

第2章 Matlab基础应用

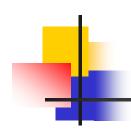


Matlab 变量

- □ Matlab 变量的命名规则
 - 以字母开头
 - 后面可以跟字母、数字和下划线
 - 长度不超过 63 个字符(6.5 版本以前为 19 个)
 - 区分字母的 大小 写
- □ Matlab 语句的通常形式

变量 = 表达式

表达式是用运算符将有关运算量连接起来的式子, 其结果被赋给赋值号 "="左边的变量



系统预定义变量

- □ 系统预定义的变量
 - ◆pi: 圆周率π,其值为 imag(log(-1))
 - ◆inf, Inf : 无穷大
 - ◆ nan, NaN: Not-a-Number, 一个不定值, 如 0/0
 - ◆ eps : 浮点运算相对精度
 - ◆ i, j : 虚部单位,即 $\sqrt{-1}$

注: 应尽量避免给系统预定义变量重新赋值!

□ 特殊变量 ans



nargin 函数的输入变量数目 nargout 函数的输出变量数目 realmax 最大的可用正实数 realmin 最小的可用正实数



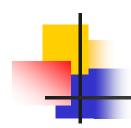
查看已定义的变量

- who 显示工作空间中的所有变量
- whos 显示变量的详细属性

>> who						
Yo	ur	var	iab	les	are:	
A	a	b	С	x		

>> whos			
Name	Size	Bytes	Class
A	4x 3	96	double array
a	4x1	32	double array
ь	3 x1	24	double array
С	1 x 1	8	double array
x	1x5	40	double array

Grand total is 25 elements using 200 bytes



矩阵的输入

□ Matlab 的操作对象 —— 矩阵

□ 定义矩阵: 直接输入法

例: \rightarrow A = [1 2 3; 4 5 6; 7 8 9]

- 矩阵用方括号 "[]" 括起
- 矩阵同一行中的元素之间用 空格 或 逗号 分隔
- 矩阵行与行之间用 分号 分开
- 直接输入法中,分号可以用 回车 代替
- >> 是命令提示符,不用输入
- 回车 运行所输入的命令



矩阵元素赋值

□ 矩阵元素可以是任何数值表达式

```
例: \Rightarrow x=[-1.3, sqrt(3), (1+2+3)*4/5]
```

□ 矩阵元素的单独赋值

```
例: \rightarrow x (5) = abs (x (1))
```

动态定维功能: Matlab 会自动扩展向量的长度,

并将未赋值部分置零

例: \rightarrow x(5)=abs(x(6))??



矩阵元素赋值

□ 大矩阵可以把小矩阵作为其元素

```
例: >> A=[A ; 11 12 13]
```



⑦ 如何在原矩阵的右边添加一列?

```
A=[2 3 5;3 4 1;0 9 7;1 2 3]
c=[1;2;3;4]
A=[A c]
```



矩阵元素的引用

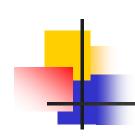
□ 单个元素的引用

```
例: >>A(2,3)
```

● 利用小括弧和元素所在的位置(下标)

```
x(i): 向量x中的第i个元素
```

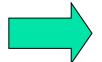
A(i, j): 矩阵 A 中的第 i 行,第 j 列元素



矩阵元素的引用

□ 多个元素的引用:冒号的特殊用法

a:b:c



产生一个由等差序列组成的向量:

- a 是首项, b 是公差, c 确定最后一项;
- 若 b = 1, 则 b 和其前面的冒号可以省略

例: >> x=1:2:5 例: >> x=2:1:5 例: >> x=3:2:1 >> y=1:2:6 >> y=2:5

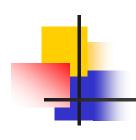


矩阵元素的引用

```
例: >>x(1:3)
>>A(3,1:3)
```

- A(i:j, m:n) 表示由矩阵 A 的第 i 到第 j 行和第 m 到第 n 列交叉线上的元素组成的子矩阵
- 可利用冒号提取矩阵 的整行或整列

```
例: >>A(1, :)
>>A(:, 1:3)
>>A(:, :)
```



矩阵操作

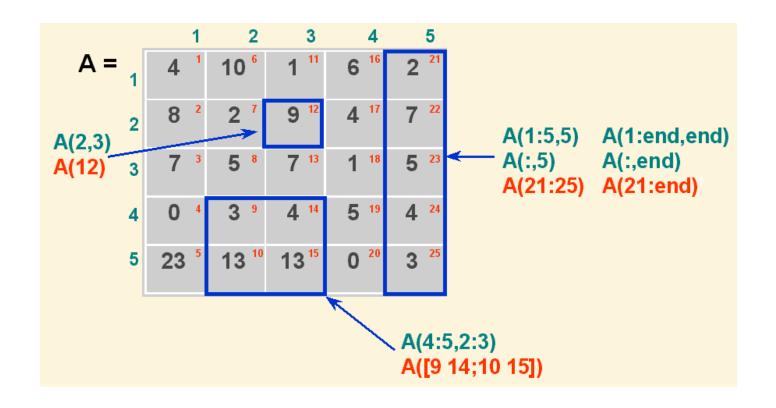
- □ 提取矩阵的部分元素: 冒号运算符
 - ◆ A(:) A 的所有元素
 - ◆ A(:,:) 矩阵A的所有元素
 - ◆ A(:,k) A的第k列, A(k,:) A的第k行
 - ◆ A(k:m) A 的第 k 到第 m 个元素
 - ◆ A(:,k:m) A 的第 k 到第 m 列组成的子矩阵

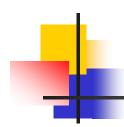
自己动手

- (7) A(:) 与 A(:,:) 的区别?
- ② 如何获得由 A 的第一、三行和第一、二列组成的子矩阵?

4

矩阵的序号编址:按列计数。





矩阵操作

□从大矩阵中提取一个子矩阵

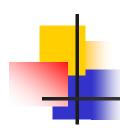
```
A(I,J)
其中 I=[i_1, i_2, ..., i_m], J=[j_1, j_2, ..., j_m]
```

表示由 A 的第 i_1 , i_2 , ..., i_m 行和第 j_1 , j_2 , ..., j_m 列组成的子矩阵。

```
例: >>A([1,3,4], [2,3])
```

□删除指定的行或列

```
例: >>A(3,:) = []
例: >>A(:, [1,2,4]) = []
```



特殊矩阵

- □ 一些特殊矩阵的生成
 - Matlab 提供了一些函数,用于生成一些常见的特殊矩阵



常见矩阵生成函数

zeros(m,n)	生成一个 m 行 n 列的零矩阵, $m=n$ 时可简写为 $zeros(n)$
ones(m,n)	生成一个 m 行 n 列的元素全为 1 的矩阵, $m=n$ 时可写为 ones(n)
eye(m,n)	生成一个主对角线全为 1 的 m 行 n 列矩阵, $m=n$ 时可简写为 $eye(n)$,即为 n 维单位矩阵
diag(X)	若 X 是矩阵,则 diag(X) 为 X 的主对角线向量 若 X 是向量, diag(X) 产生以 X 为主对角线的对角矩阵
tril(A)	提取一个矩阵的下三角部分
triu(A)	提取一个矩阵的上三角部分
rand(m,n)	产生 $0\sim1$ 间均匀分布的随机矩阵 $m=n$ 时简写为 $\mathrm{rand}(n)$
randn(m,n)	产生均值为 0 ,方差为 1 的标准正态分布随机矩阵 $m=n$ 时简写为 $randn(n)$

其它特殊矩阵生成函数: magic、hilb、pascal 等



3、复数矩阵

不需要特殊的处理,可以直接进行。

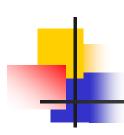
复数可以表示为: a=10-9i

注意: 虚部与虚数单位之间不能留空格

函数:complex

z = complex(3,4)

 $z=complex(3) \iff z=complex(3,0)$



2.1 Matlab 数值运算

- □ 数学运算符
 - + 加法
 - - 减法
 - * 乘法
 - / 和 \ 除法(右除和左除)
 - ^ 幂运算
- □ 命令分隔符: 逗号和分号



运算规则: 从左到右; 先乘除后加减; 乘方运算符最高

(1) 两矩阵加减

A、维数相同时,进行加减运算时,对应的元素进行加减;

B、矩阵与标量加减,用矩阵中的每个元素都与标量进行加减运算;

(2) 两矩阵相乘

A、前提是前一矩阵的列等于后一矩阵的行,与数学约 定一样;

B、矩阵与标量相乘,用矩阵中的每个元素都与标量进行相乘;

- 4
 - (3) 矩阵中的元素对元素的相乘: .*
 - (4) 矩阵中的元素对元素的相除: ./ .\
 - (5) 乘方运算

z=x.^y x,y均为向量: z(i)=x(i) ^y(i)

x为向量,y为标量: z(i)=x(i) ^y

x为标量,y为向量: z(i)=x^y(i)



矩阵基本运算

- □矩阵的除法:/、\ 右除和左除
 - 若 A 可逆方阵,则

• 通常,矩阵除法可以理解为

当A和B行数相等时可进行左除 当A和B列数相等时可进行右除



矩阵的乘方

◆ A 是方阵, p 是正整数
 A^p 表示 A 的 p 次幂, 即 p 个 A 相乘。

◆ 若 a 是标量,
$$D = \begin{pmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_n \end{pmatrix}$$

$$a \wedge D = \begin{pmatrix} a \wedge d_1 & 0 & \dots & 0 \\ 0 & a \wedge d_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a \wedge d_n \end{pmatrix}$$



矩阵转置

- □ 矩阵的转置与共轭转置
 - ◆′ 共轭转置
 - ◆ .′ 转置,矩阵元素不取共轭

点与单引号之间不能有空格!

```
例:
```

```
>> A = [1, 2; 3i, 4i]
>> B = A'
>> C = A.'
```



矩阵的旋转和排序

□ 矩阵的旋转

- ◆ fliplr(A) 左右旋转
- ◆ flipud(A) 上下旋转

矩阵的排序: sort函数, 它将矩阵按照升序排列。

◆ rot90(A) 逆时针旋转 90 度; rot90(A,k) 逆时针旋转 k×90 度

```
例:
```

```
>> A = [1 2 3;4 5 6]
>> B = fliplr(A)
>> C = flipud(A)
>> D = rot90(A)
>> E = rot90(A,-1)
```

注意矩阵旋转与转置的区别!



改变矩阵的形状

□ 改变矩阵的形状: reshape(A,m,n)

将矩阵元素按 列方向 进行重新排列成一个 m×n 的新矩阵

新矩阵的元素个数必须与原矩阵元素个数相等!

```
例: \rightarrow A = [1,2,3; 4,5,6; 7,8,9; 10,11,12]
    >> B = reshape (A, 4, 3)
```

```
\rangle\rangle C = reshape (A, 2, 6)
```

```
>> C = reshape (A, 5
```



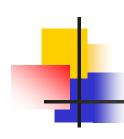
查看矩阵的大小

- □ 查看矩阵的大小: size、length
 - 返回矩阵 A 的行数和列数 \bullet [m,n]=size(A)
 - 返回矩阵 A 的行数 \rightarrow m1=size (A,1)
 - 返回矩阵 A 的列数 \rightarrow m²=size (A,2)

```
例: \rightarrow A = [1,2,3; 4,5,6]
    >> size (A)
     >> size (A, 1)
     >> size (A, 2)
```

- 返回向量 x 的长度 length(x)
- 等价于 max (size (A)) length (A)

```
>> length (A)
\rightarrow x = [5:2:10]; length(x)
```



分号和续行符

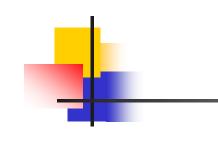
□ 分号和续行符的作用

- 若不想在屏幕上输出结果,可以在语句最后加分号
- 如果语句很长,可用续行符 "…"(三个点)续行 续行符的前面最好留一个空格



部分函数

函数名	含义
abs	绝对值或者复数模
sqrt	平方根
real	实部
imag	虚部
conj	复数共轭
round	4舍5入到整数
fix	舍入到最接近0的整数
floor	舍入到最接近-∞的整数
ceil	舍入到最接近∞的整数



函数名	含义
sign	符号函数
sin	正弦
cos	余弦
tan	正切
asin	反正弦
acos	反余弦
atan	反正切
atan2	第四象限反正切

函数名	含义
sinh	双曲正弦
cosh	双曲余弦
tanh	双曲正切
exp	自然指数
log	自然对数
log10	以10为底的对数

1

2.2 关系和逻辑运算

1、关系操作符

(1)MATLAB常用的关系操作符有:

<(小于)、<=(小于或等于)、

>(大于)、>=(大于或等于)、

==(等于)、~=(不等于)。

(2)MATLAB的关系操作符可以用来比较两个大小相同的数组,或者比较一个数组和一个标量。在与标量比较时,结果和数组大小一样。

(3)如果满足指定的关系,返回1,否则返回0。

```
>>a=4:0.5:7;
>> b=a>5
b =
    0    0    0    1    1    1    1
>>c=a(a>5)
c =
    5.5000   6.0000   6.5000   7.0000
```

```
»d=find(a>5)

d =
    4    5    6    7
```

子矩阵的查找使用find命令完成,它返回关系表达式为真的下标。



2、逻辑操作符

(1)逻辑操作符定义了一种与或非的关系表达式。

MATLAB的逻辑操作符有

- (2)所有逻辑运算符连接的两个操作数或者同维、同大小,或者其中一个为标量。
- (3)逻辑运算符都是对元素的操作,每个非零元素都当作"1"处理,逻辑运算的结果是由1和0构成的矩阵。

例如:

与逻辑运算有关的函数函数:

- (1) all(A): c=all(x),则
 - c=1,向量x中全为非零元素;
 - c=0,向量x中含有零元素
- (2) any(A): c=any(x), 则
 - c=1,向量x中含有非零元素;
 - c=0,向量x中全为零元素



2.4 时间和日期

- 1、日历函数:calendar
- 2、时间函数:clock
- 3、查看CPU 时间:cputime
- 4、日期函数:date
- 5、秒表定时:

tic

toc

2.5退出和保存工作空间:

1、save:工作空间中的所有变量保存在磁盘上名为matlab.mat的文件中。

2、save [文件名] [变量名]

将指定的变量保存在指定的文件中。

如: >>save temp x y z

把xyz这3个变量保存在文件temp.mat中。

3、下次加载MATLAB时,可以利用load命令将保存在文件中的变量恢复到工作空间中,其格式为:

(1) load

将保存在matlab.mat中的变量装入到matlab空间中。

(2) load [文件名] [变量名]

从指定文件中将指定的变量装入到matlab的工作空间中。

如: >>load temp x

从文件temp.mat 中只将变量x装入到 matlab的工作空间中。

2.6 MATLAB数据类型

数值类型、字符型、稀疏型、单元型(<mark>元胞</mark>)、 结构型、逻辑类型

最常使用的是数值类型和字符型;稀疏型用于稀疏矩阵;单元型和结构型用于编写大型软件;8位型(6.0以上版本:16位型、32位型)用于图象处理。

1、数值类型(double):

在缺省情况下,当结果为整数,作为整数显示; 当结果为实数,以小数后4位的精度近似显示。

(>>format short;>>format long)

□ Matlab 提供四种带符号整型和四种无符号整型

数据类型	取值范围	转换函数
带符号8位整型	[-27, 27 -1]	int8
带符号 16 位整型	[-2 ¹⁵ , 2 ¹⁵ -1]	int16
带符号 32 位整型	[-231 , 231 -1]	int32
带符号 64 位整型	[-2 ⁶³ , 2 ⁶³ -1]	int64
无符号8位整型	[0 , 28 -1]	uint8
无符号 16 位整型	[0 , 2 ¹⁶ -1]	uint16
无符号 32 位整型	[0 , 2 ³² -1]	uint32
无符号 64 位整型	[0 , 2 ⁶⁴ -1]	uint64



不同的整型数据之间不能运算,例如: int8(22)*int16(5)

2、字符型(char):

```
>> a=['sss';'bbb']
a =
sss
bbb
```

(1) 字符数组的定义:字符数组就是字符串,字符串中的每一个字符在系统内部都相应地表示一个数值。

```
a='You are welcome!'
```

```
a =
You are welcome!
size(a)
ans =
1 16
```

相关函数

```
A、函数disp用来显示字符串: disp('Hello')
Hello
 B、判断一个变量是否为字符型数组,可以用函数
class或ischar。
 x=class(a)
 y=ischar(a)
 \mathbf{x} =
 char
 y =
```

(2) 字符与数值的相互转换:

```
double: 将一个字符串转换为一个数值
   char: 将数值转换为字符串
 s='Good morning!'
 s1=double(s)
 s2=char(s1)
 S =
 Good morning!
 s1 =
   71 111 111 100 32 109 111 114 110 105
110 103 33
 s2 =
 Good morning!
```

(3) 二维字符数组:

创建二维字符数组时,每行的**长度要相等**。如果字符串长度不等,可以用空格充填较短的字符串。

```
例如:

s=['abc ';'defg']

s =

abc

defg
```

用char函数创建二维字符数组比较方便。当字符串长度不等时,char函数可以自动地在较短的字符中加上一定数量的尾部空格,使其与最长字符串的长度相等。

例如:

```
y=char('abc','defg','abcde')
 z1=length(y)
 z2=size(y)
y =
abc
defg
abcde
z1 =
z^2 =
       5
```

A、strcmp: 判别两个字符串是否相等

B、strncmp: 判别两个字符串的前n个字符是否相等

s1='glisten';s2='glitter';

a1=strcmp(s1,s2)

a2=strncmp(s1,s2,3)

a1 =

0

a2 =

s1='This is a good example'
strrep(s1,'good','great')
strrep(s1,'bad','great')

函 数	功能	函 数	功 能
stremp	比较字符串	strrep	替换字符串
strempi	忽略大小写比较 字符串	strnemp	比较字符串的前n个字 符
upper	转换为大写	lower	转换为小写
blanks	产生空字符串	deblank	删除字符串中的空格
strmatch	查找匹配的字符 串	findstr	在一个字符串中查找另 一字符串
strjust	对齐字符数组, 包括左对齐,右 对齐和居中	strtok	返回字符串中第一个分隔符(空格,回车和 Tab键)前的部分

A=[' aaa ';'bbbbb']
B=strjust(A,'center')

注: 尾部 的 格

(5) 字符串的合并

□ 水平合并

◆ 直接使用中括号

```
str1=['hello ', 'world!']
```

◆ 使用 strcat 函数

```
str2=strcat('hello ','world!')
```

- 使用 strcat 时,自动去除原字符串结尾处的空格
- 水平合并得到的是一个更长的字符串

□ 垂直合并

◆ 直接使用中括号

```
str3=['Matrix '; 'Laboratory']
```

◆ 使用 strvcat 函数

```
str4=strvcat('Matrix','Laboratory')
```

- 在中括号中用分号实现垂直合并,必须保证每个字符串的长度相等,否则需用空格补齐
- 用 strvcat 合并, 自动为较短的字符串补充空格
- 垂直合并得到的是一个字符数组

>>speye(4) 生成单位稀疏矩阵

ans =

 $(1,1) \qquad 1$

(2,2) 1

(3,3) 1

(4,4) 1

sparse 把满矩阵转化为稀疏矩阵

调用格式: sparse(A)

sparse(i,j,s,m,n)

i和j为矢量,指矩阵中非0元素的行号和列号,s是一个全部元素为非0的矢量,m和n分别为输出稀疏矩阵的行数和列数

把稀疏矩阵转换为满阵 full (2017/10/12)



A=[0 0 6;0 0 0;1 2 0] [i,j,s]=find(A) [m,n]=size(A)

S=sparse(i,j,s,m,n)

find(A)找到非零值的行、列、值。

4、单元型(cell):把不同类型的数据存储在一起的数组称为单元数组。单元数组中的每个元素是其他任意一种数据类型,通常可以将相关的尺寸大小不同的数据组合在一起。

```
A=eye(2)
c=cell(size(A))
A =

1 0
0 1

c =

[] []
[] []
```

D =

'Good'

```
给单元数组赋值
D=cell(1,3)
D{1,1}='Good'; D{1,2}=[1 2;4 5];D{1,3}=1+4i;
D
D = [] []
```

[2x2 double] [1.0000 + 4.0000i]



EX2: 利用直接输入的方法创建单元数组,单元数组的下标用圆括号括起来,右边是单元数组的内容,用花括号括起来。

(2) 单元数组的显示

```
A、直接显示
Aa
Aa =
 'GOOD'
                     200]
  [5.0000- 6.0000i]
             [] [2x2double]
B、利用函数显示
celldisp(Aa)
celldisp(Aa,'Bb') Bb表示显示的名字
C、利用下标显示
```

Aa(3,2)或 Aa{3,2} _二者差异

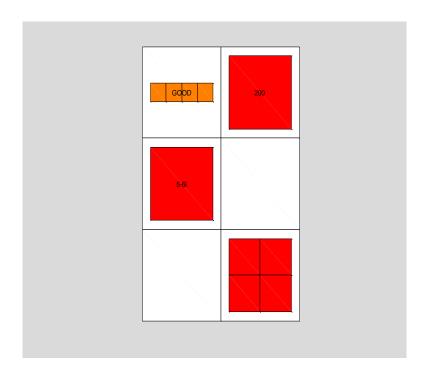
```
Aa\{1,1\} =
GOOD
Aa\{2,1\} =
 5.0000 - 6.0000i
Aa{3,1} =
Aa\{1,2\} =
 200
Aa\{2,2\} =
Aa{3,2} =
     5
```





D、利用图形显示单元数组

cellplot(Aa)
cellplot(Aa,'legend')

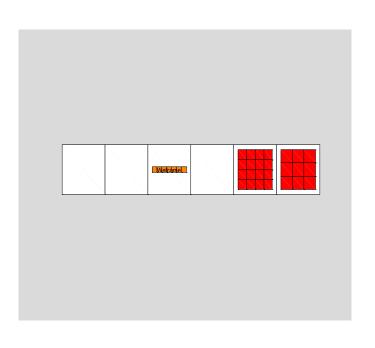


```
X=cell(2,3)
X{1,2}='Welcome';
X{1,3}=ones(4);
X{2,3}=[1\ 2\ 3;4\ 5\ 6;7\ 8\ 9]
Da = sum(X\{1,3\})
Db=sum(X\{2,3\})
X =
X =
       'Welcome' [4x4 double]
               [3x3 double]
Da =
Db =
```

(4) 单元数组的变形

```
E=reshape(X,1,6)
cellplot(E)
```

```
E = \begin{bmatrix} \end{bmatrix} [] 'Welcome' [] [4x4 double] [3x3 double]
```



(5) 字符型单元数组:由于字符矩阵要求 矩阵的每一行的长度相等,需要在字符串的 尾端加入空格,在读取的时候,又常常将末 尾的空格去掉,很不方便。由于单元数组允 许不同的单元内有不同类型和长度的数据, 问题就可以避免。

```
cellstr:将标准的字符型数组转换为字符型单元数组。
EX1:
ne=['aaa';'bb '];
cellstr(ne)
ans =
    'aaa'
    'bb'
```



5、结构型(struct):把不同类型的数组存储在一起的数组称为结构数组。结构数组相当于数据库的记录。结构数组中不同类型的数组是通过不同的域名来区分的。

在MATLAB的结构数组中,每个元素都是由不同的数据类型所组成的,不同的数据类型分别放在不同的数据区域里,称为结构数组的域。

```
(1) 建立结构数组
A、利用函数struct建立结构数组
调用格式:
s=struct('field 1',values 1,'field 2',values 2,...)
例:
sxx=struct('name',{'zhao','Li','wang'},'age',{20,21,29},'address',['changda']) %注意:[]所有的address都赋相同的值
sxx(1)
sxx(2)
```



B、利用点号(.)运算符创建结构数组

```
sss.name='zhou';
sss.age='10';
sss.address='changda';
SSS
sss.name='zhao'
SSS =
    name: 'zhao'
    age: '10'
  address: 'changda'
sss(3).address='chang an'
SSS =
1x3 struct array with fields:
  name
  age
  address
```

相关函数

A、getfield: 获取结构数组中某个域的内容

B、setfield: 设置结构数组中某个域的内容

C、rmfield: 删除结构数组的域

D、isfield: 判断某个变量域是否是一个结构数组的域

E、isstruct: 判断某个变量域是否是一个结构数组

6、逻辑数据类型:

创建逻辑型数据的函数

logical:将任意类型的数组转变为逻辑类型数组,

非零为真,零为假

true: 产生逻辑真值数组

false: 产生逻辑假值数组

Aa=true(3,4)

```
    在使用true或者false函数创建逻辑类型数组时,若

 不指明参数,则创建一个逻辑类型的标量 >> a=true >> isnumeric(c)
例 >> a=true
                  ans =
  a =
                                在MATLAB中有些函数
  >> b=false
                  >> islogical(a)
                                以is 开头,这类函数是
  b =
                                用来完成某种判断功能
                  ans =
    0
                                的函数。
  >> c=1
                               isnumeric(*):判断输入
                  >> islogical(b)
  c =
                  ans =
                                的参数是否为数值类型
                               islogical(*):判断输入的
  >> isnumeric(a)
                  >> islogical(c)
                               参数是否为逻辑类型
  ans =
                  ans =
```



上机作业

- 1、写出可以完成下列任务的Matlab命令(函数):
 - (1) Matlab中的圆周率 π 约等于多少?
 - (2) Matlab中的浮点运算相对精度是多少?
 - (3) 如何知道命令 realmax, realmin 的具体用法?
 - (4) 如何知道函数 max 的作用? 并举例加以说明。

2、回答以下问题:

- (1) Matlab 是什么含义?
- (2) Matlab 命令 who 与 whos 有什么区别?
- (3) 如何使用 Matlab 命令补全功能?
- (4) Matlab 命令 clear、clc 与 home 有什么区别?



上机作业

- 3、写出可以完成下列任务的Matlab命令:
 - (1) 生成 5 阶的单位阵和 8 阶均匀分布的随机矩阵
 - (2) 生成一个列向量 $x = [1, 3, 5, 7, 9, \dots, 99]$
 - (3) 生成以x 的前 8 个元素为对角线的矩阵 A
 - (4) 生成一个与 任意矩阵A 同维的正态分布的随机矩阵 B
 - (5) A为3*4矩阵,B为3*5的矩阵,计算 A 的转置 与 B 的下三角部分的乘积 C
 - (6) 删除 C 的第三行和第二列
 - (7) 生成由 B 的第 2、4、5 行和第4、1 列组成的子矩阵 D