

# Matlab 实例教程 1

## 基础入门手册

既是简洁实用的入门教程  
又是基本编程函数/方法/思路的必备查阅手册

发布日期: 2022.6.3

## 目录

第一章 入门.....	4
1.1 认识软件.....	4
1.2 创建变量.....	5
1.3 调用函数.....	5
1.4 脚本编程.....	6
1.5 语法高亮.....	6
第二章 矩阵.....	7
1.6 创建矩阵.....	7
1.7 创建矩阵的函数.....	7
1.8 连接矩阵.....	8
1.9 矩阵索引.....	9
1.10 矩阵元素修改.....	10
1.11 重塑矩阵.....	10
1.12 矩阵属性.....	11
1.13 特殊矩阵形式.....	12
1.14 多维数组.....	13
1.15 性能优化.....	14
第三章 运算符.....	14
1.16 算术运算.....	14
1.17 算术常用函数.....	16
1.18 逻辑运算.....	17
1.19 关系运算.....	18
1.20 运算符的优先级.....	19
1.21 兼容性.....	20
第四章 数据类型.....	21
1.22 整数型.....	21
1.23 浮点型.....	22
1.24 Inf 和 NaN .....	23
1.25 显示格式.....	24
1.26 字符型.....	25
1.27 特殊字符.....	26
1.28 混合连接的类型.....	27
1.29 混合运算的类型.....	28
1.30 字符串型.....	29
1.31 缺失字符串 .....	30
1.32 格式化文本.....	31
1.33 字符型与数值型的转换.....	31
1.34 元胞数组.....	32
1.35 元胞数组的修改.....	33
1.36 结构数组.....	34
1.37 结构数组的处理.....	36
1.38 表.....	37

1.39 表的数据处理.....	38
1.40 表的读入写出.....	39
1.41 日期时间型.....	39
1.42 缺失数据的处理.....	40
1.43 类型识别.....	41
第五章 脚本和函数.....	43
1.44 循环和条件语句.....	43
1.45 函数文件.....	43
1.46 函数句柄.....	44
1.47 输入参数数量.....	44
1.48 变量名称.....	45
1.49 实时脚本.....	46
1.50 保存和载入.....	46
1.51 预设和界面调整.....	47
1.52 绘图.....	47
第六章 符号运算.....	48
1.53 符号对象.....	48
1.54 符号函数和符号矩阵.....	49
1.55 基本运算.....	49
1.56 使用假设.....	50
1.57 清除假设.....	51
1.58 可变精度算术.....	52
1.59 运算精度的选择.....	52
1.60 数值型的转换.....	53
1.61 查找符号变量.....	54
1.62 变量替代.....	54
1.63 长表达式的缩写.....	55
1.64 符号函数绘图.....	56
End .....	56

## 第一章 入门

写在前面：本文档原为《freexyn 编程实例视频教程系列 1: Matlab 基础入门》课程文档，后经韩松岳同学悉心整理并添加程序和注释，形成本文，现全网免费发布，版权所有，仅供个人学习交流使用，其中不足之后，欢迎补充指正；若程序注释没有完全表达其含义，可对照视频学习。

### 1.1 认识软件

#### 1. 软件简介

Matlab 是 matrix 和 laboratory 组合缩写，意为矩阵实验室

主要功能和用途：

用于科学和工程计算的高级语言；

用于调整运行环境进行迭代探索、设计和问题解决；

用于可视化数据的图形表达和用于创建自定义绘图的工具；

用于曲线拟合，数据分类，信号分析，控制系统调整等；

用于各种工程和科学应用的附加工具箱；

用于自定义用户界面构建应用程序；

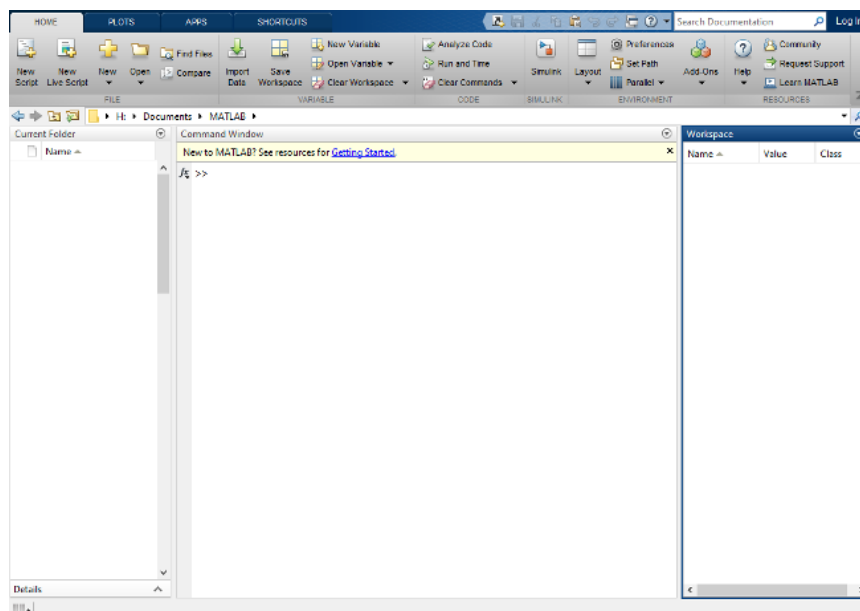
用于接口 C/C++、JAVA、.NET、Python、SQL 和 Excel 等；

#### 2. 软件安装

软件安装过程略

#### 3. 认识界面

启动 Matlab，默认的界面布局



当前文件夹：访问文件的路径，也是工作目录

命令窗口：输入命令并运行，由提示符“>>”指示

工作区：显示创建或导入的数据或变量

历史命令窗口：记录命令窗口中执行过的命令

## 1.2 创建变量

### 1. 在 Matlab 中创建和使用变量

作者: freexyn 整理/注释: 韩松岳

### 2. 认识命令

whos

clear

### 3. 说明

Matlab 中创建变量不需要声明，类型默认使用双精度浮点型  
创建变量后在命令窗口立即显示结果，并在工作区添加该变量  
当不指定输出变量时，默认使用 `ans` 存储计算结果  
程序语句以分号结尾，执行运算后不显示输出结果  
命令窗口中回车键起运行程序功能  
若输入一条语句后换行，但不想立刻运行，用 `Shift+Enter`  
允许一行输入多个语句，用逗号或分号分隔每个语句  
快速回调之前执行过的语句，可以用方向键的上键和下键  
清除或者不执行当前语句，用 `Esc` 键退出  
可以选中部分程序，然后按回车键直接运行这部分程序  
创建数组或矩阵，用中括号把数值括起来

## 1.3 调用函数

### 1. 几个常用函数，说明 Matlab 调用函数的方法

### 2. 认识函数

求最大值 `max`

正弦函数 `sin`

显示字符 `disp`

清除屏幕 `clc`

### 3. 说明

Matlab 提供了很多函数方便直接调用，调用格式：

`[out1,out2]=function(input1,input2)`

调用时把输入参数括在括号内，有多个输入参数时，用逗号分隔  
相应的，有多个输出值时，用逗号分隔，并用中括号括起来

### 4. 命令语法和函数语法的对偶性

当不需要输出，且所有输入都是字符向量

以下两种语法是等效的：

`functionName (input1, ..., inputN) %函数语法`

`functionName input1 ... inputN %命令语法`（缺点：参数必须是字符型，而不能是数值型）

## 1.4 脚本编程

### 1. 使用脚本文件编程的方法

### 2. 说明

#### 2.1 概念

脚本文件是包含一系列的命令和函数的程序文件，扩展名是.m

当需要把多个程序语句组合起来完成一项总体运算和功能时，常用脚本文件，方便存储、管理和重用，是 Matlab 里普遍的编程方式

#### 2.2 创建脚本的方法


%在当前工作目录右键-新建文件-脚本；

%菜单栏，“主页” - “新建脚本”。

%使用 edit 命令；

#### 2.3 运行脚本的方法

要运行的脚本文件必须存储在当前工作目录或者搜索路径上

%在编辑器中点击运行按钮；

%快捷键 F5；

%当前工作目录选中该脚本右键-运行。

%在命令窗口中输入文件名称回车运行；

## 1.5 语法高亮

### 1. Matlab 显示不同程序元素的默认方式

作者：freexyn 整理/注释：韩松岳

### 2. 说明

#### 2.1 语法高亮

Matlab 会把不同的程序字段用不同的颜色区分出来

关键字是蓝色的，如 if

注释是绿色的，如%注释

字符向量是紫色的，如'adsaf'

未终结的字符向量是栗色的，如'ads

程序中无效字符是红色的

命令窗口输出时错误提示是红色的

#### 2.2 使用注释

%在文本开头添加百分号%

%快捷键，选中文本按 Ctrl+R，取消注释 Ctrl+T

## 第二章 矩阵

### 1.6 创建矩阵

#### 1. 创建矩阵

#### 2. 认识创建方法

基本运算符 `[]`

冒号运算符 `:`

用逗号或空格分隔同一行的元素

用分号或回车键分隔不同的行

创建序列 `linspace`

#### 3. 说明

矩阵是一个二维的数据阵列

Matlab 是一个基于矩阵的计算环境，最基本的数据结构是矩阵

单个数值也存储为矩阵，在这种情况下，矩阵的大小为 1 乘 1

再例如 'Hello World' 这样的字符元素是 1×1 的矩阵

也支持有两个以上维度的数据结构，即 n 维数组

#### 4. 实例演示

```
%1_6
[1 2 3] %空格分隔元素
[1,2,3] %逗号分隔元素
[1 2 3
    4 5 6] %回车创建不同行
[1 2 3;4 5 6] %分号分隔不同行

1:5 %行向量: 1 至 5
1:4.5 %取整到 4
4:1 %空的行向量 (错误写法)
4:-1:1 %4 至 1 行向量 (正确写法)
2.5:0.3:3.2 %按 0.3 增长至边界值内

linspace(1,5,5) %参数分别为起点、终点和元素个数
linspace(1,5) %不指定元素个数, 则默认 100 个元素
```

### 1.7 创建矩阵的函数

#### 1. 介绍一些创建矩阵的常用函数

作者: freexyn 整理/注释: 韩松岳

#### 2. 认识函数

全 0 矩阵 `zeros`

全 1 矩阵 `ones`

单位矩阵 `eye`

对角矩阵 `diag`

魔方矩阵 `magic`

随机矩阵 `rand`

上三角矩阵 `triu`

下三角矩阵 `tril`

### 3. 实例演示

```
%1_7
zeros(1,4) %1 行 4 列全 0 矩阵
zeros(3,4) %3 行 4 列全 0 矩阵
zeros(4) %4 阶全 0 矩阵
ones(1,4) %1 行 4 列全 1 矩阵
eye(3) %3 阶单位矩阵
eye(2,3) %2 行 3 列单位矩阵
magic(3) %3 阶魔方矩阵
magic(30)
a=rand(1,4) %1 行 4 列随机矩阵
b=rand(4) %4 阶随机矩阵
diag(b) %提取对角线元素
diag(a) %a 为向量，则将 a 元素扩展为 n 阶矩阵
triu(b) %提取上三角形元素
tril(b) %提取下三角形元素
tril(b,1) %添加偏移量参数，1 正数往右上
tril(b,2)
tril(b,-1) %向左下偏移 1，再取下三角元素
```

## 1.8 连接矩阵

### 1. 矩阵的连接

### 2. 认识连接方法

基本连接符`[]`

水平连接 `horzcat`

垂直连接 `vertcat`

平铺复制 `repmat`

对角分块 `blkdiag`

任意维度连接 `cat`

### 3. 说明

矩阵连接是通过连接一个或多个矩阵来创建一个新矩阵的过程

与创建类似，逗号或空格实现水平连接，分号实现垂直连接

连接后的矩阵要仍然保持矩形结构才能实现连接

也就是说，水平连接矩阵，每个矩阵必须具有相同的行数

垂直连接时，每个矩阵必须具有相同的列数

### 4. 实例演示

```
%1_8
a=[1 2;3 4]
b=[5 6;7 8]
```



```
[a,b] %矩阵水平连接
[a;b] %垂直连接
horzcat(a,b) %水平连接函数
vertcat(a,b) %垂直连接函数
repmat(a,1,3) %平铺复制为 1 行 3 列矩阵
repmat(a,3,3)
blkdiag(a,b) %对角分块重组函数：将矩阵 a 和 b 整体分别视为对角线上元素进行重组
e=[1 2]
cat(1,e,e) %任意维度连接：维度、矩阵 e、矩阵 e。1 维按列纵向连接
cat(2,e,e) %2 维：横向连接
cat(3,e,e) %3 维：页面方向（三位方向）
```

## 1.9 矩阵索引

### 1. 矩阵的索引

### 2. 认识函数

获取矩阵的行列数 `size`

### 3. 概念

一个矩阵里有多个元素，要想访问或修改其中的元素，使用索引  
索引 3 种方式：

- ①组合索引 `A(i,j)`，也称下标索引
- ②线性索引 `A(i)`，按列优先的顺序依次向下索引
- ③逻辑索引，在逻辑为真的位置返回矩阵的元素

### 4. 说明

组合索引和线性方式可以互换

访问多个元素或不连续的元素可以把索引写成矩阵的形式

逻辑索引的维度必须与矩阵的维度相同

索引超出矩阵范围的元素会报错

### 5. 实例演示

```
%1_9
a=magic(3) %3 阶魔方矩阵
a(2,3) %组合索引：索引第 2 行第 3 列元素（2 个参数，用逗号分隔）
a(3) %线性索引：列优先索引（1 个参数）

size(a) %获取矩阵行列维度
sub2ind([3,3],2,3) %组合索引转化为线性索引，三个输入参数为：矩阵维度、组合索引行位置、组合索引列位置
[row,col]=ind2sub(size(a),8) %线性索引转化为组合索引

% a(3,4) %超出范围报错（初学者常犯错误）

a(1,:) %冒号：表示任意行或任意列
a(:,1) %任意行的第 1 列
a(:,:)

a(1,2:3) %第 1 行，第 2-3 列
```

```

a(1,[1 3]) %第 1 行, 第 1 列和第 3 列
a(1,end) %end 表示最后一个: 第 1 行最后 1 列
a([1 3]) %线性索引: 矩阵 a 第 1 个和第 3 个元素
a([1 2;4 5]) %提取第 1、2 和第 4、5 个元素并分别放置在矩阵第 1 行和第 2 行
a(:) %按列优先转变为单个数字序列, 返回一列值

b=[1 2;3 4]
c=[true false;true false] %逻辑数组结果用 1/0 表示
b(c) %逻辑索引: 逻辑值为真返回对应元素, 为假不返回

```

## 1.10 矩阵元素修改

### 1. 矩阵元素的修改、添加和删除

#### 2. 说明

通过指定矩阵索引修改、添加或删除相应元素  
 索引超出矩阵范围的元素会报错, 但可以赋值  
 删除元素后的矩阵仍然要保持矩形, 否则报错  
 必要时, 对超出矩阵范围的索引赋值需要预分配内存

#### 3. 实例演示

```

%1_10
a=[1 2 3;4 5 6;7 8 9]
a(2,2)=10
a(2)=10
a(3,4)=10 %超出矩阵范围索引会报错, 但可以赋值。扩展为索引维度再对该索引元素赋值, 其余
扩充位置填充元素 0
a(3:4,4:5)=[1 2;3 4] %通过组合索引赋值, 先扩展维度再赋值

a(:,2)=[] %任意行第 2 列变为空: 删除
% a(1,2)=[] %矩阵单个元素删除报错, 删除单个元素无法保持矩阵维度
a(2)=[] %线性索引先按列优先扩展为单个数字序列, 删除单个元素可行

```

## 1.11 重塑矩阵

### 1. 重塑矩阵包括矩阵的元素重排、旋转、翻转、移位、排序等

#### 2. 认识函数

元素重排 reshape

旋转 rot90

左右翻转 fliplr

上下翻转 flipud

翻转 flip

循环移位 circshift

排序 sort

按行排序 sortrows

判断是否排序 issorted

### 3. 实例演示

```
%1_11
%freexyn
a=[1 2 3 4;5 6 7 8]
reshape(a,4,2) %元素重排，参数：矩阵名称、重排行数、重排列数。重排规则：按列优先重排
为所需维度
% reshape(a,3,3) %元素数量不符报错
rot90(a) %矩阵旋转：逆时针 90 度

fliplr(a) %矩阵左右翻转
flipud(a) %矩阵上下翻转
flip(a,1) %矩阵任意方向翻转：参数 1 为方向，1 维方向是上下翻转
flip(a,2) %2 维方向是左右翻转

circshift(a,[0 2]) %矩阵循环移位：矩阵、移动步长（用向量表示行数、列数），移动方向为
坐标轴正方向
circshift(a,[-1 2])

sort(a) %矩阵元素的列排序：默认升序排列（行不保持）
sort(a,'descend') %参数：降序排列
issorted(a) %判断是否按升序已排列
issorted(a,'descend')

b=magic(3)
sortrows(b) %按行（保持）排序：将行视为整体，不指定参数则默认按每行第 1 元素大小升序排
列
sortrows(b,2) %指定参数 2，即按每行第 2 个元素排序
sortrows(b,3)
```

## 1.12 矩阵属性

### 1. 矩阵的属性

描述矩阵的信息，包括大小、长度、元素数目和维度等

### 2. 认识函数

大小 size

长度 length

元素数目 numel

维度 ndims

### 3. 实例演示

```
%1_12
a=[1 2 3 4;5 6 7 8]
size(a) %矩阵行数列数
length(a) %矩阵长度：取行数、列数中的最大值
length(a') %矩阵转至后，长度仍为 4
numel(a) %返回元素个数
ndims(a) %矩阵维度数：2 即行和列（2 维度）
ndims(a(1,:)) %组合索引提取出第 1 行索引再计算维度，即行向量的维度，结果 2 因仍为行和
列（2 维度）
ndims(a(1,1)) %提出 1 行 1 列元素计算维度，结果仍为 2，因单元素认为 1*1 矩阵（2 维度）
```

## 1.13 特殊矩阵形式

### 1. 特殊的矩阵形式：空矩阵、标量和向量

作者：freexyn 整理/注释：韩松岳

### 2. 认识函数

判断空矩阵 `isempty`

判断标量 `isscalar`

判断向量 `isvector`

### 3. 概念

#### 3.1 空矩阵

有一个或多个等于零的维度 ( $0 \times 0$ ,  $0 \times 1$ ,  $1 \times 0$ )

#### 3.2 标量

维度为  $1 \times 1$  的矩阵，在 Matlab 中显示为单个实数或复数

#### 3.3 向量

维度为  $1 \times n$  或  $n \times 1$  的矩阵，在 Matlab 中显示为一个行或一个列

### 4. 实例演示

```
%1_13
%freexyn
a=[] %空矩阵
size(a) %获取行列数
length(a) %矩阵长度
numel(a) %元素个数
ndims(a) %矩阵维度
isempty(a) %判断是否空矩阵
zeros(0,1) %全 0 矩阵指定一个维度为 0

a=2 %创建标量
size(a)
length(a)
numel(a)
ndims(a)
isempty(a)
isscalar(a) %判断是否为标量

a=[1 2 3] %创建向量
size(a)
length(a)
numel(a)
ndims(a)
isempty(a)
isscalar(a)
isvector(a) %判断是否向量
```

## 1.14 多维数组

### 1. 多维数组的创建和索引方法

### 2. 认识函数

维度简化 squeeze

### 3. 概念

Matlab 中具有多于两个维度的数据阵列被称为多维数组

多维数组是普通二维矩阵的扩展

如图为  $3 \times 3 \times 2$  的三维数组（维度名称分别为行、列、页面）

			11	12	13
1	2	3	14	15	16
4	5	6	17	18	19
7	8	9			

### 4. 实例演示

```
%1_14
a=[1 2 3;4 5 6;7 8 9]
b=[11 12 13;14 15 16;17 18 19]
c=a
c(:, :, 2)=b %数组 c 任意行任意列的第 2 个页面赋值数组 b。显示时，对高维数组进行拆分，拆分为 2 维数组显示
ndims(c) %数组维度
size(c) %行、列、页面数
d=c %3 维数组赋值给 d
d(:, :, :, 2)=c %d 的任意行、列、页面的第 4 个维度的第 2 索引赋值一个 3 维数组 c。显示时，高维数组依次拆分，遍历循环显示每个 2 维数组
ndims(d) %4 维
size(d) %每个维度大小：3 3 2 2

%% 用创建矩阵的函数创建高维数组
zeros(2,2,2) %3 维全 0 数组
repmat(10,[2 2 2]) %矩阵平铺重塑：将标量 10 平铺重塑为 2 行 2 列 2 页面的数组
reshape(c,[2 3 3]) %对高维数组进行重塑：c 是 3*3*2 数组重塑为 2*3*3 数组

c(1,[1 2],2) %组合索引：访问数组 c 中第 1 行、第 1 和 2 列中、第 2 个页面的元素
d(:, :, 1, 1) %（4 维数组）组合索引：任意行、列、第 1 页面、第四维度 1 的元素
c(1,1) %对于 3 维数组使用低维索引，索引第 1 行第 1 列，不指定页面，也可索引。高维数组使用低维索引，会自动用 1 补齐末尾索引
a(1,1,1) %低维数组用高维索引。原理：Matlab 中任何数组都可理解为无限尾随 1 的高维数组
e=ones(2,2,1,1,1,1,1) %全 1 矩阵：2 行 2 列，后面尾随 1 无实际意义
e=ones(2,2,1,1,1,2,1) %全 1 矩阵：2 行 2 列，第 6 维为 2，会将 3、4、5、7 维度初始化为 0
squeeze(e) %维度简化：简化高维数组中不必要的维度
```

## 1.15 性能优化

### 1. 在处理大型数组时，兼顾 Matlab 运算性能的优化

#### 2. 说明

使用大型数组时，尽量避免创建不必要的副本

处理数组容量不断变化的问题时，合理的进行预分配内存

把代码放入程序文件中，比在命令窗口中，运算效率高

#### 3. 实例演示

```
%1 15
a=magic(3)
a(3,3)=10
b=a %数据一样，变量名称不同，因此 a、b 变量同时指向了同一矩阵
a(3,3)=15 %执行该命令，先预分配内存并复制一份 a 的原始值，保留在上个语句 b 中，然后再改变 a 的元素值
%预分配内存：输入变量时会初始化分配所需内存，而后续编程中变量变化需要更多内存时，需额外内存支持，可能面临分布式的内存分布，调用效率受到影响，因此需预分配内存
a=zeros(300,300) %初始化 300*300 全 0 矩阵，Matlab 会分配相应内存，在该维度内对元素进行相应修改替换，确保在完整内存中高效运行
% a=zeros(30000,30000)
```

## 第三章 运算符

### 1.16 算术运算

#### 1. 算术运算（arithmetic）主要指加减乘除、幂和舍入等运算

#### 2. 说明

Matlab 有两种不同类型的算术运算：数组运算和矩阵运算

数组运算基于元素的运算，支持任意向量、矩阵和多维数组

矩阵运算遵循线性代数的规则

字符（.）区分矩阵运算和数组运算

数组运算和矩阵运算的加减法则相同，所以.+和.-是不必要的

运算数之一为标量时，乘法法则相同，所以.\*是不必要的

运算数都为标量时，除法法则相同，所以./是不必要的

#### 3. 两种运算符

数组运算符汇总表

运算符	功能	描述	函数
+	加法	A+B	plus
+	一元加	+A	uplus
-	减法	A-B	minus

运算符	功能	描述	函数
-	一元减	-A	uminus
.*	乘法	A.*B 是对应元素的乘积	times
.^	幂	A.^B 是对应元素 A(i,j)的 B(i,j)次幂	power
./	右除	A./B 是对应元素的 A(i,j)/B(i,j)	rdivide
.\	左除	A.\B 是对应元素的 B(i,j)/A(i,j)	ldivide
.'	转置	A.'是数组 A 的转置	transpose

矩阵运算符汇总表

运算符	功能	描述	函数
*	矩阵乘法	A*B	mtimes
\	矩阵左除	$x = A \backslash B$ 是等式 $Ax = B$ 的解	mldivide
/	矩阵右除	$x = B/A$ 是等式 $xA = B$ 的解	mrdivide
^	矩阵的幂	如果 B 是一个标量, A^B 是 A 的 B 次幂; 对于其他值 B, 计算涉及特征值和特征向量	mpower
'	复共轭转置	A'是 A 的转置。对于复矩阵, 是复共轭转置	ctranspose

#### 4. 实例演示

```
%1_16
a=[1 2;3 4]
a+a %数组加法
a-a %数组减法
a.*a %数组乘法: 对应元素乘积
a*a %矩阵乘法: 线性代数
a./a %数组除法: 对应元素相除
a/a %矩阵除法: 得出单位矩阵对角线为 1
a.^3 %数组幂: 单个元素分别求幂
a^3 %矩阵幂: 即 a*a*a
a*a*a

a*5 %有标量作为计算数时, 数组和矩阵乘法法则相同
a.*5 %因此可用矩阵乘法*替代数组乘法
a' %复共轭转置
a.' %转置: 行数变列数

sin(a) %对 a 中每个元素求 sin (调用函数大多执行数组运算)
[sin(1) sin(2);sin(3) sin(4)] %与上式相同
exp(a) %数组的指数运算: 函数调用, 同上述 sin 运算
expm(a) %矩阵的指数幂运算: 函数后加 m, matrix

(-1)^(1/2) %负数开方产生虚数单位
sqrt(-1) %同上
(5+2i)*(5-2i) %复数运算
```

## 1.17 算术常用函数

### 1. 介绍一些算术运算的常用函数

### 2. 认识函数

加 plus

减 minus

乘 times

除 rdivide

矩阵乘法 mtimes

矩阵左除 mldivide

求和 sum

乘积 prod

舍入 round

向-inf 舍入 floor

向 inf 舍入 ceil

向 0 舍入 fix

模操作 mod

### 3. 实例演示

```
%1_17
%freexyn
a=[1 2;3 4]
a+a %数组加法
plus(a,a) %数组加法（函数形式）
minus(a,a) %数组减法
times(a,a) %数组乘法：单个元素对应相乘
rdivide(a,a)
mtimes(a,a) %矩阵乘法
mldivide(a,a) %矩阵除法

sum(a) %矩阵 a 求和：按列操作
sum([1 2 3 4]) %行向量求和：所有元素加和
sum([1 2 3 4]') %转置为列向量：同为所有元素加和
prod(a) %数组 a 中元素求乘积：按列相乘；该函数若输入行、列向量则所有元素乘积

round(2.6) %舍入：就近舍入即四舍五入
floor(2.6) %向负无穷舍入
ceil(2.3) %向正无穷舍入
fix(2.6) %向 0 舍入
fix(-2.6)

mod(5,2) %余数（模操作）：被除数、除数
rem(5,2) %求余数（大多情况与 mod 相同）
mod(-5,2) %被除数为负数时，结果不一样
rem(-5,2)
```



## 1.18 逻辑运算

### 1. 逻辑型和逻辑运算

逻辑型（logical）数据是用数字 1 和 0 分别表示真（true）或假（false）的状态

### 2. 认识函数

逻辑型 logical

真 true

假 false

判断逻辑型 islogical

全为真 all

是否为真 any

### 3. 说明

Matlab 中一些运算会返回逻辑值，表示一个条件是否被满足

可以使用这些逻辑值来索引数组或执行条件代码

逻辑运算符

符号	功能	函数
&	逻辑与	and
	逻辑或	or
~	逻辑非	not
&&	短路的逻辑与	
	短路的逻辑或	

逻辑真值表

A	B	and A & B	or A   B	xor xor(A,B)	not ~A
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

### 4. 实例演示

```
%1_18
a=[0 1 2 -1]
logical(a) %创建逻辑型: 0 为假返回逻辑 0, 其他非 0 数字都是真返回 1

islogical(a)
islogical(b)

true & false %逻辑与
true | false %逻辑或
```

```

~true
1 & 0
2 & 0
~100

1 && 0 %短路的与：功能与“与”相同
1 || 0
% [1 0] && [0 0] %短路的与、或只适用于标量运算
[1 0] & [0 0] %逻辑与可以用于数组

all([0 1 1]) %判断全为真
all([1 1 1])
any([0 1 1]) %判断任一为真，有一个真则结果为真，全为假结果为假
any([1 1 1])
any([0 0 0])

c=~a %a取非再赋值给c，则c为逻辑数组[1 0 0 0]
a(~mod(a,2)) %a中偶数余数取到0再取非则为真返回1，逻辑真被索引到返回相应偶数元素
true & [] %结果为空的逻辑数组，涉及到空矩阵的任何逻辑运算结果都是空逻辑数组

```

## 1.19 关系运算

### 1. 关系运算

关系运算（relational）使用“小于”，“大于”和“不等于”等运算符定量地比较运算数，比较的结果是一个逻辑数组，在关系为真的位置显示 1

作者：freexyn 整理/注释：韩松岳

### 2. 认识函数

查找元素 find

### 3. 关系运算符

关系运算符		
符号	功能	函数
==	判断相等	eq
~=	判断不等于	ne
>	判断大于	gt
>=	判断大于等于	ge
<	判断小于	lt
<=	判断小于等于	le

## 4. 实例演示

```

%1_19
a=[1 2 3]
a>1 %分别判断元素是否大于1，满足关系返回逻辑值1，否则0
a>a %返回0 0 0
a>=a %每个数都等于自己，返回三个1

```

```

[]>[] %关系运算中只要存在空矩阵，结果都返回空的逻辑数组
[]==[]
[]==2
%[]==[1 2 3] %报错，矩阵维度不一致无法比较（进行关系运算）

b=3+4i %复数关系运算
c=3+5i
c<=b %非等于关系的比较，只比较实部 3
c==b %返回 0。说明：复数关系运算中，==与~=会比较实部和虚部（上述例子中 3 和 3、4 和 5），其他非等于关系的比较，只比较实部

a>1 & a<3 %将两个逻辑结果进行“与”运算，结果仍为逻辑值
tf=a>1 & a<3 %将上述结果赋值给变量 tf
a(tf) %使用逻辑索引，提取矩阵中满足特定条件的元素

index=find(a>1 & a<3) %返回输入变量中条件为真的元素的线性索引
a(index) %使用线性索引，提取矩阵中满足特定条件的元素

a(a>1 & a<3)=10 %使用逻辑索引，修改满足特定条件的元素的值

```

## 1.20 运算符的优先级

### 1. Matlab 中各类、多种运算符组合使用时的优先运算顺序

#### 2. 优先级顺序

可以任意组合使用算术运算符、关系运算符和逻辑运算符等形成的表达式进行运算，Matlab 进行运算处理的顺序取决于每个运算符的优先级。在每个优先级中，运算符具有相同的优先级，并从左到右进行处理。Matlab 运算符的优先级规则从最高到最低排序如下

括号 ()

转置 (.)，幂(^)，复共轭转置 (')，矩阵的幂(^)

一元减的幂(.^-)，一元加的幂(.^+)，逻辑非的幂(.^~) 一元减的矩阵的幂(^-)，一元加的矩阵的幂(^+)，逻辑非的矩阵的幂 (^~)。

一元加(+),一元减(-),逻辑非(~)

乘法(\*),右除(/),左除(\),矩阵的乘法(\*),矩阵的右除(/),矩阵的左除 (\)

加法 (+), 减法(-)

冒号(:)

关系运算符(<),(<=),(>),(>=),(==),(~=)

逻辑与(&)

逻辑或(||)

短路逻辑与(&&)

短路逻辑或 (||)

### 3. 实例演示

```

%1_20
1>=1+1 %四则运算高于关系运算符
(1>=1)+1
1:2+3 %四则运算高于冒号运算符
(1:2)+3
1|1&0 %逻辑与高于逻辑或
(1|1)&0

```

## 1.21 兼容性

### 1. 基本运算中数组大小的兼容性（2016b）

#### 2. 说明

这里兼容性是指，两个大小不同的数组是否能够进行运算

两个完全相同大小的数组可以运算

其中之一是标量的两个大小不同的数组可以运算

一个是行向量，一个是列向量，可以运算

一个是矩阵，一个是具有相同行数的列向量，可以运算

一个是矩阵，一个是具有相同列数的行向量，可以运算

一个是矩阵，一个是具有相同行数和列数的三维数组，可以运算

这里的兼容性运算主要指数组的四则运算

#### 3. 实例演示

```
%1_21
a=[1 2;3 4]
b=[1 2]
c=[3;4]
a+a
a+2 %矩阵与标量运算：将标量扩展为前面矩阵的兼容性大小，再遵循数组四则运算
a.*2
a*2
b+c %先扩展为兼容性大小：b 复制行、c 复制列，再运算
a+b
a+c
a.*b
a.*c

% a*b %无法运算，不满足矩阵乘法
a*c %可以运算，但并非兼容性运算，而是满足矩阵乘法

d=cat(3,a,a) %参数 3 是在 3 维方向连接两个矩阵 a 和 a
a+d %不同维度的数组也可兼容性运算
a.*d %可兼容性运算

bsxfun(@plus,a,d) %低版本兼容性运算函数：第一输入参数表示运算符，后面输入参数表示运算数
```

## 第四章 数据类型

### 1.22 整数型

数值型（numeric type），包括整型和浮点型

#### 1. 整数类型的创建方法、取值范围和运算应用

#### 2. 认识函数

8 位整数 int8

无符号 8 位整数 uint8（其他见下表）

取值范围 intmax/intmin

测试类型 class

#### 3. 概念

Matlab 支持整数数据的 1 个，2 个，4 个和 8 个字节的存储，再分为有符号和无符号，总共 8 个整型类别。

整型类型		
整型类别	取值范围	函数
有符号的 8 位整数	$-2^7$ 至 $2^7 - 1$	int8
有符号的 16 位整数	$-2^{15}$ 至 $2^{15} - 1$	int16
有符号的 32 位整数	$-2^{31}$ 至 $2^{31} - 1$	int32
有符号的 64 位整数	$-2^{63}$ 至 $2^{63} - 1$	int64
无符号的 8 位整数	0 到 $2^8 - 1$	uint8
无符号的 16 位整数	0 至 $2^{16} - 1$	uint16
无符号的 32 位整数	0 至 $2^{32} - 1$	uint32
无符号的 64 位整数	0 到 $2^{64} - 1$	uint64

#### 4. 实例演示

```
%1_22
int8(10) %有符号整型
int8(10.5) %只能存储整数、四舍五入
intmax('int8') %获取有符号整型最大值 127
intmin('int8') %获取有符号整型最小值-128
intmax('uint8')
intmin('uint8')

int8(130) %130 超出最大值范围，则返回最大值 127
int8(500) %返回 127
int8(-130) %-128

int8(1)*int8(5) %整型运算：结果为整型 5
int8([1 2 3])*int8(5) %整型数组，遵循兼容性运算
a=int8([1 2 3])*2.3 %整型数组*双精度浮点型，四舍五入存储整数
class(a) %测试 a 类型，int8
```

## 1.23 浮点型

### 1. 浮点型的创建和转换方法、取值范围、运算和精度问题

作者: freexyn 整理/注释: 韩松岳

### 2. 认识函数

双精度 double

单精度 single

判断浮点型 isfloat

取值范围 realmax/realmin

精度 eps

### 3. 说明

浮点型分为单精度浮点型和双精度浮点型

IEEE 浮点数算术标准 (IEEE 754) 是 IEEE 二进位浮点数算术标准 (IEEE Standard for Floating-Point Arithmetic) 的标准编号, 等同于国际标准 ISO/IEC/IEEE 60559。该标准由美国电气电子工程师学会 (IEEE) 计算机学会旗下的微处理器标准委员会 (Microprocessor Standards Committee, MSC) 发布, 是最广泛使用的浮点数运算标准, 为许多 CPU 与浮点运算器所采用。

知识扩展: 这个标准定义了表示浮点数的格式(包括负零-0)与反常值(denormal number), 一些特殊数值(无穷与非数值(NaN)), 以及这些数值的“浮点数运算符”, 它也指明了四种数值修约规则和五种例外状况(包括例外发生的时机与处理方式)。

任何存储为 double 格式的值都需要 64 位。

double 类型

位	用法
63	符号 (0=正数, 1=负数)
62 至 52	指数
51 至 0	小数位 (分数位)

任何存储为 single 格式的值都需要 32 位。

single 类型

位	用法
31	符号 (0=正数, 1=负数)
30 至 23	指数
22 至 0	小数位 (分数位)

### 4. 浮点型的精度问题

Matlab 中几乎所有的运算默认都是以符合 IEEE 754 标准的双精度算法执行的, 由于计算机只能以有限的精度表示数字 (双精度要求 52 个尾数位), 对于数值运算, 这种浮点型表示值与其真实值存在微小的差异

所以,如果浮点算术运算的结果不如预期的那样精确,甚至有时会产生数学上非直觉的结果,则很可能是由于计算机硬件的限制造成的,硬件可能没有足够的位来表示结果的完全准确性,因此截断了结果值的部分精度(例如 32 位的电脑)

但这并不是 Matlab 中的错误,运用 IEEE 754 标准所执行的所有计算都受到影响,其中包括用 C 或 FORTRAN 等

## 5. 实例演示

```
%1_23
%freexyn
double(1) %创建双精度浮点型 1
a=1 %默认双精度浮点型,再赋值给 a

isfloat(a) %判断是否浮点型
class(a) %测试变量 a 的数据类型
whos a %测试变量 a 的数据类型等信息

single(1) %单精度 1
realmax('double') %双精度浮点数的最大取值
[-realmax,-realmin,realmin,realmax] %完整的双精度浮点型取值范围(-0 和 0 之间取不到任何值)
[-realmax('single'),-realmin('single'),realmin('single'),realmax('single')] %单精度浮点型取值范围,上式未指定则默认取双精度浮点型范围

%% 浮点型与整型转换方法
c=int8(1) %整型 1
c1=double(c) %整型 c 转化为双精度,数据类型创建方法也是转化方法
c2=single(c) %转化为单精度
int8(c1) %双精度转化为整型
whos c c1 c2 %查看数据类型:双精度 8 字节精度最高

%% 精度问题相关
sin(pi) %会保留部分精度的小数再进行计算,因此结果会有误差(计算机有限位数存储有限精度)
sin(sym(pi)) %sym 将 pi 转化为符号型再求 sin 值,结果是 0 更精确
eps %默认求 1 的精度,求得的值为 1 附近的误差值
1
1+eps %结果 1,因为硬件所限导致的精度问题,无法表达 eps 更小的值
eps(10) %求 10 附近的精度
eps(100) %求 100 附近的精度
eps(single(100)) %单精度 100 附近的精度

(4/3-1)*3-1 %4/3 并非完整精确的分数,而是有限存储空间内的近似值,因此结果并非 0
(sym(4/3)-1)*3-1 %返回 0,引入符号型获得精确 4/3 值
```

## 1.24 Inf 和 NaN

### 1. 介绍 inf 和 nan 的含义和用法

### 2. 认识命令

无穷大 inf

非值 nan

判读无穷大 isinf

判断非值 isnan

### 3. 说明

用特殊值“inf”表示无穷大，比如零除和溢出等这样的运算会产生无穷大，这导致结果太大而不能用传统的浮点值表示

用特殊值“NaN”表示既不是实数也不是复数的“非数字”的值  
Matlab 里，inf 与 inf 相等判断为真，nan 与 nan 相等判断为假

### 4. 实例演示

```
%1_24
realmax %浮点型最大取值
realmax+1e30 %最大取值再加上 10^30 并未返回无穷大值
realmax+1e300 %加上 10^300 返回无穷大值。说明：10^30 相对于 realmax 较小，因此忽略；
10^300 相对于 realmax 不能忽略。

a=inf
1/0 %无穷大
1e309 %超过存储范围，返回无穷大
isinf(exp(1000)) %返回逻辑 1，判断无穷大
inf==inf %1 真
inf+10 %返回 Inf
inf-1e308 %返回 Inf
inf+inf %返回 Inf。
inf/inf %nan。无穷大参与的运算，结果通常为无穷大，仅本情况例外

0/0 %nan
a=nan %nan
isnan(a) %nan
nan==nan %假
nan>0
nan<0
nan~=nan %涉及 nan 的关系运算结果通常为假
```

## 1.25 显示格式

### 1. 设置命令行窗口数值的显示格式

### 2. 认识函数

format

### 3. 说明

数值格式仅影响数字显示在命令行窗口输出中的方式  
而不是 Matlab 计算或保存它们的方式  
指定的格式仅应用于当前 Matlab 会话

### 4. 实例演示

```
%1_25
a=[1/3 1.23e-5] %默认显示 4 位小数
format short %短格式默认 5 位
a
format long %长格式默认 15 位
```



```

a
format short e %科学计数法
a
format rational %有理数
a
format hex %16 进制
a
format shortG %短格式基础上紧凑格式
a

get(0,'format') %获取当前格式, 0 为句柄值, 表示当前会话
set(0,'format','short') %set 设置格式为 short
a
format short
format %short 为默认格式, 因此可不打出来

```

## 1.26 字符型

在 Matlab 中有两种表示文本的方法: 字符型和字符串型

### 1. 字符型 (Characters) 数组的创建、连接、转换和运算

#### 2. 认识函数

字符型 "

判断字符型 ischar

转成字符 char

转成字符 num2str

转成数值 str2num

交集 intersect

并集 union

#### 3. 说明

字符型一般用来存储和处理文本数据

字符数组是一个字符序列

字符向量把字符存储为 1 乘 n 的向量, 是常用形式

#### 4. 实例演示

```

%1_26
a='123' %创建 3 个字符数组
b='freexyn'
c='自由未知数'
size(c) %查看行列数
ischar(a) %判断字符型
'I'm fine' %字符型内单引号处理: 改用双引号('')
c(1:3) %索引字符向量的元素
c(6)='.' %通过索引修改字符向量的元素

[b,c] %字符连接: 中括号水平连接
strcat(b,c) %水平连接函数
strvcat(b,c) %垂直连接函数: 若列数不同自动使用空格补齐

```

```

char('d') %字符 d 转换（创建）为字符型
'd' %通常使用该方法
char(100) %双精度转换为字符型：基于 Unicode 码
char([97 98 99 100]) %转换 1 行 4 列的数值向量，结果 a b c d
double('a') %将字符型 a 转换为双精度数值型
char('12','0') %char 函数输入多个参数时，会将其纵向连接为多行，并相应转换为字符型
char('12',100)
char('12',100,'123')
char(['12',100],'123') %参数用中括号括起来表示在同一行
char(100) %转换为字符 d：根据 unicode 码
num2str(100) %'100'，把 100 转化为'100'字符型
str2num('100')
r=80
disp(['r:',num2str(r)]) %打印（显示）信息：r:80，数值跟随变量自动变化

%% 字符型数字运算
a='d'
b='100' %1 行 3 列字符向量：1 0 0
a+a %字符型加法：首先转换成数值型，再运算，返回结果 200
a+b %1 0 0 字符向量按 ASCII 码转换为数值型分别为 49 48 48，再分别与 d 转换的 100 相加，
返回 149 148 148
char(49) %数值型转化为字符型：'1'
str2num('100')+str2num('100')

%% 集合运算
a='123' %字符向量
b='1245'
union(a,b) %并集
intersect(a,b) %交集

```

## 1.27 特殊字符

### 1. 特殊字符识别和处理

### 2. 认识函数

判断字母 isletter

判断空格 isspace

判断特定字符 isstrprop

空字符 blanks

字符调整 strjust

删除空格 deblank

删除空格 strtrim

### 3. 实例演示

```

%1_27
a='abc 123'
isletter(a) %判断字母
isspace(a) %判断空格
find(isspace(a)) %对空格进行线性索引
a(find(isspace(a)))=[] %将 a 字符中的空格删除

%% 判断特定字符
b='12 ab_AB, '

```

```

isstrprop(b,'alpha') %判断特定字符 字母
isstrprop(b,'lower') %判断特定字符 小写字母
isstrprop(b,'upper') %判断特定字符 大写字母
isstrprop(b,'digit') %判断特定字符 数字
isstrprop(b,'punct') %判断特定字符 标点
isstrprop(b,'wspace') %判断特定字符 空格

%% 空字符的处理
c=blanks(7) %创建空字符函数，创建 7 个空字符
c(3:5)='aaa'
strjust(c,'left') %strjust 字符调整，将 c 中空字符调至左边
deblank(c) %删除尾随空字符
strtrim(c) %删除前后空字符，但不删除中间的空字符

```

## 1.28 混合连接的类型

### 1. 多个类型的数据混合连接后的数据类型

### 2. 组合类型列表

组合类型列表

类型	字符型	整型	单精度	双精度	逻辑型
字符型	字符型	字符型	字符型	字符型	无效
整型	字符型	整型	整型	整型	整型
单精度	字符型	整型	单精度	单精度	单精度
双精度	字符型	整型	单精度	双精度	双精度
逻辑型	无效	整型	单精度	双精度	逻辑型

### 3. 说明

总体上，除逻辑型外，连接后的类型遵循向下转换的原则  
逻辑型除了与自身连接是逻辑型，与其他连接转换成其他型

### 4. 实例演示

```

%1 28
[100 single(100)]
[100 single(100) int8(100)]
[100 single(100) int8(100) 'd']
[100 true] %双精度与逻辑型连接

%% 整型内部连接
[int8(100) int16(100)]
[int16(100) int8(100)] %遵循最左侧整型类型的原则
[int8(-100) uint8(100)]
[uint8(100) int8(-100)] %右边由有符号变为无符号，仅取到最小值 0

%% 混合连接
[true pi int32(10) single(1.23) uint8(345)] %pi 为双精度，int32 为整型，混合连接遵循最左侧整型数据类型原则，后面超出 int32 的数据会相应变化
[true pi single(1.23) uint8(345)] %结果取 uint8

```

## 1.29 混合运算的类型

### 1. 多个类型的数据混合运算后的数据类型

作者: freexyn 整理/注释: 韩松岳

### 2. 混合运算类型列表

混合运算类型列表		
运算数类型	运算数类型	结果类型
double	int(uint)	int(uint)
	single	single
	double	double
	char	double
	logical	double
single	single	single
	char	single
	logical	single

### 3. 说明

总体上, 除逻辑型和字符型, 混合运算类型遵循向下转换的原则  
逻辑型除了与自身运算是逻辑型, 与其他运算转换成其他型

### 4. 实例演示

```
%1_29
%freexyn
%% 混合运算
10+single(10) %双+单=单
10+int8(10) %双+整=整
%[10 10]+int8(10) %双精度数组无法与整型相加, 报错

10+'d' %双精度+字符型=双精度
'd'+ 'd' %字符型+字符型, 先转换为双精度再运算
[10, 'd'] %复习上节, 连接: [双精度 字符型] =字符型
10+true %双精度+logical=双精度

%% 整型内部运算
% int8(10)+int16(10) %不同类型整型无法相加运算
[int8(10),int16(10)].*[int8(10),int8(10)] %整型数组乘法
% [int8(10),int16(10)]*[int8(10);int8(10)] %矩阵乘法, 前面列数等于后面行数, 报错: 矩阵乘法不完全支持整数类, 仅支持参数之一为标量的情况
```

## 1.30 字符串型

### 1. 字符串（Strings）数组的创建、比较、索引和运算

### 2. 认识函数

创建 strings

判断 isstring

字符串长度 strlen

### 3. 说明

字符串是一个字符序列，常见的，存储一个 1 乘 n 的字符向量

字符串数组是由多个字符串作为元素组成的数组

从 Matlab2016b 开始，可以使用字符串类型

从 2017a 开始，可以使用双引号创建字符串

### 4. 实例演示

```
%1_30 字符串型创建、比较、索引和运算
%% 字符串和字符串数组的创建
s="自由未知数" %字符串
isstring(s) %判断字符串型
ischar(s) %判断字符串型
s=["123","abcd";"自由未知数","%$%#"] %字符串数组
strings(2,3) %创建字符串数组函数，2行3列空字符串数组
strings(0,3) %空的0*3数组

%% 字符串型的比较
c='' %空字符串向量
s="" %1*1字符串
size(c)
size(s)
isempty(c)
isempty(s)
c='freexyn' %字符串向量（数组）
s="freexyn" %1*1字符串
size(c)
size(s)

%% 字符串型的连接
['123','ab'] %字符串向量连接，结果为更大的字符串向量
["123","ab"] %字符串连接，结果为字符串数组
%['123';'ab'] %行数列数不同，报错
["123";"ab"] %可行

%% 补充
length(c) %字符串向量长度
length(s) %字符串数组长度
strlen(s) %字符串元素的长度

%% 字符串型索引
s=["123","abcd";"自由未知数","%$%#"]
s(1,2) %常规组合索引
s{1,2} %索引字符串的元素内容，返回字符串向量
```

```
s{1,2}(1:2) %索引元素内容后，用二级索引获取子段

%% 字符串型运算
s+"x" %视为矩阵添加标量，添加到每个元素中
s+'x' %结果同上
```

## 1.31 缺失字符串

### 1. 字符串的转换、缺失字符串的创建和应用

作者: freexyn 整理/注释: 韩松岳

### 2. 认识函数

转换 string

缺失值 missing

判断缺失 ismissing

### 3. 说明

缺失值表示数据中不可靠或不可用的点

不同类型中缺失值的表达不同，数值型用 NaN，字符串里用<missing>

missing 从 Matlab2017a 开始推出

字符串数组扩展时，缺失元素用<missing>自动填充

### 4. 实例演示

```
%1_31
%% 字符串的转换
string(100) %双精度转换为字符串型
char(100) %双精度转换为字符型，依据 Unicode 码操作
string('100') %字符型转换为字符串型
char("100") %字符串型转化为字符型

%% 缺失值
missing %通用的缺失值函数，适用于任何类型
string(missing) %将missing转化为字符串类型的缺失值
ismissing(["", "abc", string(missing)]) %判断数组3个元素是否为缺失值
string(missing)==string(missing) %缺失值与任何数值比较都为假，除特例比较不相等
string(missing)=="x"
string(missing)~=string(missing) %特例

%% 缺失值的运算
string(missing)+string(missing) %缺失值的任何运算结果都是缺失值
string(missing)+"x"

%% 字符串数组的扩展
s="x" %创建字符串数组
s(2,3)="x" %扩展为2行3列的数组，并将第2行第3列元素赋值为''x''
```

## 1.32 格式化文本

### 1. 简单介绍格式化文本的用法

### 2. 认识函数

sprintf

### 3. 说明

格式化文本，是具有特定显示格式的文本形式，包括字段宽度、显示精度、特殊标志和辅助符号等

普通文本和数值需要按照特定格式显示和输出时，会用到

示例，`sprintf('%f\n|.2f\n|.2f',pi*ones(1,3))`

### 4. 实例演示

```
%1_32
sprintf('%f\n|.2f\n|.2f',pi*ones(1,3)) %第一个参数为格式设置，第二个参数为
对应的数值，此处，3 组格式设置对应 3 个值
%每行起始为竖线|，%是必须的字段，f 代表浮点型，将浮点型值转化为文本，\n 为转义符代表回车
%.2 代表小数点后保留 2 位精度
%8.2 总长度 8，小数点后保留 2 位，长度不够时按实际长度
```

## 1.33 字符型与数值型的转换

### 1. 字符型和数值型之间的相互转换

### 2. 认识函数

字符转数值 `uintN` %依据 Unicode 码转换

数值转字符 `char` %依据 Unicode 码转换，ASCII 码与 Unicode 码并非相同概念，ASCII 较基础，应用范围小，主要表示键盘上字母符号等，Unicode 码包含 ASCII 码，囊括多国语言字符。

数值转字符串 `string`

字符转数值 `str2num`、`str2double`

数值转字符 `num2str`、`int2str`

十和二进制互换 `bin2dec`、`dec2bin`

十和十六进制互换 `hex2dec`、`dec2hex`

十和其他进制互换 `base2dec`、`dec2base` %十进制必然是数值型，其他任意进制都使用字符型表达

### 3. 实例演示

```
%1_33
%% 字符与数值转换
uint8('ab') %字符'ab'转换为 8 位整型的整数
uint16('ab') %字符'ab'转换为 16 位整型的整数
uint8('是') %ASCII 无汉字，Unicode 包含
uint16('是')
```

```

2^16 %16 位最大值
uint32('是')
%相反功能
char([97 98]) %数值转换为字符型
string([97 98]) %转换成字符串数组

%% 字符型转换为数值型
str2num('100') %字符向量转换为数值型
str2double('100') %同上
str2num("100") %字符串转换为数值型
str2double("100") %同上
%str2num(['100','100']) %该函数无法将字符串数组转换为数值型
str2double(['100','100']) %字符串数组转换为数值型
%相反功能: 数值转换为字符
num2str(100) %双精度数值转换为字符
num2str(1.2345,3) %第 2 个参数设置前面转换后保留的精度
num2str(100,'%5.2f') %格式化文本, 单引号中进行设置
int2str(1.23) %浮点型转为字符

%% 不同进制之间转换
bin2dec('1000') %二进制转换为十进制
dec2bin(8) %十进制转换为二进制
hex2dec('A') %十六进制的 A 转换为十进制
dec2hex(10) %十进制转换为十六进制
base2dec('10',2) %任意进制转换为十进制: '10'为转换数值, 2 为二进制
base2dec('10',8) %八进制的 10 转化为十进制
base2dec('10',20)
dec2base(2,2) %十进制的 2 转化为二进制
dec2base(8,8)
dec2base(9,8) %十进制的 9 转化为八进制

```

## 1.34 元胞数组

### 1. 元胞数组的创建、索引和转换方法

### 2. 认识函数

创建方法: `a={}` 和 `cell` (与 Python 中字典结构相似)

判断 `iscell`

索引 `()` 和 `{}`

转换 `cell2mat`、`mat2cell`

### 3. 概念

元胞数组 (cell array) 是一种具有容器特性的数据类型, 每个元素可以包含任何类型的数据

### 4. 说明

元胞数组创建和扩展时默认填充元素是空矩阵 `[]`

元胞数组不需要完全连续的内存, 但每个元素需要连续的内存  
对大型的元胞数组, 增加元素数量可能导致 Out of Memory 错误  
因此, 必要时, 元胞数组需要初始化和预分配内存



## 5. 实例演示

```
%1_34
%% 元胞数组创建、2 种预分配内存方法
a={}
b={1,2,magic(3)
    'a',["a","b"],[]} %分别双精度标量、3 阶魔方矩阵、字符、字符串数组、空矩阵。2 行
用回车分割，或用分号分割
cell(2,3) %创建 2 行 3 列元胞数组
c=cell(100,100) %对大型元胞数组，预分配内存
d={} %初始化后（也可不初始化，因 Matlab 使用变量不需提前声明，因此也可直接赋值）
d{100,100}=[] %用数组扩展的方式，将第 100 行第 100 列元素设置为空，其他未指定元素默认
用空矩阵填充，完成预分配内存
iscell(a) %判断元胞数组

%% 元胞数组索引方法（可类比字符串数组索引）
b(1,3) %索引元胞数组的元素
b{1,3} %索引元胞数组的元素的内容
b{1,1:3} %索引 1 行 1-3 列元素，未指定输出变量，默认 ans
[v1,v2,v3]=b{1,1:3} %索引 3 个元素值，并赋值给 3 个变量
v=b{1,1:3} %当仅指定 1 个输出变量，则只返回第 1 个值
b{1,3}(1:2) %二级索引

%% 元胞数组数据连接
%[b{1,1:3}] %维度不同无法连接
[b{1,1:2}] %2 个索引到的元素，连接为 1*2 矩阵

%% 元胞数组转换：每个元胞元素类型相同且大小相同才可转换
%cell2mat(b) %b 没有明确结果类型因此报错
m=cell2mat({1 2}) %数值型可以转换为矩阵
n=cell2mat({'a','b'})
%cell2mat({'a',"b"}) %该转换函数不支持字符串数组
mat2cell(m,1) %第 2 个参数 1 表示 1 行 1 列元胞数组，将整个矩阵看做元胞数组元素
mat2cell(m,1,[1 1]) %第 3 个参数指定了列数为 2 列，每列个数分别为 1
mat2cell(n,1)
mat2cell(n,1,[1 1])
```

## 1.35 元胞数组的修改

### 1. 元胞数组的修改、添加、删除和连接

### 2. 认识函数

连接[]和{}

### 3. 说明

元胞数组的子数组或元素也是元胞型的，其元素内容是本身类型

## 4. 实例演示

```
%1_35
%% 元胞数组修改
b={1,2,magic(3)}
```

```
'a',["a","b"],[]} %理解: 元胞数组是大箱子, 用大括号表示, 内部的元素是盒子, 盒子
也是元胞型, 盒子与箱子遵循 Matlab 预设规则, 能修改的就是打开盒子往里面装内容, 内容可以是
任意大小任意格式的内容。
% b(1,1)=100 %元胞数组数据修改, 小括号索引出第 1 行第 1 列元胞型, 100 双精度型无法赋值
(小括号索引到盒子), 报错
b(1,1)={100} %赋值也应包装成盒子(元胞型)再赋值
b{1,1}=100 % (大括号索引到盒子的内容) 所以使用任意数值赋值即可

%% 元胞数组添加、删除
b(3,4)={8} %b 本身是 2 行 3 列, 如此扩展为 3 行 4 列, 第 3 行第 4 列为{8}其余扩展元素默认
空矩阵填充
% b(1,1)=[] %报错, 删除 1 个元素无法保持矩阵矩形
b(end,:)=[] %最后一行删除, 可行
b{1,1}=[] %大括号索引到盒子里的内容, 并置为空(删除), 盒子仍存在

%% 元胞数组连接
%中括号连接: 把元胞数组打开重排
[{1 2 3},{ 'a' }] %水平连接
% [{1 2 3};{ 'a' }] %列数不同, 报错
[{1 2 3};{ 'a', [], []}] %列数相同
%大括号连接: 元胞数组视为整体重排(可理解为元胞数组的嵌套, 元胞数组内可容纳任意类型, 包
括元胞型自身)
{{1 2 3},{ 'a' }} %水平连接
{{1 2 3};{ 'a' }} %纵向排列
{{1 2 3},{ 'a' }, {}, [], string(missing), uint16(10)} %只要占用元胞数组内的盒子,
都会显示
```

## 1.36 结构数组

### 1. 结构数组的创建、索引和预分配内存

### 2. 认识函数

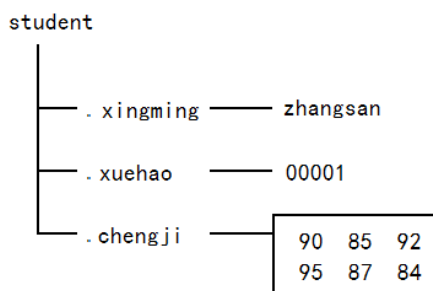
创建 struct

判断 isstruct

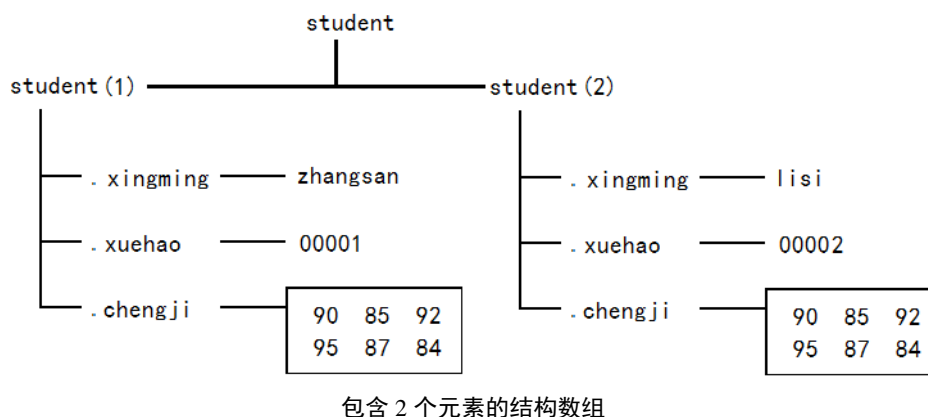
运算符 .

### 3. 概念

结构(structure array)是一种具有容器特性的数据类型, 它使用称为字段的数据容器对相关数据进行分组, 每个字段可以包含任何类型或大小的数据。(与元胞数组描述类似, 都是容器型数据类型, 组织结构不同)



包含 1 个元素的结构数组



#### 4. 创建方法（2 种方法）

数组名.字段名=字段值，遍历所有字段名赋值

数组名=struct（‘字段名’，‘字段值’...）

#### 5. 说明

所有元素都具有相同数量的字段和相同的字段名称

字段未指定的默认值为[]（默认为空矩阵，与元胞数组相同）

结构数组不需要完全连续的内存，但每个字段需要连续的内存

对于大型的结构数组，增加字段的数量或字段中数据的数量可能会导致 Out of Memory 错误

因此，必要时，结构数组需要初始化和预分配内存

#### 6. 实例演示

```
%1 36
%% 单元素结构数组创建
%方法 1: (分别列出字段信息，元素名与字段信息用圆点(.)分隔)
student.xingming='zhangsan'; %student 是结构数组名，后跟字段名，等号右侧为字段值
student.xuehao='00001';
student.chengji=[1 2 3;4 5 6]
%方法 2: (struct 函数创建，分别列出每条字段信息，最后赋值给变量名)
stu=struct('xingming','zhangsan','xuehao','00001','chengji',[1 2 3;4 5 6])

%单元素结构数组的操作
isstruct(stu) %判断结构数组
stu.chengji %字段索引
stu.chengji(1:3)%2 级索引，与元胞数组类似

%% 多元素结构数组
%方法 1: (在前面基础上添加)
student(2).xingming='lisi';
student(2).xuehao='00002';
student(2).chengji=[7 8 9;1 2 3]
%方法 2:
stu(2)=struct('xingming','zhangsan','xuehao','00001','chengji',[1 2 3;4 5 6])
%多元素结构数组操作
student.chengji %若不指定哪个元素，则显示所有该字段名的值
student(2).chengji %指定第 2 个元素，显示该元素的该字段值
```

```
% student.chengji(1:3) %2 个及以上元素的结构数组不能直接使用二级索引，因未指定元素，
无法确定索引的结果
student(2).chengji(1:3)
%预分配内存
st(100,100)=struct('a',[]) %边界思想，设置结构数组的边界元素为字段名为'a'且字段值
为[]的结构数组，其他元素值默认填充为空[]
st.a %索引 st 的字段'a'，返回 100 个空矩阵
```

## 1.37 结构数组的处理

### 1. 结构数组的连接、嵌套、引用变量值和访问字段值

作者: freexyn 整理/注释: 韩松岳

#### 2. 说明

结构数组必须具有相同的字段名才能连接，元素数目可以不同  
为某元素添加字段，其他所有元素也具有了该字段，默认值为[]

#### 3. 实例演示

```
%1_37
%freexyn
%创建 3 个结构数组 student/stu/st
student.xingming='zhangsan';
student.xuehao='00001';
student.chengji=[1 2 3;4 5 6];
stu=struct('xingming','zhangsan','xuehao','00001','chengji',[1 2 3;4 5 6]);
student(2).xingming='lisi';
student(2).xuehao='00002';
student(2).chengji=[7 8 9;1 2 3]
stu(2)=struct('xingming','zhangsan','xuehao','00001','chengji',[1 2 3;4 5 6])
st(100,100)=struct('a',[])

%% 结构数组的连接
[student,stu] %同为 1*2 结构数组，并且字段名相同，结果为 1*4 结构数组
% [student,st] %维度不同且字段名不同，报错

%% 嵌套
student(1).stu=stu %嵌套，理解：创建字段名 stu，并把原结构数组 stu 作为字段值赋值给
该字段名。给结构数组 student 的第 1 个元素 student(1)添加 stu 字段，其他所有元素（如
student(2)）也会具有该字段
student(1).stu %索引，查看第 1 个元素的 stu 字段名里的字段值（字段内容），结果为原 stu
数组
student(2).stu %索引，结果是空矩阵，因为未指定该字段的值
%多级索引打开内部嵌套的字段值（理解：结构数组的索引，用圆点运算符(.)层层打开字段值，访
问所需内容）
student(1).stu(1) % student 第一个元素的 stu 字段的第一个元素的值
student(1).stu(1).chengji %进一步获取该元素的 chengji 字段的值
student(1).stu(1).chengji(1:3) %进一步获取成绩值的第 1-3 元素

%% 预留字段（添加新字段）
yuliuziduan='nianling' %通过预留变量名的形式，给元素添加字段，字段名的赋值需字符型。
```

```

stu(1).(yuliuziduan)='nan' %给包含多个元素的结构数组通过小括号+预留变量名的方式添加字段时，要指定具体元素，等号右侧赋字段值
%用途：当后面需要修改字段名时，不需再每处修改，而只修改预留变量所赋的值即可，树状图思想，预留变量作为中间值
stu(1).(yuliuziduan) %小括号引用该预留变量代表的字段名，并索引该字段名的字段值，即nan

%% 获取结构数组字段值
student.chengji %索引 student 所有元素的 chengji 字段的值
v=student.chengji %当只有一个输出参数 v 时，只返回第 1 个值
[v1,v2]=student.chengji %若返回所有字段值，则指定相同个数的输出参数
v=[student(1).chengji,student(2).chengji] %将多字段值存储在同一矩阵中，则先将字段值提取再矩阵连接。要求字段值数据类型相同，横纵连接符合矩阵连接的维度要求

```

## 1.38 表

### 1. 表类型数据的创建、索引和自身属性的用法

xingming	xuehao	chengji	
{ 'zhangsan' }	{ '1001' }	89	95
{ 'lisi' }	{ '1002' }	90	87
{ 'wangwu' }	{ '1003' }	88	84

3\*3 table 数组

### 2. 认识函数

创建 table

判断 istable

属性.Properties

### 3. 概念

表（table）具有容器特性的数据类型，可以方便的存储混合类型的数据，可以使用数字或命名索引访问数据以及元数据（例如变量名称，行名称，描述和变量单位等）

### 4. 说明

表由行和列组成

通常，表的列代表不同的变量，行代表不同的变量值

不同变量须具有相同数量的变量值，即行数须相同，否则不完整

表的索引方法有两种，下标索引和字段索引

### 5. 实例演示

```

%1_38
%% 表格创建
xingming={'zhangsan';'lisi';'wangwu'} %用元胞数组的形式，创建表的每一列
xuehao={'1001';'1002';'1003'}
chengji=[89 95;90 87;88 84]
t=table(xingming,xuehao,chengji) %表格的创建：table 函数+小括号输入变量

```

```

istable(t) %判断

%% 表格索引
t(1:2,2:end) %下标索引: 1-2 行, 2-最后一列
t.xingming %字段索引: 索引该字段(列名称)所指向的整列数据
t.chengji(2,1) %组合索引: 成绩字段里第 2 行第 1 列
t.age=[20;19;21] %使用字段索引为表格添加变量(字段)名称, 并赋值。默认添加至最后一列
size(t) %表格维度

%% 表格属性的应用
t.Properties %表格的属性数据, 是结构数组
t.Properties.VariableNames %二级索引获取属性里的变量名称
t.Properties.RowNames={'1','2','3'} %给行名称赋值。注: 赋值应与属性值数值类型相同(属性是元胞数组, 则赋值也以元胞数组形式)
%用途: 赋予行名称后, 可以作为索引使用
t('1','xuehao') %索引 1 行 xuehao 列的数据, 形式类似于下标索引, 下标即矩阵下标 ij, 而该索引以行与列的名称索引。
t(1,2) %下标索引, 1 行 2 列
t({'1','2'},{'xingming','age'})

```

## 1.39 表的数据处理

### 1. 表的编辑(排序查找提取删除)、计算、与结构数组转换

### 2. 认识函数

统计 summary

与结构数组转换 table2struct、struct2table

与元胞数组转换 table2cell、cell2table

### 3. 实例演示

```

%1_39
xingming={'zhangsan';'lisi';'wangwu'};
xuehao={'1001';'1002';'1003'};
chengji=[89 95;90 87;88 84];
t=table(xingming,xuehao,chengji)

%% 表格的统计
summary(t) %无法统计字符型元胞数组, 可以对 chengji 双精度型进行统计处理(按列统计)
mean(t.chengji) %字段索引再 mean 函数(默认维度按列即字段名求均值)
mean(t.chengji,2) %2 表示第 2 维度即按行求均值
t.pingjunzhi=mean(t.chengji,2) %使用字段索引并赋值

%% 表的排序、查找
sortrows(t,'xingming') %按行排序, 表格 t 按字段'xingming'按字母升序
t(:,[1 2 4 3]) %用索引排序: 任意行, 列将 3 和 4 列互换
% t.pingjunzhi=[] %删除, 用字段索引赋空值删除该行
tf=t.pingjunzhi>90 %查找平均值大于 90 的学生信息, 返回逻辑值;
t(tf,:) %使用 tf 作为逻辑索引, 索引表格中符合条件的学生的信息

%% 数据转换
t
s=table2struct(t) %表格转换为结构数组后, 每一行都变为 1 个元素, 因此是 3 行 1 列的结构数组

```

```
s(1) %查看第1个元素的值
struct2table(s) %反向转换
c=table2cell(t) %表格转换为元胞数组后，变量名称消失，变量值转换为元胞数组，其中多列
的双精度值会自动拆分为多个单列表示
cell2table(c) %反向转换
```

## 1.40 表的读入写出

### 1. 表与外部文件的读入和写出

### 2. 认识函数

读入 readtable

写出 writetable

### 3. 实例演示

```
%1_40
xingming={'zhangsan';'lisi';'wangwu'};
xuehao={'1001';'1002';'1003'};
chengji=[89 95;90 87;88 84];
t=table(xingming,xuehao,chengji)

writetable(t,'student.txt') %写出数据到txt文件
stu=readtable('student.txt') %从txt文件读入数据
stu.chengji=[stu.chengji_1,stu.chengji_2] %上面写出再读入后，多列数据自动拆分，
把拆分后的数据恢复成原先数据
stu.chengji_1=[]
stu.chengji_2=[]
writetable(t,'student.xls') %写出数据到Excel文件
stu=readtable('student.xls') %从Excel文件读入数据
```

## 1.41 日期时间型

### 1. 日期时间型的概念及其简单应用

作者: freexyn 整理/注释: 韩松岳

### 2. 认识函数

日期时间 datetime

持续时间 duration

### 3. 概念

日期时间型 (Dates and Time) 数据具有灵活的显示格式和高达毫微秒的精度，并且可以处理时区、夏令时和平闰年等特殊因素

日期时间型数据有以下三种表示方式

**Datetime** 型，表示日期时间点，是存储日期和时间数据的主要方法，它支持算术运算，排序，比较，绘图和格式化显示

**Duration** 型，表示日期时间的持续长度

**CalendarDuration** 型 (略)

## 4. 实例演示

```
%1_41
%freexyn
datetime(2018,8,8) %创建日期时间型, 输入参数为: 年、月、日
t=datetime(2018,8,8,12,0,0) %输入参数为: 年、月、日、时、分、秒
d=duration(3,2,0) %创建持续时间型, 时长 3 小时 2 分 0 秒
years(1)
days(1)
hours(1)

%% 运算
t2=t+d %时间点和持续时间的运算结果仍为时间点
t3=t-d
t-days(4)
hours(1)+minutes(30)
t2>t %时间点的比较, 时间越晚, 则越大
t3>t
hours(1)>minutes(30) %持续时间的比较, 时间越长, 则越大

%% 显示格式
t %时间点的显示格式设置
datetime(t,'Format','y-MM-dd') %时间点显示格式, 使用 format 属性设置
datetime(t,'Format','y-MM-dd HH:mm:ss eeee') %y M d H m s e 分别代表年、月、
日、时、分、秒、星期
d %持续时间显示格式设置
duration(d,'Format','m') %m 表示分钟, 另外, h 表示小时、s 表示秒

%% 补充
[y m d]=ymd(t) %函数 ymd 获取时间点 t 中的年月日信息赋值给相应变量, 另外, 时分秒 hms
同理
dateshift(t,'start','day',0:2) %时间推移方法获取时间序列, start 表示返回一天的起
点即 0 点, 0:2 推移 2 天
char(t) %日期时间型转换为字符型, 转换后可用字符型规则处理数据
NaN %datetime 型的数据, 表示非时间, 即缺失值。
```

## 1.42 缺失数据的处理

### 1. 各类型缺失数据的创建、判断、替换、移位和处理方法

#### 2. 认识函数

替换 standardizeMissing

替换为 fillmissing

位置'MissingPlacement'

忽略'omitnan'

移除 rmmissing

#### 3. 实例演示

```
%1_42
%% 各类数据缺失值的创建
a=[nan 1 2 3] %数值型缺失值
s=[string(missing) "a" "b"] %字符串型缺失值
```



```

t=[NaN datetime(2018,8,8)] %时间型缺失值
%missing 函数可创建不同数值类型的缺失值
aa=[missing 1 2 3]
ss=[missing "a" "b"]
tt=[missing datetime(2018,8,8)]
isnan(a) %判断数值型
ismissing(a) %判断缺失值
ismissing(s)
ismissing(t)

%% 缺失值的替换
standardizeMissing(a,[2 missing]) %变量中参数替换为缺失值
standardizeMissing(s,["b" missing])
standardizeMissing(t,[datetime(2018,8,8) missing])
fillmissing(a,'constant',0) %变量中缺失值替换成参数,'constant'和0表示把缺失值
替换为常数0
fillmissing(s,'constant',"fill")
fillmissing(t,'constant',datetime(2019,9,9))

%% 缺失值的移位(排序)
sort(a,'MissingPlacement','last') %把变量a中的缺失值移位到最后

%% 缺失值的运算
max(a) %忽略nan求最大值
sin(a) %nan的sin值就是nan
sum(a) %求和返回nan值
sum(a,'omitnan') %忽略nan
sum(rmmissing(a)) %移除a中的缺失值

```

## 1.43 类型识别

### 1. 判断数据的类型和类别

例如数值型（整数、浮点数、实数、无穷数、有限数、nan 等）、字符（串）型、结构数组、元胞数组、表、函数句柄等

作者：freexyn 整理/注释：韩松岳

### 2. 认识函数

变量信息 whos

类型 class

无穷大 isinf

非值 isnan

数值型 isnumeric

实数 isreal

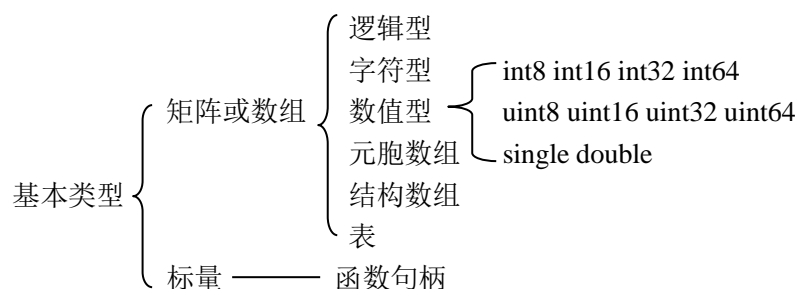
有限值 isfinite

综合判断 isa

字符向量元胞数组 iscellstr

### 3. 简单总结数据类型

Matlab 的基本类型（16 个）



#### 4. 实例演示

```

%1_43
%freexyn
x=1
whos x
class(x)
isnan(x)
isnan(nan)
isinf(x)
isinf(1e309)

isnumeric(x)
isnumeric('a')
isreal(x)
isreal(1+2i)
isfinite(x)
isfinite(1e309)

isa(x,'numeric') %判断数值型
isa(x,'integer') %判断整数
isa(int16(1),'integer') %判断 int16 (1) 是否为整数
isa(x,'int8') %判断 x 是否为 8 位整型
isa(int8(1),'int8')
isa(x,'float') %判断浮点型
isa(x,'double') %判断双精度浮点型
isa(x,'single') %判断单精度浮点型
isa(x,'logical') %判断逻辑型
isa(true,'logical')
isa(x,'char') %判断字符型
isa('a','char')
isa(x,'string') %判断字符串型
isa('a','string')
isa('b','string')
isa(x,'struct') %判断结构数组
isa(x,'table') %判断表数组
isa(x,'cell') %判断元胞数组
isa(x,'function_handle') %判断函数句柄
isa(x,'datetime') %判断日期时间型
isdatetime(x)
iscellstr({'11'}) %判断元胞数组是否由字符构成 (是)
iscellstr({11}) %同上 (不是, 是双精度数值)

```

## 第五章 脚本和函数

### 1.44 循环和条件语句

#### 1. 循环语句和条件语句的用法

#### 2. 说明

循环语句: for, while

条件语句: if, switch

#### 3. 实例演示

```
%1_44
for x=1:5 %简单 for 程序实例
    y=x+5
end
[1:5]+5 %矩阵运算方式

if 1<2 %简单 if 程序实例
    disp('true')
else
    disp('false')
end
```

### 1.45 函数文件

#### 1. 函数文件的创建和应用

#### 2. 认识函数

function

#### 3. 说明

函数定义语法示例, function y = myfun(x) (也可以 end 结尾)

在命令行中, 通过使用函数名并输入参数实现调用该函数

函数文件类型包括局部函数、嵌套函数、私有函数和匿名函数

一个函数文件中可以包含用于多个函数的代码

函数工作区与基础工作区是分开的

因此, 命令行和函数之间不能相互访问对方的变量

**Tips:** 与脚本文件同为.m 文件, 但是文件类型不同, 图标也不同

#### 4. 实例演示

```
%1_45
%函数定义语法: function 关键字应放在函数文件中第一个可运行的程序行
function s=mianji(d) %主函数的函数名必须与函数名相同
r=banjing(d) %函数内部运算产生的变量与基础工作区不相通, 因此, 函数和基础工作区通过传
递输入/输出参数的方式交换数据
s=pi*r*r; %函数体: 即计算内容
```

```
end

%% 函数文件内创建局部函数 (R2016b 以上支持)
function r=banjing(d) %子函数, 多个函数须有 1 个主函数
r=d/2;
end
```

## 1.46 函数句柄

### 1. 函数句柄及匿名函数的用法

#### 2. 说明

##### 2.1 函数句柄

函数句柄是一种存储函数关联项的数据类型

通过在函数名称前添加一个 @ 符号为函数创建句柄

##### 2.2 匿名函数 (同样用@, 用法与函数句柄类似)

##### 2.3 功能函数

可以使用函数句柄作为功能函数的输入参数(可理解为功能函数用来处理函数句柄这类数据类型)

典型的功能函数包括 fplot (对一个函数绘图) 和 fzero (求一个函数零点) 等

### 3. 实例演示

```
%1_46
fx=@sin %创建句柄并赋值给变量
fplot(fx) %函数句柄用法: 作为参数传递给功能参数进行绘图
%函数句柄作用: 提供了一种方式, 将整个函数作为参数传递给另一个函数进行处理
% fplot(sin) %不用函数句柄的方式, 报错
mianji=@(r)pi*r*r %匿名函数用法: @(自变量)函数表达式, 再赋值给函数名称(句柄名称)
%匿名函数作用: 相较于函数文件, 不必单独创建函数文件, 节约存储和维护成本, 适用于具有单个
表达式的函数
mianji(1)
fplot(mianji) %匿名函数作为参数进行绘图
```

## 1.47 输入参数数量

### 1. 输入参数的数量和可变数量输入

作者: freexyn 整理/注释: 韩松岳

#### 2. 认识函数

输入 nargin

可变输入 varargin (输出改为 out)

### 3. 实例演示

```
function y=fx1_47(varargin) %若输入参数数量不固定, 则不需单个设置参数变量, 而用
varargin
y=nargin; %Matlab 内置函数 nargin
```

```
end
%例如, cube(a) 求立方体体积/cube(a,b,c) 求长方体体积, 可通过 if 语句判断输入参数数量,
然后进行相应运算。
```

## 1.48 变量名称

### 1. 检查变量名称是否存在或是否与关键字冲突

### 2. 认识函数

判断关键字 iskeyword

判断存在 exist

### 3. 说明

有效的变量名称

有效的变量名称以字母开头, 后跟字母、数字或下划线

Matlab 变量名称对字母大小写是区分的, A 和 a 是不相同的变量

不能使用与 Matlab 关键字冲突的变量名称, 例如 if、end 等

尽量避免使用与函数名相同的变量名, 如 clear、sin 等

如果无意中创建了冲突的变量, 使用 clear 清除

在 Matlab 中使用函数或变量时, 可能会提示以下错误消息:

未定义的函数或变量“a”

原因可能是下列之一

%没有定义改变量, 或者已经清除掉了

%该函数的名称拼写错误

%包含函数的文件名和函数名是不一样的

%该函数的搜索路径已被更改

%该函数所属的工具箱未安装

%该功能是您没有许可证的工具箱的一部分

%其他情况 (具体分析)

### 4. 实例演示

```
%1_48
iskeyword('a') %判断字符'a'是否为关键字
iskeyword('if')
iskeyword('clc')
% if=1 %关键字作为变量, 报错
clc=1 %内置函数名作为变量, 功能被覆盖
clc
clear %清除变量后, clc 恢复
clc
exist('a') %判断是否存在变量'a'
a=1
exist('a')
```

## 1.49 实时脚本

### 1. 简单介绍实时脚本的用法

### 2 实时脚本

实时脚本是一个交互式文档，它在一个称为实时编辑器的环境中将 Matlab 代码与嵌入式输出、格式化文本、方程和图像组合到一起

实时脚本使用实时脚本文件格式存储在扩展名为.mlx 的文件中

Matlab R2016a 及以上版本支持实时脚本

2.1 在实时脚本中插入文本

2.2 在实时脚本中插入方程

2.3 在实时脚本中编辑图窗

2.4 以实时脚本方式打开现有脚本

2.5 将实时脚本另存为脚本，或者导出为 PDF

### 3. 实例演示

```
%1_49
%该程序在实时脚本中运行
三次函数的用法 %实时脚本中的文字
创建三次函数

$$f(x) = x^3 + x^2 + x + 1$$
 %实时脚本中的方程和公式
f=@(x)x.^3+x.^2+x+1
f(1)
%xxxx
三次函数的图像
fplot(f)
title('三次函数')
```

## 1.50 保存和载入

### 1. 保存和载入

1.1 保存工作区的变量和数值，重启 Matlab 后再载入数据

1.2 把变量创建和保存过程用日志文件记录下来

### 2. 认识函数

保存 save

载入 load

创建日志 diary

### 3. 说明

工作区变量是临时存储，退出 Matlab 会丢失

可以把当前工作区数据保存到一个.mat 的文件中，方便存储传输

使用日志文件可以记录 Matlab 命令窗口的所有会话内容

日志文件可以使用文本方式打开和编辑

## 4. 实例演示

```
%1_50
diary fx1_50 %创建日志文件
a=1
b=2
save fx1_50 %存储该文件
diary off %关闭日志文件
%通过在命令行窗口调用 load 函数，可以加载日志文件数据
```

## 1.51 预设和界面调整

### 1. 简单介绍预设和界面调整的用法

### 2. 说明

#### 2.1 调整行间距

#### 2.2 分页面输出

#### 2.3 长语句换行输入

#### 2.4 停止运行

停止正在运行的程序，使用 **Ctrl+C** 或 **Ctrl+Break**

在 Mac 系统上，使用 **Command+**。

#### 2.5 预设

主页-环境-预设-Matlab

### 3. 实例演示

```
%1 51
%freexyn
a=1
format compact %format 函数设置数值显示格式，compact 命令取消结果在命令行窗口中显示的空行
format loose %恢复结果显示的空行
rand(1,100)
% more on %分页输出便于观察，回车按行浏览，空格按页，Q 退出
rand(1,100)
% more off

a=[1.23456789;... %长语句换行输入
   2.23456789]
b=1
```

## 1.52 绘图

### 1. 绘制 $\sin(x)$ 在 $[0 \ 2\pi]$ 的图像

### 2. 认识函数或命令

作图 plot

x 轴标题 xlabel

y 轴标题 ylabel

图像标题 title

### 3. 实例演示

```
%1_52
x=[1 2]
y=[1 2]
plot(x,y,'*') %线型使用'*'
x=0:pi/100:2*pi;
y=sin(x)
plot(x,y)
xlabel('xxxx') %x 轴标题
ylabel('yyyy') %y 轴标题
title('sin(x)') %图像标题
```

## 第六章 符号运算

### 1.53 符号对象

#### 1. 符号数值、变量和表达式的创建方法

作者: freexyn 整理/注释: 韩松岳

#### 2. 认识函数

sym、syms

#### 3. 说明

符号数学工具箱

符号数学工具箱引入了一种特殊的数据类型 - 符号对象

该数据类型包括符号数字, 符号变量, 符号表达式和符号函数, 还包含符号矩阵及上述变量组成的符号数组。

符号数学工具箱提供求解、绘图和操作符号数学方程的功能

#### 4. 实例演示

```
%1_53
%freexyn
1/6 %双精度浮点型 1/6, 在 Matlab 中使用小数表达
x=sym(1/6) %创建符号变量, 分数形式表达
class(x) %测试 x 类型为符号对象
sin(pi) %对双精度浮点 pi 值求 sin 值, 有舍入误差
sin(sym(pi)) %对符号对象数值 pi 求 sin 值, 完整表达

%% 创建符号变量
sym('x') %方法 1
syms x %方法 2 声明后可直接使用, 无输出结果, 但工作区会存储
syms x y z
sym('x',[1 5]) %方法 1 的形式也可赋值多个符号变量, 并自动编号

%% 创建符号表达式
f=2*x+1
g=x*y+z
```



## 1.54 符号函数和符号矩阵

### 1. 符号函数和符号矩阵的创建和应用

### 2. 说明

符号函数和符号矩阵的用法与常规的矩阵和函数相同，差异在于处理的数据类型不同，用符号对象运算时，需要首先声明符号对象。

### 3. 实例演示

```
%1_54
syms x %声明符号对象 x
f=2*x+1 %符号表达式
f(x)=2*x+1 %符号函数
f(1) %结果是符号对象 3，与双精度 3 不同
syms f(x,y) %直接创建（声明）没有函数表达式的函数
f(x,y) %调用
f(1,2) %运算结果

%% 符号矩阵
syms a b c d %声明符号变量
A=[a b;c d] %符号矩阵，结果每行用中括号单独表达
B=[a 1;c 3] %混合类型符号矩阵
sum(A) %符号矩阵运算，与常规矩阵算法相同
sum(B)
sym('x',[2 2]) %创建 2 行 2 列矩阵，自动添加下标
sym('x%d%d',[2 2]) %分别引用 2 个下标
sym('x%d2018%d',[2 2]) %2 个下标中间添加数字

%% 普通矩阵转化为符号矩阵
C=hilb(3) %创建 3 阶希尔伯特矩阵
sym(C) %转化为符号矩阵
```

## 1.55 基本运算

### 1. 符号对象的算术运算、关系运算和逻辑运算

### 2. 认识函数

isAlways %判断符号表达式是否为真

### 3. 说明

符号对象的基本代数运算与浮点型数据的运算大体相同

### 4. 实例演示

```
%1_55
1+1
sym(1)+1
syms a b x %声明符号变量
a+1
a+b
```

```

f=a+b+1
f+x
g(x)=a*x+b %函数运算
g(10)
m=[a b;b a] %符号矩阵运算
m+1
m.*2 %每个元素都乘 2
m*m %矩阵乘法

%% 关系运算
1<2 %浮点型关系运算返回逻辑值
sym(1)<2 %先转换为符号型再运算，返回符号型表达式
isAlways(sym(1)<2) %用符号表达式的判断函数，返回逻辑值
a<b %未知参数关系运算
f=a<b %表达式赋值给 f，以便进行运算
isAlways(f) %警告并返回逻辑假，因 a b 都是未知参数
isAlways(abs(a)>=0)
isAlways(abs(a)*abs(b)>=abs(a*b))
isAlways(abs(a)>=0 | 1>2) %第一项为真，取或运算后，结果为真
isAlways(abs(a)>=0 & 1>2) %结果为假

a | b
a & b

```

## 1.56 使用假设

### 1. 符号变量使用假设

作者: freexyn 整理/注释: 韩松岳

### 2. 认识函数

设置 assume

添加 assumeAlso

显示 assumptions

### 3. 说明

在符号数学工具箱中，符号变量默认是复数变量

若要运算中不使用全体复数域，可以为变量添加假设指定范围

可以通过假设设置变量属于集合：整数、正数和实数

### 4. 实例演示

```

%1_56
%freexyn
syms x
assumptions(x) %x 不存在假设，属于复数域
assume(x>=0) %设置假设
assumptions(x) %查看假设
% assume(x<=5)
% assumptions(x) %会覆盖前面假设内容
assumeAlso(x<=5) %追加假设
assumptions(x)

%% 设置 x 属于集合

```

```

assume(x,'integer') %假设 x 属于整数
assumptions(x)
assume(x,'positive') %假设 x 为正数
assume(x,'real') %假设 x 属于实数
assumptions(x)

%% 符号变量声明时同时设置假设
sym('x')
sym('x','real') %创建（声明）符号变量并假设属于实数
assumptions(x)
syms y positive %另一方法，创建（声明）符号变量并假设属于正数
assumptions(y)
assumptions %不给定参数时，会显示所有假设

%% 设置假设的用法
solve(y+1==0,y) %solve 函数用来解方程，
solve(x+1==0,x)

```

## 1.57 清除假设

### 1. 清除假设和重置符号引擎

### 2. 认识函数

reset

### 3. 说明

符号变量和它们的假设是分开存储的

符号引擎工作空间通常是空的，Matlab 工作空间记录符号变量并在需要时把他们传递到符号引擎工作空间进行计算

符号引擎空间存储了所有的关于符号变量的假设，这些假设会影响方程的求解、化简和变换

清除变量和清除假设是两个过程，运算结束后注意清除假设

### 4. 实例演示

```

%1_57
syms x y z positive %假设不存在工作区中，而在符号引擎工作空间中
assumptions %查看所有假设
whos %查看所有变量
clear x %清除变量 x，并不会清除掉假设
assumptions
whos
% assume(x,'clear') %清除假设：变量 x 是访问和调用 x 假设的桥梁，变量若被清除了，则无法访问或清除假设
syms x %重新声明假设 x
assume(x,'clear')
assumptions
whos
assume([x y],'clear') %可同时清除多个假设，以数组形式
assumptions
reset(symengine) %重制符号引擎工作空间：清理所有符号假设
assumptions

```

## 1.58 可变精度算术

### 1. 可变精度算术的应用

### 2. 认识函数

vpa

### 3. 说明

默认的, Matlab 双精度浮点数使用 16 位数字精度  
而符号数学工具箱的 vpa 函数, 提供了无限大的可变精度  
它默认使用 32 位数字精度, 32 位指的是有效数字的位数

### 4. 实例演示

```
%1_58
pi %默认双精度浮点型pi值, 16位精度, 默认short格式小数点后4位
vpa(pi) %pi转换为可变精度数值, 32位精度
vpa(pi)+1 %先转换为可变精度数值再运算
vpa(pi,100) %通过输入第2个参数指定精度
vpa(sqrt(2),100)

%% 让整个运行环境使用可变精度
digits %获取当前运行环境的精度
i=digits(100) %设置运行环境精度100, i返回设置前的精度值
digits
vpa(pi) %显示pi的可变参数值, 已变为100位
vpa(pi)+1
digits(32)
digits
vpa(pi)
```

## 1.59 运算精度的选择

### 1. 分别在以下三种算术条件下求 $\sin(\pi)$ 的值

符号运算

可变精度运算

双精度浮点型运算

### 2. 说明

#### 2.1 符号算术

默认的, 符号数学工具箱使用确切的数字, 进行精确的符号计算

#### 2.2 可变精度算术

是符号数学工具箱的功能, 是符号计算的近似数值计算  
通过控制数值显示的有效位数实现可变精度  
默认 32 位, 运算速度稍快, 内存消耗量依赖于设定的精度



## 1.61 查找符号变量

### 1. 查找符号变量

作者: freexyn 整理/注释: 韩松岳

### 2. 认识函数

symvar

### 3. 说明

在符号表达式、符号函数和符号矩阵中查找符号变量

symvar 函数自动把查找到的符号变量按字母顺序排列并输出

如果不是查找所有符号变量，那么靠近字母 x 的变量优先查找

特别的，对于符号函数，作为函数输入参数的符号变量优先查找

大多应用在当没有指定运算变量时，用于确定默认符号变量

### 4. 实例演示

```
%1_61
%freexyn
syms a b n t x y
f=sin(a*x+b)+x^n+log(y) %符号表达式
symvar(f) %查找函数表达式 f 中的符号变量
symvar(f,1) %优先查找靠近 x 的符号变量
symvar(f,2) %查找 2 个符号变量
f(t)=sin(a*t+b)+x^n+log(y) %符号函数
symvar(f) %优先查找自变量，然后，查找靠近 x 的变量
symvar(f,1)
symvar(f,2)
m=[a b n;t x y] %符号矩阵
symvar(m)
symvar(m,2) %查找 2 个靠近 x 的变量
```

## 1.62 变量替代

### 1. 符号变量的代入和替代

### 2. 认识函数

subs

### 3. 说明

把符号变量替代为数值

把符号变量替代为矩阵

替换符号矩阵中的元素

### 4. 实例演示

```
%1_62
syms x y
```

```

f=x+y
subs(f,x,1) %将符号表达式 f 中变量 x 替换为 1
subs(f,y,2)
subs(f,1) %符号表达式 f 中默认符号变量替换为 1, 默认符号变量即 x, 或者靠近 x 最近的变量
subs(f,x,y) %x 替换为 y
subs(f,[x y],[1 2]) %将 x 和 y 分别替换成 1 和 2
subs(f,x,[1 2]) %x 替换为矩阵 [1 2]

m=[x y x;y x y] %2 行 3 列的符号变量
m(1,3)=sym('a') %下标索引替换法
subs(m,x,sym('a')) %x 替换为 a
subs(m,m(1,1),sym('a'))
subs(m,x,x^2+x+1+y) %将 x 替换为符号表达式
subs(m,x,[2*x 2*y])

```

## 1.63 长表达式的缩写

### 1. 长表达式的缩写

### 2. 认识函数

pretty  
subexpr

### 3. 说明

长表达式可能含有多个相同的子表达式

可以把这样的子表达式替换并缩写

pretty 使用内部算法来选择要缩写的子表达式, 可以嵌套缩写

subexpr 函数只做一个子表达式的缩写, 它不支持嵌套缩写

它默认使用变量 sigma 来缩写子表达式

### 4. 实例演示

```

%1_63
syms x
s=(sqrt(5)+x)/2
pretty(s)
ss=(s^2+s+1)*(s^2+s-1)/((s^2-s-1)*(s^2-s+1)^2)
pretty(ss) %嵌套缩写, 将表达式中重复出现的子表达式缩写并拎出说明, 子表达式符号无法更改
[s1,t]=subexpr(ss,'t') %嵌套缩写, 返回表达式名称可指定, 且是单行表达

```

## 1.64 符号函数绘图

### 1. 简单介绍符号函数、方程和表达式的绘图方法

### 2. 认识函数

fplot

### 3. 实例演示

```
%1_64
% fplot(@sin) %函数句柄调用
% fplot(sin) %直接传递函数名,报错
% fplot(sin(x)) %函数名传入自变量,报错
%%Matlab 是一个综合运行环境,上述为 Matlab 中基础工作模块中的函数

%% 符号函数工具箱中,也有 fplot
% Matlab 会根据输入数值类型区别调用工具箱,输入数值型会调用基础模块,输入符号型调用符号
函数工具箱
syms x
fplot(sin(x)) %符号工具箱中可以直接调用函数名称
fplot(exp(x))
fplot(sin(x)+x^4*log(x))
fplot(sin(x)+x^4*log(x),[0 10]) %第二个参数指定绘图范围

% f=x^2+x+1 %符号表达式
% fplot(f) %传入符号表达式绘图
f(x)=x^2+x+1 %符号函数
fplot(f) %传入符号函数绘图
```

本系列教程结束

欢迎交流和留言

作者/旺旺/ UP: freexyn (邮箱: freexyn@163.com)

整理/注释: 韩松岳 (邮箱: colincooper@sina.com)

建议、提问、合作、供稿等,请发邮件

End