+，-，\*，/，^，==,&&,||均是正常使用  
xor(0,1) %异或 结果：1  
~= %表示不等号  
% 是注释作用  
PS1('>>') %会隐藏Octave命令窗口中每一行的版本信息  
; %表示该句在按下回车键后不会输出 抑制打印输出

>>a=3

a=3

>>a=3;

>>a

a=3

------------

>>c=(3>1);

>>c %输出结果会是c=1 这里的1表示为真

c=1

------------

>>a=pi; %pi表示圆周率

>>a

a=3.1416

>>disp(a); %这里的disp相当于打印

a=3.1416

>>disp(sprintf('2 demical is %0.2f',a))

2 demical is 3.14

>>format long

>>a

a=3.14159265358979

>>format short

>>a

a=3.1416

---------------

A=[1 2;3 4;5 6] %会输出一个3行2列的矩阵，内容是A中的内容，矩阵中的；相当于矩阵中的换行

A=[1 2;

3 4;

5 6

] %同上

A=[1 2 3] %1行3列 内容：矩阵中的内容

A=1:0.1:2 %会得到一个1行11列的矩阵，从1开始，到2结束，中间每次的步长是0.1

>>A=1:6

A=

1 2 3 4 5 6

ones(2,3) %生成一个2行3列的矩阵，矩阵中的内容都是1

2\*ones(2,3) %生成一个2行3列的矩阵，矩阵中的内容都是2

zeros(2,3) %生成一个2行3列的矩阵，矩阵中的内容都是0

rand(2,3) %生成一个2行3列的矩阵，矩阵中的内容都是从0-1之间的随机数

randn(2,3) %生成一个2行3列的矩阵，矩阵中的内容都是来自三个值:一个平 值为高斯分布、方差或等于1的标准偏差

————————————————

w=-6+sqrt(10)\*randn(1,1000);

hist(w) %会绘制一个直方图 这里的方差就是10，标准差是10的平方根

hist(w,50) %绘制一个直方图，直方图的的条数为50

eys(4) $会生成4行4列的单位矩阵

A=[1 2;3 4;5 6]

size(A) %输出A的尺寸大小，3 2

s=size(A) %使用某个变量接收的话，该变量会变成相应的矩阵，内容是A的尺寸大小

size(s) %输出的是s的尺寸大小，1 2

length(A) %会返回最大维度 这里是3

pwd %会输出当前Octave的路径

cd 路径 %会转到指定的路径位置

list %会列出该路径下所有的文件

>>load 文件名 %会将文件中的内容读到Octave中

>>load('文件名') %同上

>>who %会显示当前Octave中存储的变量

>>变量名 %会显示该变量中内容，包括刚才读入的文件名，输入文件名的话，会显示文件中内容

>>whos %会显示Otave中存储的变量详细信息，包括:size(维度),Bytes(占据的空间),Class(里面的数据类型)

>>clear 变量名 %会删除指定为变量

可以将指定变量中的内容存储到新的变量中

>>v=Price(1:10) %会将Price中1-10条的内存存储到v中

>>save hello.mat v; %会将v变量中的内容存成一个名为：hello.mat的文件，文件中的内容就是v中的内容

>>clear %按下回车键会清空所有的变量

>>save hello.txt v -ascii %会将v中的内容按照ascii的形式存储到hello.txt中

A=[1 2;3 4;5 6]

>>A(3,2) %索引到第3行第2列，输出6

>>A(2,:) %会输出第2行所有列的元素

>>A(:,2) %会输出第2列所有行的元素

>>A([1 3],:) %会输出第1行、第3行所有列的元素

>>A(:,2)=[10;12;13] %会将第2列替换成10;12;13

>>A=(A,[100;200;300]) %会在最右边添加一列，100;200;300

>>A(:) %会输出A中的所有元素，并放入同一列中进行输出

A=[1 2;3 4;5 6]

B=[11 12;13 14;15 16]

>>C=[A B] %会将AB横向连接起来

>>C=[A;B] %会将AB纵向连接起来

[A,B] 与[A B] 结果一样

A=[1 2;3 4;5 6]

B=[11 12;13 14;15 16]

C=[1 1;2 2]

>>A\*C %表示A,C两个矩阵相乘

>>A .\*B %表示A矩阵中的内容和B矩阵中的内容对应相乘

>>A .^2 %表示A中元素的平方 .一般表示元素位运算

>>1 ./A %表示对A中的元素求倒数

>>exp(A) %表示以A中的元素为基准，求每个e的次方

>>abs(A) %对A中的每个元素求绝对值

>>-A %相当于-1\*A

>>A+ones(length(A),2) %对A中的所有元素加1 length(A)是求的A的最大维度

>>A+1 %也是相当于A中的没个元素加1

>>A' %A的转置。这里的单引号是左单引号

a=[1 15 2 0.5]

>>val=max(a) %输出的就是该矩阵中元素的最大值

val=15

>>[val,ind]=max(a) %输出的是矩阵中的最大值以及最大值的索引

val=15

ind=2

>>a<3 %将矩阵中的元素分别和3进行比较，小于3为真，输出1；大于3为假，输出0

ans= 1 0 1 1

>>find(a<3) %将会返回输出小于3的索引下标

ans= 1 3 4

>>A=magic(3) %会输出一个3行3列的矩阵，成为魔幻矩阵，但是这个矩阵的每一行、每一列、每%条对象上的元素加起来均相等

a=[1 15 2 0.5]

>>sum(a) %求和

>>prod(a) %求矩阵中元素的乘积

>>floor(a) %将矩阵的元素向下四舍五入

>>ceil(a) %将矩阵的元素向上四舍五入

>>rand(3) %生成一个3行3列的矩阵，内容：0-1

>>max(rand(3),rand(3)) %生成一个3行3列的矩阵，但是每一元素去的是相同位置上的最大值

A=[8 1 6;3 5 7;4 9 2]

>>max(A,[],1) %输出每一行(第一维度)的最大值

ans=8 9 7

>>max(A,[],2) %输出每一列(第二维度)的最大值

ans=

8

7

9

>>max(A) %没有指出，默认情况是输出每一列的最大值

>>max(max(A)) %输出的矩阵的中的最大的一个值

同样可以先把矩阵的中内容转换成一列，再求矩阵中的最大值

>>max(A(:))

对于这个魔幻矩阵

>>sum(A,1) %求解的是的第一维度(每行)的和

>>sum(A,2) %求解的是的第二维度(每列)的和

至于每一个对角线的和的求法如下：

>>eye(3) %先创建一个3\*3的单位矩阵

>>A .\*eye(3) %矩阵中对应位置的内容相乘，得出矩阵对角线上的内容

>>sum(sum(A .\*eye(3))) %就会求出矩阵对角线上的和 这是求得是从左上到右下对角线的和

>>sum(sum(A .\*flipub(eye(3)))) %求的是左下到右上的对角线的内容 flipub作用就是将对角线反转

>>temp=pinv(A) %求的是A的逆矩阵

>>temp\*A %会求出一个单位矩阵

-----------------------------------

>>t=[0:0.1:0.98]

>>y1=sin(2\*pi\*4\*t)

>>plot(t,y1) %会画出几个sin的正弦函数

>>y2=cos(2\*pi\*4\*t)

>>plot(t,y2) %会画出几个sin的余弦函数

但是这两个函数并不会同时存在，后面的会覆盖上一个

下面是将两个函数同时画在一张图上

>>plot(t,y1)

>>hold on;

>>plot(t,y2,'r') %r 表示红色 此时两幅图就会显示在同一图上

>>xlabel('time') %设置x轴的标签名

>>ylabel('value') %设置y轴的标签名

>>legend('sin','cos') %会在合适分别为两幅图的进行区分表明

>>title('my plot') %设置该图的名字

>>cd C:\User\Desktop; print -dpng 'myPlot.png' %会先转到桌面，并将该图片存储为png格式

>>close %会关闭打开的图

>>figure(1);plot(t,y1) %绘制第一张图，并打开

>>figure(2);plot(t,y2) %绘制第二张图，并打开

>>subplot(1,2,1); %将图分成1行两列的小格子，现在是占据第一个格子

>>plot(t,y1) %将会在第一个格子上显示相应的图

>>subplot(1,2,2); %将图分成1行两列的小格子，现在是占据第二个格子

>>plot(t,y2) %将会在第二个格子上显示相应的图

>>axis([0.5 1 -1 1]) %会将此图横轴坐标范围变成0.5~1 纵轴坐标范围变成-1~1

>>clf; %会清空整个图像上的所有内容

>>A=magic(5)

>>imagesc(A) %会绘制一个5\*5的色彩格子，每个格子对应不同的坐标范围

>>imagesc(A),colorbar,colormap gray; 会绘制一张图，同时在右侧有一个颜色柱，该柱是灰色的

============================

循环：

>>for i=1:10,

> v(i)=2^i

> end;

>>v %就会纵向输出v的值

>>indices=1:10

>>indices %就会横向输出1~10的值

>>for i=indices,

> disp(i)

> end; %此时的输出结果和上面的相同

>>i=1

>>while i<+5,

> v(i)=100;

> i=i+1;

> end;

>>v %前五项输出100，将上面的v的强五项覆盖

>>i=1

>>while true,

> v(i)=999;

> i=i+1;

> if i==6,

> break;

> end;

> end;

>>v %前五项输出999，将上面的v的强五项覆盖

>>v(1)=2

>>if v(1)==1,

> disp('this is one');

> elseif v(1)==2,

> disp('this is two');

> else

> disp('this is three')

> end;

this is two

退出Oatave的话可以使用:exit或quit