print(**'hello world!'**)  
  
**if** 3 > 0:  
 print(**'OK'**)  
 print(**'yes'**)  
  
x = 3; y = 4 *# 不推荐，还是应该写成两行*print(x + y)  
---------------------------------  
print(5 / 2) *# 2.5*print(5 // 2) *# 丢弃余数，只保留商*print(5 % 2) *# 求余数*print(5 \*\* 3) *# 5的3次方*print(5 > 3) *# 返回True*print(3 > 5) *# 返回False*print(20 > 10 > 5) *# python支持连续比较*print(20 > 10 **and** 10 > 5) *# 与上面相同含义*print(**not** 20 > 10) *# False*-------------------------------------  
[root@room9pc01 day01]*# cat login.py*username = input(**'username: '**)  
print(**'welcome'**, username)  
print(**'welcome '** + username)  
  
  
  
-------------------------------------  
print(**'hello world!'**)  
print(**'hello'**, **'world!'**) *# 逗号自动添加空格*print(**'hello'** + **'world!'**) *# 加号表示字符拼接*print(**'hello'**, **'world'**, sep=**'\*\*\*'**) *# 单词间用\*\*\*分隔*print(**'#'** \* 50) *# \*号表示重复50遍*print(**'how are you?'**, end=**''**) *# 默认print会打印回车，end=''表示不要回车*+++++++++++++++++++++++++++++++++++++  
number = input(**"请输入数字: "**) *# input用于获取键盘输入*print(number) *# input获得的数据全是字符*print(number + 10) *# 报错，不能把字符和数字做运算*+++++++++++++++++++++++++++++++++++++  
0o开头是8进制数：0o23  
0b开头是2进制数：0b10  
Ox开头是16进制数：0x23  
Linux权限表示为8进制数，python更改Linux权限  
  
**import** os *#导入os模块*os.chmod(**"/root/1.py"**0o644) *#调用os模块中的chmod命令设置/root/1.py的权限为644*引号的使用  
  
  
sentence = **'tom\'s pet is cat'** *#转义符号，默认找第一个引号闭合*sentence = **"tom\'s pet is cat"** *#单双引号结合解决这个问题*sentence = **"tom said:\"helloword\""** *#输出tom said:"helloword"*words = **"""  
hello  
the  
word"""** && print(words) *# 三引号何以换行输出*words = **"hello\nthe\nword"** && print(words)  
*#和上一个输出一样，print将\n解释为换行*>>> py = **"python"** *#定义一个变量*>>> py *#查看变量***'python'**>>> py[0] *#取变量的第一个字符***'p'**>>> len(py) *#统计变量共有多少个字符*6  
>>> py[-1] *#去变量最后一个字符***'n'**>>> py[6] *#取变量第7个字符，没有第7个报错*>>> py[2:4] 从第二个字符取到第4个字符，不包含第四个  
**'th'**>>> py[2:] 从第二个字符取到最后一个字符  
**'thon'**>>> py[:2] 从第0个取到第二个，不包含第二个  
**'py'**>>> py[:] 取全部字符  
**'python'**>>> py[::2] 从第0个开始隔一个取一个字符，每两个取一个  
**'pto'**>>> py[1::2] 从第1个开始隔一个取一个字符，每两个取一个  
**'yhn'**>>> py[::-1] 从右向左取值  
**'nohtyp'**>>> py[3::-1] 在第三个字符从右向左取值  
**'htyp'**>>> **'python'**[1] 取常量的字符  
**'y'**

**>>> py + ' is good' 变量与常量拼接**

**'python is good'**

**>>> py \* 3 将变量代表的东西重复三编**

**'pythonpythonpython'**

**数字：**

**基本数字类型**

**• int:有符号整数**

**• bool:布尔值**

**– True:1**

**– False:0**

**• float:浮点数**

**• complex:复数**

**列表：**

**列表是一个容器，什么都可以放**

**>>> alist = [10,20,30,'bob','xixi',[1,2,3]]**

**定义列表，列表中包含一个小列表**

**>>> len(alist) 查看列表共有几项**

**6**

**>>> alist[-1] 取最后一项，最后一项是列表，还可以继续取**

**[1, 2, 3]**

**>>> alist[-1][-1] 取最后一项，最后一项是列表，在取小列表的最后一个**

**3**

**>>> alist[-2][2] 取列表中常量的第二个字符**

**'x'**

**>>> alist[3:5] 取列表当中第三到第五个值，不包含第五个**

**['bob', 'xixi']**

**>>> alist[-1]=100 修改列表的最后一个值**

**>>> alist**

**[10, 20, 30, 'bob', 'xixi', 100]**

**>>> 10 in alist 判断10是在alist列表中**

**True**

**>>> 'bob' in alist 判断bob是在alist列表**

**True**

**>>> 80 not in alist 判断80是在alist列表中**

**True**

**>>> 'o' in py 判断o是否在py变量中**

**True**

**>>> 'po' in py 判断po是否在py变量中**

**False**

**列表字典是可变的**

**例： blist = alist**

**更改blist 后alist也会发生改变，这种其实相当于指向了存储地点的统一内存，不让他们存在于同一内存**

**clist = alist[:] 将值取出来赋值给clist从新开辟内存,id查看存在什么地方**

**>>> blist = alist**

**>>> clist = alist[:]**

**>>> id(alist)**

**139641945175176**

**>>> id(blist)**

**139641945175176**

**>>> id(clist)**

**139641945176072**

**>>>**

**元组的定义及操作**

**可以认为元组是“静态”的列表**

**元组一旦定义,不能改变**

**>>> atuple = (1, "tom", 2, "alice")**

**>>> atuple[2] 取出第二个值**

**‘tom‘**

**>>> atuple[2][1] 在第二个值上取第2个字符**

**‘o‘**

**>>> 'tom' in atuple**

**True**

**>>> atuple[0] = 3**

**Traceback (most recent call last):**

**File "<stdin>", line 1, in <module>**

**TypeError: 'tuple' object does not support item assignment**

**字典：**

**字典是一个key=value(键=值，对应形式的，没有顺序，通过key向外取值)**

**>>> adict = {"name":"bob","age":18}**

**定义字典**

**>>> adict['name'] 取值**

**'bob'**

**>>> len(adict) 几对键值**

**2**

**>>> 'bob' in adict 值不可以判断**

**False**

**>>> 'name' in adict 字典判断key**

**True**

**按存储模型分类**

**– 标量类型:数值、字符串**

**– 容器类型:列表、元组、字典**

**按更新模型分类:**

**– 可变类型:列表、字典**

**– 不可变类型:数字、字符串、元组**

**按访问模型分类**

**– 直接访问:数字(无法切片)**

**– 顺序访问:字符串、列表、元组**

**– 映射访问:字典（取key出values）**

**判断语句**

py=**'python'  
if 'th' in** py: in && not in  
 print(**'this is '**,py)  
  
**if** 3 > 0 :  
 print(**'yes'**)  
 print(**'ok'**)  
**if** 10 **in** [10,20,30]:  
 print(**'ok'**)  
**if** -0.0: 任何值为0的数字都是False  
 print(**'yes'**)  
**if** [1,2]: 非空对象都是True  
 print(**'yes'**)  
**if ''**: 空对象都是False  
 print(**'ok'**)

**if ' '**:  
 print(**'ok'**) 空格字符也是字符，非空，为True

**判断用户名密码**

**import** getpass

导入getpass模块，

getpass.getpass():让用户输入密码“密文输入”

getpass.getuser() :返回执行用户的登录名

username=input(**'username:'**)  
password=getpass.getpass(**'password:'**)  
   
**if** username==**'bob' and** password==**'123456'**:  
 print(**'Longin successful'**)  
**else**:  
 print(**'Login error'**)

**猜数字脚本**

**import** random

导入random模块  
num = random.randint(1,10)  
you = int(input(**'get numer:'**))  
**if** num == you :  
 print(**"ok"**)  
**elif** num < you :  
 print(**'big numer'**)  
**else**:  
 print(**'small numer'**)

**Random模块：**

**random.random()**函数是这个模块中最常用的方法了，它会生成一个随机的浮点数，范围是在0.0~1.0之间。

**random.uniform()**正好弥补了上面函数的不足，它可以设定浮点数的范围，一个是上限，一个是下限。

**random.randint()**随机生一个整数int类型，可以指定这个整数的范围，同样有上限和下限值，python random.randint。

**random.choice()**可以从任何序列，比如list列表中，选取一个随机的元素返回，可以用于字符串、列表、[元组](http://www.iplaypy.com/jichu/tuple.html" \t "/root/文档\\x/_blank)等。

**random.shuffle()**如果你想将一个序列中的元素，随机打乱的话可以用这个函数方法。

**random.sample**random.sample(sequence, k)，从指定序列中随机获取指定长度的片断。sample函数不会修改原有序列

代码如下:

list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
slice = random.sample(list, 5)  # 从list中随机获取5个元素，作为一个片断返回  
print slice  
print list  # 原有序列并没有改变

**条件表达式**

**• Python 在很长的一段时间里没有条件表达式(C ? X :Y),或称三元运算符,因为范·罗萨姆一直拒绝加入这样的功能**

**• 从Python 2.5集成的语法确定为: X if C else Y**

**剪刀石头布：**

**import** random  
all\_choicces=[**'石头'**,**'剪刀'**,**'布'**]  
win\_list=[[**'石头'**,**'剪刀'**],[**'剪刀'**,**'布'**],[**'布'**,**'石头'**]]  
computer=random.choice(all\_choicces)  
player=input(**'请出拳:'**)  
print(**"你出的是：%s ，电脑出的是：%s"**%(player,computer))  
**if** player == computer:  
 print(**"平局"**)  
**elif** [player,computer] **in** win\_list:  
 print(**"你赢了"**)  
**else**:  
 print(**"你输了"**)

**优化版：**

**import** random  
a=**'(0)石头\n(2)剪刀\n(3)布\n请选择:'**all\_choicces=[**'石头'**,**'剪刀'**,**'布'**]  
win\_list=[[**'石头'**,**'剪刀'**],[**'剪刀'**,**'布'**],[**'布'**,**'石头'**]]  
computer=random.choice(all\_choicces)  
b=int(input(a))  
player=all\_choicces[b]  
  
print(**"你出的是：%s ，电脑出的是：%s "**%(player,computer),end=**''**)  
**if** player == computer:  
 print(**"\033[32;1m平局\033[0m"**)  
**elif** [player,computer] **in** win\_list:  
 print(**"\033[31;1m你赢了\033[0m"**)  
**else**:  
 print(**"\033[31;1m你输了\033[0m"**)

**If 多次比较**

**score = int(input('分数: '))**

**if score >= 60 and score < 70:**

**print('及格')**

**elif 70 <= score < 80:**

**print('良')**

**elif 80 <= score < 90:**

**print('好')**

**elif score >= 90:**

**print('优秀')**

**else:**

**print('你要努力了')**

**While循环**

*#!/usr/local/bin/python3*sum=0  
con=1  
**while** sum < 101:  
 sum += con  
 con += 1  
print(sum)

**While中break的使用**

*#!/usr/local/bin/python3***while True**:  
 yn=input(**'Continue(y/n)'**)  
 **if** yn **in** [**'n'**,**'N'**]:  
 **break** print(**'running...'**)

**取100以内的偶数之和continue**

*#!/usr/local/bin/python3*sumer=0  
oushu=0  
**while** oushu < 100:  
 oushu += 1  
 **if** oushu % 2 == 0:  
 sumer +=oushu  
 **else**:  
 **continue**print(sumer)

**优化**

*#!/usr/local/bin/python3*sumer=0  
oushu=0  
**while** oushu < 100:  
 oushu += 1  
 **if** oushu % 2 : //为1为真  
 **continue** sumer +=oushu  
print(sumer)

**-------------------------------------剪刀石头布再优化**

**import** random  
a=**'(0)石头\n(1)剪刀\n(2)布\n(3)退出游戏\n请选择:'**all\_choicces=[**'石头'**,**'剪刀'**,**'布'**]  
win\_list=[[**'石头'**,**'剪刀'**],[**'剪刀'**,**'布'**],[**'布'**,**'石头'**]]  
**while** 1 :  
 computer=random.choice(all\_choicces)  
  
 yonghu=input(a)  
 **if** yonghu **not in** [**'0'**,**'1'**,**'2'**,**'3'**]:  
 print(**"输入有误，程序以退出ing"**)  
 **break** b=int(yonghu)  
 **if** b == 3:  
 **break** player=all\_choicces[b]  
  
 print(**"你出的是：%s ，电脑出的是：%s "**%(player,computer),end=**''**)  
 **if** player == computer:  
 print(**"\033[32;1m平局\033[0m"**)  
 **elif** [player,computer] **in** win\_list:  
 print(**"\033[31;1m你赢了\033[0m"**)  
 **else**:  
 print(**"\033[31;1m你输了\033[0m"**)