

《数学实验》第2讲

主要内容:

MATLAB语言的基本语法

数组的创建与使用

MATLAB运算符: 算术、关系、逻辑

MATLAB变量的数据类型

基本输入与格式化输出操作函数: input、disp、sprintf

常用函数: sum、mean、max/min、find、sort、length/size

已知矩阵 A , B , 计算矩阵 C 。

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, B = \begin{bmatrix} 2 & 0 & 3 \\ 4 & 1 & 6 \\ 0 & 5 & 2 \end{bmatrix}, C = AB$$

编写MATLAB程序：

```
A=[1 2 3; 4 5 6; 7 8 9]; %变量A存储矩阵A  
B=[2 0 3; 4 1 6; 0 5 2]; %变量B存储矩阵B  
C=A*B; %变量C为矩阵A与矩阵B的乘积.
```

通过程序了解MATLAB语言的一些基本语法，如：

1. 创建矩阵可用方括号`[]`，矩阵的不同行可用分号分隔，同一行元素用空格分隔。
2. 矩阵乘法运算符用“`*`”；
3. 赋值语句用“`=`”；
4. 注释语句，用`%`开头。

赋值语句

●基本语法

变量名=表达式

示例：

```
a=[2 5 6 7 9];
```

```
a(2)=10
```

• 其他赋值语句，如：

变量名=函数名(输入参数列表)

[变量名列表]=函数名(输入参数列表)

示例： [V1,V2,V3]=myfun(M1,M2)

表达式语句

- 注： 一个语句可以只有表达式
系统自动将表达式的结果赋值给MATLAB内部变量"ans"。

示例：

```
>> a=fix(10*rand(1,5)) %赋值语句， 赋值给变量a
```

a =

9 8 1 4 8

```
>> a-0.5 %表达式语句， 结果赋值给ans
```

ans =

8.50000 7.50000 0.50000 3.50000 7.50000

语句的分隔符

1. 语句分隔符: 分号或逗号;
2. 语句的末尾不使用分号时, 系统会显示执行结果。

示例:

```
a=3; b=4, c=a^2; a(1,4)=b
```

运行结果:

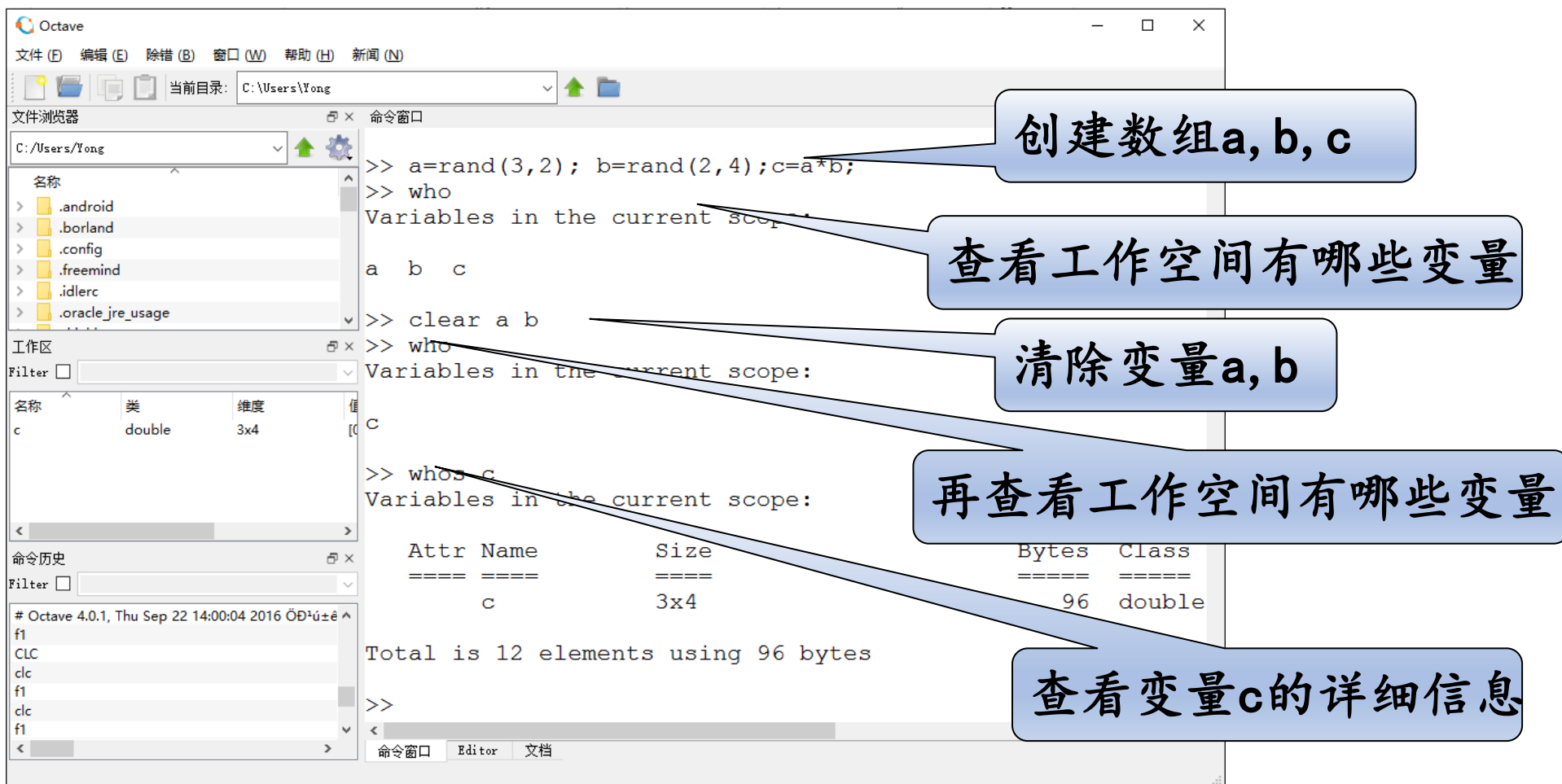
```
b =    4
```

```
a =    3    0    0    4
```

常用命令、快捷键

- `clear` 清除工作空间中的变量
- `clear` 变量名列表
- 示例: `clear A B` 清除变量A, B
- `clc` 清除命令窗口内容
- `who` 列出当前工作空间所有变量名称
- `whos` 列出当前工作空间变量更多信息 (维数, 占用内存字节数等)
- `whos` 变量名列表
- 示例: `whos v1 v2` 列出变量v1, v2的更多信息
- 快捷键: 向上方向键、向下方向键
用于浏览命令窗口历史命令、语句

常用命令示例:



The screenshot shows the Octave environment with the following commands and outputs:

```
>> a=rand(3,2); b=rand(2,4); c=a*b;
>> who
Variables in the current scope:

a  b  c

>> clear a b
>> who
Variables in the current scope:

c

>> whos c
Variables in the current scope:

Attr Name      Size      Bytes  Class
=====
      c         3x4         96  double

Total is 12 elements using 96 bytes
```

Annotations (callouts) point to specific parts of the screenshot:

- 创建数组a, b, c (Create arrays a, b, c) points to the first command.
- 查看工作空间有哪些变量 (Check which variables are in the workspace) points to the first `who` command.
- 清除变量a, b (Clear variables a, b) points to the `clear a b` command.
- 再查看工作空间有哪些变量 (Check which variables are in the workspace again) points to the second `who` command.
- 查看变量c的详细信息 (Check detailed information of variable c) points to the `whos c` command.

创建数组

- 1. 使用方括号
- 同一行的元素用“空格或逗号”分隔
- 不同行的元素用“分号或换行”分隔
- **示例：**

```
x=[1 3 5 6 4]
```

```
y=[1, 3, 5, 6, 4]
```

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

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


创建数组

- **2. 冒号操作符**

- 用于创建行向量 $a:\text{step}:b$ 其中 $a:b$ 等同于 $a:1:b$
- a 为起始值, step 为增量, b 用于判断向量终点值.

- **示例:**

$x=1:5$ 表示 $x=[1\ 2\ 3\ 4\ 5]$, 增量默认为1

$x=1:2:9$ 表示 $x=[1\ 3\ 5\ 7\ 9]$

$x=10:-2:1$ 表示 $x=[10\ 8\ 6\ 4\ 2]$

创建数组

- 3. `linspace(a,b,n)`
- $n-1$ 等分区间 $[a, b]$ 的节点组成的行向量(含区间端点 a, b)

示例:

编写程序将区间 $[-2, 2]$ 四等分, 将区间端点、中间分隔节点这5个数由小到达排列成一个数组。

编写程序:

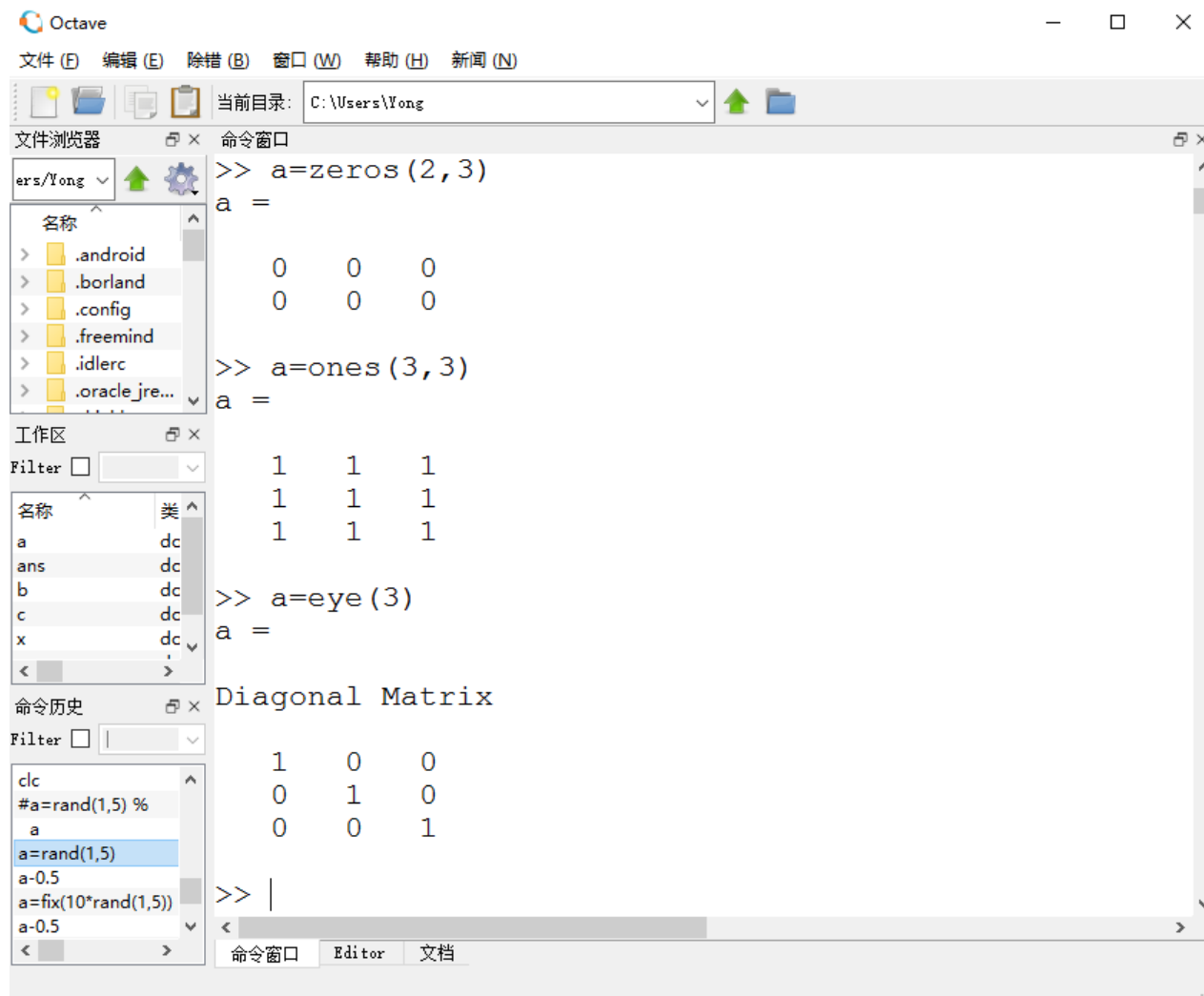
```
x=linspace(-2, 2, 5)
```

表示 $x = [-2 \quad -1 \quad 0 \quad 1 \quad 2]$

注意: 如果要产生一个区间上的均匀节点, 并且指定所产生数组的元素个数, 则使用`linspace`更为方便。

创建数组

- 4. 拼接
 - 示例格式1: `[A B]` 横向拼接要求A,B行数相同,
 - 示例格式2: `[A; B]` 纵向拼接, 要求A,B列数相同.
 - 示例: `z=[rand(2,3), rand(2,2)]`
- 5. 空矩阵 `[]` 产生一个空矩阵
 - 示例: `a=[]`
- 6. 调用函数创建
 - `a = zeros(m, n)` 产生一个m行、n列的零矩阵;多用于变量的初始化
 - `a = ones(m, n)` 产生一个m行、n列的元素全为1的矩阵
 - `a = eye(m, n)` 产生一个m行、n列的单位矩阵



The screenshot shows the Octave software interface. The command window displays the following commands and their outputs:

```
>> a=zeros(2,3)
a =
    0    0    0
    0    0    0

>> a=ones(3,3)
a =
    1    1    1
    1    1    1
    1    1    1

>> a=eye(3)
a =
    1    0    0
    0    1    0
    0    0    1
```

The left sidebar shows the file browser and workspace. The workspace contains variables 'a', 'ans', 'b', 'c', and 'x'. The command history shows the sequence of commands entered.

函数zeros, ones, eye
用法运行示例

提取和修改数组中的元素

- 通过数组下标访问：(1)下标为大于等于1 的整数；(2)下标不能越界
- 常用语法： 示例： $x(i)$, $x(a : b : c)$, $x([a \ b \ c \ d])$, $x(i,j)$
- 获取子阵：
- 获取某一行 $A(r, :)$ 第r行;
- 获取某一系列 $A(:, c)$ 第c列
- 获取子阵 $A(\text{行下标集}, \text{列下标集})$

示例：

```
x=[1 2 3 4 5; 6 7 8 9 10;11 12 13 14 15] %创建矩阵
```

```
y1=x([1, 2], :)% 取第1,2行
```

```
y2=x([2 3],[1 3 4])% 取第2,3行,第1,3,4列
```

提取和修改数组中的元素

- 语法: 数组名(:) 取数组中所有元素组成列向量
- 示例:

```
>> A=rand(3,2)
A =
    0.30030    0.52739
    0.63670    0.96702
    0.69932    0.35359

>> v=A(:)
v =
    0.30030
    0.63670
    0.69932
    0.52739
    0.96702
    0.35359

>>
```

提取和修改数组中的元素

- **修改元素：**用赋值语句修改。如果赋值语句右侧不是一个标量，则要求赋值语句两侧表示的数组维数要相同，否则造成维数不一致的错误。

参考用法1: $A(i,:) = B(k,:)$,

参考用法2: $A([1\ 2],:)=V$

- **示例：**

```
>> A=rand(3,2),B=rand(4,2)
A =
    0.77811    0.95721
    0.41612    0.87731
    0.48357    0.61014

B =
    0.11186    0.48241
    0.92981    0.38390
    0.19615    0.58141
    0.70188    0.30491

>> A([1 2],:)=B([3 4],:)
A =
    0.19615    0.58141
    0.70188    0.30491
    0.48357    0.61014
```

删除数组元素操作

- 操作方式：将空矩阵赋值给相应子阵达到删除目的。
- 用法：

$A(i_1:i_2, :) = []$ %删除A由i1:i2指定的行

$A(:, j_1:j_2) = []$ %删除A由j1:j2指定的列

删除数组元素操作示例

- **示例:**

```
x=[1 5 9; 2 6 10; 3 7 11; 4 8 12]
```

```
%创建4行3列矩阵x
```

```
y=x([1 3 4],[2 3])
```

```
% 取第1,3,4行,第 2,3列赋值给y
```

```
z =x([1 3 5])
```

```
%取x第1,3,5个元素赋值给z
```

```
x([1 2],:)= [ ];
```

```
% 删除x的第1、2行
```

```
x
```

运行结果显示:

```
x = 1    5    9
     2    6   10
     3    7   11
     4    8   12

y = 5    9
     7   11
     8   12

z = 1    3    5
x = 3    7   11
     4    8   12
```

end在存取数组元素方面的特殊用法

- 用法：end在下标表达式中表示最后一个下标值
- 如果end出现在一个向量的下标中，则表示向量的元素个数。
- 如果end出现在一个矩阵的行下标位置，则表示矩阵的行数。
- 如果end出现在一个矩阵的列下标位置，则表示矩阵的列数。
- 示例： `x=[1 5 9; 2 6 10; 3 7 11; 4 8 12];`
`x(end,2)= 0;` x%将矩阵x的最后一行第2列元素赋值为0
运行结果：

x =	1	5	9
	2	6	10
	3	7	11
	4	0	12

end在存取数组元素方面的特殊用法

- 其他例子:

- 操作向量示例

```
t = rand(1,10);
```

```
x1 = t(1:end-1) %取第1个-倒数第2个
```

```
x2 = t(end-2:end)%取倒数第3个-倒数第1个
```

- 操作矩阵示例

```
A = rand(3)
```

```
B = A(1:end-1, :)%取A的第1行-倒数第2行
```

```
C = A(:, [2:end]) %取A的第2列-倒数第1列
```

- 算术运算符:

- 矩阵之间的运算符 $+$ $-$ $*$ $/$ \backslash $^$
- 数组对应元素之间 $.*$ $./$ $.\backslash$ $.^$
- 标量之间的运算符 $+$ $-$ $*$ $/$ $^$ \backslash

- 算术运算符: 标量之间的运算符:

- 标量加减乘除: $+$ $-$ $*$ $/$ \backslash $^$
- 示例:
- $a=5;$
- $3*\text{sqrt}(a^3)$
- $3/6$ % 3除以6, 表达式的值为0.5
- $3\backslash 6$ % 3的倒数乘以6, 表达式的值为2

- 算术运算符: 矩阵操作运算符:

- 矩阵转置 B'
- 矩阵加减: $A+B$, $A-B$, A 与 B 维数相同或其中之一为标量
- 矩阵相乘: $A*B$, A 与 B 为矩阵或其中之一为标量
- 矩阵左除: $A\backslash B$, 当 A 为方阵表示: $A^{-1}B$
- 矩阵右除: A/B , 当 B 为方阵表示 AB^{-1} , 或 B 为标量
- 矩阵幂: A^n , A 为方阵
- 举例:
- $A = \text{rand}(3,3); B = \text{rand}(3,3);$
- $C1 = A\backslash B; C2 = A/B$
- $E1 = C1 - \text{inv}(A)*B$ %inv函数求矩阵的逆
- $E2 = C2 - A*\text{inv}(B)$ %E1,E2 理论上为零矩阵

算术运算符:

数组对应元素操作运算符 `.* ./ \.^`

- 数组对应元素计算:
- 数组相乘: `C=A.*B`
- 数组右除: `C=A./B;`
- 数组左除: `C=A.\B`
- 数组幂: `C=A.^B`
- **要求:** A, B同维数或其中之一为标量

算术运算符:

数组对应元素操作运算符 `.* ./ \.^`

• 示例:

- `A=rand(3,4);`
- `B=rand(3,4);`
- `A.*B, A./B, A.\B, A.^B`
- `T1=A.*2;` %以A的每个元素与2相乘构造数组
- `T2=A.^2;` % 以A的每个元素2次方构造数组
- `T3=2./A ;` % 以A的每个元素的倒数乘以2构造数组
- `T4=2.\A ;` % 以2的倒数乘以A的每个元素构造数组

练习

1. 判断正误:

变量A, B分别表示3行4列矩阵A和4行3列矩阵B, 则两矩阵相乘应该使用表达式A.*B.

- 判断: (_____)
- 此叙述应判定为“错”

正确答案:

A*B

2. 选择题

已知两向量 $a=[2,3,1,0]$, $b=[7,2,5,3]$, 计算两向量对应元素相乘所得向量c, 应采用下列 (_____) 中的代码.

- A. $a=[2,3,1,0]$, $b=[7,2,5,3]$, $c=a*b$
- B. $a=[2,3,1,0]$, $b=[7,2,5,3]$, $c=b*a$
- C. $a=[2,3,1,0]$, $b=[7,2,5,3]$, $c=a.*b$
- D. $a=[2,3,1,0]$, $b=[7,2,5,3]$, $c=b.*a$

正确答案:

(C, D)

❖ 实践题(5 min)

生成矩阵 $A=[1\ 2\ 3;4\ 5\ 6;7\ 8\ 9]$, $B=[2\ 1\ 3;4\ 5\ 7;6\ 7\ 8]$, $C=[1\ 3\ 4;2\ 3\ 4]$ 和向量 $x=[7; 2; 3]$.请MATLAB编程计算:

- 1)矩阵A与向量x相乘, 并用符号d输出结果
- 2)矩阵A与B对应元素点除, 并用D输出结果
- 3)矩阵A与B对应元素点乘, 并用E输出结果
- 4)矩阵A的每个元素均3次方, 并用F输出结果
- 5)用右除命令实现 CA^{-1} , 并用G输出结果
- 6)求矩阵的逆 B^{-1} , 并用W输出结果

要求:

- 1.请用脚本文件编写程序(文件名设置为demo1.m, 存放在桌面新建文件夹test1中)
- 2.一次性输出上述所有5题结果

关系运算符

- 举例说明: 假设 $a=3$; $b=7$;

符号	名称	表达式	结果
<	小于	$t=a<3$	$t=0$
<=	小于或者等于	if $a\leq 3$, $\text{disp('a}\leq 3\text{'})$; end	
>	大于	$a>b$	0
>=	大于或者等于	$a\geq b$	0
==	等于	$a==b$	0
~=	不等于	$a\sim=b$	1

示例: 分析下列语句中数组t1, t2的值

• $x=[4 \quad 9 \quad 1 \quad 2 \quad 1]$;

• $y=[1 \quad 8 \quad 5 \quad 5 \quad 1]$;

• $t1=x\geq 2$

• $u=x-y$

• $t2=x-y\geq 1$

运行输出:

$t1 = \quad 1 \quad 1 \quad 0 \quad 1 \quad 0$

$u = \quad 3 \quad 1 \quad -4 \quad -3 \quad 0$

$t2 = \quad 1 \quad 1 \quad 0 \quad 0 \quad 0$

逻辑运算符

- 逻辑运算的值:

为0 (代表“假”) 或1 (代表“真”)

- 三种运算符:

与(and) &

或(or) |

非(not) ~

- 示例: 如果为 $a>3$ 且 $b>3$ 则显示handle 1, 否则显示handle 2.

```
a=3; b=7;  
if a>3 & b>3,  
    disp('handle 1')  
else  
    disp('handle 2')  
end
```

分析: $a>3$ 为0 (假), $b>3$ 为1 (真)
二者不同时为“真”
所以 $a>3$ & $b>3$ 的值为0 (假)。

运行输出:

handle 2

数据类型

- 主要的数据类型: double char sym struct cell
- `a=rand(3); b='Li San';` %a为double型, b为char型
- `syms x, y=1 + x^2` %x,y为sym类型;对y赋值的语句含符号对象
- `F.name='li San', F.birth=1999, F.src=rand(3)` %F为struct型

• `whos a b x y F`

Name	Size	Bytes	Class	Attributes
F	1x1	620	struct	
a	3x3	72	double	
b	1x6	12	char	
x	1x1	112	sym	
y	1x1	112	sym	

查看变量类型

使用**whos**命令

用法: whos 变量名列表

示例:

```
a=rand(1,2); b='abc'
```

```
whos a b
```

运行输出:

Name	Size	Bytes	Class
Attributes			
a	1x2	16	double
b	1x3	6	char

使用函数: **class**

用法: **class(变量名)**

该函数返回变量的数组类型的char型数组, 如'double', 'char'。

示例:

```
a=rand(3); b='abc'
```

```
class(a), class(b)
```

运行结果:

```
ans = double
```

```
ans = char
```

cell数组基本用法

- 创建数组用法:

`a=cell(m,n)`

- 存取cell数组用法示例:

`a{i}` i 为下标

`a{i,j}` i,j 分别为行、列下标

特点: 一个cell数组中的元素的类型
可以互不相同

运行示例:

```
>> a=cell(2,3)
```

```
a =
```

```
    []    []    []
```

```
    []    []    []
```

```
>>
```

```
a{1,1}='abc';a{1,2}=rand(3);a{1,3}=cell(1,2);
```

```
>> a
```

```
a =
```

```
    'abc'    [3x3 double]    {1x2 cell}
```

```
    []
```

```
    []
```

```
    []
```

cell数组基本用法

例1. 假设有2条公交线路，第1条公交线路站点编号依次为101, 102, 103, 104；第2条公交线路站点编号依次为201, 202, 102, 203, 204, 205。请分别用double型数组和cell型数组存储这些信息。

程序实现：

```
D=[ 101 102 103 104 0 0 %第1条公交线路信息  
    201 202 102 203 204 205];%第2条公交线路信息  
A{1}=[101 102 103 104]; %第1条公交线路信息  
A{2}=[201 202 102 203 204 205];%第2条公交线路信息
```

输入函数input

第一种用法:

input(提示信息字符数组)
用于输入一般类型数据

•示例:

•g=input('输入您的成绩: ')

•运行输出:

•输入您的成绩: 95

•g = 95

第二种用法:

input(提示字符串, 's')

用于输入字符数组 (含第2个参数's')

•示例:

•name=input('输入您的姓名: ', 's')

•运行输出:

•输入您的姓名: Li San

•name = Li San

显示数组的元素

- 数组显示函数`disp(变量名)`
- 特点：显示数组内容，但不输出变量名
多用于调试程序时显示数组内容

- 示例：

```
a=rand(1,3);
```

```
disp(a)
```

```
a
```

- 运行输出：

```
0.6441    0.6872    0.7481
```

```
a =
```

```
0.6441    0.6872    0.7481
```

格式化输出函数sprintf

- 功能：将数据格式化输出为字符串
- 用法：str = sprintf(formatSpec,A1,A2,...,An)
- 将数组A1,A2,...,An按照参数formatSpec格式化为字符串赋给str.
- %d 格式化整数 %f 格式化浮点数
- %c 格式化单个字符 %s 格式化字符串数组
- 百分号符号字符后可以加个**整数**, 用以限定输出化为字符串的长度, 例如:
%5d, %5s。

• 示例:

```
a=[pi,sqrt(2)];name='Li San';grade=[86 95 89];  
s1= sprintf('%0.5f',pi)%将该实数化为5个小数位字符串
```

```
s2= sprintf('%10.6f', a)%将a化为10个字符长,含6个小数位的字符串
```

```
s3=sprintf('%8s%3d%3d%3d',name,grade)
```

运行结果:

```
s1 = 3.14159
```

```
s2 = 3.141593 1.414214
```

```
s3 = Li San 86 95 89
```


❖ 实践题(5 min)

- 1) 利用input函数输入0.952并赋值给变量g; 输入字符串'matlab'赋值给w;
- 2) 建立一个2行2列的cell数组C; C的1行1列位置放入3阶单位阵; C的1行2列位置放入2阶全1矩阵; C的2行1列位置放入g; C的2行2列位置放入w。
- 3) 用disp函数显示C{1,2}; 用sprintf函数显示C{2,1}并保留两位小数
- 4) 建立结构体F, 分别对结构体存入数字F.number = 0.3;字符串F.charn = 'abc'; 2阶单位矩阵F.mat = eye(3)
- 5) 命令窗口调用F.number(回车)、C{2,2} (回车)

要求:

1.请用脚本文件编写程序(文件名设置为demo2.m, 存放在桌面新建文件夹test1中)

一、求和函数sum

用法：

- `s=sum(v)` 求向量`v`中元素的和
- `s=sum(A,1)` 或 `s=sum(A)` 求矩阵`A`中每列的和，返回成1个行向量
- `s=sum(A,2)` 求矩阵`A`中每行的和，返回成1个列向量

示例：

```
a= fix(1000*rand(3,5))/1000;  
s=sum(a), s2=sum(a,1) % s,s2完全相同  
t =sum(a',1)  
t2= sum(a,2)% t,t2元素值相同，差别是什么
```

运行输出：

```
s = 1.00700 0.34800 2.46700 0.87400  
1.97500  
s2 = 1.00700 0.34800 2.46700 0.87400  
1.97500  
t = 1.8430 2.2620 2.5660  
t2 =  
1.8430  
2.2620  
2.5660
```

二、求平均值mean

用法:

- $s = \text{mean}(v)$ 求向量 v 中元素的平均值
- $s = \text{mean}(A, 1)$ 或 $s = \text{mean}(A)$ 求矩阵 A 中每列的平均值，返回成1个行向量
- $s = \text{mean}(A, 2)$ 求矩阵 A 中每行的平均值，返回成1个列向量

示例:

```
X = [1 2 3; 3 3 6; 4 6 8; 4 7 7]
```

```
v1 = mean(X, 1)
```

```
v2 = mean(X, 2)
```

运行输出:

```
X = 1 2 3
```

```
3 3 6
```

```
4 6 8
```

```
4 7 7
```

```
v1 = 3.0000 4.5000 6.0000
```

```
v2 =
```

```
2
```

```
4
```

```
6
```

```
6
```

三、求最大值max，求最小值min

用法：

[v, I]=max(x)

- 如果x为向量，v为向量中的最大元素；I为最大元素在x中的下标。
- 如果x为矩阵，v为每列的最大元素组成的行向量，I则为每列最大元素的行下标组成的向量。

[v, I]=min(x)

- 如果x为向量，v为向量中的最小元素；I为最小元素在x中的下标。
- 如果x为矩阵，v为每列的最小元素组成的行向量，I则为每列最小元素的行下标组成的向量。

运行示例：

```
>> x=[5 1 4 7 3]
x =    5    1    4    7    3
```

```
>> [v,I]=max(x)
v =    7
I =    4
```

```
>> [v,I]=min(x)
v =    1
I =    2
```

四、length和size

length(x)

- 如果x为向量，则返回x的元素个数；
- 如果x为矩阵，则返回矩阵x行数和列数的最大值。

size(x)

- 如果x为向量、矩阵，则返回行数、列数组成的行向量。

运行示例：

```
>> x=[3 2 4 5 1]
```

```
x =    3    2    4    5    1
```

```
>> t1=length(x), t2=size(x)
```

```
t1 =    5
```

```
t2 =    1    5
```

五、排序函数sort

用法:

- `[B, I]=sort(v)` 对向量`v`中元素排序, `B`为按递增排序后的元素; `I`为排序后数组`B`中的元素在原数组`v`中的位置下标.

示例:

```
a=[1 5 6 7 4 9]
v=sort(a)
[v2,idx2]=sort(a)
v_byidx = a(idx2)
```

按升序排列:

```
a=[1 5 6 7 4 9]
v1=sort(a)
v1_c=sort(a, 'ascend') %升序排列
```

按降序排列:

```
a=[1 5 6 7 4 9]
v=sort(a, 'descend') %降序排列
```

六、查找函数find

用途：

- find函数用于查找数组中的非零元素位置、元素值.
- 结合逻辑表达式可以返回所需要元素的位置下标.

例1：

```
a=[1 5 6 7 4 9];
```

```
idx=find(a>=6)
```

运行输出：

```
idx =
```

```
    3    4    6
```

例2:

已知两个数组。请找出数组x与y相同位置的元素值之差不超过2的元素对的个数。例如:

$x = [1 \ 9 \ 0 \ 7 \ 8]$, $y = [8 \ 0 \ 3 \ 2 \ 8]$.

```
x = [1    9    0    7    8];  
y = [8    0    3    2    8];  
idx=find((x>=y-2) & (x<=y+2)  
)  
s=length(idx)
```

运行输出:

idx = 5
s = 1

例3:

```
x = [1    9    0    7    8];  
y = [8    0    3    2    8];  
s = sum((x>=y-2) &  
(x<=y+2) )
```

运行输出:

s = 1

在逻辑运算中若运行:

$(x \geq y-2) \ \& \ (x \leq y+2)$

则输出:

ans =

0 0 0 0 1

❖ 实践题(5 min)

生成矩阵 $A=[1\ 2\ 3;4\ 5\ 6;7\ 8\ 9]$, $b=[2\ 1\ 3\ 4\ 5]^T$, $C=[1\ 3\ 4;2\ 3\ 4;3\ 4\ 5]$ 和向量 $x=[7, 2, 3]$.请MATLAB编程计算:

- 1)找出向量 x 最大元素和下标, 分别用 d 和 idx 输出
- 2)找出矩阵 A 最大元素, 并用 e 输出结果
- 3)对向量 b 进行降序排序, 分别用 g 和 $idx2$ 输出排序结果和相应坐标
- 4)找出矩阵 A 元素大于等于矩阵 C 的元素下标 (存为 $idx3$) 和相应元素 (存为 h)

要求:

1.请用脚本文件编写程序(文件名设置为**demo3.m**, 存放在桌面新建文件夹**test2**中)

MATLAB语言的基本语法

数组的创建与使用

MATLAB运算符: 算术、关系、逻辑

MATLAB变量的数据类型

基本输入与格式化输出操作函数: input、disp、sprintf

常用函数: sum、mean、max/min、find、sort、length/size