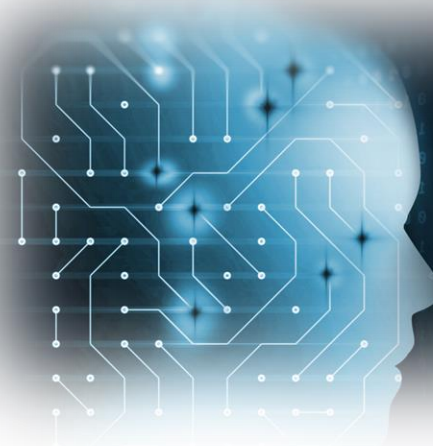
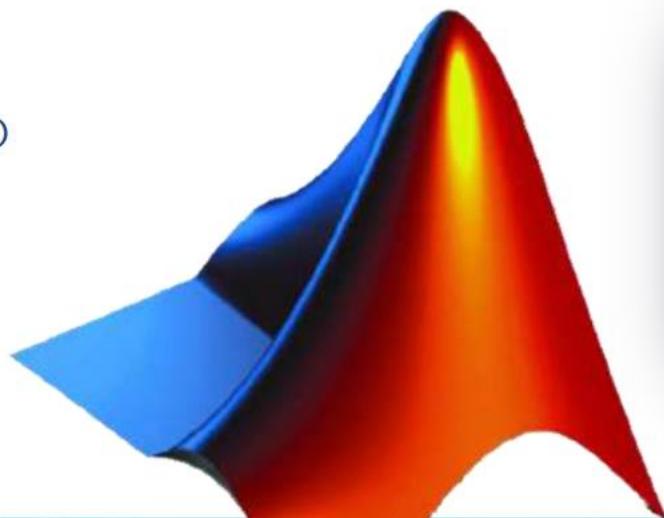




Programming Way to AI

心理学编程

MATLAB®



陈骥
心理与行为科学系

ji.chen@zju.edu.cn

课程目标

- This course is an **introductory** level course for **beginners**.
- The purpose of this course is to introduce you to some of the **basic commands and features** of MATLAB.
- Cultivate interest, pave the way for **data science** and **AI research** in psychological science

Course agenda

- 基本介绍，工作界面及基础命令
- 数学函数(build-in Mathematical functions)
- 数据类型和结构
- 矩阵和数组：创建、重组、索引
- 变量的读入和储存，脚本创建、保存和执行
- 编程撰写的基本语句和代码
 - 循环控制语句、条件控制语句
- 数据可视化和基础绘图

需关注部分

- 1.变量类型与定义
- 2.条件与循环语句
- 3.基础作图可视化
- 4.图形界面设计
5. 函数与代码规范

官方教程和网络资源：

<https://ww2.mathworks.cn/help/matlab/getting-started-with-matlab.html>

INTRODUCTION

- The name MATLAB stands for **MATrix LABoratory**
 - It is good at dealing with matrices
 - Vendor's website: <http://www.mathworks.com>
- Advantages of MATLAB
 - **Ease** of use
 - **Powerful** built-in routines and toolboxes
 - Good visualization of results
 - Popularity in both academia and industry
- Disadvantages of MATLAB
 - Can be slow but can be improved
(MATLAB is an interpreted language)
 - Must be licensed (it's not free)

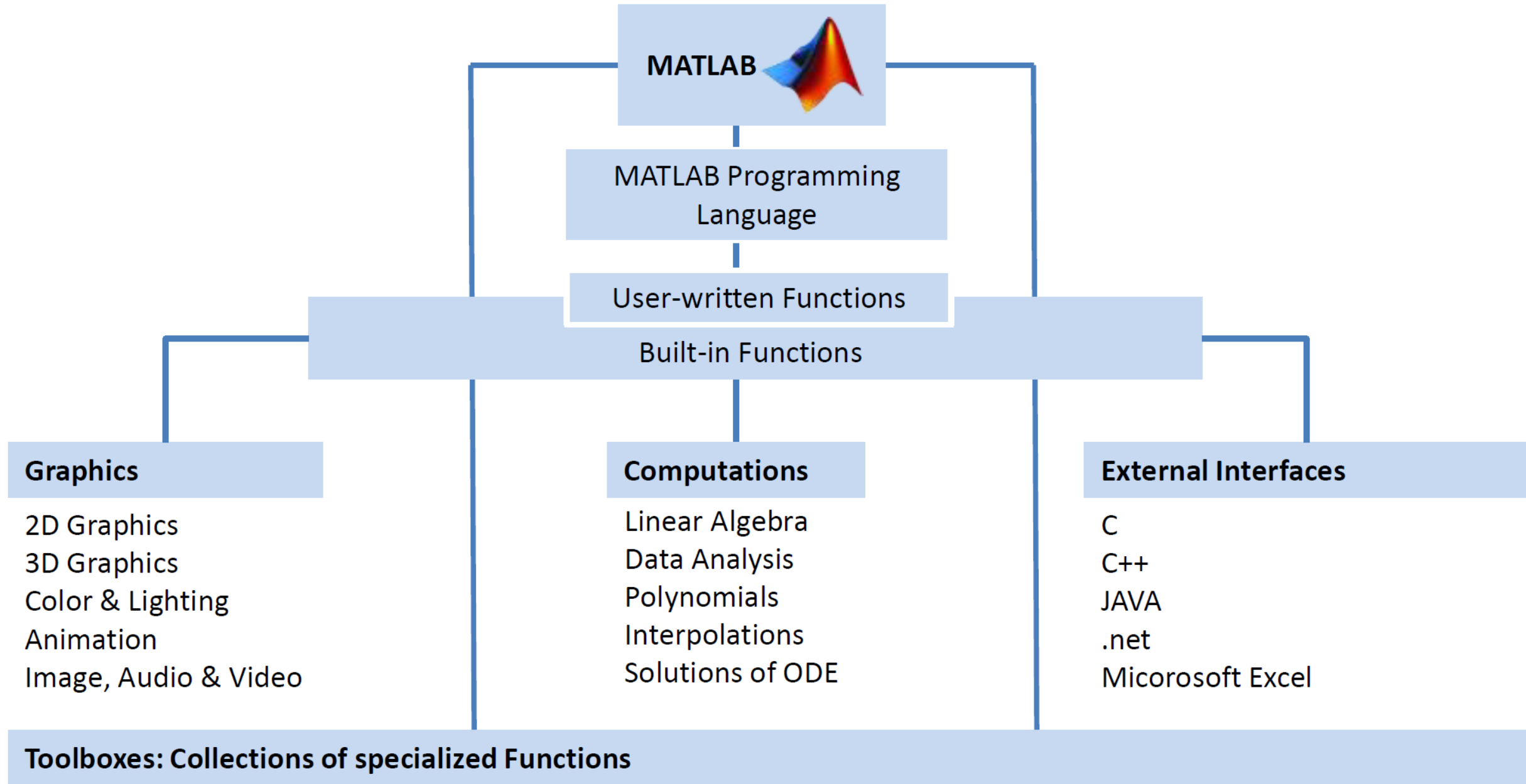
MATLAB

Is an *integrated technical computing environment* that combines *numeric computation, advanced graphics and visualization*, and a *high-level programming language*.

The MathWorks, Inc.

- ✓ High-level language of technical computing 计算
- ✓ Development environment for engineers, scientists 开发
- ✓ Interactive tools for design, problem solving 设计
- ✓ Mathematical function libraries 函数库
- ✓ Graphics and data visualization tools 可视化
- ✓ Custom GUIs 图形界面
- ✓ External Interfaces: C, C++, Fortran, Java, COM, Excel, .NET 多语言接口

MATLAB product family



Matlab Desktop

The screenshot shows the MATLAB 7.11.0 (R2010b) Desktop environment. The interface includes a menu bar (File, Edit, Debug, Parallel, Desktop, Window, Help), a toolbar, and a Command Window. The Current Folder browser on the left shows the directory structure, with the current folder set to `/nas02/home/m/a/markreed/matlab`. The Command Window displays the output of the `whos` command, listing variables and their sizes, classes, and attributes. The Workspace browser on the right shows the current workspace variables, including `danom`, `danomInterp`, `danomse`, `danomseInterp`, `dataarray`, `datafile`, `h`, `lag`, `lagd`, `lags`, `lagt`, `lagyrs`, `mytitle`, `naofile`, `naoindex`, `naolag`, `naorunavg`, `naoyrs`, `origaxis`, `p`, `r`, `rsqlagd`, `rsqlags`, `rsqlagt`, `runavgd`, `runavgdse`, `runavgs`, `runavgsse`, `runavgt`, `runavgtse`, `sanom`, `sanomInterp`, `sanomse`, `sanomseInterp`, `tanom`, `tanomInterp`, `tanomse`, `tanomseInterp`, `windowSize`, `years`, and `yzero`. The Command History browser at the bottom right shows the sequence of commands entered, including `help matlab`, `plot`, `scatterplot`, `plotit`, `help`, `matlab`, `rdms`, `plotregions`, `test_netcdf`, `nao_corr`, `who`, `whos`, `nao_corr`, `who`, `whos`, `years`, `runavt`, `runavgt`, `whos`, `nao_corr`, and `whos`.

Current Directory

Current Folder

Workspace

Command Window

Command History

m file comment

mat file overview

Command window 操作说明

- 在命令窗口【Enter】键提交命令执行。
- 计算结果中的“ans”是英文“answer”的一种缩写，其含义就是“运算答案”。ans是Matlab的一个预定义变量，在没有指定输出变量名时储存命令行输出，并自动匹配数据类型。
- 如果语句以分号结束，MATLAB 会执行计算，但不在命令行窗口中显示输出。
- 按向上 (↑) 和向下箭头键 (↓) 可以重新调用以前的命令。在空白命令行中或在键入命令的前几个字符之后按箭头键。例如，要重新调用命令 `a = 1`，请键入 `a`，然后按向上箭头键。如需扩大所选范围以同时包含多个语句，请按 **Shift+↑**。

在命令窗口中，可发出创建变量和调用函数的命令

例如，通过在命令行中键入以下语句来创建名为 `a` 的变量：

`a = 1`

MATLAB 将变量 `a` 添加到工作区，并在命令行窗口中显示结果

常见通用命令

命令

含义

clc

清除命令窗口的显示内容

clear

清除Matlab工作区中保存的变量

who或whos

显示Matlab工作区中的变量信息

dir

显示当前工作目录的文件和子目录清单

cd

显示或设置当前工作目录

help或doc

获取在线帮助

quit或exit

关闭/推出MATLAB



浙江大学
ZHEJIANG UNIVERSITY

MATLAB数学函数

— Arithmetic Operator & Their Precedence

Operations	Operators	Examples
Addition	+	>> 5 + 3
Subtraction	-	>> 5 - 3
Multiplication	*	>> 5 * 3
Right Division	/	>> 5/3
Left Division	\	>> 5\3 = 3/5
Exponentiation	^	>> 5^3 (means $5^3 = 125$)

Precedence Order	Operators
1	Parentheses (). For nested parentheses, the innermost are executed first.
2	Exponentiation, ^
3	Multiplication, *; Division, /, \
4	Addition, +; Subtraction, -

Trigonometry			
Radian			Degree
Basic	Hyperbolic	Inverse	Basic
sin	sinh	asin	sind
cos	cosh	acos	cosd
tan	tanh	atan	tand
cot	coth	acot	cotd
sec	sech	asec	secd
csc	csch	acsc	cscd

Exponential

exp
log
log10
log2
sqrt

Complex

abs
conj
imag
real
angle

Discrete Math

factorial
gcd
lcm
primes

Matrix

det
trace
rank
inv
expm
sqrtm

■ Example 1

Calculate $z = e^{-a} \sin(x) + 10\sqrt{y}$
for $a=5$, $x=2$, $y=8$

■ Example 2

$\log(142)$, $\log_{10}(142)$

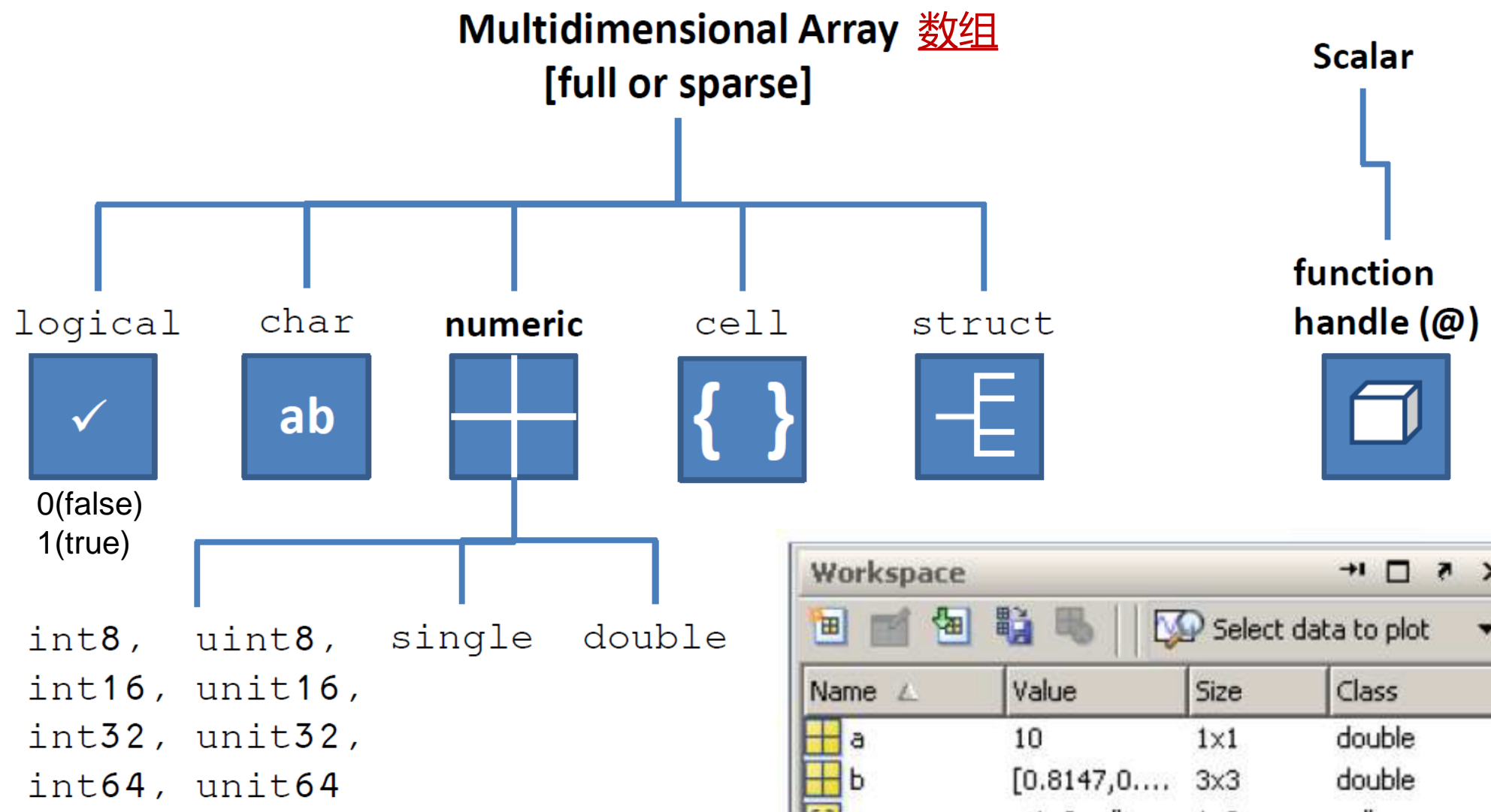


浙江大学
ZHEJIANG UNIVERSITY

MATLAB数据类型

Basic Data and Variables

数据类型和结构



Workspace

Select data to plot

Name	Value	Size	Class
a	10	1x1	double
b	[0.8147,0....	3x3	double
c	<1x2 cell>	1x2	cell
d	<1x5 struct>	1x5	struct

变量命名规则

单个数值变量

Matlab的单个数值默认采用double类型，可以带小数点和负号。 例如：3 -10 0.001 1.3e10 1.256e-6

- 变量名、函数名对字母的大小写是敏感的。如myVar与myvar表示两个不同的变量
- 变量名第一个字母必须是英文字母
- 变量名可以包含英文字母、下划线和数字
- 变量名不能包含空格、标点
- 变量名最多可包含63个字符（6.5及以后的版本）

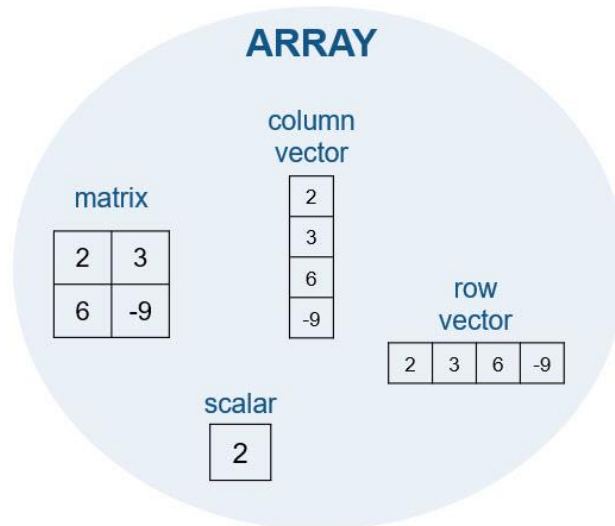
» 数据类型

- 数值型 *numeric* 数组

包括有符号和无符号整数、单精度和双精度浮点数，默认以双精度浮点形式存储所有数值。

标量 2; 行向量 [2 3 6] or [2,3,6];

列向量 [2;3;6]; 矩阵 [2 3;6 1]



Double	双精度数组
Single	单精度数组
Int	有符号整数数组
Unit	无符号整数数组

- 字符(char)和字符串(string)数组

char是一个字符序列，将短文本片段存储为字符向量。如 `c = 'Hello World'`

string是文本片段的容器。如 `str = "Greetings friend"`

strings: 创建不包含字符的字符串数组

```
str = strings(4)
```

```
str = 4x4 string
""    ""    ""    ""
""    ""    ""    ""
""    ""    ""    ""
""    ""    ""    ""
```


字符数组和字符串数组异同

char
C = 'Hello, world'

string
C = string('Hello, world')

1.表现形式上

在MATLAB的显示当中，char变量是用单引号，而string变量用的是双引号。

```
1 | c = 'char';%字符
2 | s = "string";%字符串
```

2.相同用法

在很多需要 **字符串** 变量的地方，例如disp、fprintf、fullfile等函数，两者可以互换。

```
1 | disp('hello world!');
2 | disp("hello world!");
3 |
4 | fprintf('a+b=%d \r\n', 10);
5 | fprintf("a+b=%d \r\n", 10);
```

字符串则是一个变量，如果需要对字符串进行造作，可以使用字符串相关函数，例如strcmp、strcat等。

4.两者转换

利用 **char()** 和 **string()** 函数可以对字符串和字符变量相互转换。

3.差异

- 字符变量可以被索引，而字符串变量不可以被索引。因此字符变量可以被当做一个由字符组成的一维向量。

```
1 | c = 'char';
2 | for i = 1:4
3 |     c(i)
4 | end
```

```
str = 'The rain in Spain.';
C = strsplit(str)
```

```
C = 1x4 cell
    {'The'}    {'rain'}    {'in'}    {'Spain.'}
```

```
data = '1.21, 1.985, 1.955, 2.015, 1.885';
C = strsplit(data, ',')
```

```
C = 1x5 cell
    {'1.21'}    {'1.985'}    {'1.955'}    {'2.015'}    {'1.885'}
```

Overview

5

Scalar of 1 X 1

5	7	4
---	---	---

Row Vector of 1 X 3

5
7
4

Column vector of 3 X 1

5	10	56
6	4	6
7	78	86

One Dimensional
Matrix of 3 X 3

5	10	56	85	23
6	4	6	23	2
7	78	86	53	6

Two Dimensional
Matrix of 3 X 5

5	10	56	85	23	89	0
6	4	6	23	2	0	
7	78	86	53	6		

Three Dimensional
Matrix of 3 X 5

Creation

创建向量

Array

Constant spacing : (colon operator)

Creating a vector with constant spacing by specifying the *first term*, the *spacing*, and the *last term*:

`variable_name = [m:q:n]`

等差序列
q为公差

m = first term

q = spacing

n = last term

or

`variable_name = m:q:n`

Constant spacing linspace

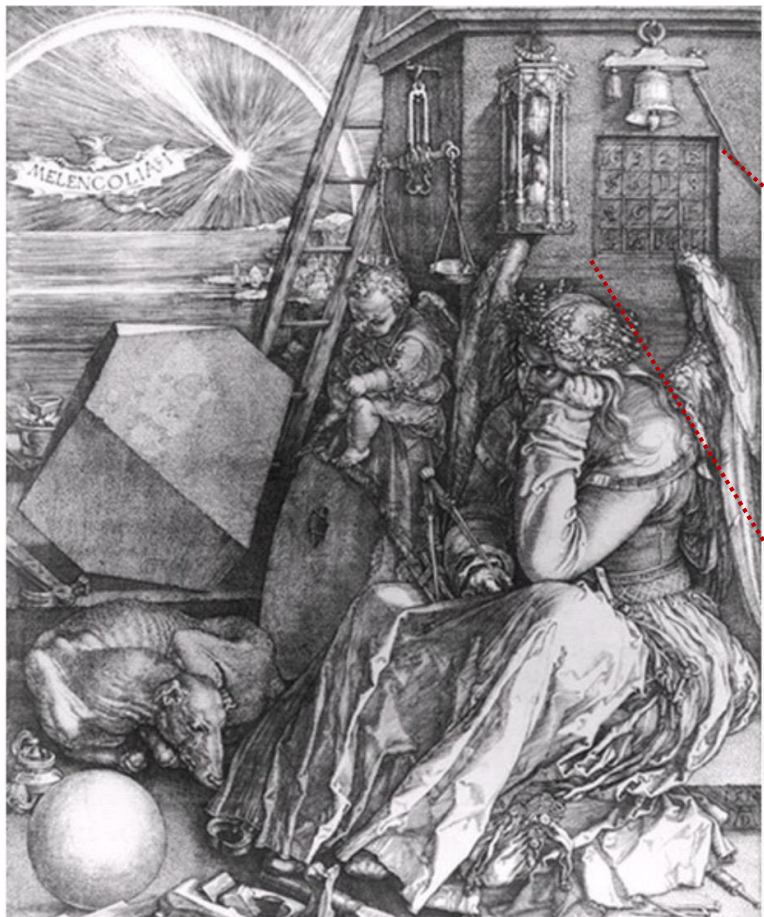
Creating a vector with linear (equal) spacing by specifying the *first* and *last terms*, and the *number of terms*:

`variable_name = linspace(xi, xf, n)`

xi = first element

xf = Last element

n = number of elements



阿布列西特·杜勒早期作品
《文艺复兴雕版画——忧郁者I》

Magic(4)

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

每一行，每一列，和对角线加起来均等于34

sum(A,dim): 当dim为1时，该函数等同于sum(A); 当dim为2时，返回一个列向量，其第i个元素是A的第i行的各元素之和。

prod(A,dim): 当dim为1时，该函数等同于prod(A); 当dim为2时，返回一个列向量，其第i个元素是A的第i行的各元素乘积。

mean(A,dim); median(A,dim)

median(A,3)?

矩阵赋值:

```
A(1)=1;
A(2)=2;
```

- 大矩阵可以把小矩阵作为其元素

```
A = [A ; 3 4];
```

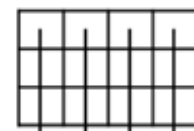
创建矩阵

➤ 直接创建

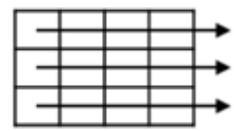
```
A = [1 2 3; 4 5 6; 7 8 9]
```

%或者

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



Dimension = 1



Dimension = 2

用空格 ' ' 把元素间隔开
用逗号 ',' 把元素间隔开
是等价的

```
mat1 = [1 2 3];
mat2 = [1,2,3];
```

- 矩阵行与行之间用 分号 分开

```
mat3 = [ 1 2 3; 4, 5, 6];
```

mat3 =

```
1 2 3
4 5 6
```

➤ 采用函数建立矩阵

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

>> y = reshape(x,3,4) % reshape用法
y =
     1     4     7    10
     2     5     8    11
     3     6     9    12
```

Reshape创建矩阵

```
>> z = 1:4
z =
     1     2     3     4

>> diag(z) % 创建对角矩阵
ans =
     1     0     0     0
     0     2     0     0
     0     0     3     0
     0     0     0     4
```

创建对角矩阵

```
eye(size(A)) % 产生与A矩阵同阶的单位矩阵
zeros(m,n) % 产生0矩阵
ones(m,n) % 产生1矩阵
rand(m,n) % 产生随机元素的矩阵
size(a) % 返回包含两个元素的向量。
length(a) % 返回向量的长度。
```

➤ 矩阵元素的索引

- `A(1)` 引用A的第一个元素;
- `A(i)` : 向量 A 中的第 i 个元素;
- `A(i,j)` : 矩阵 A 中的第 i 行, 第 j 列元素; 多个元素的引用:

Quiz: 矩阵中的元素索引值是如何排序的?

先列再行

`A(1, :)` 冒号提取整行整列

- `A(i:j, m:n)` 提取第i行到j行, m列到n列子矩阵

`A(:)` 将矩阵转换为一个列向量

➤ 创建三维数组

方法1

```
for i=1:2
    for j=1:2
        for k=1:2
            A(i,j,k)=i+j+k;
        end
    end
end
```

$A(:, :, 1)$ 第一页上的矩阵

$A(:, :, 2)$ 第二页上的矩阵

Now try entering `median(A,3)`

方法2 先输入一个二维数组

```
D2=[1,2,3;4,5,6;7,8,9];
D3(:,:,1)=D2;
D3(:,:,2)=2*D2;
D3(:,:,3)=3*D2;
```

$D3(:,:,2) =$

1	2	4	6
2			
3	8	10	12
4			
5	14	16	18

方法3

Cat命令: $C = \text{cat}(\text{dim}, A1, A2, A3, A4, \dots)$, dim表示创建数组的维度, $A1, A2, A3, A4$ 表示的是各维度上的数组。在命令窗口中输入以下代码即可创建一个简单的三维数组:

```
D2=[1,2,3;4,5,6;7,8,9];
C=cat(3,D2,2*D2,3*D2);
```


矩阵的基本运算:

- (1) 矩阵转置;
- (2) 矩阵加和减;
- (3) 矩阵乘法;
- (4) 矩阵除法 $A \setminus b = \text{inv}(A) * b$;
- (5) 矩阵的乘方 a^2

加法

$$\begin{bmatrix} 1 & 4 & 2 \\ 2 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \end{bmatrix} = \begin{bmatrix} 1+0 & 4+0 & 2+5 \\ 2+7 & 0+5 & 0+0 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 7 \\ 9 & 5 & 0 \end{bmatrix}$$

矩阵的加法满足下列运算律(A, B, C都是同型矩阵):

$$A + B = B + A$$

$$(A + B) + C = A + (B + C)$$

减法

$$\begin{bmatrix} 1 & 4 & 2 \\ 2 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \end{bmatrix} = \begin{bmatrix} 1-0 & 4-0 & 2-5 \\ 2-7 & 0-5 & 0-0 \end{bmatrix} = \begin{bmatrix} 1 & 4 & -3 \\ -5 & -5 & 0 \end{bmatrix}$$

矩阵的加减法和矩阵的数乘合称矩阵的线性运算

乘法——矩阵叉乘(*)

A的每一行中的数字对应乘以B的每一列的数字把结果相加起来

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}, B = \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix}, C = \begin{bmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix}$$

$$C_{11} = A_{11}B_{11} + A_{12}B_{21}$$

$$C_{12} = A_{11}B_{12} + A_{12}B_{22}$$

$$C_{21} = A_{21}B_{11} + A_{22}B_{21}$$

$$C_{22} = A_{21}B_{12} + A_{22}B_{22}$$

两行两列的矩阵×两行一列的矩阵=两行一列的矩阵

$$\begin{bmatrix} 5 & 6 \\ 8 & 3 \end{bmatrix} \times \begin{bmatrix} 6 \\ 3 \end{bmatrix} = \begin{bmatrix} \\ \end{bmatrix}$$

A

B

C

必须满足A矩阵的列的数量等于B矩阵的行的数量
行数量为A的行数量, 列数量为B的列数量

转置 $A = A'$

$$\begin{bmatrix} 2 & 4 & 3 \\ 0 & -2 & 8 \end{bmatrix}^T = \begin{bmatrix} 2 & 0 \\ 4 & -2 \\ 3 & 8 \end{bmatrix}$$

$$(A^T)^T = A$$

$$(\lambda A)^T = \lambda A^T$$

$$(AB)^T = B^T A^T$$

在没有共轭的情况下,
对于逻辑和非数值输入,
产生相同结果。

共轭转置 $A = A'$

$$A = \begin{bmatrix} 3+i & 5 \\ 2-2i & i \end{bmatrix} \quad \bar{A} = \begin{bmatrix} 3-i & 5 \\ 2+2i & -i \end{bmatrix} \quad A^* = \begin{bmatrix} 3-i & 2+2i \\ 5 & -i \end{bmatrix}$$

转置不更改这
些元素的虚部
的符号

矩阵点乘(.*)

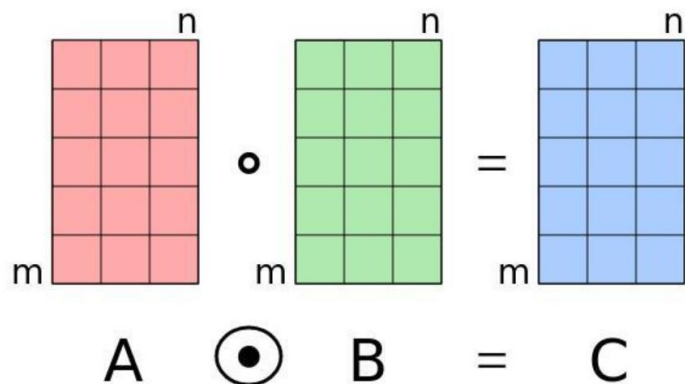
各个矩阵对应元素相乘

$$y = wx = \begin{bmatrix} w_{11} \\ w_{21} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} = \begin{bmatrix} w_{11}x_{11} & w_{11}x_{12} & w_{11}x_{13} \\ w_{21}x_{21} & w_{21}x_{22} & w_{21}x_{23} \end{bmatrix}$$

or:

$$y = wx = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} = \begin{bmatrix} w_{11}x_{11} & w_{12}x_{12} & w_{13}x_{13} \\ w_{21}x_{21} & w_{22}x_{22} & w_{23}x_{23} \end{bmatrix}$$

点乘对象的行数必须相等，前者的列数必须与后者相等，或为1



维度相同时
哈达玛积

example :

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \odot \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} & a_{13}b_{13} \\ a_{21}b_{21} & a_{22}b_{22} & a_{23}b_{23} \\ a_{31}b_{31} & a_{32}b_{32} & a_{33}b_{33} \end{pmatrix}$$

向量点乘：内积(inner product), 数量积, 点积(dot product)

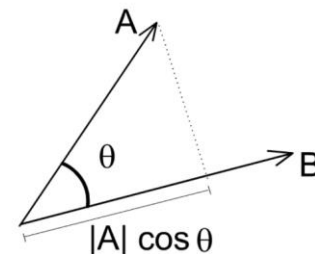
设向量 $A=[a_1, a_2, \dots, a_n]$, $B=[b_1, b_2, \dots, b_n]$
则向量A和B的内积表示为：

其结果
为某一
数值

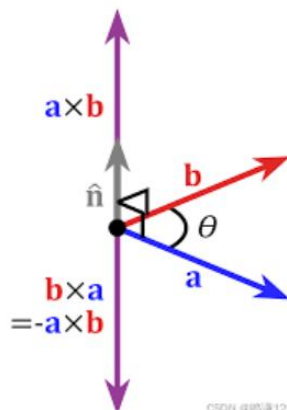
$$A \cdot B = a_1 \times b_1 + a_2 \times b_2 + \dots + a_n \times b_n$$

$$A \cdot B = |A| \times |B| \times \cos \theta$$

$|A|$ 和 $|B|$ 分别是向量A和B的模
 θ 是向量A和向量B的夹角



向量叉乘：向量积、矢积、外积



运算结果是一个向量而不是一个标量
所得向量与这两个向量 (a, b) 垂直

$$\vec{a} = (x_1, y_1, z_1), \vec{b} = (x_2, y_2, z_2), \vec{a} \text{ 与 } \vec{b} \text{ 之间夹角为 } \theta$$

$$\vec{a} \times \vec{b} = (y_1 z_2 - z_1 y_2, z_1 x_2 - x_1 z_2, x_1 y_2 - y_1 x_2)$$

$$|\vec{a}| |\vec{b}| \sin(\theta)$$

向量范数

- L_0 范数

$\|x\|_0 \stackrel{def}{=} \text{向量中非零元素的个数}$

其在matlab中的用法:

```
1 sum( x(:) ~= 0 )
```

- L_1 范数

$\|x\|_1 \stackrel{def}{=} \sum_{i=1}^m |x_i| = |x_1| + \cdots + |x_m|$, 即向量元素绝对值之和

其在matlab中的用法:

```
1 norm(x, 1)
```

- L_2 范数

$\|x\|_2 = \left(|x_1|^2 + \cdots + |x_m|^2 \right)^{1/2}$, 即向量元素绝对值的平方和后开方

其在matlab中的用法:

```
1 norm(x, 2)
```

- L_∞ 范数

- 极大无穷范数

$\|x\|_\infty = \max\{|x_1|, \cdots, |x_m|\}$, 即所有向量元素绝对值中的最大值

其在matlab中的用法:

```
1 norm(x, inf)
```

- 极小无穷范数

$\|x\|_\infty = \min\{|x_1|, \cdots, |x_m|\}$, 即所有向量元素绝对值中的最小值

其在matlab中的用法:

```
1 norm(x, -inf)
```

Quiz: 大家自己创建一个向量，并在Matlab中操作体会

矩阵范数

分为诱导范数和元素形式范数

1. 诱导范数

- L_1 范数 (列和范数)

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m \{|a_{ij}|\}, \text{ 即所有矩阵列向量绝对值之和的最大值}$$

其在matlab中的用法:

```
1 norm(A,1)
```

- L_2 范数

$$\|A\|_2 = \sqrt{\lambda_i}, \text{ 其中 } \lambda_i \text{ 为 } A^T A \text{ 的最大特征值。}$$

其在matlab中的用法:

```
1 norm(A,2)
```

- L_∞ 范数 (行和范数)

$$\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n \{|a_{ij}|\}, \text{ 即所有矩阵行向量绝对值之和的最大值}$$

其在matlab中的用法:

```
1 norm(A,inf)
```

2. "元素形式"范数

- L_0 范数

$$\|A\|_0 \stackrel{def}{=} \text{ 矩阵的非零元素的个数}$$

其在matlab中的用法:

```
1 sum(sum(A ~= 0))
```

- L_1 范数

$$\|A\|_1 \stackrel{def}{=} \sum_{i=1}^m \sum_{j=1}^n |a_{ij}|, \text{ 即矩阵中的每个元素绝对值之和}$$

其在matlab中的用法:

```
1 sum(sum(abs(A)))
```

- L_∞ 范数

$$\|A\|_\infty = \max_{i=1, \dots, m; j=1, \dots, n} \{|a_{ij}|\}, \text{ 即矩阵的各个元素绝对值的最大值}$$

其在matlab中的用法:

```
1 max(max(abs(A)))
```

- L_F 范数

$$\|A\|_F \stackrel{def}{=} (\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2)^{1/2}, \text{ 即矩阵的各个元素平方之和后开方}$$

其在matlab中的用法:

```
1 norm(A,'fro')
```

Quiz: 大家自己创建一个矩阵，并在Matlab中操作体会

» 数据类型

- 结构体数组 struct

字段的数据容器: 将相关数据组合在一起的一种数据类型

每个字段都可以包含任意类型或任意大小的数据

结构体数组中数据包含在可按名称访问的字段中

输入:

```
patient(1).name = 'John Doe';  
patient(1).billing = 127.00;  
patient(1).test = [79,75,73;180,178,177.5;220,210,205];
```

输出显示:

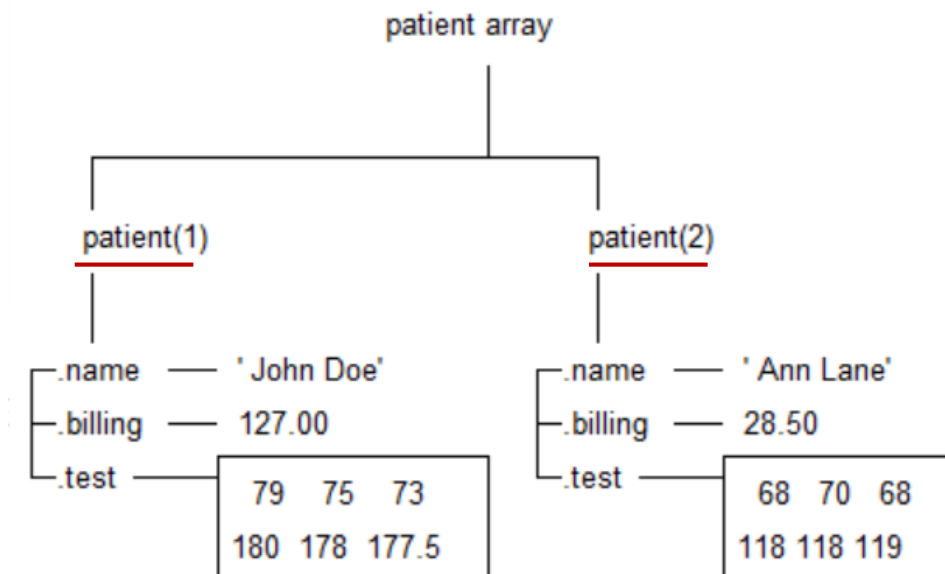
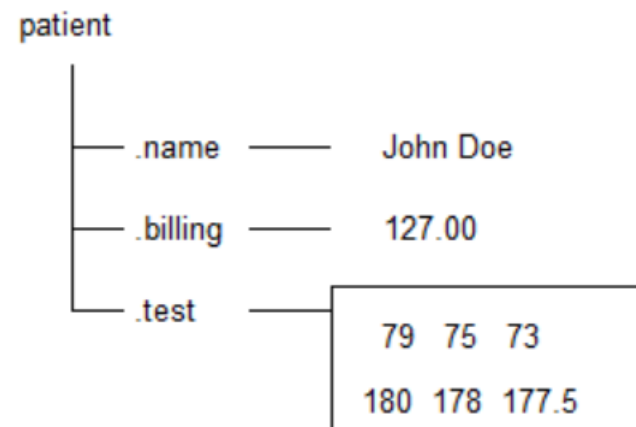
```
patient = struct with fields:  
    name: 'John Doe'  
    billing: 127  
    test: [3x3 double]
```

输入:

```
patient(2).name = 'Ann Lane';  
patient(2).billing = 28.50;  
patient(2).test = [68,70,68;118,118,118,119;172,170,169];
```

输出显示:

```
patient=2x3 struct  
    name  
    billing  
    test
```



» 数据类型

- 元胞数组cell

可以包含各种类型和大小的数据的数组，每个元胞都可以包含任意类型的数据。

```
c = {42, rand(5), "abcd"}
```

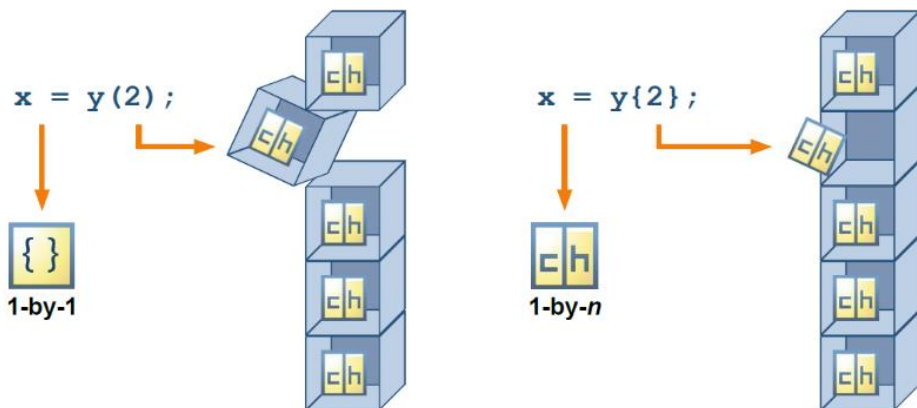
```
c =
```

```
1×3 cell array
```

```
{[42]} {5×5 double} {"abcd"}
```

通过将索引括在圆括号 () 中可以引用元胞集

使用花括号 {} 进行索引来访问元胞的内容



```
>> c(1,1)  
ans =  
  
1×1 cell 数组  
  
{[42]}
```

```
>> c{1,1}  
ans =  
  
42
```

创建空元胞

```
C = cell(3)
```

```
C=3×3 cell array
```

{0×0 double}	{0×0 double}	{0×0 double}
{0×0 double}	{0×0 double}	{0×0 double}
{0×0 double}	{0×0 double}	{0×0 double}

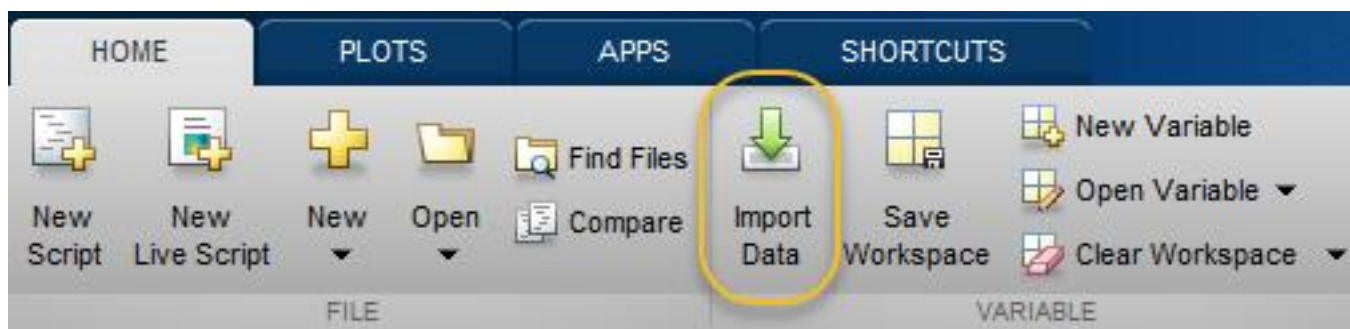


浙江大学
ZHEJIANG UNIVERSITY

变量的读入和储存

» 变量的读入和储存

- 从MATLAB界面中导入
 - 常见的标准文件格式：txt, csv, xls, xlsx, jpg, png

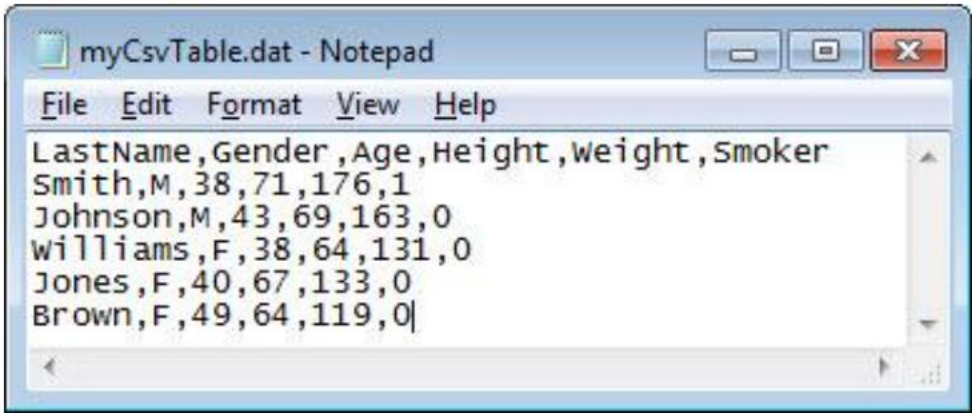


读取步骤

1. 选择Home选项卡中的Import data选项，或双击当前路径下的文件
2. 选择要加载的数据，缺失值被导入为NaN。
3. 指定加载数据集的方式：表、向量、矩阵或文本数据等
4. **单击Import Selection。**

变量的读入和储存

- 使用命令读文本文件或表
- T = readtable(filename,name,value)



T = readtable(filename)

T=5x6 table

LastName	Gender	Age	Height	Weight	Smoker
{'Smith' }	{'M'}	38	71	176	1
{'Johnson' }	{'M'}	43	69	163	0
{'Williams' }	{'F'}	38	64	131	0
{'Jones' }	{'F'}	40	67	133	0
{'Brown' }	{'F'}	49	64	119	0

readtable	基于文件创建表
writetable	将表写入文件

T = readtable(filename) 通过从文件中读取列向数据来创建表。

readtable 基于文件的扩展名确定文件格式：

- .txt、.dat 或 .csv（带分隔符的文本文件）
- .xls、.xlsb、.xlsm、.xlsx、.xltm、.xltx 或 .ods（电子表格文件）
- .xml（可扩展标记语言 (XML) 文件）
- .docx（Microsoft® Word 文档文件）
- .html、.xhtml 或 .htm（超文本标记语言 (HTML) 文件）

示例： filename = 'myCsvTable.dat';

从R2020a开始可以检测数据类型，丢弃多余的标题行，并填充缺失值

T = readtable(filename,'Format','%s%s%u%f%f%s')

元胞数组： %s, double： %f, uint32： %u



• 使用命令写文本文件或表

语句: `T = writetable(filename,name,value)`

示例:

```
LastName = {'Sanchez';'Johnson';'Li';'Diaz';'Brown'};  
Age = [38;43;38;40;49];  
Smoker = logical([1;0;1;0;1]);  
varNames = {'Last Name','Age','Smoker (1 or 0)'};  
T = table(LastName,Age,Smoker,'VariableNames',varNames)
```

T=5x3 table

Last Name	Age	Smoker (1 or 0)
{'Sanchez'}	38	true
{'Johnson'}	43	false
{'Li' }	38	true
{'Diaz' }	40	false
{'Brown' }	49	true

```
writetable(T,'sampletable.txt')
```


» 变量的读入和储存

- 使用命令读写文本文件或表

部分名称-值参数 (name,value)

Name		Value
FileType	文件类型	'spreadsheet' 'text' 'delimitedtext' 'fixedwidth' 'xml' 'worddocument' 'html'
ReadRowNames	读取第一列以作为行名称的指示符	false (默认) true 0 1
TextType	导入的文本数据的类型	'char' 'string'
VariableNamingRule	保留变量名称的标志	"modify" (默认) "preserve"
MissingRule	管理缺失数据的过程	'fill' (默认) 'error' 'omitrow' 'omitvar'
Delimiter	字段分隔符 (仅限文本文件)	任何有效字符 (如逗号 ';' 或句点 '.')
VariableWidths	变量的字段宽度 (仅限文本文件)	正整数值向量
Range	要读取的工作表的部分 (仅限电子表格文件)	由逗号分隔的对组指定的矩形区域



浙江大学
ZHEJIANG UNIVERSITY

循环和控制

» 编程撰写： 循环控制及条件控制语句

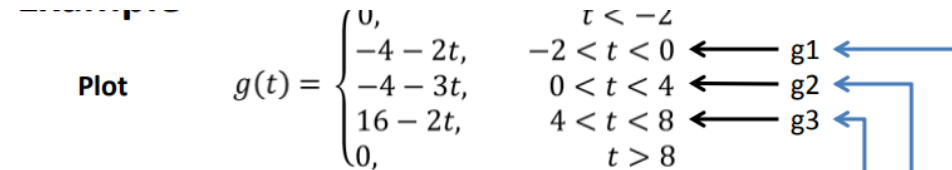
- 循环及条件控制语句： 语言语法

if,elseif,else	条件为 true 时执行语句
switch,case,otherwise	执行多组语句中的一组
for	用来重复指定次数的 for 循环
while	条件为 true 时重复执行的 while 循环
try,catch	执行语句并捕获产生的错误
break	终止执行 for 或 while 循环
return	将控制权交还给调用脚本或函数
continue	将控制传递给 for 或 while 循环的下一迭代
pause	暂时停止执行 MATLAB
parfor	并行 for 循环
end	终止代码块或指示最大数组索引

Conditional expression consisting of relational and/or logical operators.

Relational Operators	
Operations	Operators
Less than	<
Greater than	>
Less than or equal to	<=
Greater than or equal to	>=
Equal to	==
Not Equal to	~=

Logical Operators	
Operations	Operators
And	&&
Or	
Not	~



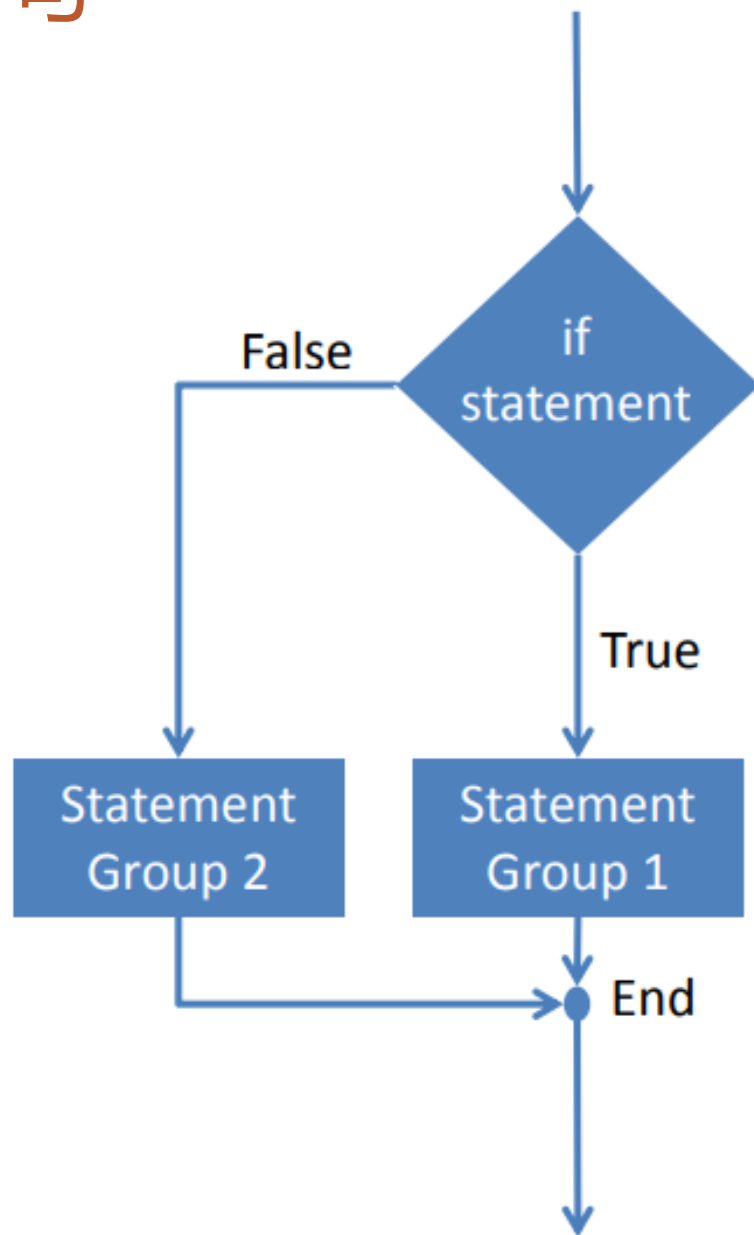
```

1- t = linspace(-5,10);
2- g1 = -4 - 2*t;
3- g2 = -4 + 3*t;
4- g3 = 16 - 2*t;
5- g = g1.*(-2<t & t<=0)...
6-     + g2.*(0<t & t<=4)...
7-     + g3.*(4<t & t<=8);
8- plot(t,g)

```

*if, elseif, else*控制语句

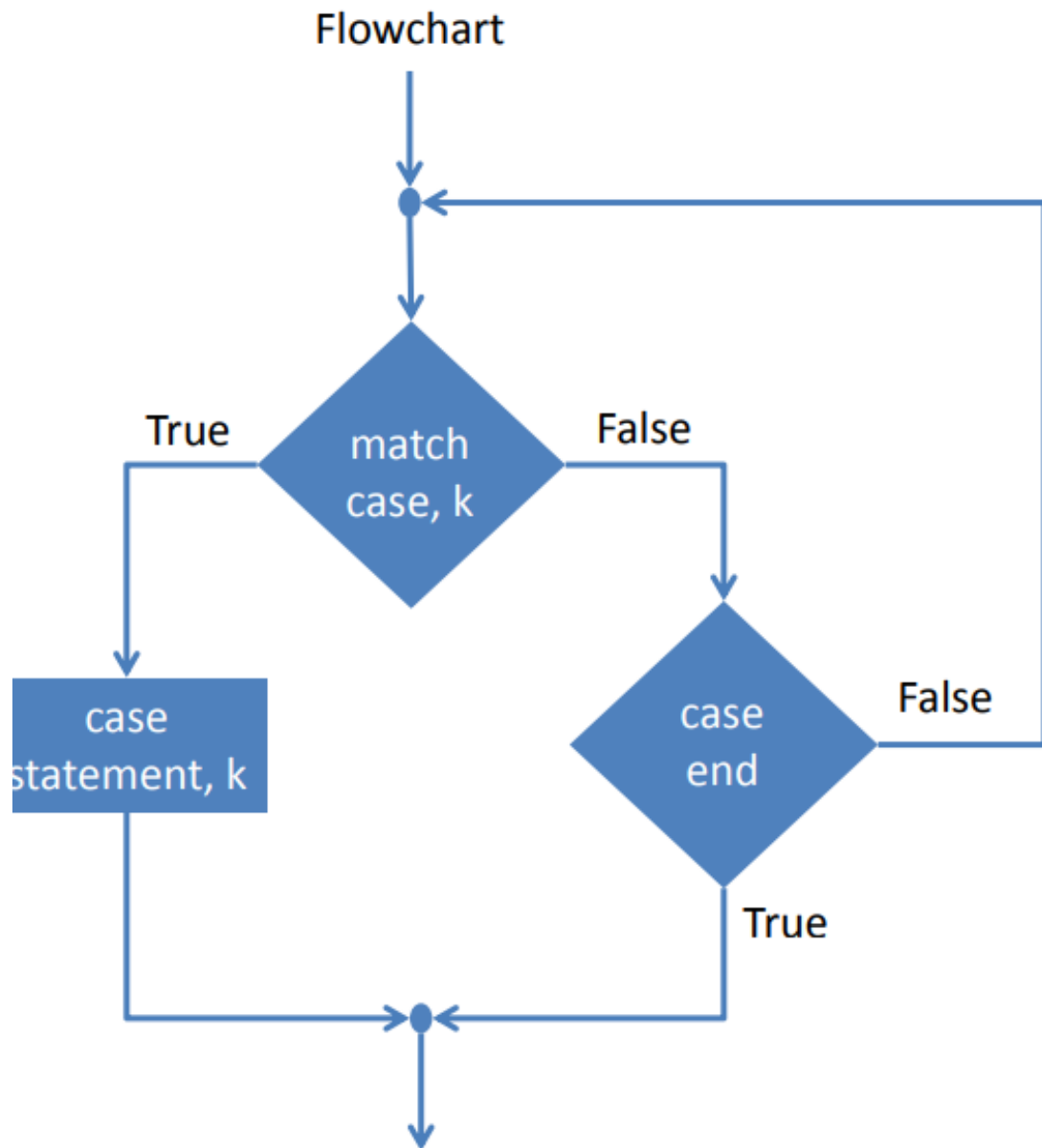
如果是这个情况
那执行1
反之，则执行2



if结构语句格式:

```
if 条件1
    语句组(statement) 1
elseif 条件2
    语句组 2
...
elseif 条件m
    语句组 m
else
    语句组n
end
```

switch控制语句



switch结构语句格式:

switch 表达式
case 表达式1
语句组1

...

case 表达式m
语句组m
otherwise
语句组 n

end

```
1 %示例代码
2 n = input('请输入一个数字: ');
3
4 switch n
5     case -1
6         disp('此值是-1')
7     case 0
8         disp('此值是0')
9     case 1
10        disp('此值是1')
11    otherwise
12        disp('其他数值')
13 end
```

switch 块会测试每个 case, 直至一个 case 表达式为 true。case 在以下情况下为 true:

- case_expression == switch_expression.
- 对于元胞数组 case_expression, 元胞数组的至少一个元素与 switch expression 匹配

case后用{}而不是括号（）包裹

try控制语句

try语句

try语句是一种试探性执行语句，其语句格式为：

```
try
    语句组1
catch
    语句组2
end
```

try语句先试探性执行语句组1，如果在执行过程中出现错误，转去执行语句组2.

不能在一个try块中使用多个catch块，但可以嵌套完整的try/catch块。

```
m = rand(3,4);
n = magic(5);
try
```

```
    a = m*n;
```

*%当程序碰到 $a = m*n$;
错误后，就会跳转到catch
里面的语句，继续执行disp(a)*

```
catch
    disp(size(m))
    disp(size(n))
```

```
end
disp(m)
```

根据要求，现场编写条件控制语句达到目的

练习1:

分别用if和switch多分支语句计算税款,用户输入货价，输出相应的税款：

货价<2000，免税；

货价在2000到5000之间，超过2000部分抽税2%；

5000以上，除2%以外，5000以上抽税5%，加收手续费60元。