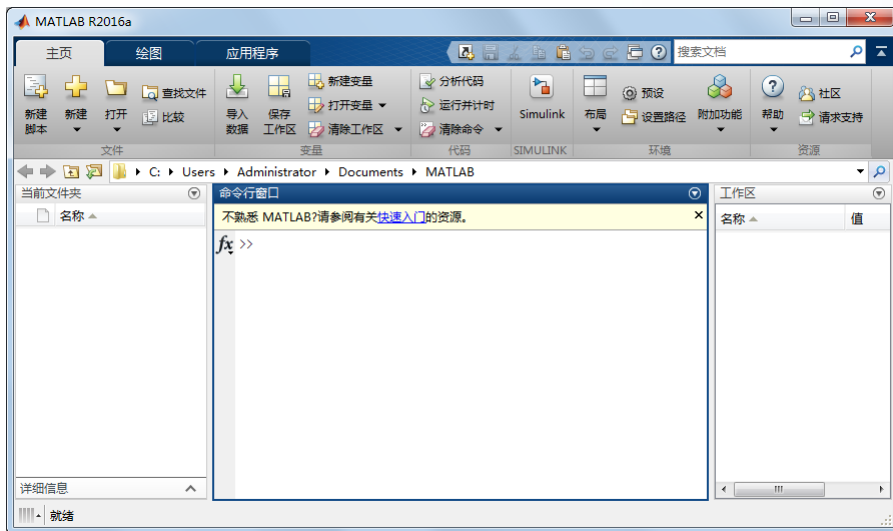


1.1 MATLAB系统环境

- ❑ MATLAB操作界面的组成
- ❑ MATLAB的搜索路径

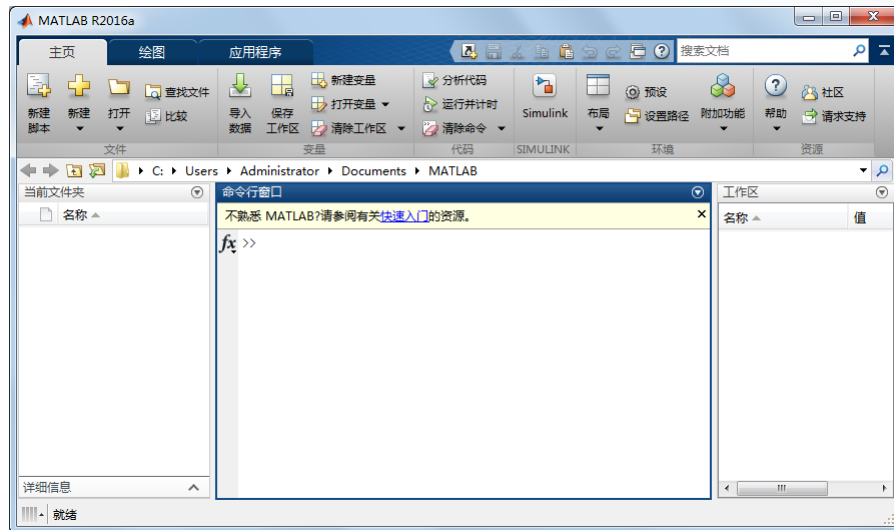
1. MATLAB操作界面的组成

- ❑ MATLAB主窗口
- ❑ 命令行窗口
- ❑ 当前文件夹窗口
- ❑ 工作区窗口



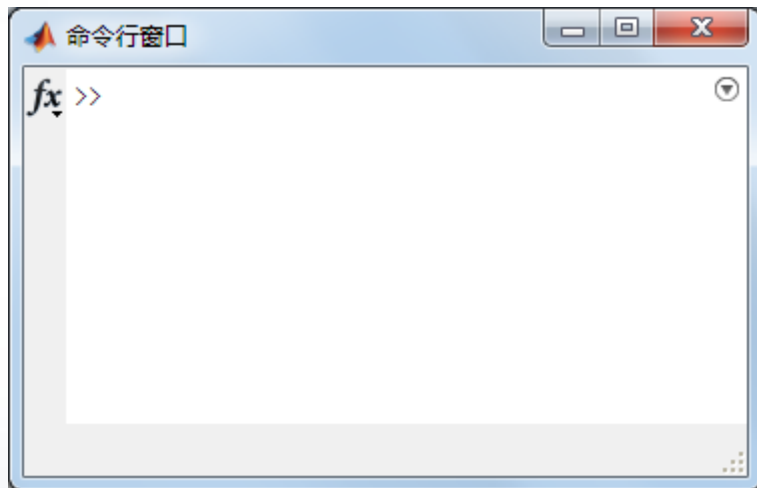
(1) MATLAB主窗口

- ❑ 功能区
- ❑ 快速访问工具栏
- ❑ 当前文件夹工具栏

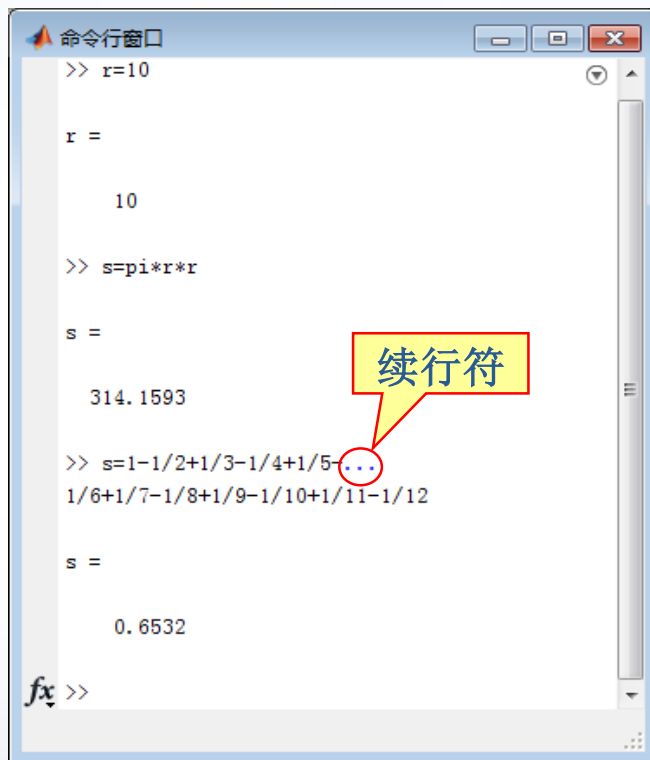


(2) 命令行窗口

命令行窗口用于输入命令并显示命令的执行结果。



(2) 命令行窗口



The screenshot shows the MATLAB Command Window with the following content:

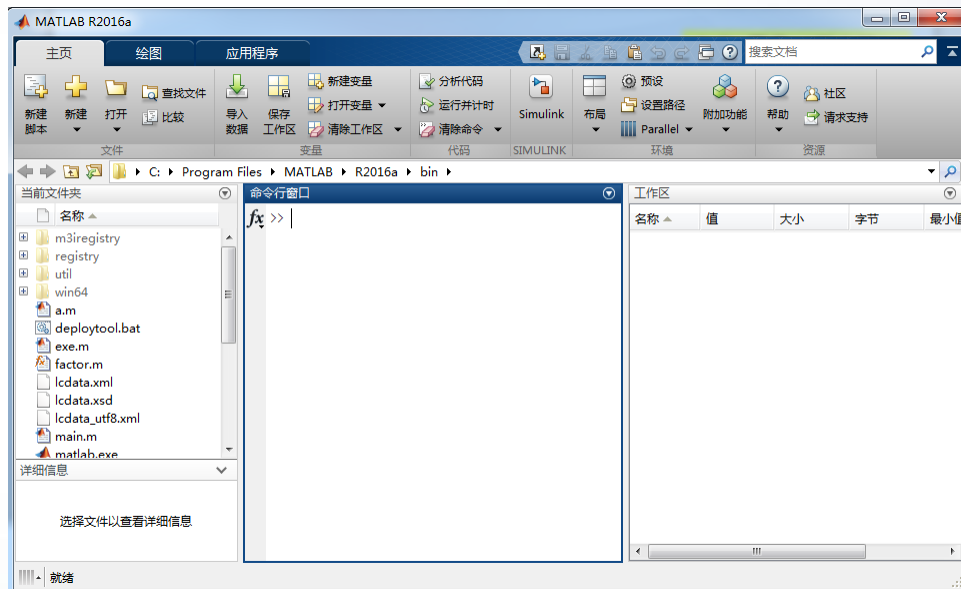
```
>> r=10  
  
r =  
  
    10  
  
>> s=pi*r*r  
  
s =  
  
    314.1593  
  
>> s=1-1/2+1/3-1/4+1/5-...  
    1/6+1/7-1/8+1/9-1/10+1/11-1/12  
  
s =  
  
    0.6532  
  
fx >>
```

A red circle highlights the ellipsis (...) in the input line, and a yellow callout box with the text "续行符" (Continuation Character) points to it.

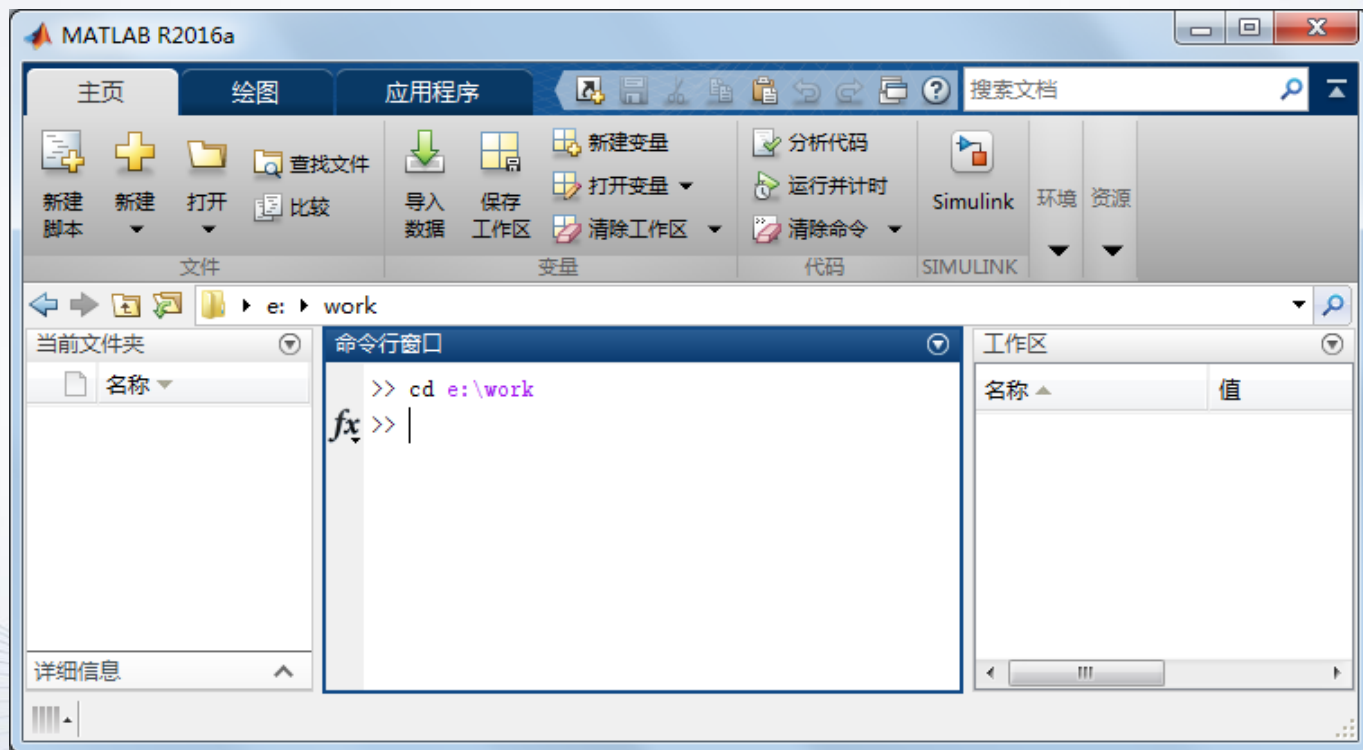
(3) 当前文件夹窗口

如何设置当前文件夹呢？

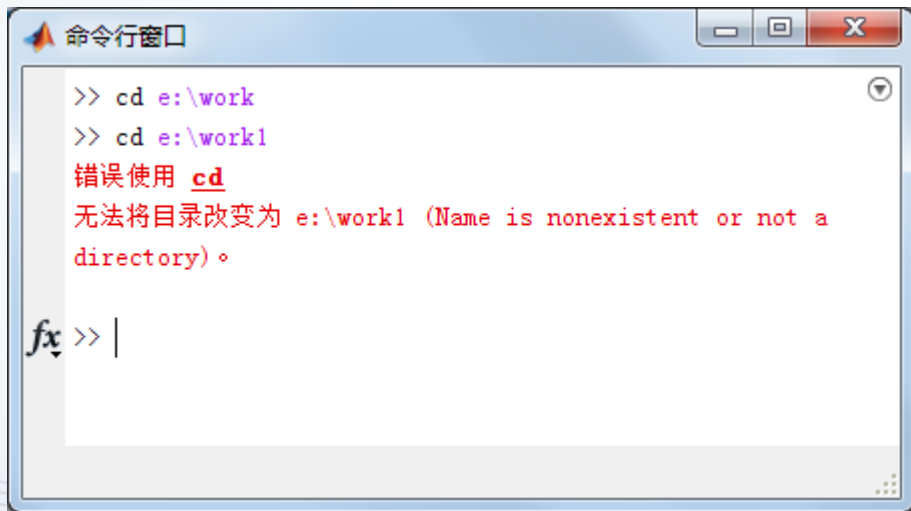
- ❑ 在当前文件夹工具栏或当前文件夹窗口中选择某文件夹为当前文件夹
- ❑ 使用cd命令



(3) 当前文件夹窗口



(3) 当前文件夹窗口

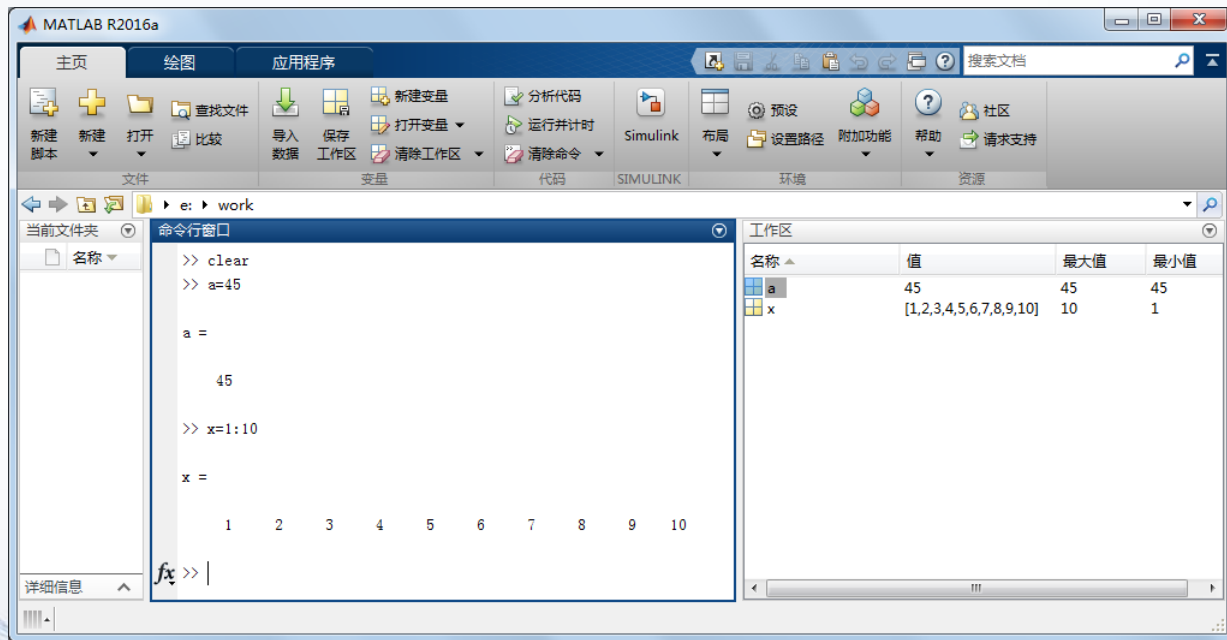


命令行窗口

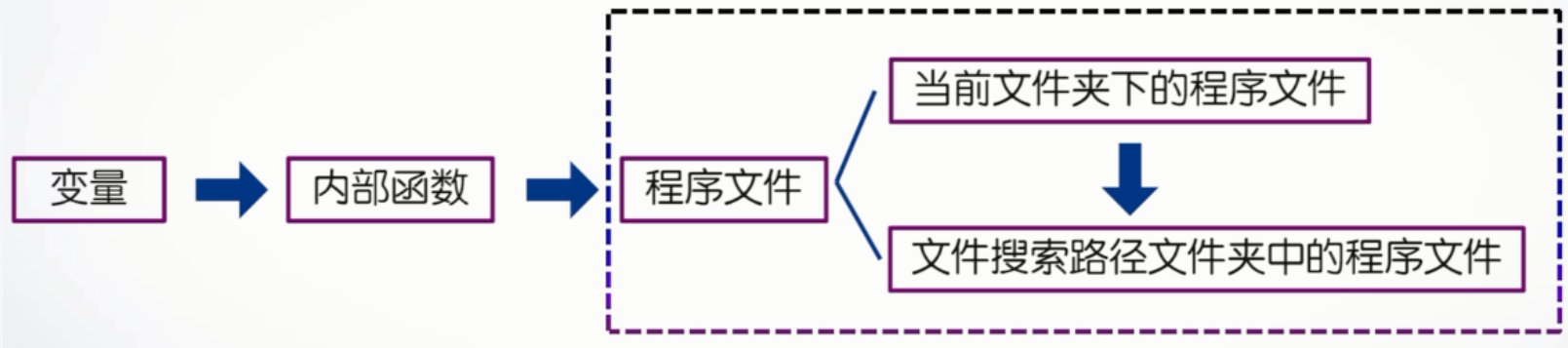
```
>> cd e:\work  
>> cd e:\work1  
错误使用 cd  
无法将目录改变为 e:\work1 (Name is nonexistent or not a  
directory)。  
fx >> |
```

注意：一定要先建立文件夹，再将这个文件夹设为当前文件夹。

(4) 工作区窗口



2. MATLAB的搜索路径



命令对象操作举例。

```
>> clear
```

```
>> x=sin(1)
```

```
x =
```

```
    0.8415
```

```
>> sin=[1, 2, 3, 4, 5]
```

```
sin =
```

```
    1    2    3    4
```

```
    5
```

```
>> x=sin(1)
```

```
x =
```

```
    1
```



如果这时要调用sin函数，该如何操作？

可以在MATLAB工作区窗口删除定义的sin变量，那么再用sin就代表内部函数了。

□ 用path命令设置文件搜索路径。例如：

```
>> path(path, 'e:\work')
```

□ 用对话框设置文件搜索路径。





- ❑ 如果在当前文件夹和搜索路径文件夹下建立了一个同名的M文件，那么在命令行窗口输入文件名时，执行哪个文件？
- ❑ 如果用户建立的文件既没有保存在当前文件夹下，也没有保存在文件搜索路径中，那么在命令行窗口输入文件名时，会出现什么信息？

第一个问题根据MATLAB的搜索路径，程序文件的搜索顺序是首先在当前文件夹中搜索，然后再在文件搜索路径中的文件夹中搜索，所以执行的是当前文件夹下的M文件。第二个问题，同样根据MATLAB的搜索路径，如果用户建立的文件既没有保存在当前文件夹下，也没有保存在文件搜索路径中，那么MATLAB就找不到这个文件，系统会给出错误提示信息。

1.2 MATLAB数值数据

- 数值数据类型的分类
- 数值数据的输出格式
- 常用数学函数

1. 数值数据类型的分类

- 整型
- 浮点型
- 复数型

1. 数值数据类型的分类

(1) 整型

- ❑ 无符号整数：无符号8位整数、无符号16位整数、无符号32位整数、无符号64位整数。
- ❑ 带符号整数：带符号8位整数、带符号16位整数、带符号32位整数、带符号64位整数。

1. 数值数据类型的分类

(1) 整型

- ❑ 无符号8位整数数据范围：00000000~11111111 ($0 \sim 2^8 - 1$)。
- ❑ 带符号8位整数数据范围：10000000~01111111 ($-2^7 \sim 2^7 - 1$)。

1. 数值数据类型的分类

(1) 整型

```
>> x=int8(129)
```

```
x =
```

```
127
```

```
>> x=uint8(129)
```

```
x =
```

```
129
```

带符号8位整型
数据的最大值
是127

(2) 浮点型

- ❑ `single`函数：将其他类型的数据转换为单精度型。
- ❑ `double`函数：将其他类型的数据转换为双精度型。

```
>> class(4)
```

```
ans =
```

```
double
```

```
>> class(single(4))
```

```
ans =
```

```
single
```


(3) 复型

复型数据包括实部和虚部两个部分，实部和虚部默认为双精度型，虚数单位用i或j来表示。

❑ real函数：求复数的实部

❑ imag函数：求复数的虚部

```
>> 6+5i
```

```
ans =
```

```
6.0000 + 5.0000i
```

```
>> 6+5j
```

```
ans =
```

```
6.0000 + 5.0000i
```

2. 数值数据的输出格式

format命令的格式:

format 格式符

```
>> format long
```

```
>> 50/3
```

```
ans =
```

```
16.666666666666668
```

```
>> format
```

```
>> 50/3
```

```
ans =
```

```
16.6667
```

注意: format命令只影响数据输出格式, 而不影响数据的计算和存储。

3. 常用数学函数

(1) 函数的调用格式为:

函数名(函数自变量的值)

```
>> A=[4, 2;3, 6]
```

```
A =
```

```
     4     2
```

```
     3     6
```

```
>> B=exp(A)
```

```
B =
```

```
54.5982    7.3891
```

```
20.0855  403.4288
```

函数在运算时是将函数逐项作用于矩阵的每个元素上，所以最后运算的结果就是一个与自变量同型的矩阵。

(2) 常用函数的应用

① 三角函数有以弧度为单位的函数和以角度为单位的函数，如果是以角度为单位的函数就在函数名后面加“d”，以示区别。

```
>> sin(pi/2)
```

```
ans =
```

```
1
```

```
>> sind(90)
```

```
ans =
```

```
1
```

② abs函数可以求实数的绝对值、复数的模、字符串的ASCII码值。

```
>> abs(-4)
```

```
ans =
```

```
4
```

```
>> abs(3+4i)
```

```
ans =
```

```
5
```

```
>> abs('a')
```

```
ans =
```

```
97
```

③用于取整的函数有fix、floor、ceil、round。

```
>> round(4.7)
```

```
ans =
```

```
5
```

```
>> fix(-3.2)
```

```
ans =
```

```
-3
```

```
>> floor(3.6)
```

```
ans =
```

```
3
```

```
>> ceil(-3.8)
```

```
ans =
```

```
-3
```

round函数是按照四舍五入的规则来取整。

ceil是向上取整，取大于等于这个数的第一个整数。

floor是向下取整，取小于等于这个数的第一个整数。

fix是固定取靠近0的那个整数，也就是舍去小数取整。

④函数应用举例。

□ 分别求一个三位正整数的个位数字、十位数字和百位数字。

```
>> m=345;
```

```
>> m1=rem(m, 10)
```

```
m1 =
```

```
5
```

```
>> m2=rem(fix(m/10), 10)
```

```
m2 =
```

```
4
```

```
>> m3=fix(m/100)
```

```
m3 =
```

```
3
```

□ 求[1, 100]区间的所有素数。

```
>> x=1:100;  
>> k=isprime(x);  
>> k1=find(k);  
>> p=x(k1)
```

p =

1 至 13 列

2 3 5 7 11 13 17 19 23 29 31 37 41

14 至 25 列

43 47 53 59 61 67 71 73 79 83 89 97

1.3 变量及其操作

- ❑ 变量与赋值语句
- ❑ 预定义变量
- ❑ 变量的管理



1. 变量与赋值语句

在MATLAB中，变量名是以字母开头，后接字母、数字或下画线的字符序列，最多63个字符。

```
>> 12my=10
12my=10
↑
错误：不应为 MATLAB 表达式。

>> _my=10
_my=10
↑
错误：输入字符不是 MATLAB 语句或表达式中的有效字符。

fx >>
```

- ❑ 变量名区分字母的大小写。
- ❑ 标准函数名以及命令名一般用小写字母。

赋值语句两种格式:

- 变量=表达式
- 表达式

计算表达式 $\frac{5+\cos 47^\circ}{1+|x-y|}$ 的值，并将结果赋给变量z，然后显示计算结果。

其中， $x = \sqrt{7} - 2i, y = e^{\frac{\pi}{2}}$ 。

```
>> x=sqrt(7)-2i;  
>> y=exp(pi/2);  
>> z=(5+cosd(47))/(1+abs(x-y))
```

z =

1.4395



2. 预定义变量

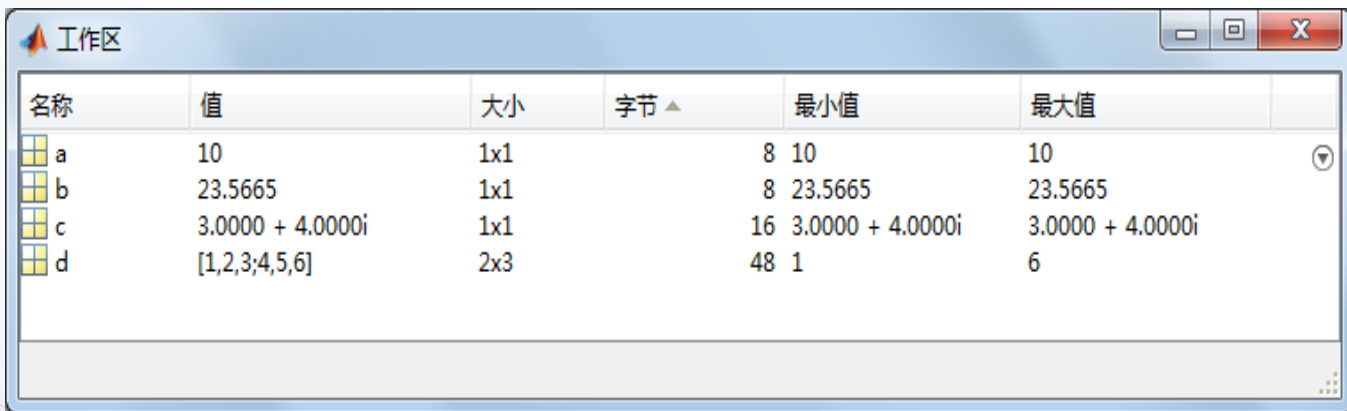
预定义变量是在MATLAB工作空间中驻留，由系统本身定义的变量。

- `ans`是默认赋值变量
- `i`和`j`代表虚数单位
- `pi`代表圆周率
- `NaN`代表非数



3. 变量的管理

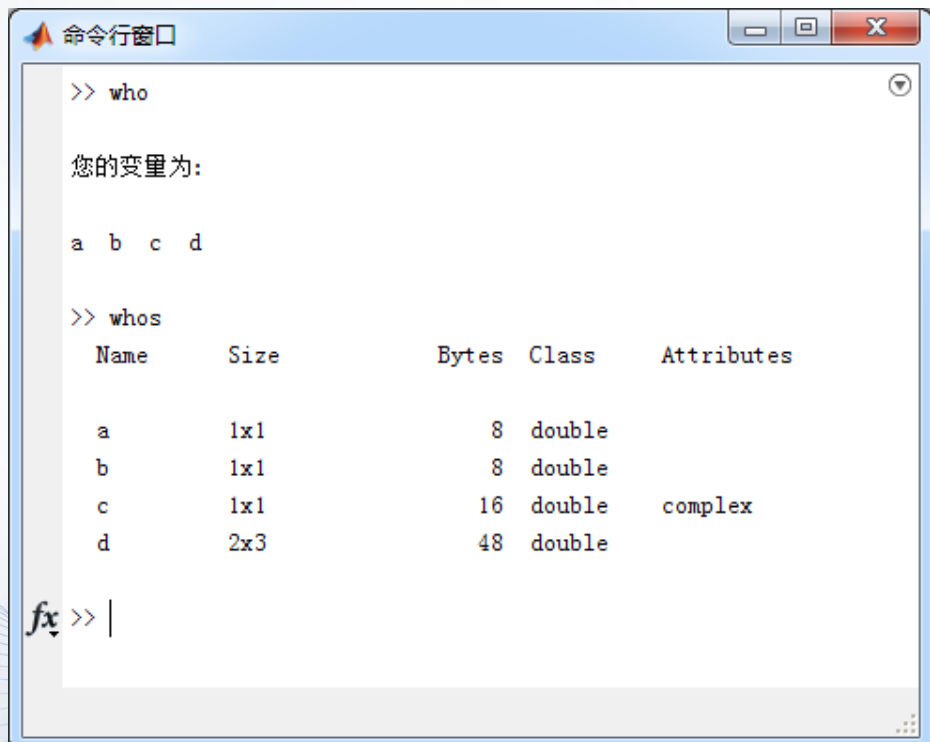
(1) 内存变量的删除与修改



名称	值	大小	字节 ▲	最小值	最大值
a	10	1x1	8	10	10
b	23.5665	1x1	8	23.5665	23.5665
c	3.0000 + 4.0000i	1x1	16	3.0000 + 4.0000i	3.0000 + 4.0000i
d	[1,2,3;4,5,6]	2x3	48	1	6



who命令与whos命令



```
命令窗口
>> who

您的变量为:

a b c d

>> whos

Name      Size      Bytes  Class  Attributes
-----
a         1x1         8  double
b         1x1         8  double
c         1x1        16  double  complex
d         2x3        48  double
```

fx >> |

(2) 内存变量文件

用于保存MATLAB工作区变量的文件叫做内存变量文件，其扩展名为.mat，也叫MAT文件。

□ save命令：创建内存变量文件。

□ load命令：装入内存变量文件。

```
>> save mydata a x
```

```
>> load mydata
```


1.4 MATLAB矩阵的表示

- 矩阵的建立
- 冒号表达式
- 结构矩阵和单元矩阵

1. 矩阵的建立

(1) 利用直接输入法建立矩阵:将矩阵的元素用**中括号**括起来,按矩阵行的顺序输入各元素,同一行的各元素之间用**逗号或空格**分隔,不同行的元素之间用**分号**分隔。

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

```
A =
```

1	2	3
4	5	6
7	8	9

(2) 利用已建好的矩阵建立更大的矩阵：
一个大矩阵可以由已经建立好的小矩阵拼接而成。

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

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

```
>> C=[A, B;B, A]
```

C =

1	2	3	-1	-2	-3
4	5	6	-4	-5	-6
7	8	9	-7	-8	-9
-1	-2	-3	1	2	3
-4	-5	-6	4	5	6
-7	-8	-9	7	8	9



可以用实部矩阵和虚部矩阵构成复数矩阵

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

```
>>C=[6, 7, 8;9, 10, 11];
```

```
>>A=B+i*C
```

A =

1.0000 + 6.0000i	2.0000 + 7.0000i	3.0000 + 8.0000i
4.0000 + 9.0000i	5.0000 + 10.0000i	6.0000 + 11.0000i

2. 冒号表达式

格式:



省略步长e2，则步长为1。例如，
 $t=0:5$ 与 $t=0:1:5$ 等价。

```
>> t=0:1:5
```

```
t =
```

```
0     1     2     3     4     5
```

格式:

linspace(a,b,n)

第1个
元素

元素
总数

最后
元素

当n省略时，自动产生100个元素。

```
>> x=linspace(0,pi,6)
```

```
x =
```

0

0.6283

1.2566

1.8850

2.5133

3.1416

3. 结构矩阵和单元矩阵

(1) 结构矩阵

由结构数据构成的矩阵就是结构矩阵，结构矩阵里的每个元素就是结构数据类型。

格式为：

结构矩阵元素. 成员名=表达式

```
>> a(1).x1=10; a(1).x2='liu'; a(1).x3=[11, 21;34, 78];  
>> a(2).x1=12; a(2).x2='wang'; a(2).x3=[34, 191;27, 578];  
>> a(3).x1=14; a(3).x2='cai'; a(3).x3=[13, 890;67, 231];
```

(2) 单元矩阵

建立单元矩阵和一般矩阵相似，直接输入就可以了，只是单元矩阵元素用大括号括起来。

```
>> b= {10,'liu',[11,21;34,78];12,'wang',[34,191;27,578];...  
14,'cai',[13,890;67,231]}
```

```
b =
```

[10]	'liu'	[2x2 double]
[12]	'wang'	[2x2 double]
[14]	'cai'	[2x2 double]

1.5 矩阵元素的引用

- 矩阵元素的引用方式
- 利用冒号表达式获得子矩阵
- 利用空矩阵删除矩阵的元素
- 改变矩阵的形状

1. 矩阵元素的引用方式

(1) 通过下标来引用矩阵的元素

$A(3, 2)$ 表示A矩阵第3行第2列的元素。

```
>> A(3, 2)=200
```



1. 矩阵元素的引用方式

(1) 通过下标来引用矩阵的元素

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

```
>> A(4, 5)=10
```

```
A =
```

1	2	3	0	0
4	5	6	0	0
0	0	0	0	0
0	0	0	0	10

注意：如果给出的行下标或列下标大于原来矩阵的行数和列数，那么MATLAB将自动扩展原来的矩阵，并将扩展后没有赋值的矩阵元素置为0。

(2) 通过序号来引用

- ❑ 在MATLAB中，矩阵元素按列存储，即首先存储矩阵的第一列元素，然后存储第二列元素， \dots ，一直到矩阵的最后一列元素。
- ❑ 矩阵元素的序号就是矩阵元素在内存中的排列顺序。


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

```
A =
```

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

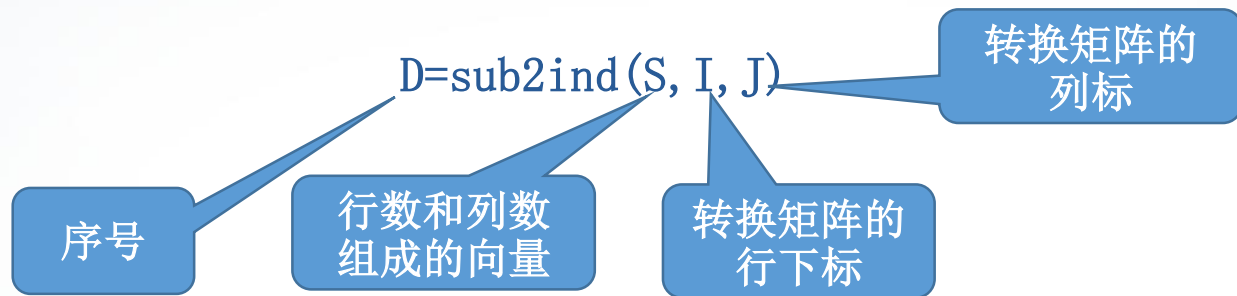
```
>> A(3)
```

```
ans =
```

```
     2
```

序号与下标是一一对应的，以 $m \times n$ 矩阵A为例，矩阵元素 $A(i, j)$ 的序号为 $(j-1) \times m + i$

sub2ind函数：将矩阵中指定元素的行、列下标转换成存储的序号。调用格式为：





sub2ind函数举例。

```
>> A=[1:3;4:6]
```

```
A =
```

```
    1    2    3
```

```
    4    5    6
```

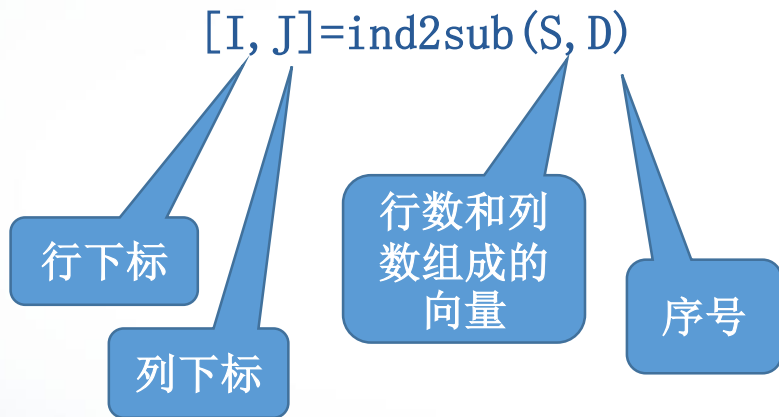
```
>> D=sub2ind(size(A), [1,2;2,2], [1,1;3,2])
```

```
D =
```

```
    1    2
```

```
    6    4
```

ind2sub函数：将把矩阵元素的序号转换成对应的下标，其调用格式为：



ind2sub函数举例。

```
>> [I,J]=ind2sub([3,3],[1,3,5])
```

```
I =
```

```
     1     3     2
```

```
J =
```

```
     1     1     2
```

2. 利用冒号表达式获得子矩阵

子矩阵是指由矩阵中的一部分元素构成的矩阵。

$A(i,:)$	第 <i>i</i> 行的全部元素
$A(:,j)$	第 <i>j</i> 列的全部元素
$A(i:i+m,k:k+m)$	第 <i>i</i> ~ <i>i</i> + <i>m</i> 行内且在第 <i>k</i> ~ <i>k</i> + <i>m</i> 列中的所有元素
$A(i:i+m,:)$	第 <i>i</i> ~ <i>i</i> + <i>m</i> 行的全部元素



```
>> A=[1, 2, 3, 4, 5;6, 7, 8, 9, 10;11, 12, 13, 14, 15]
```

```
A =
```

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

```
>> A(1:2, :)
```

```
ans =
```

```
     1     2     3     4     5
     6     7     8     9    10
```

```
>> A(2:3, 1:2:5)
```

```
ans =
```

```
     6     8    10
    11    13    15
```

end运算符：表示某一维的末尾元素下标。

```
>>A=[1, 2, 3, 4, 5;6, 7, 8, 9, 10;11, 12, 13, 14, 15;16, 17, 18, 19, 20];
```

```
>>A(end, :)
```

```
ans =
```

```
    16    17    18    19    20
```

```
>> A([1, 4], 3:end)
```

```
ans =
```

```
     3     4     5  
    18    19    20
```

3. 利用空矩阵删除矩阵的元素

空矩阵是指没有任何元素的矩阵。

```
>> x=[]
```

```
x =
```

```
 []
```

X是一个空矩阵。



```
>> A=[1, 2, 3, 0, 0;7, 0, 9, 2, 6;1, 4, -1, 1, 8]
```

```
A =
```

1	2	3	0	0
7	0	9	2	6
1	4	-1	1	8

```
>> A(:, [2, 4])=[]
```

```
A =
```

1	3	0
7	9	6
1	-1	8

4. 改变矩阵的形状

`reshape(A, m, n)`：在矩阵总元素保持不变的前提下，将矩阵A重新排成 $m \times n$ 的二维矩阵。

注意：`reshape`函数只是改变原矩阵的行数和列数，但并不改变原矩阵元素个数及其存储顺序。

```
>> x=[23, 45, 65, 34, 65, 34, 98, 45, 78, 65, 43, 76];
```

```
>> y=reshape(x, 3, 4)
```

y

23	34	98	65
45	65	45	43
65	34	78	76

$A(:)$: 将矩阵A的每一列元素堆叠起来, 成为一个列向量。

```
>> A = [-45, 65, 71; 27, 35, 91]
```

```
A =
```

```
    -45    65    71  
     27    35    91
```

```
>> B=A(:)
```

```
B =
```

```
    -45  
     27  
     65  
     35  
     71  
     91
```

$A(:)$ 等价于 $\text{reshape}(A, 6, 1)$ 。

1.6 MATLAB基本运算

- 算术运算
- 关系运算
- 逻辑运算

1. 算术运算

(1) 基本算术运算

基本算术运算符：+ (加)、- (减)、* (乘)、/ (右除)、\ (左除)、[^] (乘方)。

❑ MATLAB的算术运算是在矩阵意义下进行的。

❑ 单个数据的算术运算只是矩阵运算的一种特例。

注意，MATLAB的运算是在矩阵意义下进行的，单个数据的算术运算只是矩阵运算的一种特例。

加减运算

- ❑ 若两矩阵同型，则运算时两矩阵的相应元素相加减。
- ❑ 若两矩阵不同型，则MATLAB将给出错误信息。
- ❑ 一个标量也可以和矩阵进行加减运算，这时把标量和矩阵的每一个元素进行加减运算。

乘法运算

- ❑ 矩阵A和B进行乘法运算，要求A的列数与B的行数相等，此时则称A、B矩阵是可乘的，或称A和B两矩阵维数和大小相容。
- ❑ 如果两者的维数或大小不相容，则将给出错误信息，提示用户两个矩阵是不可乘的。

除法运算

在MATLAB中，有两种矩阵除法运算：右除/和左除\。

如果A矩阵是非奇异方阵，则 B/A 等效于 $B \cdot \text{inv}(A)$ ， $A \backslash B$ 等效于 $\text{inv}(A) \cdot B$ 。



```
>> A=[1, 2, 3;4, 2, 6;7, 4, 9];  
>> B=[4, 3, 2;7, 5, 1;12, 7, 92];  
>> C1=B/A  
C1 =  
    -0.1667    -3.3333     2.5000  
    -0.8333    -7.6667     5.5000  
    12.8333    63.6667   -36.5000  
>> C2=A\B  
C2 =  
     0.5000    -0.5000    44.5000  
     1.0000     0.0000    46.0000  
     0.5000     1.1667   -44.8333
```

```
>> 3/4  
ans =  
     0.7500  
>> 4\3  
ans =  
     0.7500  
>> a=[10.5, 25]  
a =  
    10.5000    25.0000  
>> a/5  
ans =  
     2.1000     5.0000  
>> 5\a  
ans =  
     2.1000     5.0000
```

乘方运算

一个矩阵的乘方运算可以表示成 A^x ，要求A为方阵，x为标量。

```
>> A=[1, 2, 3;4, 5, 6;7, 8, 0];
```

```
>> A^2
```

```
ans =
```

30	36	15
66	81	42
39	54	69

(2) 点运算

- 点运算符: `.*`、`./`、`.\`和`.^`。
- 两矩阵进行点运算是指它们的对应元素进行相关运算，要求两矩阵同型。



```
>> A=[1, 2, 3;4, 5, 6;7, 8, 9];  
>> B=[-1, 0, 1;1, -1, 0;0, 1, 1];  
>> C=A.*B
```

C =

-1	0	3
4	-5	0
0	8	9

```
>> D=A*B
```

D =

1	1	4
1	1	10
1	1	16

当 $x=0.1$ 、 0.4 、 0.7 、 1 时，分别求 $y=\sin x \cos x$ 的值。

```
>> x=0.1:0.3:1;
```

```
>> y=sin(x).*cos(x);
```



2. 关系运算

- ❑ 关系运算符：<(小于)、<=(小于或等于)、>(大于)、>=(大于或等于)、==(等于)、~=(不等于)。
- ❑ 当两个比较量是标量时，直接比较两数的大小。若关系成立，关系表达式结果为1，否则为0。

```
>> 3>4
```

```
ans =
```

```
0
```

```
>> x=5
```

```
x =
```

```
5
```

```
>> x==5
```

```
ans =
```

```
1
```


- ❑ 当参与比较的量是两个同型的矩阵时，比较是对两矩阵相同位置的元素按标量关系运算规则逐个进行，最终的关系运算的结果是一个与原矩阵同型的矩阵，它的元素由0或1组成。
- ❑ 当参与比较的一个是标量，而另一个是矩阵时，则把标量与矩阵的每一个元素按标量关系运算规则逐个比较，最终的关系运算的结果是一个与原矩阵同型的矩阵，它的元素由0或1组成。



建立3阶方阵A，判断A的元素是否为偶数。

```
>> A = [24, 35, 13; 22, 63, 23; 39, 47, 80]
```

```
A =
```

```
    24    35    13  
    22    63    23  
    39    47    80
```

```
>> P = rem(A, 2) == 0
```

```
P =
```

```
     1     0     0  
     1     0     0  
     0     0     1
```

3. 逻辑运算

逻辑运算符： $\&$ (与)、 $|$ (或)和 \sim (非)。

设参与逻辑运算的是两个标量 a 和 b ，那么运算规则为：

- $a\&b$ a 、 b 全为非零时，运算结果为1，否则为0。
- $a|b$ a 、 b 中只要有一个为非零时，运算结果为1。
- $\sim a$ 当 a 为零时，运算结果为1；当 a 为非零时，运算结果为0。

3. 逻辑运算

```
>> 3<4 & 6>5
```

```
ans =
```

```
1
```

```
>> ~(9==1)
```

```
ans =
```

```
1
```

```
>> ~9==1
```

```
ans =
```

```
0
```

- 若参与逻辑运算的是两个同型矩阵，那么将对矩阵相同位置上的元素按标量规则逐个进行运算，最终运算结果是一个与原矩阵同型的矩阵，其元素由1或0组成。
- 若参与逻辑运算的一个是标量，一个是矩阵，那么将在标量与矩阵中的每个元素之间按标量规则逐个进行运算，最终运算结果是一个与矩阵同型的矩阵，其元素由1或0组成。

水仙花数是指各位数字的立方之和等于该数本身的三位正整数。
求全部水仙花数。

```
>> m=100:999;  
>> m1=rem(m,10);  
>> m2=rem(fix(m/10),10);  
>> m3=fix(m/100);  
>> k=find(m==m1.*m1.*m1+m2.*m2.*m2+m3.*m3.*m3)  
k =  
    54    271    272    308  
>> s=m(k)  
s =  
   153   370   371   407
```


1.7 字符串处理

- 字符串的表示
- 字符串的操作

1. 字符串的表示

在MATLAB中，字符串是用单引号括起来的字符序列。

```
>> xm='Central South University'  
>> xm(1:3)  
ans =  
Cen
```

若字符串中的字符含有单引号，则该单引号字符要用两个单引号来表示。

```
>> 'I'm a teacher.'  
ans =  
I'm a teacher.
```

建立多行字符串，形成字符串矩阵。

```
>> ch=['abcdef' ; '123456'];  
>> ch(2,3)  
ans =  
3
```

建立一个字符串向量，然后对该向量做如下处理：

- ① 取第1~5个字符组成的子字符串。
- ② 将字符串倒过来重新排列。
- ③ 将字符串中的小写字母变成相应的大写字母，其余字符不变。
- ④ 统计字符串中小写字母的个数。

```
>> ch='ABc123d4e56Fg9';  
>> subch=ch(1:5)  
subch =  
ABc12  
>> revch=ch(end:-1:1)  
revch =  
9gF65e4d321cBA
```

```
>> k=find(ch>='a' & ch<='z')  
k =  
3 7 9 13  
>> ch(k)=ch(k)-('a'-'A')  
ch =  
ABC123D4E56FG9  
>> length(k)
```

```
ans =  
4
```



2. 字符串的操作

(1) 字符串的执行

字符串

格式: `eval(s)`

```
>> t=pi;  
>> m='[t, sin(t), cos(t)]';  
>> y=eval(m)  
y =  
    3.1416    0.0000   -1.0000
```


(2) 字符串与数值之间的转换

- ❑ `abs`和`double`函数都可以用来获取字符串矩阵所对应的ASCII码数值矩阵。
- ❑ `char`函数可以把ASCII码矩阵转换为字符串矩阵。

```
>> s1='MATLAB';  
>> a=abs(s1)  
a =  
    77    65    84    76    65    66  
>> char(a+32)  
ans =  
matlab
```

(3) 字符串的比较

字符串的比较有两种方法：利用关系运算符或字符串比较函数。

- ❑ 关系运算符比较：两个字符串里的每个字符依次按ASCII值大小逐个进行比较，比较的结果是一个数值向量，向量中的元素要么是1，要么是0。

```
>> 'www0' >= 'W123'  
ans =  
     1     1     1     0
```

□ 字符串比较函数用于判断字符串是否相等，有4种比较方式，函数如下：

- `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个字符是否相等，如果相等，返回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1，否则返回0。
- `strncmpi(s1, s2, n)`：在忽略字符串大小写前提下，比较两个字符串前n个字符是否相等，如果相等，返回1，否则返回0。



```
>> strcmp('www0', 'w123')
```

```
ans =
```

```
0
```

```
>> strncmpi('Www0','w123',1)
```

```
ans =
```

```
1
```

(4) 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。

```
>> p=findstr('This is a test!','is')
```

```
p =
```

```
     3     6
```

```
>> p=findstr('is','This is a test!')
```

```
p =
```

```
     3     6
```

```
>> result=strrep('This is a test!','test','class')
```

```
result =
```

```
This is a class!
```