python习题班

笔记本:	zymapple的笔记本		
创建时间:	2018/11/11 22:51	**更新时间:**	2018/11/20 1:34
作者:	519799178@qq.com		
URL:	https://jingyan.baidu.com/article/f3e34a12a25bc8f5ea65354a.html		

1.操作系统
-管理软硬件资源，给运行软件提供一个统一调用平台
-操作系统分类

- windows
- Unix
    - Minix, Linus-Linux
- MacOS

-如果内存大于等于8G，尝试使用虚拟机
-否则，windows
-如果是苹果系统，那就苹果系统
-虚拟机

- 虚拟的机器，是软件模拟出来的软件环境，是一个完全独立的系统
- 我们的虚拟机是xubuntu16.04　　LTS（长期支持版本）
- 使用步骤：
    - 下载打包好的虚拟环境，解压
    - 解压后会得到一个VDI格式的文件
    - 下载virtualbox，还有一个（vmware）
    - 容易出现的问题：
        - 没有64位系统以供选择：打开bios虚拟化
        - 打开后一直黑屏没反应：集成系统是64位，你集成的是62位

-自行安装
-anaconda和pycham

#GIT - 起源
--个人文件的版本控制
-- 多方协作
--linus做出了git
--2008年github上线
--关于版本控制的产品：

    - 集中式：cvs， svn
    - 分布式：github（开源），bitbuckets（私人）
    - 分享精神：blog+github

-git安装
-windows:
-mac os:
-linux: Ctrl+ait+t (打开终端窗口) 'sudo su'(这个命令是切换到管理员账户）网上安装的打包环境的密码是‘ddd123’

-GIT基本使用
- 创建仓库

- pwd （显示当前目录）
- cd（进入目录）
- mkdir learngit（创建一个目录，目录名为learngit）
- git init（初始化一个仓库）
- ll ， ls ， ll -a ，ls -ah ，ll -a ./.git ， （查看文件目录）
-
- touch file.txt（新建文件）
- vi file.txt(写入)
- cat file.txt (查看文件) vi file.txt(查看文件)
- git add <file>（添加至缓存区）
- git add . 添加文件目录下所有文件到缓存区
- git commit -m 'message for desc': (添加至仓库，一般注释信息要求强制添加)
- git push -u origin master 更新本地仓库文件到绑定的远程仓库
- git remote -v 查看绑定的远程仓库
- git remote add origin https://自己的仓库url地址 \ 将本地的仓库关联到github上
- q（退出）/wq（退出）
- git status :(显示git当前的状态)
- git rm file.txt: 删除信息
    - 删除后需要用 git commit -m 'xxxxxxxx' 来告诉仓库删除了文件（把删除这个动作提交给仓库）
- git log :查看日志
- git reset --hard 版本号
- 更全的GIT命令在下面这里
- http://www.cnblogs.com/chenwolong/p/GIT.html

-github or Bitbuckets
**我的github账号是 ewq654321 密码 qqwwaa654321
-github
-bitbuckets:使用方法类似

- 克隆XXgit到本地

****如何解决failed to push some refs to git

https://jingyan.baidu.com/article/f3e34a12a25bc8f5ea65354a.html

-python数据类型的内置函数
-str连接操作
-str乘法操作
-str当成列表来用

- a = '123456789'
- print(a[1]) = 2
- print(a[-1]) = 9
- print(a[1:5]) = 2345

-字符串方法

- capialize() 首字母大写，其余小写
- title() 将每个单词的首字母变为大写，其余变为小写
- upper() 将所有字母变为大写字母
- lower() 将所有字母变为小写
- swapcase() 大小写互换
- len() 计算字符串长度-len函数计算长度是计算字符数。一个汉字算一个长度单位
- find() 查找指定字符串，找不到返回-1，找到后返回索引值
- index() 查找指定字符串，找不到报错
- count() 计算字符串出现的次数
- startswith() 检测是否以指定字母开头-返回布尔值
- endswith() 检测是否以指定字母结束

- isspace() 检测字符串是否是空字符串
- isupper() 检测所有字母是否是大写字母
- islower() 检测所有字母是否是小写字母
- istitle() 检测是否以指定标题显示-每个单词首字母大写
- isalpha() 检测字符串是否是字母组成-返回布尔值（被字母包裹的汉字算作字母）
- isalnum() 检测字符串是否由字母和数字组成（不能出现其他字符，除了被包裹起来的汉字）-返回布尔值
- split() 用指定字符切割字符串-返回又字符串组成的列表
- splitlines() 用换行符切割字符串
- join() 将列表里的元素用指定字符串连接
- ljust() 指定字符串长度，内容靠左，不足的位置用指定字符串填充，默认空格
- rjust()指定字符串长度，内容居中，不足的位置用指定字符串填充，默认空格
- center() 指定字符串长度，内容靠右，不足的位置用指定字符串填充，默认空格
- strip() 去掉左右两边指定字符，默认是去掉空格
- lstrip() 去掉左侧指定字符，默认空格
- rstrip() 去掉右侧指定字符，默认空格
- maketrans() 生成用于字符串替换的映射表（两个参数长度要一致）----做一个映射表，有点像指点的键值对
    - translate() 进行字符串替换----与上方的maketrans联合使用

-list

- append() 向列表末尾追加新元素
- copy() 复制列表
- count() 计算某个元素在列表中出现的次数
- extend() 将一个列表继承另一个列表
- index() 获取值在列表中的索引
- insert() 在指定位置前插入元素
- pop() 根据索引移除列表内一个元素，不给索引默认移除最后一个元素
- remove() 移除列表中指定的值
- reverse() 列表原地翻转
- sort() 排序，默认从小到大排列

-tuple

- count() 计算某个元素在元组中出现的次数
- index()获取值在元组中的索引

-dict

- clear() 清除整个字典
- copy() 复制字典
- fromkeys() 按照指定的序列创建键值对中的键，值都是相同的
- get() 根据键获取指定的值-找不到如果有默认值则返回默认值，如果没有默认值，则返回None--找不到不会报错
- items() 键字典变成类似于元组的形式方便遍历
- pop() 移除字典中指定元素
- popitem() 移除字典的键值对，返回移除的键和值
- setdefault() 在字典里添加一个元素
- update() 修改字典中的值，用一个字典更新另一个字典

-set

- set() 工厂函数，创建集合
- add() 向集合中添加元素
- clear() 清空集合
- copy() 复制集合
- pop() 随机弹出一个元素（集合没有顺序，所以是随机弹出）
- remove() 删除集合中的某个值，如果值不在集合中会报错
- discard() 删除集合中的某个值，如果值不在集合中，也不会报错
- difference() 差集
- difference_update() 与上一个的区别是，上面的函数返回一个新的集合，这个函数是把原来集合覆盖掉

-编码问题

- 本质上计算机只能识别01代码
- 如何用一长串01代码表示复杂的信息

-编码简史

- 二进制
  - -bit：一个0或者1的二进制数字
  - -byte：八个01代码
- 第一阶段：ASCII
- 第二阶段：百花齐放，GB，BGK，BIG5 ，Latin1，JIS，
  - Latin1:兼容欧洲大多数语言

    - 韩国，台湾：BIG5
    - 中国：BGxxxx
    - 日本：JIS
    - ANSI-MSCS(Multi-bytes charecter set 多字节字符集)
    - 以上全部兼容ASCII码
- 第三个阶段：Unicode(ISO)

-编码表示方法

- ASCII-american standard code for information interchange
    - 所有的控制字符 （包括回车，删除等）编码在0-31范围之间以及127
    - 所有标点符号，英文大小写放在32-126之间
    - 预留128-255之间的位置（00000000-最高位被预留了）
    - 0xxx xxxx 是它的编码形式
- Latin1
    - 0-127的所有位置不动，那么可以兼容ASCII，二进制位0xxx xxxx
    - 128-255位置全部用完，二进制位1xxx xxxx
        - 128-159之间为控制字符
        - 160-255为文字符号
        - 其中包括了西欧语言，希腊语，泰语，阿拉伯语，希伯来语
    - 欧元符号（用Latin1表示不出来，当时欧元符号还没有诞生）
- GBxxxxxxxx
    - GB2312
        - 如果一个字节中第一位为0，那么这就是一个ASCII字符
        - 如果一个字节中第一位为1，那么这个是汉字，认定需要两个自己才能表示一个编码的文字
        - 这个码表中包含汉字6763个和非汉字图形字符682个
        - 还有很多的空间没有用到，没用到的全部预留了。
        - 0xxx xxxx：表示为ASCII字符
        - 1xxxxxxx 1xxxxxxx：表示为汉字
- GBK
    - 在GB2312基础上添加汉字
    - 兼容GB2312和ASCII
    - 0xxx xxxx：表示为ASCII字符
    - 1xxxxxxx 1xxxxxxx：表示为汉字
- GB18030
    - 2/4位混编

-Unicode码表

- 只是一个码表，具体实现没有规定
- 0-0x10FFFF来映射这些字符，最多可以容纳1114112个字符
- 中文编码范围是4E00-9FCF，其中9FC4-9FCF之间的区间没有使用
- 上述区间全都是汉字，不包含全角字符，不包含特殊文字
- UTF=UnicodeTransformationFormat
- UTF-8 UTF-16 UTF-32
    - UTF-16是早期遗留问题
    - UTF-32浪费空间
- UTF8 可变字长

-UCS-2

- UCS=Univeraslcharacterset,通用字符集
- UCS-2与Unicode相同
- 采用2个字节，定长的表示每一个字符，所以总计可以表示2**16个字符

-UCS-4

- 第一字节：表示组（group）128个
- 第二字节：表示平面（plane）256个
- 第三字节：表示行（row）256个
- 第四字节：表示马尾（cell）256个
- 如果USC-4前两个字节为0，这就是USC-2

-常用概念

- 编码/解码：由人类可直接读取信息转换成bytes格式的，叫编码
- 大尾（BigEndian）和小尾（LittleEndian）
    - ‘汉’ -->6c49
    - 6c49-->大尾（BigEndian）
    - 496c-->小尾（LittleEndian）
- BOM
    - UTF-8没有字节顺序问题（字节表示唯一标识符）
    - UTF-16会出现问题
    - BOM

-python编码问题

- str

- bytes
- bytearray

-ord（）查看码位
-chr（）查看码位

-python文件默认utf-8编码，如果特殊需要，需要声明

- 声明放在第一行，或者第二行
- '''#-*- coding: windows - 1252-*-'''
- 读写文件默认utf-8,可以指定
- code point 方式比较字符串，可能会带来问题
    - 重音符号的表示
    - 使用 unicodedata.normalize函数

-Python源码中出现了解码错误，那么会产生SyntaxError异常
-其他情况下，如果发现编码解码错误，那么会产生UnicodeEncodeError异常

-循环，函数