个变量，可以使用declare语句

declare variable\_name datatype DEFAULT default\_value;

### 存储过程\_2

### 函数\_1

1. 内置函数-字符串函数
2. 查看字符的ascii码值ascii(str)，str是空串时返回0

select ascii(‘a’);

1. 查看ascii码值对应的字符char(数字)

select char(97);

1. 拼接字符串concat(str1,str2...)

select concat(12,34,’ab’)

1. 包含字符个数length(str)

select length(‘abc’)

1. 截取字符串

left(str.len)：返回字符串str的左端len个字符

right(str,len)：返回字符串str的右端len个字符

substring(str,pos,len)： 返回字符串str的位置pos起len个字符

1. 去除空格

ltrim(str)：返回删除了左空格的字符串str

rtrim(str)：返回删除了右空格的字符串str

trim(方向，remstr from str)：返回从某侧删除remstr后的字符串str；方向包括both、leading、trailing，表示两侧、左、右

1. 替换字符串replace

select replace(‘abc123’,’123’,’def’);

1. 大小写转换

lower(str)

upper(str)

select lower(‘aBcD’)

1. 日期时间函数
2. 获取子值，值为整数类型，函数如下：

year（date）：返回date的年份（范围在1000-9999）

month（date）：返回date中的月份数值

day（date）：返回date中的日期数值

hour（date）：返回date的小时数（范围在0到23）

minute（date）：返回date的分钟数（范围在0到59）

second（date）：返回date的秒数（范围在0到59）

1. 日期计算，使用+-运算符，数字后面的关键字为year、month、day、hour、minute、second

3、自定义函数

创建自定义函数

delimiter $$

create function 函数名称（参数列表） returns 返回类型

begin

sql语句

end

$$

delimiter；

### 函数\_2

存储过程和存储函数的区别：

1. 函数的限制比较多，比如不能使用临时表，只能使用表变量，存储过程限制比较少
2. 存储过程实现的功能逻辑相对来讲要复杂一些，而函数的实现功能针对性强一点
3. 返回值不同，函数是必须要返回值的，且仅仅返回一个结果集；存储过程可以没有返回值，但是能返回结果集
4. 调用的语法不同，函数经常会嵌入到sql中使用，通过select 函数名()；存储过程通过call语句去调用

### Python操作数据库\_1

1. Python DB-API 概述
2. Python标注数据库接口为Python DB-API，Python DB-API为开发人员提供了数据库应用编程接口。
3. Pymysql是在Python3.x版本中用于连接Mysql服务器的一个实现库，Python2中则使用mysqldb。
4. Pymysql遵循Python数据库API v2.0规范，并包含了pure-Python Mysql客户端库。
5. 安装Pymysql：pip install pymysql
6. 连接数据库

（1）数据库准备，连接数据库之前，请确保已经创建类python数据库，以及students表。

创建Connection 对象，用于建立与数据库的连接。

（2）Connection链接对象拥有的方法

close关闭连接，连接数据库跟打开文件一样，操作完成之后需要关闭，否则会占用连接。

commit()提交，pymysql默认开启事物，所以每次更新数据看都要提交

rollback()回滚，事物回滚

cursor()返回Cursor对象，用于执行sql语句并获得结果

（3）获取cursor对象

cur=conn.cursor( ) #cursor对象用于执行sql语句

1. cursor对象拥有的方法

close()关闭cursor对象

execute（operation[，parameters]）执行语句，返回受影响的行数，可以执行所有语句fetchone()获取查询结果集的第一个行数据，返回一个元组

fetchall()执行查询时，获取结果集的所有行，一行构成一个元组，再将这些元组装入一个元组返回。

### Python操作数据库\_2

### 小结