## 01\_Linux基础

1.操作系统的作用？

操作系统是配置在计算机硬件上的第一层软件，主要作用是管理好硬件设备。

2.Linux中根目录和家目录分别用什么表示？

/表示根目录、~表示家目录

3.Linux中相对路径和绝对路径？

以根目录（/）或者家目录（~）开始的路径称为绝对路径，反之称为相对路径

4.常用终端命令的介绍。

ls list 查看当前文件夹下的内容

pwd print wrok directory 查看当前所在文件夹

cd [目录名] change directory 切换文件夹

touch [文件名] touch 如果文件不存在，新建文件

mkdir [目录名] make directory 创建目录

rm [文件名] remove 删除指定的文件名 删除目录的话需要加上-r选项

clear clear 清屏

5.在Linux中以.开头的目录或者文件为隐藏文件，想要查看，可以通过ls –a 显示全部的文件。

6.cd命令常见的参数

Cd 直接回车是去到当前用户的家目录

Cd ~ 同样是去家目录

Cd . 当前目录

Cd ..返回上一级目录

Cd /返回根目录

Cd – 最近两个工作目录之间来回切换

7.拷贝命令cp

Cp 源文件 目标文件

如果想拷贝目录需要加-r选项

8.移动命令mv

Mv 源文件 目标文件

9.删除文件目录命令rm

Rm 默认只能删除文件

删除目录需要制定-r选项

10.查看文件内容的两个命令cat和more

Cat 显示全部内容，适合内容比较少的文档，-n显示行号 -b显示非空行号

More 是分屏显示，适合内容比较多的文档

11.重定向 >和>>

>代表的是覆盖

>>代表的是追加

12.grep查找命令

Grep 要查找的内容 文件

-n 显示行号

-v取反 -i忽略大小写

13.管道的作用

命令1 | 命令2

把前一个命令的输出当成第二个命令的输入

常见的管道命令 grep more

Ps aux | grep xx

## 02\_Python基础\_day01

1. Python名言

人生苦短，我用Python（Life is short,you need Python）

2. Python创始人吉多·范罗苏姆（龟叔），1991年第一版Python开源了。

3. Python的设计哲学

优雅、明确、简单

4. Python的优缺点

优点：完全面向对象的语言、拥有强大的标准版、社区提供了大量的第三方模块

缺点：运行速度不如编译性语言、国内市场小、中文资料匮乏

5. Python的版本介绍

Python2.x是过去，最后一个稳定版是2.7

Python3.x是现在和未来的趋势，最新的是3.6

6. 执行Python的三种方式

①解释器。Python/python3 Python文件

python xxx.py

python3 xxx.py

②交互模式运行。

在终端输入python或者python3 直接回车即进入官方提供的交互模式，但此方法不能保存代码，不适合运行太大的程序，比官方更好用的交互模式是ipython

③使用集成环境IDE运行

Pycharm可以运行python程序

## 02\_Python基础\_day02

1. Python中注释的作用？单行和多行注释  
   在程序中对某些代码进行标注说明，增强程序的可读性。  
   单行注释：以#号开头，再加一个空格，后面跟上注释内容  
   TODO注释：# TODO 注释内容 备忘功能，可以记录待开发的程序  
   多行注释：一对连续的三个双引号  
   ”””注释内容“””
2. 计算机的三大件  
   CPU、内存、硬盘
3. 程序执行原理？  
   程序没运行之前保存在硬盘上，运行之后加载到内存，然后CPU执行内存中程序代码
4. 变量的作用？如何定义变量？  
   变量是用来存储数据的，在 Python 中，每个变量在使用前都必须赋值，变量赋值以后该变量才会被创建，变量在定义的时候不需要指定类型，系统会根据值自动推导类型
5. 变量有哪些类型？如何查看变量类型？  
   变量分为**数字型和非数字型**  
   数字型：**整型（int）、浮点型（float）、布尔型（bool）和复数型**（用于科学计算）  
   非数字型：**字符串（string）、列表（list）、元组（tuple）、字典（dict）**

使用type函数查看数据类型

1. 格式化输出的格式有哪些？  
   % 被称为 格式化操作符，专门用于处理字符串中的格式

%s 字符串

%d 有符号十进制整数，%02d 表示输出的整数显示位数，不足的地方使用 0 补全

%f 浮点数，%.2f 表示小数点后只显示两位

%% 输出 %

## 02\_Python基础\_day03

1. 什么是标识符？标识符的定义规则？  
   程序员定义的变量名、函数名。  
   **标示符可以由 字母、下划线 和 数字 组成**

**不能以数字开头**

**不能与关键字重名**

1. Python中关键字（keyword）的概念？以及如何查看关键字？  
   Python内部已经占用的标识符，具有特殊的功能和含义  
   import keyword   
   print(keyword.kwlist)
2. if判断语句语法规则？  
   if 条件1:  
    满足条件1要做的事情  
   elif 条件2：  
    满足条件2要做的事情  
   else:  
    其他
3. 逻辑运算符有哪三种？它们有什么特点？  
   逻辑或：or 全假则为假，一真则为真  
   逻辑与：and 全真才为真，一假则为假  
   逻辑非：not 非真即假，非假即真
4. 如何生产一个随机整数  
   import random  
   random.randint(1,10)
5. 程序的三大流程  
   顺序：从上往下，顺序一行一行执行  
   分支：就是有不同的条件判断，决定要执行的分支  
   循环：让特定的代码重复执行，只要你满足我的条件，就一直循环，不满足则退出
6. 赋值运算符  
   一个等号（=）表示赋值运算符，还可以把运算符和等号连在一起，更简化，例如+=、-=、\*=、/=….
7. 循环  
   初始条件设置 —— 通常是重复执行的 计数器

while 判断条件是否满足:  
条件满足时，做的事情1

处理条件(计数器 + 1)  
条件不满足则跳出循环  
  
一定要注意**修改计数器，防止出现死循环**

## 02\_Python基础\_day04

1. while循环的基本格式  
   初始化计数器  
   while 条件:  
    满足条件要重复执行的代码  
    ……  
    修改计数器
2. 明白break和continue两个关键字的作用，两者的区别是什么？  
   break和continue都是专门用在循环中关键字，  
   break是退出当前循环，不再执行当前循环后续代码  
   continue是退出本次循环，继续下一次循环  
   break和continue都只针对当前循环有效
3. 字符串中常见的转义字符有哪些？  
   \t制表符，输出文本时保持垂直方向对齐  
   \n换行符 \\输出反斜杠  
   \r回车 \’输出单引号 \”输出双引号
4. 函数的概念和作用？  
   函数，就是把具有独立功能的代码块组织为一个小模块，在需要的时候调用，函数的命名符合标识符的命名规则，函数只有先定义才可以被调用。  
   函数的作用，在开发程序时，使用函数可以**提高编写的效率以及代码的重用**，避免重复造轮子
5. 形参和实参分别是什么？  
   形参指的是形式参数，在**定义**函数的时候，用来接收参数的，可以在函数内部作为变量使用  
   实参指的是实际参数，在**调用**函数的时候，用来把数据传到函数内部。
6. 函数返回值关键字是什么？有什么注意事项？  
   函数返回值是return关键字，函数完成工作后，最后给调用者的一个结果。  
   一旦执行到return，函数内部return后续的代码将不再执行,并且不会输出内容。

## 02\_Python基础\_day05

1. 列表的概念以及基本格式  
   列表用于存储一串信息，是使用频率最高的数据类型。(数组)  
   列表用[]定义，数据之间使用逗号隔开，列表的索引从0开始，有序的数据集合。列表一般都保存相同的数据类型。
2. 列表的添加、修改、统计或者删除某个元素  
   列表的添加有三种方式：insert(key, value)、append(value)、extend(list)  
   修改时通过下标重新赋值即可，列表[下标] = 新的值  
   查找下标是index(值)，默认是查找第一次出现的索引位置  
   删除方法：remove(数据)、pop(下标)、clear()，也可以通过del关键字删除
3. 列表的排序  
   统计：len(列表) count(值)获取数据出现的次数
4. 列表和元组的区别？  
   列表和元组非常类似，**列表的值可以修改，元组的值不可以修改**  
   列表用[]定义，元组使用()定义
5. 访问元组中某个元素  
   元组[索引]
6. 字典的定义格式  
   字典使用{}定义，一般用来描述一个物体的完整信息，用键值对来存储数据，键是索引，值是数据，值可以是任意类型，但键必须是**字符串、数字或者元组（不可变类型）**
7. 字典和列表的不同  
   列表是有序的对象集合，字典是无序的对象集合  
   列表定义通过[]，字典定义通过{}

## 02\_Python基础\_day06

1. 字符串的定义  
   字符串就是一串字符，用一对儿单引号或者双引号括起来
2. 字符串切片  
   切片就是使用索引值，从一个大的字符串中切出小的字符串，适用于字符串、元组、列表。基本语法是 字符串[开始索引:结束索引:步长]，默认步长为1，索引从0开始，如果倒序切片，索引从-1开始。
3. Python内置的公共方法  
   len、del、max、min、cmp
4. 列表”+”和extend、append的区别  
   两个列表相加得到了一个全新的列表  
   extend(列表)，是给原来的列表又扩展了一个新的列表  
   append(数据)，是给原来的列表追加了新元素，如果传的是列表，则把列表当成一个数据追加给原来的列表。

列表1 += 列表2

1. 成员运算符  
   in、not in
2. for else 完整语法

for 变量 in 集合:

循环体代码

else:

**没有通过 break 退出循环，循环结束后，会执行的代码**

## 02\_Python基础\_day07

1. 变量的引用  
   变量中记录数据的地址，就叫做引用。在Python中函数的参数和返回值都是引用传递，可以使用id()函数查看变量在内存中地址。
2. 不可变类型和可变类型  
   不可变类型：在内存中的数据不允许被修改，包含：数字类型（整型、浮点型、布尔型、复数型），元组，字符串  
   可变类型：在内存中的数据可以被修改，包含：列表，字典，这个修改必须是通过**方法**来修改的，如果给不可变类型重新赋值，一样会改变变量的引用。
3. 局部变量和全局变量  
   局部变量：在函数**内部定义**的变量，只能在函数内部使用，函数执行完毕之后将被系统回收，生命周期也将结束。  
   全局变量：在函数**外部定义**的变量，所有函数内部都可以使用。在函数内部不能直接修改全局变量的值，如果修改了就相当于又重新创建了一个新的局部变量。可以通过global关键字声明一下，明确表明这个是全局变量。全局变量的定义要在所有函数定义的上面，这样可以保证所有函数都可以使用。建议以g\_或者gl\_为前缀命名全局变量。
4. 函数缺省参数  
   定义函数时，可以给某个参数指定一个默认值，具有默认值的参数就叫做缺省参数，调用函数的时候如果没有传缺省参数的值，那么会调用参数的默认值。缺省参数放在后面。
5. 多值参数  
   有时可能需要 一个函数 能够处理的参数 个数 是不确定的，这个时候，就可以使用 多值参数

python 中有 两种 多值参数：

参数名前增加 一个 \* 可以接收 元组， \*args

参数名前增加 两个 \* 可以接收 字典， \*\*kwargs keyword

1. 函数的递归  
   函数调用自身的编程技巧称为递归，说白了就是函数自己调用自己，递归必须有终止条件，否则将出现死循环。就好比打开一扇门里面还有门，再打开还有门，一直打开全部的门，直到没门了，然后一层一层退出来。

## 03\_面向对象\_day01

1. 面向对象的简称？  
   面向对象编程 —— Object Oriented Programming 简写 OOP，是一种编程思想或者方式
2. 面向对象和面向过程的区别？  
   面向过程是早期的一个编程思想，所有的步骤从头到尾逐步实现，把某些功能独立的代码块封装成函数，然后顺序的调用不同的函数，注重步骤与过程，不注重职责分工，适合开发简单的项目。  
   面向对象**相比函数是一个更大的封装**，面向对象注重对象和职责，不同的对象承担不同的职责，适合开发比较复杂的项目。
3. 类和对象的概念？  
   **类和对象是面向对象的两大核心概念**，  
   类是对一群具有相同**特征**或者**行为**的事物的一个统称，是抽象的，不能直接使用。就好比图纸或者模板，根据这个模板可以创建很多个对象  
   **对象是由类创建出来的一个具体存在，可以直接使用。**
4. **类和对象的关系？  
   类是模板，对象是根据类这个模板创建出来的，应该 先有类，再有对象**

**类只有一个，而对象可以有很多个，不同的对象之间属性可能会各不相同**

**类中定义了什么属性和方法，对象中就有什么属性和方法，不可能多，也不可能少**

1. 属性和方法是什么？  
   对对象的特征描述，通常可以定义成属性

对象具有的行为（动词），通常可以定义成方法

1. 如何定义类并创建对象？  
   class 类名:

def 方法1(self, 参数列表):

pass

类名符合**大驼峰**命名，方法第一个参数必须是**self**，self 就表示 当前调用方法的对象自己在方法内部，可以通过 self. 访问对象的属性，也可以通过 self. 调用其他的对象方法  
创建一个对象如下：  
对象变量 = 类名() 自动分配内存空间，自动初始化方法（定义属性）

1. \_\_init\_\_()方法的作用  
   初始化方法，在对象创建的时候会自动执行，主要用来**定义类的属性**
2. \_\_str\_\_()方法的作用  
   返回对象的描述信息，print 函数输出对象变量时，能够打印自定义的内容，必须返回字符串
3. \_\_del\_\_()方法的作用  
   对象被从内存中销毁前，会被自动调用
4. 面向对象的三大特点  
   **封装**、继承、多态
5. 如何理解面向对象的封装？  
   面向对象编程的第一步，就是要把**属性和方法封装到类**中，外界使用类创建对象，然后让对象调用方法，对象方法的细节都封装在类的内部。

## 03\_面向对象\_day02

1. 私有属性、私有方法；伪私有属性和方法  
   私有属性：对象不希望公开的属性，语法是在属性前面加上\_\_属性名  
   私有方法：对象不希望公开的方法，语法是在方法前面加上\_\_方法名  
   伪私有属性和方法是指在Python中并没有真正意义上的私有，而是把属性或者方法名前面加上了\_类名，可以通过\_类名\_\_属性名的形式访问，但是不推荐。
2. 继承的作用，以及语法  
   继承实现了代码的重用，子类可以拥有父类的所有方法和属性。  
   class 子类名(父类名):  
    pass

继承具有传递性，子类可以拥有所有与子类有继承关系的父类方法和属性。

1. 重写父类方法，如何调用重写的父类方法？  
   当父类的方法实现不能满足子类需求时，可以对方法进行**重写**(override)  
   重写父类方法有两种情况：  
   覆盖父类的方法，子类中创建一个跟父类同名的方法;  
   对父类方法进行扩展，使用super().父类方法调用，另外可以通过父类名.父类方法()的形式去访问，但是不推荐。
2. 多继承的概念以及语法  
   多继承：子类可以拥有多个父类，并且具有所有父类的属性和方法。多个父类之间应尽量避免同名的方法和属性。

class 子类名(父类名1, 父类名2...)

pass

1. 多态的概念。  
   不同的子类对象调用相同的父类方法，产生不同的执行结果

多态可以增加代码的灵活度，以**继承和重写父类方法**为前提，是调用方法的技巧，不会影响到类的内部设计

## 03\_面向对象\_day03

1. 类属性和实例属性的概念。  
   类属性是给类对象(类是一个特殊的对象)中定义的属性，通常用来记录与这个类相关的特征，类属性不会用于记录具体对象的特征。类属性的定义是在初始化方法之外的  
   实例对象的属性叫做实例属性，在初始化方法内部定义
2. 类属性和实例属性的访问  
   类属性的访问是**类名.类属性**  
   实例属性的访问是实例**对象.属性名**，也可以在类内部使用self.属性名访问
3. 类方法的定义和使用

@classmethod  
def 类方法名(cls):

pass  
需要装饰器@classmethod来标识，第一个参数必须是cls，代表当前类对象本身。可以通过**类名.类方法()**来调用

1. 静态方法的定义和使用  
   既不需要访问实例属性或者调用实例方法，也不需要访问类属性或者调用类方法。  
   @staticmethod

def 静态方法名():

pass  
需要用装饰器 @staticmethod 来标识，告诉解释器这是一个静态方法

通过**类名.静态方法()**调用

1. 类方法，静态方法和实例方法的区别  
   类方法就是针对类对象定义的方法，在类方法内部可以直接访问类属性或者调用其他的 类方法，使用@classmethod标识，需要传递cls参数，代表类对象本身。  
   静态方法是既不需要访问实例属性或者调用实例方法，也不需要访问类属性或者调用类方法。使用@staticmethod标识。

实例方法，没有特殊的修饰符，第一个参数是self，代表当前对象的引用

1. 单例模式的定义  
   只有一个实例

|  |
| --- |
| class MusicPlayer(object):  # 记录第一个被创建对象的引用  instance = None  # 记录是否执行过初始化动作  init\_flag = False  def \_\_new\_\_(cls, \*args, \*\*kwargs):  # 1. 判断类属性是否是空对象  if cls.instance is None:  # 2. 调用父类的方法，为第一个对象分配空间  cls.instance = super().\_\_new\_\_(cls)  # 3. 返回类属性保存的对象引用  return cls.instance  def \_\_init\_\_(self):  if not MusicPlayer.init\_flag:  print("初始化音乐播放器")  MusicPlayer.init\_flag = True |

1. \_\_new\_\_和\_\_init\_\_方法的调用顺序  
   先执行\_\_new\_\_方法，会返回为创建对象的引用，然后才会执行\_\_init\_\_方法
2. 了解异常的概念  
   程序在运行时，如果 Python 解释器 遇到 到一个错误，会停止程序的执行，并且提示一些错误信息，这就是异常
3. try except 捕获异常
4. else finally在异常中的作用  
   else是没有发生异常才会执行  
   finally是无论有没有异常都会执行
5. 如何抛出自定义异常（主动抛出异常）

创建一个 Exception 的对象，使用 raise 关键字抛出异常对象

## 03\_面向对象\_day04

1. 模块的概念，如何导入模块  
   每一个以扩展名py结尾的Python源代码文件都是一个模块, 在模块中定义的全局变量 、函数、类 都是提供给外界直接使用的工具，模块相当于是一个工具包

导入模块的两种方式：

① import 模块名1, 模块名2

import 模块名1

import 模块名2

② from 模块 import 工具/\*

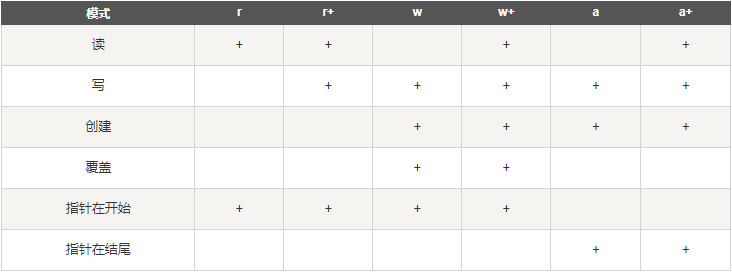
给模块起别名使用as关键字，模块别名使用大驼峰命名法

1. 理解包的概念  
   包是一个包含多个模块的特殊目录,目录下有一个特殊的文件\_\_init\_\_.py  
   使用 import 包名 可以一次性导入包中所有的模块
2. 文件的概念，文件的打开、读取、写入、关闭  
   计算机的文件，就是存储在某种长期储存设备上的一段数据  
   open 打开文件，并且返回文件操作对象

read 将文件内容读取到内存

write 将指定内容写入文件

close 关闭文件

1. 文件打开方式  
   
2. 按行读取文件内容  
   使用readline方法可以一次读取一行内容，方法执行后，会把文件指针移动到下一行，准备再次读取，适合读取大文件内容。
3. 文件的常用操作  
   rename 重命名文件 os.rename(源文件名, 目标文件名)

remove 删除文件 os.remove(文件名)

1. 目录的常用操作  
   listdir 目录列表 os.listdir(目录名)

mkdir 创建目录 os.mkdir(目录名)

rmdir 删除目录 os.rmdir(目录名)

getcwd 获取当前目录 os.getcwd()

chdir 修改工作目录 os.chdir(目标目录)

path.isdir 判断是否是目录 os.path.isdir(目录路径)

**path.isfile 判断是否是目录 os.path.isfile(文件路径)**

1. 设置文件编码为UTF-8的方式  
   Python2.x默认字符集为ASCII编码，不支持中文，Python3.x默认字符集为UTF-8编码  
   # \*-\* coding:utf8 \*-\*  
   # coding=utf8
2. eval函数的作用  
   将字符串当成有效的表达式来求值并返回计算结果

## 04\_项目实战

1. pygame的初始化和退出  
   pygame.init() pygame.quit() exit()
2. pygame.Rect(x,y,width,height) 描述矩形的类  
   包含以下属性(x、y、size、width、height、left、right、top、bottom、center、centerx、centery)
3. screen = pygame.display.set\_mode(resolution=(0,0), flags=0, depth=0) 初始化游戏显示窗口  
   返回的是一个屏幕数据对象，需要使用变量记录屏幕的结果，后续的图形都是绘制在屏幕窗口上
4. pygame.image.load(图形地址) 把图片从硬盘加载到内存
5. screen.blit(img) 方法可以在画布上绘制很多图像
6. pygame.display.update() 刷新屏幕内容显示，可以放在所有的blit方法之后
7. pygame.time.Clock 游戏时钟，可以控制刷新帧率  
   clock = pygame.time.Clock()  
   clock.tick(60) 一般在电脑上每秒绘制 60次，就能够达到非常连续高品质的动画效果
8. pygame.event.get() 可以获得用户当前所做动作的事件列表，返回的是一个列表
9. 在游戏开发中，通常把 显示图像的对象 叫做精灵 Sprite

精灵组是用来控制组内的精灵的，一个精灵组可以包含多个精灵对象，当调用精灵组对象的update()方法，可以自动调用组内每一个精灵的update()方法，调用精灵组对象的 draw(屏幕对象)方法可以将 组内每一个精灵 的 image 绘制在 rect 位置.  
pygame.sprite.Sprite —— 精灵类，存储 图像数据 image 和 位置 rect 的 对象

pygame.sprite.Group —— 精灵组类 控制所有精灵

1. 常量的定义：不会变化的值，在Python中没有真正意义的常量，只是通过命名约定

定义常量和定义变量的语法完全一样，都是使用赋值语句，常量的命名应该所有字母都使用大写，单词与单词之间使用下划线连接

1. 在 pygame 中可以使用 pygame.time.set\_timer() 来添加 定时器  
   set\_timer(eventid, milliseconds)
2. pygame.key.get\_pressed() 返回 所有按键元组
3. 碰撞检测  
   pygame.sprite.groupcollide()

两个精灵组 中 所有的精灵 的碰撞检测

pygame.sprite.spritecollide()

判断 某个精灵 和 指定精灵组 中的精灵的碰撞