[第一章．Linux 5](#_Toc531801252)

[1.系统介绍 5](#_Toc531801253)

[三大操作系统： 5](#_Toc531801254)

[linux终端工具 6](#_Toc531801255)

[计算机的组成 6](#_Toc531801256)

[Linux 简介: 6](#_Toc531801257)

[Linux的基本结构 6](#_Toc531801258)

[2.Linux下的路径 7](#_Toc531801259)

[目录树 7](#_Toc531801260)

[UNIX/linux目录树 7](#_Toc531801261)

[路径 Path 7](#_Toc531801262)

[绝对路径 7](#_Toc531801263)

[相对路径 7](#_Toc531801264)

[开始符号 7](#_Toc531801265)

[vmware 快捷键 7](#_Toc531801266)

[快捷键 7](#_Toc531801267)

[tab 键 7](#_Toc531801268)

[上下键 8](#_Toc531801269)

[ctrl+l 8](#_Toc531801270)

[3.Linux 命令 8](#_Toc531801271)

[Linux命令格式 8](#_Toc531801272)

[$date命令 8](#_Toc531801273)

[--help命令 8](#_Toc531801274)

[man 命令 帮助 8](#_Toc531801275)

[$info命令 8](#_Toc531801276)

[cd命令： 8](#_Toc531801277)

[$ pwd命令 9](#_Toc531801278)

[$ ls命令 9](#_Toc531801279)

[mkdir命令 11](#_Toc531801280)

[rmdir命令 11](#_Toc531801281)

[touch 命令() 11](#_Toc531801282)

[rm命令 11](#_Toc531801283)

[通配符 12](#_Toc531801284)

[cp 命令（复制粘贴） 12](#_Toc531801285)

[mv 命令（剪切粘贴） 12](#_Toc531801286)

[nano命令 12](#_Toc531801287)

[Cat命令 13](#_Toc531801288)

[more命令 13](#_Toc531801289)

[less命令 13](#_Toc531801290)

[find 查找命令(文件查找) 13](#_Toc531801291)

[grep命令（文本内容查找） 13](#_Toc531801292)

[gzip命令 13](#_Toc531801293)

[gunzip命令 13](#_Toc531801294)

[数据压缩算法 14](#_Toc531801295)

[打包和解包的命令 14](#_Toc531801296)

[tar 命令 14](#_Toc531801297)

[用户管理权限 14](#_Toc531801298)

[sudo命令 14](#_Toc531801299)

[exit 命令 14](#_Toc531801300)

[reboot命令 14](#_Toc531801301)

[poweroff命令 14](#_Toc531801302)

[ping命令 15](#_Toc531801303)

[useradd命令 15](#_Toc531801304)

[passwd命令 15](#_Toc531801305)

[who命令 15](#_Toc531801306)

[whoami命令 15](#_Toc531801307)

[userdel命令 15](#_Toc531801308)

[alias命令 15](#_Toc531801309)

[unalias命令 15](#_Toc531801310)

[tree 命令 15](#_Toc531801311)

[磁盘管理 16](#_Toc531801312)

[df命令 16](#_Toc531801313)

[du命令 16](#_Toc531801314)

[第二章．Python第一天 17](#_Toc531801315)

[1.Python语言简介 17](#_Toc531801316)

[Python的应用领域 17](#_Toc531801317)

[Python的优缺点 18](#_Toc531801318)

[Python官网 18](#_Toc531801319)

[Python版本 18](#_Toc531801320)

[Python的运行 18](#_Toc531801321)

[2.Python程序 18](#_Toc531801322)

[python注释： 19](#_Toc531801323)

[python程序的组成 19](#_Toc531801324)

[Python的核心数据类型： 19](#_Toc531801325)

[整型数 Int 19](#_Toc531801326)

[浮点型数 float 20](#_Toc531801327)

[复数 complex 20](#_Toc531801328)

[布尔型数 bool 20](#_Toc531801329)

[表达式 和 运算符 21](#_Toc531801330)

[表达式 expression 21](#_Toc531801331)

[运算符 21](#_Toc531801332)

[算数运算符： 21](#_Toc531801333)

[复合赋值运算符 22](#_Toc531801334)

[比较运算符 22](#_Toc531801335)

[括中分组子表达式 22](#_Toc531801336)

[混合类型自动升级 （当表达式中含有浮点数时 值也一定是浮点数） 22](#_Toc531801337)

[基本输出函数 22](#_Toc531801338)

[变量 23](#_Toc531801339)

[赋值语句 statements 23](#_Toc531801340)

[Python 中 关联/绑定/引用的含义 23](#_Toc531801341)

[Python 中的自动化内存管理 24](#_Toc531801342)

[在交互模式下查看当前作用区域内的所有变量-help 24](#_Toc531801343)

[del 语句 24](#_Toc531801344)

[is / is not 运算符 24](#_Toc531801345)

[小整数对象池 24](#_Toc531801346)

[id(x) 函数 25](#_Toc531801347)

[第三章．Python第二天 27](#_Toc531801348)

[1.数值对象的构造（创建）函数 27](#_Toc531801349)

[函数调用： 27](#_Toc531801350)

[预置数值型函数 27](#_Toc531801351)

[2.语句 statement 28](#_Toc531801352)

[显式换行 28](#_Toc531801353)

[折行符 \(反斜杠) 28](#_Toc531801354)

[隐式换行 28](#_Toc531801355)

[基本输入输出 28](#_Toc531801356)

[基本输入函数：input 28](#_Toc531801357)

[基本输出函数：print 29](#_Toc531801358)

[if 语句 30](#_Toc531801359)

[if 语句的真值表达式 30](#_Toc531801360)

[if 语句的嵌套 30](#_Toc531801361)

[条件表达式 32](#_Toc531801362)

[pass语句 32](#_Toc531801363)

[布尔运算 32](#_Toc531801364)

[布尔非操作 not 32](#_Toc531801365)

[布尔与操作 and 33](#_Toc531801366)

[布尔或操作 or 33](#_Toc531801367)

[正负号运算符 33](#_Toc531801368)

[第四章．Python第三天 36](#_Toc531801369)

[字符串 str 36](#_Toc531801370)

[字符串表示方式： 36](#_Toc531801371)

[空字符串的字面值表示方法： 36](#_Toc531801372)

[非空字符串的表示方式： 36](#_Toc531801373)

[单引号双引号的区别： 36](#_Toc531801374)

[三引号字符串 36](#_Toc531801375)

[隐式字符串字面值拼接 37](#_Toc531801376)

[转义序列代表特殊字符 37](#_Toc531801377)

[ASCII编码表 37](#_Toc531801378)

[常用ASCII编码 37](#_Toc531801379)

[序列的概念 38](#_Toc531801380)

[求序列的长度的函数 len 38](#_Toc531801381)

[raw 字符串 38](#_Toc531801382)

[字符串的运算 38](#_Toc531801383)

[+号运算符用于拼接字符串 38](#_Toc531801384)

[+= 运算符 38](#_Toc531801385)

[\* 运算符 39](#_Toc531801386)

[\*=运算符 39](#_Toc531801387)

[字符串的比较运算 40](#_Toc531801388)

[in / not in运算符 40](#_Toc531801389)

[字符串的索引操作 41](#_Toc531801390)

[索引 index 41](#_Toc531801391)

[切片 slice 41](#_Toc531801392)

[字符串序列相关的函数 42](#_Toc531801393)

[字符串编码转换函数 42](#_Toc531801394)

[整数转为字符串的函数 43](#_Toc531801395)

[字符串的构造函数 str 43](#_Toc531801396)

[python3中常用的方法 43](#_Toc531801397)

[字符串的常用方法： 43](#_Toc531801398)

# 第一章．Linux

## 1.系统介绍

### 三大操作系统：

UNIX

Aix (IBM)

Mac OS X (Apple)

IOS (Apple移动端)

Linux

ubuntu

Redhat

CenterOS

android

Windows

Win3.1

Win3.2

Win98

...

Win10

### linux终端工具

打开方法：

1.点击图标

2.输入命令gnome-terminal,终端

退出终端：

1.$ exit

2.ctrl+D快捷键

### 计算机的组成

硬件

处理器(CPU)

运行内存(RAM)

主板(总线设备)

输入输出设备(显示屏，键盘，鼠标)

外部存储设置(硬盘, U盘)

软件

操作系统软件:

Windows, Linux, UNIX

应作软件:

QQ, 微信

### Linux 简介:

1991年 发布第一个公开的版本0.02版

### Linux的基本结构

1. 应用程序

2. 标准库

3. Linux内核

4. 硬件

## 2.Linux下的路径

### 目录树

目录结构都为树形结构（目录树）

### UNIX/linux目录树

/ 根目录（unix linux 都只有一个根目录 Windows 有很多根目录 如： C盘 D盘 E盘 F盘）

路径的分隔符： /

### 路径 Path

路径是用来记录一个文本或文件夹得字符串

第一个 /（root）根

### 绝对路径

以/开头 例： /home/tarena/aid1811

### 相对路径

不以/字符开头的路径为相对路径

相对是指相对于当前的工作目录（即pwd命令显示的目录）

### 开始符号

文件名或文件名称

**.** 当前文件夹（目录）

**..**上一级文件夹（目录）

**~**用户主目录（家目录）:操作系统为每个用户创建，由用户所拥有的目录

即ls~ 等于ls/home/tarena

如：

ls -l ../../etc/passwd

ls aid1811

ls -l ~

### vmware 快捷键

ctrl + alt 释放鼠标光标

ctrl + alt + enter 全屏/退出全屏

### 快捷键

ctrl+shift+C 复制

ctrl+shift+V 粘贴

### tab 键

补全命令名或路径

### 上下键

翻出以前的命令

### ctrl+l

快速清屏，等同于clear命令

## 3.Linux 命令

### Linux命令格式

命令名【选项】 【参数】

【选项】代表里面的内容可选 如： -a， –l

### $date命令

作用：输出当前系统日期

### --help命令

作用：用于显示命令的帮助信息

格式：命令 --help

如：

ls --help

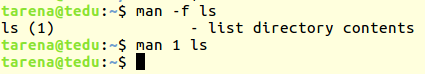
### man 命令 帮助

作用：查看命令对应的帮助手册

格式：man Linux/UNIX命令名

选项：

-f 查看所有同名的手册，再查特定手册

例：

退出键 q

### $info命令

作用：从首页开始浏览帮助手册，也可以浏览特定命令的帮助手册

### cd命令：

作用：用于改变当前的工作目录（进入某个目录）

格式：cd[目录名]

如：

cd/home/tarena

cd/

cd #切换到用户文件夹

cd .. #进入上一级目录

cd ~ #进入用户主目录

cd - #切换到进入这个文件夹之前的文件夹

### $ pwd命令

作用：用于显示当前操作路径（当前工作目录）

示例：

$ pwd

显示结果：/home/tarena

### $ ls命令

作用：显示指定目录的文件或文件夹信息

格式：

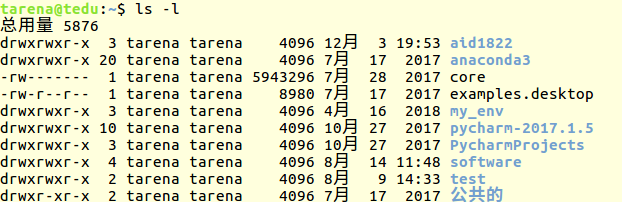
ls[选项]【文件夹名或文件名】

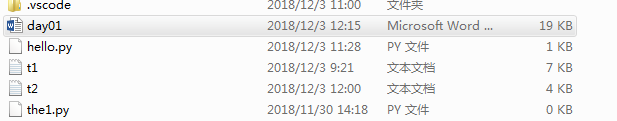
（Windows中 文件显示排列的方法）

常用选项：

**-l** 列表显示文件的详细信息

（Windows文件显示方法中的详情显示）

如图：



**-a** 显示全部文件、文件夹(显示以.开头的文件)

（Windows文件显示方法中的显示隐藏）

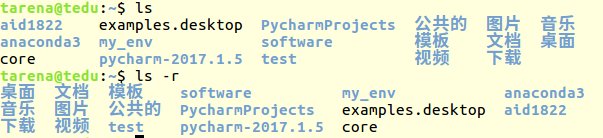
如图：



注：Linux、Unix下，以 **.** 开头的为隐藏文件

**-r**将文件以相反次序显示（原定依英文字母次序）

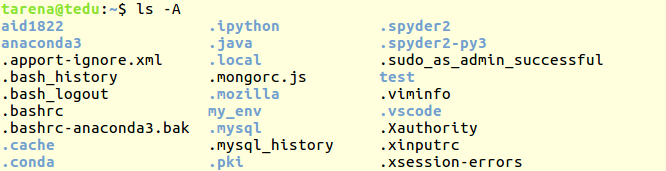
如图：



**-t**将文件依建立时间先后顺序列出

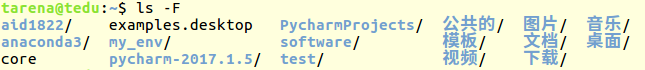
如图：

**-A** 同-a但不列出“.”（目前目录）“..”（父目录）

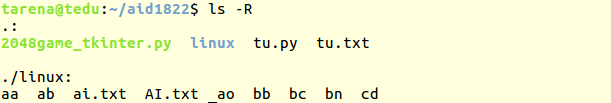
如图：

**-F** 在列出的文件名称后面加一符号

可执行文档加“\*”，目录则加“/”

 如图：

**-R** 若目录下有文件，则一下之文件亦皆依序列出

如图：

### mkdir命令

作用：创建一个或多个文件夹

选项：

-p 当父目录存在时忽略，若不存在则建立，用此参数可建立多级文件夹

如：

mkdir dir1 dir2 dir3 #多个文件夹用空格隔开

### rmdir命令

作用：删除一个或多个文件夹

选项：

-p 删除子目录，如果子目录删除后，父目录为空，则删除父目录

如：

rmdir dir1 dir2 dir3

注：用rmdir 删除文件夹时，文件夹内必须为空

### touch 命令()

作用：

1.如果文件不存在，则创建一个空文件 （长度为0字节的文件）

2.如果文件或目录存在，则用系统时间更新他的修改时间

格式：touch 文件名

示例：

touch newfile

### rm命令

作用：用于删除或删除文件夹

格式：rm [选项]文件或文件夹

常用选项

-r 递归删除文件夹内部的文件和文件夹

-i 删除前给出提示（y n）

-f 强制删除，不给提示

例：

rm -r a b c d

rm -i aa.text

rm -f bb.text

### 通配符

\* 代表0个或1个或多个任意字符

？代表任意1个字符

示例：

假设有文件 a ab ac abc aabb bc cd

rm a\* #等同于删除a ab ac abc aabb

rm a？ #等同于删除 ab ac

### cp 命令（复制粘贴）

作用：复制文件或文件夹

格式：cp【选项】 源文件或文件夹 目标文件或文件夹

常用选项

-a 复制文件夹里的内容

-d 复制文件时保留链接（类似快捷方式）

-f 覆盖已经存在的目标文件而不给出提示

-I 在覆盖目标文件之前给出提示

-p 除复制文件内容外，还把修改时间和访问权限也复制到新文件中

-r 若文件为目录文件，则复制该目录下的所有子目录和文件

-l 不复制文件，只生成链接

例 ：

cp -a text ~/桌面 #复制并粘贴到了桌面

cp -a day01.txt ~/t1.xt #在复制过程中还改了名字

### mv 命令（剪切粘贴）

作用：文件的搬移或者更名

格式：mv 源文件或文件夹 目标文件或文件夹

选项：

-I 若指定文件目录已有同名文件，则先询问是否覆盖

-f 不给提示直接覆盖

例 ：

mv test mytest #更名

mv mytest ~/

### nano命令

作用：打卡文本

注：按Ctrl+X会提示是否保存更

### Cat命令

作用：查看文本内容

格式：$ cat 文件名

注：它只能展示最后布满屏的内容，前面内容无法看见

### more命令

作用：逐行显示内容

格式：$ more 文件名

more+line从指定行号开始显示

注：Ctrl+c退出

### less命令

作用：上下滚动查看内容

格式：$ less 文件名

less+line从指定行号开始显示

注：q退出

### find 查找命令(文件查找)

作用：根据文件名等 信息查找指定文件

格式： find 路径 -name “文件名”

如:

find /etc -name passwd

### grep命令（文本内容查找）

作用:查找文件中相应的内容及文本信息

格式：grep “内容”【选项】文件名或路径

常用选项：

-n 显示行号

-r 递归搜索文件夹内的文件

例 ：

grep "tedu" -nr /etc

### gzip命令

作用：用zip压缩算法对文件进行压缩，生成压缩后.gz

格式：gzip 文件名

注：压缩后的文件名通常为.gz文件

### gunzip命令

作用：与gzip 相反 对用zip压缩算法压缩的文件进行解压缩  
格式：gunzip 文件名

示例：

$gzip a.txt

$ls

$gunzip a.txt.gz

### 数据压缩算法

分两类

有损压缩

mp3 ,jpg mp4...

无损压缩

zip .gz ,png ,gif ,rar ,xz

### 打包和解包的命令

### tar 命令

作用：对文件或文件夹进行打包的解包操作

格式：tar [选项]文件名【文件名或路径】

常用选项

-czvf 文件名 创建包并进行压缩

-xzvf 文件名 解压缩包到当前文件夹

如：

$cd ~t1.txt

$tar -czvf aid1811.tar.gz aid1811

$cp aid1811.tar.gz mytest/

$cd mytext

$tar –xzvf

### 用户管理权限

### sudo命令

作用：用超级用户root 权限来执行这些命令

格式：sudo 命令【选项】【参数】

常用选项

-i 切换到root用户

示例：

mkdir /mysext #会出错

sudo mkdir /mysext

### exit 命令

退出用户登录

### reboot命令

作用：重启

### poweroff命令

作用：关机

### ping命令

作用：测试网络连通情况，会向目标机器发送数据包来测试是否连通，是否延时

例：

$ping baidu.com

### useradd命令

作用：添加一个普通用户

选项：

-m 创建home目录

-g 所属主组

-G 指定用户在那些附加组

-s 设定默认的shell

注：需要先root

### passwd命令

作用：改变用户登录的密码

不带参数更改的时当前用户名

在root下可直接指定对应的用户进行更改

### who命令

作用：显示当前登录系统的用户

选项：

-H 显示标题栏

-l-H 显示用户登录来源

-T-H 显示终端属性

### whoami命令

作用：显示自身用户名称

### userdel命令

作用：删除用户账号（需root权限）

### alias命令

作用：设置指令的别名

例：

$ alias lx = ls

### unalias命令

作用：删除全部别名

### tree 命令

作用：以缩进的结构显示目录树

格式：tree 文件夹名

t1.txt如

tree aid1811

说明：此命令只能在linux下使用

### 磁盘管理

### df命令

作用：检查文件系统的磁盘空间占用情况

选项：

-h 将容量结果以易读的容量格式显示出来

-aT 将系统内的所有特殊文件格式及名称都列出来

例：

$ df –h 文件夹名 #将文件夹底下的可用磁盘容量以易读的容量格式显示

### du命令

作用：查看对文件和目录磁盘使用的空间，每个目录都会列出，包含隐藏文件的目录

选项：

-a 将文件的容量也列出

-sm 检查根目录底下每个目录所占用的容量

例题：

在主目录创建一个新的文件夹aid1811

先到达主目录 cd ~

新建文件夹 mkdir aid1811

在新文件夹aid1811内再创建一个新文件夹linux

到达aid1811 cd aid1811

新建文件夹linux mkdir linux

在新文件夹中创建3个文件 a b c

到达linux cd linux

创建 a b c touch a b c

删除第一个文件

删除 a rm a

练习：

1.在新文件夹aid1811内再创建一个新文件夹pbase

2.在pbase文件夹内创建 day01 day02 day03 三个文件夹

3.在day01 文件夹中创建linux code两个文件和day01.txt

4.删除aid1811/pbase/linux文件夹

5.讲aid1811/pdase/day01/day01.txt复制到day02/文件夹下 并改名为day02

练习键盘

2048game\_tkinter.py文件名

2048game\_tkinter.py 的运行方法

# 第二章．Python第一天

## 1.Python语言简介

创建时间：1989圣诞节期间

### Python的应用领域

系统运维（shell）

网络编程（搜索引擎，爬虫，服务器编程）

科学计算（Fortran77， R）

人工智能，机器人

web开发

云计算

大数据及数据库编程

教育

其他

### Python的优缺点

优点：

面向对象 ：Java c++ Python

免费

可移植

可混合编程（c c++ Java .net）

简单易学

开发效率高

应用领域广泛

开源

缺点：

与c与c++相比，执行速度不够快

不能封闭源代码

### Python官网

www.python.org

### Python版本

Python v2.7

Python v3.5

Python v3.7

### Python的运行

Python的解释执行器的类型

CPython (c语言开发)

$ python3

JPython (Java语言开发)

IronPython （.net开发）

## 2.Python程序

1.编写xxx.py文件

$ code xxx.py

2.执行Python程序

$ Python3 xxx.py (linux下执行方法)

^ ^

| |

解释执行器 被解释文件

3.执行Python程序方法2(仅用于linux、Unix)

1） 修改 xxx.py 在第一行写入

#！ /usr/bin/python3

2)为xxx.py添加属性

$chmod +x xxx.py

注：chmod 命令用来添加修改文件的权限

3) 执行：

./xxx.py

常用程序文件后缀：

.py python语言源文件

.c c语言

.cpp /.cxx /.cc /.c c++语言

.java JAVA语言源文件

在Python交互模式下执行的Python语言代码

1.进入交互模式

$ python3

>>>

2.退出交互模式：

>>> quit() 或exit() 或ctrl+d

### python注释：

Python注释是以#开头直至行尾

作用：

让注释内容不参加解释执行

### python程序的组成

程序由模块组成

模块由数据、函数、类、语句等组成

语句包含表达式

表达式建立并处理数据对象（最小单位）

### Python的核心数据类型：

数字（整型、浮点型、复数、布尔型）

字符串

列表

...

### 整型数 Int

整型数是不带有小数部分的数字，包含自然数，0 ，及负数

如：-5 0 100 9999

整数字面值得表达方式：

十进制方式表示：

如：10， 100 -99999 #注：整数可以很大

二进制表示（0b开头，后跟0或1）

如：0b111 #等同于7 0b1010 #等同于10 1\*2^3+1\*2^2+1\*2^1+1\*2^0

八进制表示方式（0o开头，后跟0~7）

如：0o177 #等同于127 0o11 #等同于9 到8进1 二进制转八进制：从右到左每3位算1位

十六进制表示方式（0x开头，后跟0-9，A-F或a-f）

注：a表示10，b表示11，...f表示15 到16进1

如： 0x11 #17 0xFF #255 二进制转十六进制：从右到左每4位算1位

练习：

计算：

0xFFFF 表示多少？

0xFFFFFFFF 表示多少？

0b1001 表示多少？

0o71 表示多少？

### 浮点型数 float

浮点型数是带有小数部分的数字（小数部分可以为0）

浮点数字面值表示方法：

小数表示法：

3.14 3.1 3.0 3. 0.14 .14

科学计数法：

格式：

小数e/E（正负号）指数

如：

6.18E-1 = 6.18\*10^-1 等同于0.618 2.9979E8 = 2.9979\*10^8 等同于299790000.0

### 复数 complex

分为两部分：

实部（real）

虚部（image）

虚部是以j或J结尾的数

如：1j (2j) 1.3+1.8j 2-2j (-100+100j)

### 布尔型数 bool

用来表示真和假两种状态类型

True 表示真（条件满足可成立）

False 表示假（条件不满足或不成立）

说明：

True 的值为1

False 的值为0

空值对象 None

None 是一个表示不存在的特殊对象

作用：

用来占位

变量解除绑定

### 表达式 和 运算符

### 表达式 expression

由一个数字或数字和运算符组成

作用：

通常让计算机做一些事情并返回结果

如： 1+2\*3 None 1+5 > 2+3 9999

### 运算符

### 算数运算符：

+ 加法 - 减法 \* 乘法 / 除法 // 地板除 % 求余 \*\* 幂运算

+ 加法：

除了数值外，字符串也可以使用 +加法，但字符串不可以与数值相加，只有字符串与字符串之间可以用

/ 除法：

除法得到的数是浮点数，不会丢弃小数部分 （整数相除也是浮点数）

例：1/3 #得0.3 8/2 #得4.0

// 地板除

除的结果去掉小数部分向下取整（当除数，被除数中含有浮点数 得到的值也为浮点数）

例 7/3 #得2.33333333 7//3 #得2

% 求余

7%3 #得1 3.25 % 1.25 #得0.25

\*\* 幂函数

格式：

x\*\*y #意为x得y次方

如： 4\*\*2 #得16 3.0\*\*4.0 #得81.0

练习

1. 一个商店买西瓜7元一个，你带96元能买几个西瓜，找零是多少？

答：个数96//7=13 找零 96%7=5

2.一个学生毕业薪资是10000元，每年涨20%，10年后它的薪水是多少？

答：10000\*（1+0.2）\*\*10

10000+(10000\*0.2)= (1 + 0.2)\*10000 #第一年

（1+0.2）\*10000+（（1+0.2）\*10000）\*0.2 =（1+0.2）\*10000\*（1+0.2）=10000\*（1+0.2）\*\*2 #第二年

所以第10年的答案为 10000\*（1+0.2）\*\*10

运算符优先级

\*\* > \* / // % > + -

### 复合赋值运算符

+= -= \*= /= //= %= \*\*=

### 比较运算符

< <= > >= == !=

语法

左表达式 < 右表达式

说明：

比较运算符返回布尔值类型的值

示例：

1 < 2 #True 1 <= 2 #True 1 == 2 #False x = 60 0 <= x <= 100 #True

### 括中分组子表达式

用（）可以将表达式分组，（）内部的表达式先进行运算

如：2\*（3\*（4+8）-1）

### 混合类型自动升级 （当表达式中含有浮点数时 值也一定是浮点数）

1+2.14 #得3.14

### 基本输出函数

print函数

格式：

print（表达式1，表达式2，表达式3....）

作用：

将表达式的结果作为终端输出

练习

1.打印下图

\*\*\*\*\*\*\*\*\*

\* \*

\* \*

\*\*\*\*\*\*\*\*\*

答：print("\*"\*9)

print("\*"," "\*5,"\*")

print("\* \*")

print("\*\*\*\*\*\*\*\*\*")

2.已知矩形长为6cm 宽为4cm用表达式表示周长和面积 并打印出来

答：print("周长为:",(6+4)\*2,"cm"," 面积为：",6\*4,"cm^2")

# vs code 注释快捷键 Ctrl + /

### 变量

什么是变量

变量是一个关联一个对象的符号

变量可以绑定一个对象，并通过变量名来使用这个对象

变量名必须是标识符

标识符的命名方法

1. 标识符必须为字母或下划线开头，后跟字母下划线或数字

A-Za-z 或 \_

2. 不能使用Python的关键字

3. 变量名区分大小写

合法示例： a a1 vvv bbb\_ccc \_aaa\_ccc \_AvCb var

不合法示例： 1a 123 $asd +a #abc -avf A#v

Python 中的关键字

True False None is del if elif else pass not ...

### 赋值语句 statements

运算符 =

赋值语句语法：

变量名 = 表达式

变量名1 = 变量名2 = 变量名3 = 表达式

变量名1 ，变量名2 ，变量名3 = 表达式1 ，表达式2 ，表达式3

作用：

创建一个变量或修改一个变量绑定的数据

赋值语句说明：

当变量不存在时，创建该变量，同时将变量绑定到这个对象上

当变量存在时，改变此时变量绑定（关联）的对象

一个变量只能定一个对象

两个变量可以同时绑定同一个对象

示例：

a = 10 #创建一个变量a 绑定10

b = 20 #创建一个变量b 绑定20

a = 30 #修改变量a 绑定10

a = b #修改变量a，绑定b的赋值

a = b = c = 1000 # a,b,c 同时绑定1000

x , y = 10 , 3.14 # x绑定10 y绑定3.14

### Python 中 关联/绑定/引用的含义

在Python中，变量没有类型

关联/绑定/引用都是指变量和一个对象的关联关系

任何变量一定会绑定一个对象

### Python 中的自动化内存管理

示例：

a = 1000

b = 2000

a = a + b

c = a

练习

指定一个半径为3cm的圆

1）计算周长为多少？

2）计算面积为多少？

答：pi = 3.14

r = 3

c = 2\*r\*pi

s = pi\*r\*\*2

print("圆的周长为：", c ,"cm")

print("圆的面积为：", s ,"cm^2")

### 在交互模式下查看当前作用区域内的所有变量-help

>>> help("\_\_main\_\_") # 两个“\_"main“\_"两个

退出键：q

### del 语句

作用：用于删除变量，同时解除与对象的关联，如果可能则释放对象

语法：

del 变量名1 ，变量名2，...

如：

x , y , z = 100,200,300

### is / is not 运算符

作用：

判断两个对象是否是同一个对象，当是同一个对象时返回True

否则返回False

is not 的作用与is相反

语法：

x is y

x is not y

注：x，y代表对象

在交互模式下：a = 10000 b = 10000 a is b 显示 False

在解释器执行下： a = 10000 b = 10000 a is b 显示 True

### 小整数对象池

CPython中，整数-5至256的数永远存在于小整数对象池中

不会释放并可重复使用

所以在CPython中，a = 5 b = 5 a is b 显示 True

### id(x) 函数

作用：返回一个对象在内存中的地址

a = b = 100 # 他们的id地址一样，所以 a is b 显示 True

练习

1.在终端中输出图形

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

答：a = "\*"

b =" "

print(b\*4+a+b\*4)

print(b\*3+a\*3+b\*3)

print(b\*2+a\*5+b\*2)

print(b+a\*7+b)

2.中国古代的秤是16两一斤，请问古代的216两是古代的多少斤，写出程序并打印？

答：x = 216

y = 16

z = x // y

o = x % y

print("为",z,"斤",o,"两")

3.从凌晨0:0:0计时，到现在过了63320秒，请问现在是几时，几分，几秒，写出程序并打印（可以用地板除和求余实现）

答：# 1分 = 60 秒 1小时 = 3600秒

a1 = 63320

a2 = a1 // 3600 # 取整小时

a3 = a1 % 3600 // 60 #取余再取整分 a1 // 60 % 60

a4 = a1 % 3600 %60 #全部取余留秒 a1 % 60

print(a2,"时",a3,"分",a4,"秒")

print(" ")

4.温度转换（华氏温度/摄氏温度/开式温度）

算法：摄氏温度 = 5.0 / 9.0 \*（华氏温度 -32）

开式温度 =摄氏温度 + 273.15

问： 100华氏温度转为摄氏温度为多少？

转为开式温度为多少？

答：b1 = 100

b2 = 5.0/9.0\*(b1-32)

b3 =b2 + 273.15

print("摄氏温度为：",b2,"开式温度为：",b3)

# 第三章．Python第二天

## 1.数值对象的构造（创建）函数

float (obj) 用字符串或数字转换为浮点数 如不给出参数返回0.0

int(x,base = 10)或int(x = 0)用失踪或字符串转换为整数，如果不给出参数则返回0 ，base为进制

int("1000.5") #会出错 int("1A",16) #按16进制看1A

complex(r = 0.0 , i = 0.0)用数字创建一个复数（实部为r,虚部为i）

complex(100,23.5) #(100+23.5j)

bool(x) 用x获取一个布尔值（True/False）

bool 返回假值得情况：

bool（x）用于显示获取x的布尔值

值 说明

None 空值

False 布尔假值

0 0.0 0j 所有的数字0

[] 空列表

'' 空字符串

() 空元祖

### 函数调用：

函数名（传参列表）

说明：

函数调用是用表达式，一定会返回一个对象的引用关系（或者返回None）

练习

将数字3.14 用变量pi绑定

将pi变量转换为整数，用变量i绑定

将pi变量与i变量相减，结果用f绑定

1.判断f会否等于0.14

2.删除所有变量

答：pi = 3.14

i =int(pi)

f = pi - i

print(f==0.14) #显示False

del pi , i , f

### 预置数值型函数

abs(x) 取x的绝对值

round(number[,ndigits])对数据进行近似的值“四舍五入”，ndigits是小数向右取整的位数，负数表示向左取整

例：round(2.957) # 得3 round(120.5564,2) # 得120.56 round(2.5) # 得2

注：[]表示里面的内容可省略

pow(x,y,z=None) 相当于x\*\*y或 x\*\*y%z

例：pow(2,3,5) #相当于2\*\*3%5

help()函数查看帮助文档

>>>help(函数名)

如：

>>>help(abs)

## 2.语句 statement

语句由一些表达式组成，通常一条语句可以独立执行来完成一部分事情并形成结果

说明：

一条语句建写在一行内

多条语句写在一行内需要分号(;)分开

示例：

x = 100+200

print(x)

#等同于 x = 100+200; print(x)

### 显式换行

### 折行符 \(反斜杠)

折行符必须放在一行的末尾，来示意解释执行器，下一行也是本行的语句

例：

x = 1+2\*3+4\

+63+123+12.35556555 #等同于 x = 1+2\*3+4+63+123+12.35556555

### 隐式换行

所有的括号内换行，称为隐式折行

括号：() [] {}

例：x = 1+2+36+456+(453+456

+4568+775\*56)+456 #等同于 x = 1+2+36+456+(453+456+4568+775\*56)+456

### 基本输入输出

### 基本输入函数：input

input('提示字符串') 返回用户输入的字符串

说明：

提示字符串可以为空

返回值

字符串

示例：

s = input('请输入半径：') #阻塞函数，等待用户输入

r = float(s)

pi =3.14

c =pi\*r\*2

print("周长为：",c)

### 基本输出函数：print

作用：将一系列的值以字符串开始输出到标准输出设备上，默认为终端

格式:

print(v,...,sep='',end='\n',flush=False)

关键字参数：

sep：两个值之间的分隔符，默认为一个空格

end：输出完毕后在字符串末尾自动追加一个字符，默认为换行符'\n'

flush：是否立即输出到屏幕上，默认为False，不立即输出

例：

print（）#打印一个空行

print（1,2,3,4） # 1 2 3 4 等同于 print(1,2,3,4, sep=' ')

print(1,2,3,4, sep='') #1234

print(1,2,3,4, sep='#') # 1#2#3#4

print(1,2,3,end='')

print(4,5,6 end='') #得到 1 2 34 5 6 在末尾没有换行

练习

输入两个整数，分别用变量x ，y绑定

1.计算这两个数的和并打印结果

2.计算这两个数的积并打印结果

3.计算x的y次方并打印

答：x=int(input("请输入x:"))

y=int(input("请输入y："))

a=x+y

b=x\*y

c=x\*\*y

print(x,'+','y','=',a)

print(x,'\*','y','=',b)

print(x,'\*\*','y','=',c)

### if 语句

作用：让程序根据条件选择性的执行某条语句或某些语句

语法：

if 真值表达式1：

语句块1

elif 真值表达式2：

语句块2

else 真值表达式3：

语句块3

说明：

elif 子句可以有0个，1个或多个

else 子句可以有0个或1个，且只能放在此if语句最后

### if 语句的真值表达式

if 100:

print('真值')

等同于：

if bool(100):

print('真值')

### if 语句的嵌套

if 语句本身是由多条子句组成的一条复合语句

if 语句本身可以作为语句嵌套到另一个复合语句内部

例：month = int(input('请输入1-12月中的一个整数：'))

if 1<=month<=12:

if month<=3:

print('你输入的是第一季度')

elif month<=6:

print('你输入的是第二季度')

elif month<=9:

print('你输入的是第三季度')

else:

print('你输入的是第四季度')

else:

print('你输错了')

练习

任意输入一个数字

1）判断这个数是否大于100

2）判断这个数是否小于0

3）判断这个数是否在50-15之间

答：num=int(input('请输入一个整数：'))

if num>100:

print(num,'大于100')

if num<0:

print(num,'小于0')

if 150>num>50:

print(num,'在50-150之间')

1.输入一个季度1-4，输出这个季度有那几个月，若果用户输入的不是1-4，则提示用户出错了

答：ji= int(input('请输入1-4个季度中的一个：'))

if ji==1:

print('第一季度包含月份为：1月，2月，3月')

elif ji==2:

print('第二季度包含月份为：4月，5月，6月')

elif ji==3:

print('第三季度包含月份为：7月，8月，9月')

elif ji==4:

print('第四季度包含月份为：11月，11月，12月')

else:

print('你输错了')

2.输入一年中的月份1-12输出这几个月在那几个季度，若果用户输入的不是1-12，则提示用户出错了

答：month = int(input('请输入1-12月中的一个整数：'))

if 1<=month<=3:

print('你输入的是第一季度')

elif 4<=month<=6:

print('你输入的是第二季度')

elif 7<=month<=9:

print('你输入的是第三季度')

elif 10<=month<=12:

print('你输入的是第四季度')

else:

print('你输错了')

练习

写一个程序，输入一个整数，用if语句计算这个绝对值，并打印（不用abs）

答： num = int(input())

if num > 0:

print(num)

elif num<0:

print(-num)

else:

print(0)

### 条件表达式

语法：

表达式1 if 真值表达式 else 表达式2

#为True运行表达式1，为False运行表示2

作用：

根据真值表达式的取值（True/False）来决定执行表达式1或表达式2 并返回结果

例：mony= int(input('请输入：'))

mony=mony - 20 if mony>100 else mony

print(mony)

练习

写一个程序，输入一个整数，用条件表达式计算这个绝对值，并打印（不用abs）

答：num = int(input('值'))

num=-num if num<0 else num

print(num)

### pass语句

作用：通常用来填充语法空白

语法：pass

例：score = int (input())

if 0<=score <=100

pass

#print('成绩合法') 必须填有表达式

else:

print('成绩不合法')

### 布尔运算

运算符

not and or

### 布尔非操作 not

语法：

not x

注：x代表表达式

作用：

对x进行布尔值取非，bool(x) 为True则返回False，否则返回True

例：

not True #False

not False #True

### 布尔与操作 and

语法：

x and y

注：x，y代表表达式

作用：

优先返回假值对象

当x的布尔值为False，返回x，否则返回y

例：true and True #True

True and False #False

False and True #False

False and False #False

100 and 3.14 #表达式为对象 因100为True 所以返回3.14

0 and 3.14 #因0为False 所以返回0

### 布尔或操作 or

语法：

x or y

作用：

优先返回真值对象

当x为True时返回x，否则返回y

例：

true and True #True

True and False #True

False and True #True

False and False #False

100 or 0 #100

0 or 0.0 #0.0

100 and 3.14 #100

### 正负号运算符

+（正号） -（负号）

一元运算符（一个元素参加运算）

语法：

+表达式

-表达式

例：

a=5

b=-a

c=+a

Ctrl+d 选择全部相同

作业

1.北京出租车计价表

收费标准：

1.3公里以内收费 13元

2.基本单价2.3元/公里（超过3公里以外）

3.空驶费：超过15公里后，每公里加收单价的50%空驶费（即3.45元/公里）

要求：输入公里数，打印除费用金额（以元为单位精确到分）

答：lu = int(input("请输入行驶公里："))

if lu <=3:

print('车费为13元')

elif 15>=lu>3:

money=13+(lu-3)\*2.3

money=round(money,2)

print('车费为',money,'元')

else:

money = 13+12\*2.3+(lu-15)\*3.45

money=round(money,2)

print('车费为',money,'元')

2.输入三个任意的数：

1）打印最大数是多少？

2）打印最小数是多少？

3）打印三个数平均值是多少？

答：a = int(input("请输入第一个数字："))

b = int(input("请输入第二个数字："))

c = int(input("请输入第三个数字："))

if a>b:

if a>c:

print('最大值为',a)

if b>c:

print('最小值为',c)

else:

print('最小值为',b)

else:

print('最大值为',c)

print('最小值为',b)

else:

if b>c:

print('最大值为',b)

if a>c:

print('最小值为',c)

else:

print('最小值为',a)

else:

print('最大值为',c)

print('最小值为',a)

z = (a+b+c)/3

print('平均值为：',z)

方法二：

答：a = int(input("请输入第一个数字："))

b = int(input("请输入第二个数字："))

c = int(input("请输入第三个数字："))

max=a

if b>max:

max=b

if c>max:

max=c

print(max)

3.BMI指数（body mass index）又称身体质量指数

BMI计算公式：BMI=体重（公斤）/身高的平方（米）

标准表：

BMI<18.5 体重过轻

18.5<=BMI<24正常范围

BMI>=24 体重过重

输入身高体重，打印出BMI值，并打印体重状况

答：ht = float(input('请输入身高：（m）'))

wt = float(input('请输入体重：（kg）'))

BMI = wt / ht\*\*2

if BMI < 18.5:

print('BMI值为：',BMI,'体重过轻')

elif 18.5<=BMI<24:

print('BMI值为：',BMI,'正常范围')

else:

print('BMI值为：',BMI,'体重过重')

# 第四章．Python第三天

### 字符串 str

作用：用来记录文字（文本）信息

### 字符串表示方式：

在非注释中凡是用引号括起来的部分都是字符串

' 单引号

" 双引号

''' 三单引号

""" 三双引号

### 空字符串的字面值表示方法：

''

""

''''''

""""""

注：空字符串的布尔测试值bool（x）为False

### 非空字符串的表示方式：

'hello'

"hello"

'''hello'''

"""hello"""

### 单引号双引号的区别：

单引号内的双引号不算结束符号

双引号内的单引号不算结束符号

示例：

print("i'm a teacher") #i'm a teacher

print('i am "bear"') #i am "bear"

print('''i'm a teacher.i am "bear"''')

### 三引号字符串

以"""或'''开头和结尾的字符串

作用：

三引号内可以包含单引号和双引号

三引号字符串中的换行符会自动转换为\n

例：

print(""" i'm a teacher

i like python

i'm studing """)

#print("i'm a teacher.\n i like python \n i'm studing")

### 隐式字符串字面值拼接

多个字符串字面值连在一起时，Python解释执行器会自动拼接为一个字符串

示例：

s= "i'm a teacher" 'i am "bear"'

print(s) #i'm a teacher"i am "bear"

### 转义序列代表特殊字符

字符串字面值中用字符反斜杠（\）后跟一个或一些字符代表特殊的一个字符

转义格式 含义

\' 单引号（'）

\" 双引号(")

\\ 一个反斜杠(\)

\n 换行符

\r 返回光标至行首（回车符）

\f 换页

\t 水平制表符

\v 垂直制表符

\b 倒退

\a 响铃

\0 空字符（字符值为0）

\0oo oo为两位八进制表示的字符

\xXX XX为两位十六进制表示的字符

\uXXXX XXXX表示四位十六进制表示的UNICODE16字符

\UXXXXXXXX XXXXXXXX表示八位十六进制表示的UNICODE32字符

例：

print('ABC\n123')

print('ABC\t123')

print('ABC\rab')

print('ABCDE\babcd')

print('==\x41\x42\x43==') #print('==ABC==')

print('==\0141\0142\0143==')

### ASCII编码表

$ man ascii

### 常用ASCII编码

字符 十进制 十六进制

'0' 48 0x30

'A' 65 0x41

'a' 97 0x61

python3中字符串默认存储的是UNICODE编码值

0x0000~ (从0开始)

### 序列的概念

字符串是序列

序列是指有先后顺序的排列

### 求序列的长度的函数 len

len(s)返回字符串中字符的个数

练习

得到如下字符串中有几个字符

1）'abcd1234' #8

2) '5\'4"' #4

3) '\"\x34\056' #3

4) '\a\b\c\d' #6 转义格式中没有\c ,\d

5) '\n\t\r\\' #4

### raw 字符串

格式：

r'字符串内容'

r"字符串内容"

r'''字符串内容'''

r"""字符串内容"""

作用：

让转义符号\无效

示例：

用字符串形成一个Windows下的路径：

C:\newfile\test.pyth

path =r'C:\newfile\test.pyth'

print(path) #C:\newfile\test.pyth

### 字符串的运算

运算符：

+ += \* \*=

### +号运算符用于拼接字符串

x='abcd'

y='1234'

z=x+y #拼接

print(z)

a=y+z

print(a)

### += 运算符

用原字符串与右侧的字符串拼接生成新的字符串，再用原变量绑定新的字符串

例：

x='abcd'

x+='123'

print(x) # 'abcd123'

### \* 运算符

生成重复的字符串

如：

x='ABCD'\*3

print(x) #ABCDABCDABCD

y=3\*'123'

print(y) #123123123

注：字符串只能和整数相乘

### \*=运算符

生成重复的字符串 再用变量重新绑定

x ='123'

x \*= 3

print(x) #123123132

练习

写一个程序，打印一个高度为4行的矩形方框

如：

请输入框形方框的宽度：10

打印如下：

##########

# #

# #

##########

注：宽度为输入数字个数的#个

答：w= int(input('请输入宽：'))

x='\*'

y=' '

print(x\*w) #print('\*'\*w)

print(x+y\*(w-2)+x)

#print('\*',' '\*(w-2),'\*',sep='')

print(x+y\*(w-2)+x)

#print('\*',' '\*(w-2),'\*',sep='')

print(x\*w) #print('\*'\*w)

#line1='\*'\*w

#line2='\*'+' '\*(w-2)+'\*'

print(lin1)

print(lin2)

print(lin2)

print(lin1)

### 字符串的比较运算

运算符：

< <= > >= == !=

规则：

一次按编码值进行两两比较，一旦不同，则比较结束，返回比较结果，

当编码值的长度完全相同数，两个字符串相等

示例：

'A' < 'B' #True

'ABC' > 'ABB' #True

'ADC' < 'ABC' #False

'ABC' >='123' #True

'AD' >='ABC' #True

'AB' < 'ABC' #True

'ABC' < 'abc' #True

'abcd'!='dcba' #True

注：从第一个字母开始比起，若第一个字母相同则比第二个

### in / not in运算符

作用：

in 用于序列，字典，集合中，用于判断某个值是否存在于容器中，如果存在则返回True，否则返回False

not in 写 in的返回结果相反

格式：

对象 in 序列

示例：

x = 'welcome to tarena!'

'to' in x #True

'hello' in x #False

'a' in x #True

练习

写一个程序，任意输入一段字符串，判断你的名字是否在这个字符串中，如果存在则打印“你的名字出现了”

否则不予理睬

答：t = input('请输入一段话：')

name = "熊"

if name in t:

print('你的名字出现了')

else:

pass

### 字符串的索引操作

### 索引 index

Python 字符串是不可以改变的序列

语法：

字符串[整数表达式]

说明：

Python 序列都可以用索引(index)访问序列中的元素

Python 序列的正向索引是从0开始的，第二个为1...最后一个为len(s)-1

Python 序列的反向索引是从-1开始的，-1代表倒数第一个，-2代表倒数第二个，第一个是-len(s)

示例：

s ='ABCDE'

print(s[1])

练习

写程序，输入一行字符串，打印出一下内容：

1）打印这串字符的第一个字

2）打印这串字符的最后一个字

3）如果这串字符的长度是奇数，打印中间这个字符

答：tt = input('请输入一段字符：')

l=len(tt)

print(tt[0])

print(tt[-1])

if l % 2 >0:

print(tt[int(l/2)])

#print(tt[ l//2 ])

### 切片 slice

作用：

从字符串序列中截取相应的元素重新组成一个字符串序列

语法：

s[(索引1):(索引2)(:(步长))]

注：（）代表其中的内容可以省略

说明：

开始索引是切片开始切下的位置，0代表的一个元素，与索引相同结束索引是切片的终止索引（不包含终止点）

步长是切片每次获取完当前元素后移动方向和偏移量

1）没有步长，相当于步长为1

2）当步长为正整数时，取正向切片：

开始索引默认为0，结束索引是最后一个元素的下一个位置

3）当步长为负整数时，取反向切片：

反向切片时，默认的起始位置为最后一个元素，终止位置为第一个元素的前一个位置

示例：

s = 'ABCDE'

a =s[1:4] #BCD

a =s[1:] #BCDE 等同于a=s[1:5]

a =s[:4] #ABCD 等同于a=s[0:4]

a =s[1:1] #''空字符串

a =s[3:0] #''空字符串

a =s[0:5:2] #ACE 等同于a=s[::2]

a =s[4:1:-2] #EC

a =s[::-1] #EDCBA

练习

1.写一个程序，输入一个字符串，把字符串的第一个字符和最后一个字符去掉，打印出处理后的字符串

答：a = input('请输入一段字符：')

print(a[1:-1])

2.输入任意一个字符串，判断这个字符串是否是回文

回文示例：

上海自来水来自海上

ABCCBA

回文是指中心对称文字

答：a = input('请输入一段字符：')

a1 = a[::-1]

if a==a1:

print('是回文')

else:

print('不是回文')

### 字符串序列相关的函数

len(x) 返回序列的长度

max(x) 返回序列的最大元素

min(x) 返回序列的最小元素

### 字符串编码转换函数

ord(c) 返回一个字符c的UNICODE编码值

chr(i) 返回i这个值对应的字符

练习

1.写一个程序，输入一个字符串，如果字符串不为空，则把这个字符串的第一个字符的编码打印出来

答：a = input('请输入一个字符：')

if a: #如果a不为空，则bool(a),为True

print(ord(a[0]))

2.写一个程序，输入一个整数值（0-65535之间的整数），打印出这个数值所对应的字符

答：a = int(input('请输入一个数字：'))

if a<65535:

print(chr(a))

### 整数转为字符串的函数

bin(i) 将整数转为二进制的字符串

oct(i) 将整数转为八进制的字符串

hex(i) 将整数转为十六进制的字符串

### 字符串的构造函数 str

str(obj='')将对象转为字符串

### python3中常用的方法

方法调用语句：

对象.方法名（方法参数）

示例：

'abc'.isalpha() #正确

123.isalpha() #报错

### 字符串的常用方法：

S.center(width[,fill]) 将原字符串居中，左右默认填充空格

S.count(sub[, start[,end]]) 获取一个字符串中子串的个数

S.find(sub[, start[,end]]) 获取字符串中子串sub的索引,失败返回-1

S.strip([chars]) 返回去掉左右char字符的字符串(默认char为空白字符)

S.replace(old, new[, count]) 将原字符串的old用new代替，生成一个新的字符串

S.startswith(prefix[, start[, end]])

返回S是否是以prefix开头，如果以prefix开头返回True,否则返回False,

S.endswith(suffix[, start[, end]])

返回S是否是以suffix结尾，如果以suffix结尾返回True,否则返回False

例：

s="hello"

s.center(20,'-')

s.count('l',0,6)

s.find('e',0,6)

s.strip('-')

s.replace('l','o',1)

s.startswith('h') #True

s.endswith('o') #True

练习

1.用字符串\*运算符，打印三角形

要求输入一个数，此整数代表左侧字符个数

如：

输入：3

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

答：m = int(input('请输入：'))

a='\*'

b='\*'\*3

c='\*'\*5

d='\*'\*7

print(a.center(7+m\*2))

print(b.center(7+m\*2))

print(c.center(7+m\*2))

print(d.center(7+m\*2))

2.输入一个字符串，把输入的字符串中的空格全部去掉

打印出处理之后的字符串内容

答：T = input('请输入一段字符串：')

if " " in T:

s=T.replace(' ','')

print(s)

3.输入三行文字，让这三行文字在一个方框内居中显示

如：

+-------------------+

| hello! |

|i'm studing python!|

| i like python |

+-------------------+

答：x = input('请输入一段字符串：')

y = input('请输入一段字符串：')

z = input('请输入一段字符串：')

lens = [len(x),len(y),len(z)]

max = max(lens)

a1=x.center(max)

a2=y.center(max)

a3=z.center(max)

print('+'+'-'\*max+'+')

print('|'+ a1 + '|')

print('|'+ a2 +'|')

print('|'+ a3 +'|')

print('+'+'-'\*max+'+')