

MATLAB基本语法

变量

变量名

保留变量不适合做变量名

变量名不应当覆盖内置函数名

变量类型

数字型变量的显示格式

MATLAB命令行

使用MATLAB进行数字运算

使用MATLAB计算数学表达式

MATLAB内置的数学函数

使用MATLAB进行矩阵运算

定义矩阵

向终端输入矩阵

使用冒号运算符创建向量

定义特殊矩阵

矩阵的索引

矩阵的操作

操作矩阵的运算符

操作矩阵的函数

学习一门技术最好的方式就是阅读官方文档,可以查看[MATLAB官方文档](#)

MATLAB基本语法

变量

- MATLAB中的变量不需要声明.
- 使用 `=` 为变量赋值

变量名

- 与大多数编程语言相同,MATLAB中的变量名是大小写敏感的.
- 变量名只能由[`0~9`, `a~z`, `A~Z`, `_`]组成,且变量名不能以数字开头.

保留变量不适合做变量名

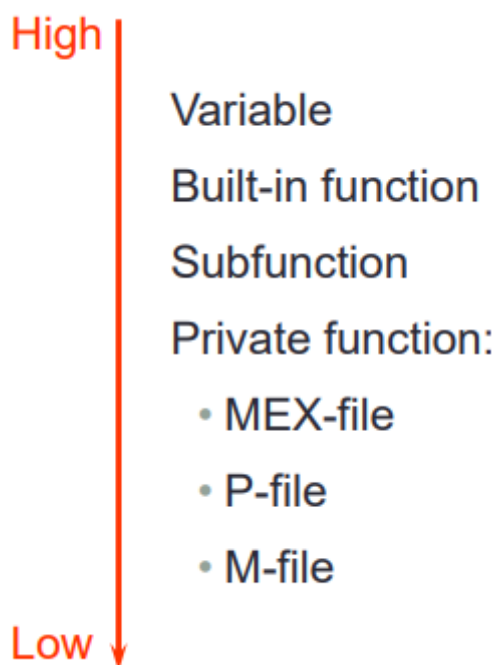
MATLAB中有一些变量有其具体意义,不适合用作变量名.

变量	意义
ans	上一句的运算的结果
i 和 j	复数算子
Inf	无穷 ∞
eps	浮点相对精度,即1.0到下一个浮点数之间的距离(值为2.2204e-16)
NaN	非数字
pi	圆周率 π

除此以外,使用 `iskeyword` 命令可以查看MATLAB语言所有的关键字,这些关键字也不允许被用作变量名.

变量名不应当覆盖内置函数名

在MATLAB中,变量的调用优先级(calling priority)高于函数,因此变量名不应该覆盖内置函数.



```
1 cos='This string.';
2 cos(8)           % 对字符串进行索引取值,得到'r'
```

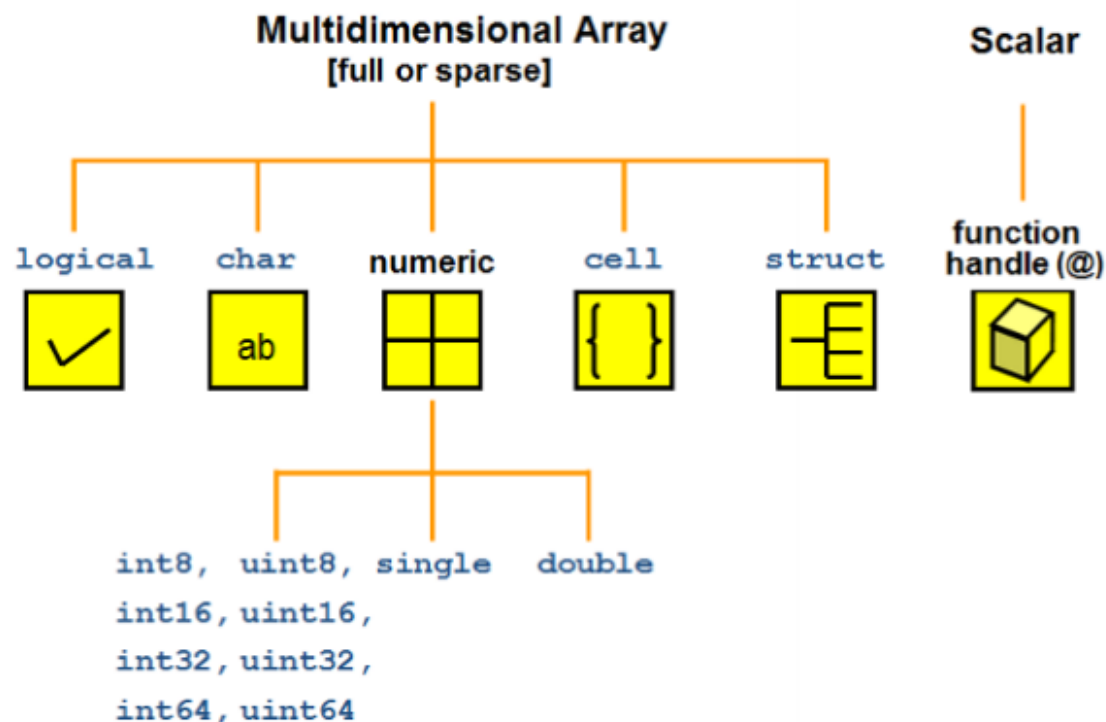
若某函数被变量名所覆盖,则调用 `clear <变量名>` 可以取消绑定在该函数名上的变量名

```
1 clear cos      % 清除绑定在cos上的变量
2 cos(8)         % 调用内置余弦函数运算得到-0.1455
```

`clear` 是一个比较危险的命令,因为该命令后若不加参数,则表示清除当前工作区内的所有变量.

变量类型

MATLAB中的变量类型有: `logical`, `char`, `numeric`, `cell`, `struct` 以及由他们组成的数组或矩阵.



数字型变量的显示格式

我们直接定义的数字型变量,默认是以 `double` 形式存储的.

我们可以通过 `format <显示格式>` 改变数字型变量的显示格式.

显示格式	说明	例子
<code>short</code>	短定点格式.显示小数点后4位	<code>3.1416</code>
<code>long</code>	长定点格式.对 <code>double</code> 类型变量显示小数点后15位,对 <code>float</code> 类型变量显示小数点后7位.	<code>3.141592653589793</code>
<code>shortE</code>	短科学计数法,显示小数点后4位.并带有科学计数法标记.	<code>3.1416e+00</code>
<code>longE</code>	长科学计数法.对 <code>double</code> 类型变量显示小数点后15位,对 <code>float</code> 类型变量显示小数点后7位.并带有科学计数法标记.	<code>3.141592653589793e+00</code>
<code>bank</code>	银行格式.显示小数点后2位.	<code>3.14</code>
<code>hex</code>	十六进制格式.	<code>400921fb54442d18</code>
<code>rat</code>	比例格式	<code>355/113</code>

MATLAB命令行

1. 使用行尾 ; 抑制输出: 在一行命令后使用 ; 抑制输出,否则运算结果将被显示在终端上.
2. 其他实用的命令:

命令	作用
<code>clc</code>	清除终端的输出
<code>clear</code>	清除当前工作区内所有变量
<code>who</code>	以简略格式显示工作区内所有变量
<code>whos</code>	以复杂格式显示工作区内所有变量

使用MATLAB进行数字运算

使用MATLAB计算数学表达式

- MATLAB常见运算符有: `+`, `-`, `*`, `/`, `^`.
- 数学表达式被计算后,其值被存入变量 `ans`.
- 运算的优先级规则:
 - 同等优先级下从左向右运算.
 - 优先级顺序(从高到低)
 - 括号 `()`
 - 乘方 `^`
 - 乘除法 `*`, `/`
 - 加减法 `+`, `-`

下面例子演示了数学表达式求值:

待求数学表达式	MATLAB命令
$\cos\left(\sqrt{\frac{(1+2+3+4)^3}{5}}\right)$	<code>cos(sqrt((1+2+3+4)^3/5))</code>
$\sin(\sqrt{\pi}) + \ln(\tan(1))$	<code>sin(sqrt(pi))+log(tan(1))</code>
$2^{3.5 \times 1.7}$	<code>2^(3.5*1.7)</code>
$e^{\sin(10)}$	<code>exp(sin(10))</code>

MATLAB内置的数学函数

- [MATLAB内置的算数运算函数](#)
 - 基本运算:
 - 加: `+`, `sum`, `cumsum`, `movsum`
 - 减: `-`, `diff`
 - 乘: `.*`, `*`, `prod`, `cumprod`
 - 除: `./`, `.\`, `/`, `\`
 - 乘方: `.^`, `^`
 - 取模运算: `mod`, `rem`, `idivide`, `ceil`, `fix`, `floor`, `round`
- [MATLAB内置的三角运算函数](#)
 - 正弦: `sin`, `sind`, `sinpi`, `asin`, `asind`, `sinh`, `asinh`

- 余弦: `cos`, `cosd`, `cospi`, `acos`, `acosd`, `cosh`, `acosh`
- 正切: `tan`, `tand`, `atan`, `atand`, `atan2`, `atan2d`, `tanh`, `atanh`
- 余割: `csc`, `cscd`, `acsc`, `acscd`, `csch`, `acsch`
- 正割: `sec`, `secd`, `asec`, `asecd`, `sech`, `asech`
- 余切: `cot`, `cotd`, `acot`, `acotd`, `coth`, `acoth`
- 斜边: `hypot`
- 转换: `deg2rad`, `rad2deg`, `cart2pol`, `cart2sph`, `pol2cart`, `sph2cart`
- [MATLAB内置的指数对数函数](#):
`exp`, `expm1`, `log`, `log10`, `log1p`, `log2`, `nextpow2`, `nthroot`, `pow2`, `reallog`, `realpow`, `realsqrt`, `sqrt`
- [MATLAB内置的复函数](#):
`abs`, `angle`, `complex`, `conj`, `cp1xpair`, `i`, `imag`, `isreal`, `j`, `real`, `sign`, `unwrap`

使用MATLAB进行矩阵运算

定义矩阵

向终端输入矩阵

在MATLAB中,使用 `[]` 将待输入的矩阵内容括起来,使用空格 或逗号 , 分隔行内变量,使用 ; 分隔每一行. 下面例子演示了矩阵的定义:

MATLAB命令	得到的矩阵
<code>[1 2 3 4]</code>	$\begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}$
<code>[1; 2; 3; 4]</code>	$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$
<code>[1 21 6; 5 17 9; 31 2 7]</code>	$\begin{bmatrix} 1 & 21 & 6 \\ 5 & 17 & 9 \\ 31 & 2 & 7 \end{bmatrix}$

使用冒号运算符创建向量

使用冒号运算符 `:` 可以创建一个长向量,其语法如下:

冒号表达式	得到的结果
<code>j:k</code>	$[j, \quad j + 1, \quad j + 2, \quad \dots, \quad j + m]$
<code>j:i:k</code>	$[j, \quad j + i, \quad j + 2i, \quad \dots, \quad j + m * i]$

下面例子演示了冒号运算符的使用:

MATLAB语句	得到的结果
<code>1:5</code>	<code>[1 2 3 4 5]</code>
<code>1:2:5</code>	<code>[1 3 5]</code>
<code>[1:5; 2:3:15; -2:0.5:0]</code>	$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 8 & 11 & 14 \\ -2 & -1.5 & -1 & -0.5 & 0 \end{bmatrix}$
<code>'a':2:'z'</code>	<code>'acegikmoqsuwyz'</code>

定义特殊矩阵

下列命令可以定义特殊矩阵

命令	得到的结果
<code>eye(n)</code>	得到一个 $n \times n$ 的单位矩阵
<code>zeros(n1, n2)</code>	得到一个 $n_1 \times n_2$ 的全0矩阵
<code>ones(n1, n2)</code>	得到一个 $n_1 \times n_2$ 的全1矩阵
<code>diag(vector)</code>	得到一个以向量 <code>vector</code> 中内容为对角线的对角矩阵

矩阵的索引

- MATLAB中的矩阵是以列先序存储的.且索引下标从1开始.
- 矩阵有两种索引方式: 按一维索引和按二维索引.对于一个一般的矩阵,其索引顺序如下:

$$\begin{bmatrix} 1 \text{或} (1, 1) & 4 \text{或} (1, 2) & 7 \text{或} (1, 3) \\ 2 \text{或} (2, 1) & 5 \text{或} (2, 2) & 8 \text{或} (2, 3) \\ 3 \text{或} (3, 1) & 6 \text{或} (3, 2) & 9 \text{或} (3, 3) \end{bmatrix}$$

- 矩阵的索引可以使用冒号`:`,表示选取所有行或所有列.
- 矩阵的索引可以是一个或两个向量,表示选中向量内的所有行或所有列.

下面例子演示了矩阵索引的规则:

原矩阵	索引	得到的结果	解释
$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$	<code>A(8)</code>	6	取矩阵第8个元素
$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$	<code>A([1 3 5])</code>	<code>[1 7 5]</code>	分别取矩阵第 <code>[1 3 5]</code> 个元素放入新矩阵的对应位置
$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$	<code>A([1 2; 3 4])</code>	$\begin{bmatrix} 1 & 4 \\ 7 & 2 \end{bmatrix}$	分别取矩阵第 <code>[1 2; 3 4]</code> 个元素放入新矩阵的对应位置
$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$	<code>A(3,2)</code>	8	取矩阵第3行第2列的元素

原矩阵	索引	得到的结果	解释
$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$	<code>A([1,2], :)</code>	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	取矩阵第 [1 2] 行和所有列的交叉项
$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$	<code>A([1 3], [1 2])</code>	$\begin{bmatrix} 1 & 2 \\ 7 & 8 \end{bmatrix}$	取矩阵第 [1 3] 行和第 [1 2] 列的交叉项

矩阵的操作

操作矩阵的运算符

运算符	操作	形式	例子
<code>+</code>	矩阵与向量相加	<code>A+b</code>	<code>[6 3] + 2 = [8 5]</code>
<code>-</code>	矩阵与向量相减	<code>A-b</code>	<code>[6 3] - 2 = [4 1]</code>
<code>+</code>	矩阵与矩阵对应位置相加	<code>A+B</code>	<code>[6 3] + [4 8] = [10 11]</code>
<code>-</code>	矩阵与矩阵对应位置相减	<code>A-B</code>	<code>[6 3] - [4 8] = [2 -5]</code>
<code>*</code>	矩阵与矩阵相乘	<code>A*B</code>	<code>[6 3] * [4 8]' = 48</code>
<code>.*</code>	矩阵与矩阵对应位置相乘	<code>A.*B</code>	<code>[6 3] * [4 8] = [24 24]</code>
<code>/</code>	矩阵与矩阵右除(等价于 <code>A*inv(B)</code>)	<code>A/B</code>	<code>[6 3] / [4 8] = 0.6</code>
<code>\</code>	矩阵与矩阵左除(等价于 <code>inv(A)*B</code>)	<code>A\B</code>	<code>[6 3] / [4 8] = [0.06667 1.3333; 0 0]</code>
<code>./</code>	矩阵与矩阵对应位置右除	<code>A./B</code>	<code>[6 3] ./ [4 8] = [1.5 0.375]</code>
<code>.\</code>	矩阵与矩阵对应位置左除	<code>A.\B</code>	<code>[6 3] .\ [4 8] = [0.6667 2.6667]</code>
<code>^</code>	矩阵与向量乘方	<code>A^b</code>	<code>[1 2; 3 4]^3 = [37 54; 81 118]</code>
<code>.^</code>	矩阵与矩阵对应位置乘方	<code>A.^B</code>	<code>[1 2; 3 4].^[1 2; 3 4] = [1 4; 27 256]</code>

操作矩阵的函数

下面对矩阵

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 6 \\ 7 & 0 & 9 \end{bmatrix}$$

进行操作以演示操作矩阵的常见函数

函数	作用	例子	结果
<code>max(A, [], dim)</code>	获取矩阵 A 的 dim 维度上的最大值	<code>max(A, [], 1)</code>	$\begin{bmatrix} 7 & 5 & 9 \end{bmatrix}$
<code>min(A, [], dim)</code>	获取矩阵 A 的 dim 维度上的最小值	<code>min(A, [], 2)</code>	$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$
<code>sum(A, dim)</code>	获取矩阵 A 的 dim 维度上的和	<code>sum(A, 1)</code>	$\begin{bmatrix} 8 & 7 & 18 \end{bmatrix}$
<code>mean(A, dim)</code>	获取矩阵 A 的 dim 维度上的平均值	<code>mean(A, 1)</code>	$\begin{bmatrix} 2.6667 & 2.3333 & 6.0000 \end{bmatrix}$
<code>sort(A, dim, direction)</code>	获取矩阵 A 的 dim 维度上按 direction 顺序排序结果	<code>sort(A, 1, 'descend')</code>	$\begin{bmatrix} 7 & 5 & 9 \\ 1 & 2 & 6 \\ 0 & 0 & 3 \end{bmatrix}$
<code>sortrows(A, column)</code>	获取矩阵 A 按照每行第 column 个元素升序排序结果.	<code>sortrows(A, 2)</code>	$\begin{bmatrix} 7 & 0 & 9 \\ 1 & 2 & 3 \\ 0 & 5 & 6 \end{bmatrix}$
<code>size(A, dim)</code>	获取矩阵 A 的 dim 维度上的形状.若不指定 dim,则返回整个矩阵的形状.	<code>size(A)</code>	$\begin{bmatrix} 3 & 3 \end{bmatrix}$
<code>find(A, n)</code>	获取矩阵 A 前 n 个非零元素的索引	<code>find(A, 5)</code>	$\begin{bmatrix} 1 & 3 & 4 & 5 & 7 \end{bmatrix}$

对于上面这些函数,除第一个参数以外,其它参数都是可选的.