Windows 安全性，稳定性都差

硬件，软件：操作系统

python开发外部服务器，linux

嵌入式：开发智能家居，机器人

操作系统是建立在硬件之上的

linux发行版——linux内核——系统调用——硬件

linux内核——终端命令——系统调用——硬件

内核只有一个，发行版可以有很多个

Linux目录结构：/home/python/...

所有用户都在home目录下，每人一个文件夹

终端命令：封装成图形界面

如ls,mkdir

帮助信息： command --help, man command

自动补全：tab

退出：ctrl+c

.开头文件是隐藏文件 ： ls -a 查看隐藏文件;ls -l 查看详细信息；ls -lh 进一步查看文件大小

\*代表任意数量字符（可以为0）；？代表一个字符

回到家目录（/home/joe）：cd/cd ~

最近两次的工作目录转换：cd -

默认cd 为当前目录下的某文件（相对路径）

创建多层目录： mkdir -p a/b/c/d

同一个工作目录下，文件与目录不可重名

删除目录以及该目录下所有内容： rm -r

树形图（只显示目录）：tree -d

复制时询问是否覆盖： cp -i ...... / mv -i ........

复制目录： cp -r ......

重命名：mv 源文件 目标文件

查看文件内容（内容少）：cat

查看文件内容（内容多）：more

编号非空行： cat -b

编号所有行： cat -n

搜索：grep

-显示行：-n

-取反：-v

-忽略大小写：-i

-指定追加：echo ... >> ...（>是输出且覆盖） 可在创建文件时编辑内容

管道：ls -lh|more

网卡相当于Sim卡，ip地址相当于手机号（为了找到某计算机）

域名即网址，是IP地址别名，方便记忆——可以通过IP地址或域名网址都可找到某网页

Ip:port 用IP地址找到计算机（服务器），用端口号找到该计算机上的应用程序

端口：

Web:80

SSH:22

远程控制服务器：ssh -p 22 user@remoteIP

远程拷贝文件：scp -P 22 01.py [user@remote:Desktop/01.py](mailto:user@remote:Desktop/01.py)

远程拷贝目录：scp -P 22 -r demo user@remote:Desktop

非对称加密算法：

公钥和私钥 公钥从客户端传到服务器

权限：rwx

硬链接数：

文件都是1；目录数量是 子目录+2（绝对路径、.）

目录有可执行权限

-sudo:substitute user do （变为root身份）

组的文件保存在 /etc/group

改变组：chgrp -R 组名 文件/目录名 （-R递归修改）

创建用户：useradd -m -g 组 用户名 （-m自动建立家目录；-g自动建立同名组）

修改附加组（增加权限）： usermod -G 组 用户名

/etc/passwd 是用来保存用户信息的文件（不可执行）

/usr/bin/passwd 是用于修改用户密码的程序

/bin 存储普通可执行文件

/sbin 存储和系统管理相关的可执行文件

切换用户:su - 用户名

改变权限：

-r:4 -w:2 -x:1

磁盘信息： df -h:剩余空间disk free ; du -h 目录名 ： 目录下文件大小 disk usage

进程信息：

-ps aux: process status

--a: 包括其他用户进程

--u ： 显示进程详细状态

--x: 显示没有控制终端的进程（使用较少）

显示CPU与内存占用高的程序：top

查找文件： find [路径] -name “名称”

创建软链接（快捷方式）： ln -s 源文件绝对路径 链接名 （-s:soft）

打包：tar -cvf 打包文件名.tar 路径

解包：tar -xvf 打包文件名.tar

打包压缩：tar -zcvf 打包文件名.tar.gz 路径

解包解压：tar -zxvf 打包文件名.tar.gz （-C）目标路径

用bz2格式压缩——>将z换成j

软件安装：apt:advanced packaging tool

编译型语言：统一编译，一次性执行（执行速度快）

解释型语言：每行翻译并执行（通过不同解释器可以跨平台）

操作系统让CPU把解释器复制到内存中，解释器读取硬盘中的程序代码，CPU翻译、执行

Python 2.x 用Python解释器

字符中有汉字，用python3解释器

过渡的第三方库，用Pytho 2.7解释器

IDE：integrated development environment

程序：处理数据

变量：存储数据（保存在内存中）

bool型：非0即True （if/while）

但在计算时True为1

input的数据类型是str

%6d:若不足6位，不足用0补齐（0不可省去）

循环有次数，需要有计数器i，且代码最后i +=1（计数一般从0开始） while i < 循环次数（无等于）

计数与索引都是从0开始

-break：在while中，满足某条件会退出（剩下的也不会执行）

-continue:在while中，满足某条件会回到最开始（剩下的不会执行）-----是为了排除某种情况 注意要在continue中重新计数（i +=1）

不换行：print（‘\*’, end=’ ’） end是替换换行而显示的内容

换行：print(‘ ’)

-while嵌套：行和列（行计数，列输出）

转义字符：

\t: 垂直对齐 end=’\t’

-import module（是文件中的一个代码文件）

函数格式：函数定义上方空2行；注释在函数下方用三个引号标注，并下方空一行；调用时可用快捷键ctrl+q查看（在其他代码中被调用时也可看到）

在函数名上可插入说明文档

参数：定义时的叫形参；调用时传入的叫实参

return之后代码不会执行

.pyc（compiled）——模块就直接编译，再调用时不用再一行行解释

-iterable 需要是一组数

-pop 默认弹出最后一个

-del 是将变量从内存中删除

函数是单独封装的，方法是对于对象而言的

元组：数据不能修改；一般用元组储存不同类型数据，列表储存同种类型；只有一个元素 （5，）

数据修改用列表，数据保护用元组

字典每个键值对占一行；合并字典：update；

字符串方法：

-isspace：空白字符：空格和制表符\t,\n,\r

unicode字符串：在键盘上无法输出

-isdecimal:数字

-isdigit:+符号

-isnumeric:+中文数字

-index 和 find 区别：find若字符串不存在，返回-1

-replace:会返回新字符串，不会修改原字符串

-split+join

逆序：num\_str[-1::-1] 开始索引是最后一个，倒序为-1

列表方法

-extend可追加列表；append是将列表作为一个独立元素加入

-for in:----else:有break就不会执行（是为了在遍历中找到某个目标）

-break的作用是找到了目标，后面的不再需要 -else是没有找到

-TODO注释 会标明任务事项（可在左下角用总视图查询）

-Shebang（#！+解释器完整路径）：放在主程序开头 +在终端给文件加执行权限

赋值（记录数据地址，类似于数据标签）后引用，就是在内存中寻找数据的地址

数据地址id（）是数字类型

调用函数是在引用内存中实参的数据地址，而非数据本身

可变类型（列表，字典）：用方法修改不会改变内存地址；但重新赋值会改变

字典的key只能为不可变类型

哈希：只接收不可变类型参数；返回整数（特征码且一一对应）

唯一性和字典的不重复性相对应：对key进行hash

函数可以访问外部的所有变量（全局变量）；但若要修改全局变量，需要global声明

-把所有全局变量放在最开头

代码结构：

-Shebang

-import

-全局变量（gl\_）

-def

-执行代码

return若有多个返回值，默认为元组，但不需要加括号；可以直接使用多个变量接收返回值（无需索引）

交换数字：

-其他变量

-算术（a=a+b,b=a-b,a=a-b）

-Python专有：a,b=b,a（右边是一个元组，默认省去括号）

列表的+=是extend方法，而非赋值

有默认值的缺省参数，需要放在所有参数最后

多值参数：\*(args)元组；\*\*(kwargs)字典

-拆包：在调用函数注明参数时需要用\*，\*\*来区分；但在函数内部直接用参数名

dir可以看到对象的所有的属性和方法

%d十进制；%x十六进制

is判断的是内存地址是否相同

面向对象：

-先架构出属性和方法（内部再区分类，对象，静态）

-静态方法也用类名调用

-不同对象的内存地址不同

-类函数的参数self就是调用该方法的对象

-\_\_str\_\_（return....）可以指定print对象的内容 一般可以替代name属性的打印

-都要有\_\_init\_\_和\_\_str\_\_

-名词属性不全都是外部传递的参数

-存在包含关系的多个类，先定义被包含的对象

-不知道初始属性的具体信息，可以用None；之后进行赋值修改

-\_\_age为私有属性，不能被外界访问，但能被类定义内其他函数调用

-若不同类内有相同的方法，可以将此方法提取出来，单独定义一个父类后继承

-对父类方法进行拓展：super().method

-子类可以通过父类的公有方法来访问父类的私有属性/方法

-若无父类，添加object

-类属性：如对创建的对象进行计数 count=0 ;\_\_init\_\_:count+=1

-类方法：@classmethod ;参数（cls）

-静态方法：不需要访问类或对象 @staticmethod ; 参数（）

-\_\_new\_\_分配空间;返回内存地址并传给\_\_init\_\_:return super().\_\_new\_\_(cls)

-创建对象时，会调用\_\_new\_\_和\_\_init\_\_

单例：

class MusicPlayer(object):

# 记录创建对象引用的内存地址

instance = None

def \_\_new\_\_(cls, \*args, \*\*kwargs):

# 1. 判断是否创建

if cls.instance is None:

# 2. 用父类object的方法分配空间

cls.instance = super().\_\_new\_\_(cls)

# 3. 返回引用的内存地址

return cls.instance

异常处理：（只在主程序中处理）

-except Exception as result: print(“未知错误 %s” % result)

-抛出异常：创建异常对象ex=Exception(“长度不够”)；主动抛出 raise ex

-import ... as ...（别名用大驼峰） DogModule

-import Module ;只需要某个方法：from ... import...

-\_\_file\_\_可以查看某模块文件地址

-\_\_name\_\_在当前文件是“\_\_main\_\_”;被当做模块导入后为模块名

-将所有需要调用的模块打包且在包内添加名为\_\_init\_\_.py的文件（pycharm 有package选项）

--在\_\_init\_\_中添加对外的模块

第三方模块：（pip是Python的包管理工具）

-sudo pip3 install pygame

-sudo pip3 uninstall pygame

保存是将内存中的内容写入磁盘中

读取文件：

file = open("README")

while True:

# read each line

text = file.readline()

if not text:

break

print(text)

file.close()

文件管理： import os

UTF-8是UNICODE编码的一种编码格式（1-6个字节表示一个字符）

-eval() 将字符串内部内容当做表达式执行，并返回结果

快捷键：

-将已有内容变为注释： ctrl+/

-向左减少缩进：shift+tab

-对于函数内部代码单步执行：step into F7

-字符串的名称rename（批量）:shift+F6

Pygame:

初始化：pygame.init()

结束（将内存中的pygame模块卸载）：pygame.quit()

游戏循环上方是初始化（创建窗口，绘制初始图像，游戏时钟,精灵和精灵组）

游戏循环下方：帧率，UI，碰撞检测，更新位置（Rect,blit,update）

-UI：事件监听，

精灵（sprite）：

-游戏初始化：创建精灵/精灵组

-游戏循环：update,draw(screen),pygame.display.update()

-属性：

--image（name）, rect, speed

-方法：

--update

常量：全部大写（替代固定的数字）

定时器：

-定义定时器常量eventid

-**初始化方法**中set\_timer（if event.type == eventid）

-**游戏循环**中，监听定时器事件

模块导入顺序：

-官方（random）

-第三方（pygame）

-应用程序

按键捕获：

-event.type == pygame.KEYDOWN and event.type == pygame.K\_RIGHT（必须一次一次的按）

-pygame.key.get\_pressed（可一直输出）

vi是Linux 中的文本编辑器

：是末行模式

i → Insert 模式，按 ESC 回到 Normal 模式.

x → 删当前光标所在的一个字符。

:wq → 存盘 + 退出 (:w 存盘, :q 退出)   （注：:w 后可以跟文件名）

dd → 删除当前行，并把删除的行存到剪贴板里

p → 粘贴剪贴板

-vi hello.py + 3（行数） 不加行数到末尾

:x， ZZ 或 :wq → 保存并退出 (:x 表示仅在需要时保存，ZZ不需要输入冒号并回车)

**末行模式：**

-:e . 编辑查看文件（类似ls）

-:n 新建文件

-:w 重命名(类似于另存为)

跳转：

gg → 到第一行。（注：相当于1G，或 :1） 数字gg/数字G

G → 到最后一行

翻页：

Ctrl + b 向上翻页

Ctrl + 飞向下翻页

H:head

M:middle

L:low

-w:向后一个单词

-b: 向前一个单词

-0：行首

-^:不是空白字符的行首

-$:行尾

{：上一段落

}：下一段落

%：可以在成对括号间切换

标记：

-mx

-’x :返回

-v:可视化模式

-V：可视化行模式 与{，}连用

-ctrl + v : 可视块模式（纵向）

-u(undo)

-ctrl + r (redo)

删除：

-x(cut)

-d(delete):与移动命令连用

-dd:删除行

-D:删除到行尾

-dw: 删除一个单词

-d5G:删到第5行

复制：

-y(copy)；y}

-nyy:复制n行

-p（paste）

文本缓冲区和剪切板不通用

替换：

-r:replace替换当前字符

-R：替换模式，一直向后替换

缩排：

右缩进：>>

左缩进：<<

重复执行：.

查找：/....

-n:下一个

-N：上一个

当前单词匹配：

-\*：向后找

-#：向前找

查找替换：(:%s/旧/新/g)

:%s/旧/新/gc （确认替换）

:s/旧/新/g（可视区替换）

插入命令：

-o:行后加入空行

-O：行前加入空行

-i：当前字符前插入

-I：行首插入

-A：行末插入

重复输入：

-数字n

-i编辑模式

-输入内容并退出

用可视化块给多行增加注释：

-ctrl + v

-I进入编辑模式

-输入内容并退出模式

分屏命令：

-:sp[文件名] 横向分屏

-:vsp[文件名] 纵向分屏

切换分屏窗口（每次先按ctrl + w）

-w:切换到下个窗口

-r:互换窗口

-c:关闭