



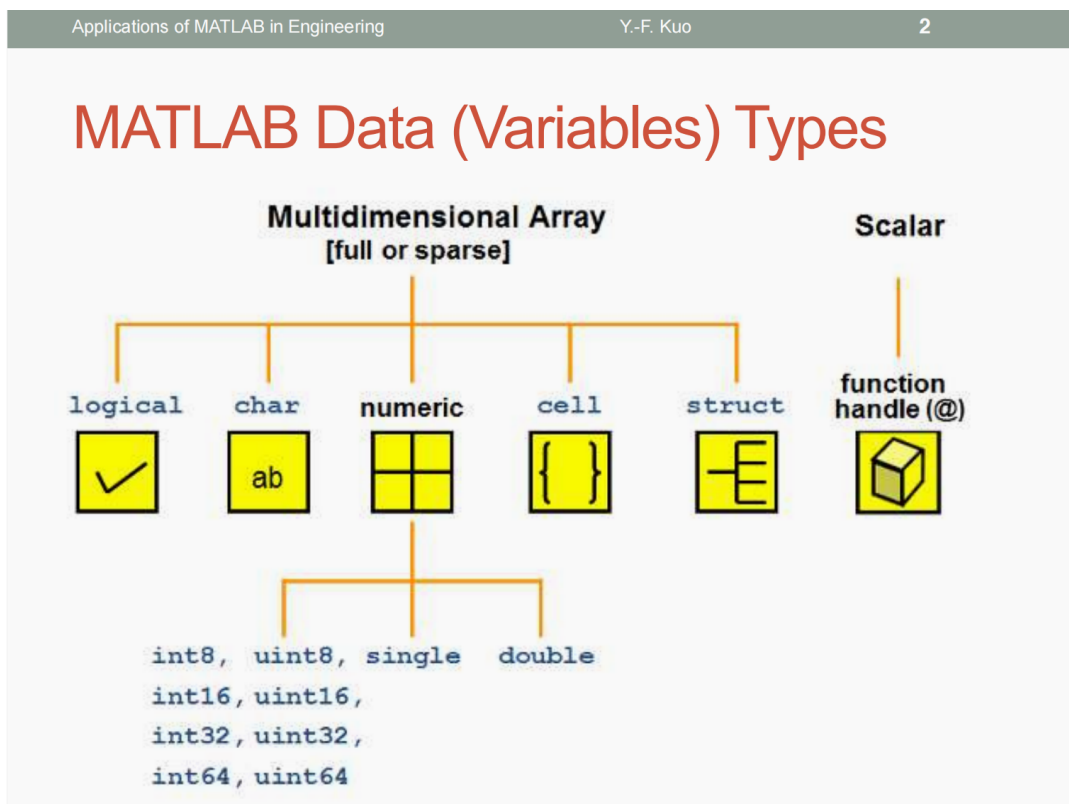
变量与档案储存

📖 Notebook

🚀 MATLAB

<https://youtu.be/n8M7d-iXMUk?si=aZFGZrr1Vufn0p85>

Variable



数据类型名称+()———转换为指定的数据类型;

例:

```
a=5.56;
int8(a);
```

```
%把double型转化为int型。
```

本质上是一个function。

Charactor and String

```
s='a';  
s1='example';  
s2='string';
```

```
s3=[s1, s2]  
%>>s3  
%output:examplestring
```

```
s4=[s1;s2];  
%>>s4  
%output:error
```

```
%原因：s1 s2的字符串长度不一样。s4是一个数组
```

逻辑运算符logical operations和

逻辑运算符'=='两侧的代码放在哪一侧都可以。

```
'a'==str;  
str=='a';  
%这两种表述都对。
```

Exercise

- Write a script that inverts any given string

```
s1='I like the letter E'  
    ↓  
s2='E rettell eht ekil I'
```

```
function s2=untitled(s1)  
    s2=s1(end:-1:1);  
    disp(s2);  
end
```

```
>>untitled('I like the letter E')
```

output:

```
ans =  
'E rettell eht ekil I'
```

Structure array结构数组

```
student.name = 'John Doe';  
student.id = 'jdo2@sfu.ca';  
student.number = 301073268;  
student.grade = [100, 75, 73; ...  
95, 91, 85.5; ...  
100, 98, 72];  
student
```

name、id、number、grade这些被称为field

结构数组的创建：

1.对域赋值创建——可以通过为结构数组的每个域分配值来创建结构数组。对域的分配是struct_name (record#) .field_name=data

```
%example
student.name='John Zheng';
student.number='3238010084';
student.score=100;

%>>student
%student=
%   name:John Zheng
%   number:3238010084
%   score:100
```

向student数组添加元素

```
student(2).name='kellie';
student(2).number=.....;
student(2).score=.....;

%>>student
%student=
% 1x2 struct array with fields:
%       name
%      number
%       score
```

2.利用函数struct创建

格式：

```
struct_name=struct('field1',value1,'field2',value2,...)
```

```
struct_name=struct('field1',{},'field2',{},...)
```

表 4-1 结构数组的相关函数

函数名	说 明	函数名	说 明
struct	创建结构数组	getfield	获取域值
isstruct	判定是否为结构数组, 是结构数组时, 其值为真	isfield	判定域是否在结构数组中, 在结构数组中时, 其值为真
fieldnames	获取结构数组域名	rmfield	删除结构数组中的域
setfield	设定域值	orderfields	域排序

Structure Functions

cell2struct	Convert cell array to structure array
fieldnames	Field names of structure, or public fields of object
getfield	Field of structure array
isfield	Determine whether input is structure array field
isstruct	Determine whether input is structure array
orderfields	Order fields of structure array
rmfield	Remove fields from structure
setfield	Assign values to structure array field
struct	Create structure array
struct2cell	Convert structure to cell array
structfun	Apply function to each field of scalar structure

nest structure

在一个结构数组中包含另一个结构数组。

采用双点号的格式: struct_name1.nest.field_name

细胞数组 Cell Array

matlab中细胞数组是一个指针, 指向细胞数组的成员。

创建细胞数组方法:

1.对细胞元素直接赋值创建

格式: cell_name{i,j}={value}

```
student{1,1}=[ '200255662'; '200551668'];  
student{2,1}={ 'xijinning'; 'liqiang'};  
student{1,2}={ 'math', 'physics'. 'maksim'};  
student{2,2}={ [1 2 3]; [4 5 6]};
```

```
%student{1,1}(1,1)='200255662'
```

(1) 花括号和圆括号在使用上的细微区别,花括号表示细胞元素的内容;圆括号表示细胞元素。这里在建立细胞数组时,是通过给细胞元素赋值来确定细胞元素的。通过以下语句,大家可以体会二者的区别。

```
>> student{1,2}  
ans =  
    '高数 1'    '英语 1'    '体育 1'    '物理 1'    '马哲'    '线代'    '制图'  
    '高数 2'    '英语 2'    '体育 2'    '物理 2'    '邓论'    '电路'    '语文'  
>> student(1,2)  
ans =  
    {2x7 cell}
```

(2) 给 student{1,2}、student{2,1}、student{2,2}和 student(1,2)、student(2,1)、student(2,2)赋值使用了细胞数组的嵌套,即这些细胞元素本身就是细胞数组。

(3) student{1,1}与 student{2,1}细胞元素值同样为字符串,但 student{1,1}每个字符串的长度相同,所以可以以字符型数组存储;而 student{2,1}各字符串的长度不同,所以改为字符型细胞数组存储。同理,student{1,2}也以字符型细胞数组存储。

(4) 细胞数组的结构图可以通过函数 cellplot 绘出,如图 4.3 所示。

2.利用函数cell创建细胞数组

```
cell_name=cell(m,n...z);
```

```
或cell_name=cell([m n]);
```

Cell Array Functions

cell	Create cell array
cell2mat	Convert cell array to numeric array
cell2struct	Convert cell array to structure array
celldisp	Cell array contents
cellfun	Apply function to each cell in cell array
cellplot	Graphically display structure of cell array
cellstr	Create cell array of strings from character array
iscell	Determine whether input is cell array
mat2cell	Convert array to cell array with different sized cells
num2cell	Convert array to cell array with consistently sized cells
struct2cell	Convert structure to cell array

Multidimensional Array 多维数组

三维: row、column、layer(层)

声明方式:

1. $A\{\dots, \dots, \dots\} = \text{VALUE}$

2. 用cat串联三维数组

$A = \text{value}, B = \text{VALUE}, C = \text{cat}(\text{dim}, A, B)$ $\text{dim} = 1(\text{column}), 2(\text{row}), 3(\text{layer})$

创建两个三维数组，并沿第三个维度串联它们。所得数组中第一个和第二个维度的长度与输入数组中对应的长度匹配，而第三个维度会扩展。

```
A = rand(2,3,4);  
B = rand(2,3,5);  
C = cat(3,A,B);  
szC = size(C)
```

```
szC = 1×3  
2     3     9
```

reshape()

reshape(array_name,row2,column2)

`B = reshape(A , sz)` 使用大小向量 `sz` 重构 `A` 以定义 `size(B)`。例如，`reshape(A,[2,3])` 将 `A` 重构为一个 2×3 矩阵。`sz` 必须至少包含 2 个元素，`prod(sz)` 必须与 `numel(A)` 相同。

示例

`B = reshape(A , sz1,...,szN)` 将 `A` 重构为一个 `sz1 × ... × szN` 数组，其中 `sz1,...,szN` 指示每个维度的大小。可以指定 `[]` 的单个维度大小，以便自动计算维度大小，以使 `B` 中的元素数与 `A` 中的元素数相匹配。例如，如果 `A` 是一个 10×10 矩阵，则 `reshape(A,2,2,[])` 将 `A` 的 100 个元素重构为一个 2×2×25 数组。

下标赋值法：多维数组采用全下标方式

```
clear all;
clc;
A=[1 2 3;4 5 6;7 8 9];
B=reshape([10:18],3,3).';
C(:,:,1)=A;
C(:,:,2)=B;
C
```

重构的后的顺序是按照重构前先列后行排序的。

Checking Variable And Variable Status

<u>isinteger</u>	Determine if input is integer array
<u>islogical</u>	Determine if input is logical array
<u>isnan</u>	Detect an element that is not a number (NaN)
<u>isnumeric</u>	Determine if input is numeric array
<u>isprime</u>	Detect prime elements of array
<u>isreal</u>	Determine if all array elements are real numbers
<u>iscell</u>	Determine if input is cell array
<u>ischar</u>	Determine if input is character array
<u>isempty</u>	Determine if input is empty array
<u>isequal</u>	Determine if arrays are numerically equal
<u>isfloat</u>	Determine if input is floating-point array
<u>isglobal</u>	Determine if input is global variable
<u>ishandle</u>	Detect valid graphics object handles
<u>isinf</u>	Detect infinite elements of array

File Access

MAT——**import:load** **export:save**

XLS/XLSX——**import:load or xlsread** **export:save or xlswrite**

save() and load()

可将工作区里的结果数据存成两种形式：

- (1) 储存成matlab才可以读的形式：.MAT文件

- Save (all) workspace data to a file:

```
clear;  a = magic(4);  
save mydata1.mat  
save mydata2.mat -ascii
```

- Load data stored in a file:

```
load('mydata1.mat')  
load('mydata2.mat', '-ascii')
```

使用纯文字打开时要使用-ascii

Low-level File I/O Function

Open and close a file:

```
fid = fopen('[filename]', '[permission]');  
status = fclose(fid);
```

表 2-13 基本数学函数

函数符号	名称或功能	函数符号	名称或功能
abs	求绝对值或复数的模	log10	以 10 为底的对数
sqrt	开平方	round	四舍五入并取整
angle	求复数相角	fix	向最接近 0 方向取整
real	求复数实部	floor	向接近 $-\infty$ 方向取整
imag	求复数虚部	ceil	向接近 $+\infty$ 方向取整
conj	求复数的共轭	rem(a,b)	求 a/b 的有符号余数
exp	自然指数	mod(c,m)	求 c/m 的正余数
ln	以 e 为底的对数	sign	符号函数
log2	以 2 为底的对数		

ln应该为log

• 28 •

表 2-14 基本三角函数

函数符号	名称或功能	函数符号	名称或功能
sin	正弦	sinh	双曲正弦
cos	余弦	cosh	双曲余弦
tan	正切	tanh	双曲正切
asin	反正弦	asinh	反双曲正弦
acos	反余弦	acosh	反双曲余弦
atan	反正切	atanh	反双曲正切

比较字符串是否相等



`strcmp(s1,s2)`: 用来比较字符串s1和s2是否相等, 如果相等, 返回结果为1, 否则返回0。

`strncmp(s1,s2,n)`: 用来比较两个字符串前n个字符串是否相等, 如果相等, 返回结果为1, 否则返回0。

`strcmpi(s1,s2)`: 在忽略字母大小写前提下, 比较字符串s1和s2是否相等, 如果相等, 返回结果为1, 否则返回0。

`strnmpi(s1,s2)`: 在忽略字符串大小写前提下, 比较字符串前n个字符是否相等, 如果相等, 返回结果为1, 否则返回0。

字符串的查找与替换



`findstr(s1,s2)`: 返回短字符串在长字符串中的开始位置

`strrep(s1,s2,s3)`: 将字符串s1中的所有子字符串s2替换为字符串s3

字符串函数



`eval(string)` 作为一个MATLAB 命令求字符串的值

`eval(try,catch)`

`blanks(n)` 返回一个n 个零或空格的字符串

`deblank` 去掉字符串中后拖的空格

`feval` 求由字符串给定的函数值

`findstr` 从一个字符串内找出字符串

`isletter` 字母存在时返回真值

`isspace` 空格字符存在时返回真值

`isstr` 输入是一个字符串，返回真值

`lasterr` 返回上一个所产生MATLAB 错误的字符串

`strcmp` 字符串相同，返回真值

`strrep` 用一个字符串替换另一个字符串

`strtok` 在一个字符串里找出第一个标记