



# 第一章 Matlab基本操作

## ----(3) 编程初步





# Matlab编程

- **1.3.1 M文件**
- **1.3.2 程序控制结构**
- **1.3.3 编程实例**



# 1.3.1 M 文件

## □ M 文件

- 用 Matlab 语言编写的程序称为 **M 文件**
- **M 文件是由若干 Matlab 命令组合在一起构成的，它可以完成某些操作，也可以实现某种算法**
- M 文件以 **.m** 为扩展名
- **文件的命名规则与变量相同！文件名应尽量与程序要表达的意义相符合，以方便今后调用**

# M 文件的建立与打开



□ M 文件是一个文本文件，可以用任何文本编辑器来建立和编辑，通常使用 Matlab 自带的 M 文件编辑器。

## □ 新建一个 M 文件

- ◆ 菜单操作 ( File → New → M-File )
- ◆ 命令操作 ( `edit M 文件名` )
- ◆ 命令按钮 ( 快捷键 )

## □ 打开已有的 M 文件

- ◆ 菜单操作 ( File → Open )
- ◆ 命令操作 ( `edit M 文件名` )
- ◆ 命令按钮 ( 快捷键 )
- ◆ 双击 M 文件



# M文件的编辑与路径设置



- ❑ 进入MATLAB的Editor/Debugger窗口来编辑程序
- ❑ 在编辑环境中，文字的不同颜色显示表明文字的不同属性。

绿色：注解；黑色：程序主体；红色：属性值的设定；蓝色：控制流程。

- ❑ 在运行程序之前，必须设置好MATLAB的工作路径



# M 文件分类

- M 文件分类（根据调用方式的不同）
  - Script: 脚本文件 / 命令文件
  - Function: 函数文件

可以直接运行的M文件

# 命令文件

## 例1 用 mesh 绘制半径为 3 的球

- 命令行方式(在命令窗口)

```
>> u=0:pi/60:2*pi;  
>> v=0:pi/60:pi;  
>> [U,V]=meshgrid(u,v);  
>> R=3;  
>> X=R*sin(V).*cos(U);  
>> Y=R*sin(V).*sin(U);  
>> Z=R*cos(V);  
>> mesh(X,Y,Z);  
>> axis equal;
```

- 编程方式：在M文件编辑器中，新建一个 M 文件`qiu.m`，内容如下：

```
u=0:pi/60:2*pi;  
v=0:pi/60:pi;  
[U,V]=meshgrid(u,v);  
R=3;  
X=R*sin(V).*cos(U);  
Y=R*sin(V).*sin(U);  
Z=R*cos(V);  
mesh(X,Y,Z);  
axis equal;
```



# 命令文件

**例2** 编写M文件, 将变量a、b的值互换

- 命令行方式(在命令窗口)

```
>> clear;  
>> a = 1:10;  
>> b = [11,12,13,14;15,16,17,18];  
>> c = a; a = b; b = c;  
>> a, b
```

- 编程方式: 在M文件编辑器中, 新建一个 M 文件 **exch.m**, 内容如下:

```
clear;  
a = 1:10;  
b = [11,12,13,14;15,16,17,18];  
c = a; a = b; b = c;  
a  
b
```



# 函数文件

## □ 函数文件由 `function` 语句引导

```
function [out1,out2,...]=函数名(in1,in2,...)  
% 注释说明部分(可选)  
函数体语句(必须)
```

- ◆ 第一行为**引导行**，表示该 M文件是函数文件
- ◆ 函数名的命名规则与变量名相同（**必须以字母开头**）
- ◆ 当输出行参多于一个时，用**方括号**括起来
- ◆ 函数必须是一个**单独的 M文件**
- ◆ 函数**文件名必须与函数名一致**
- ◆ 以**百分号**开始的语句为**注释语句**

# 函数调用

## □ 函数调用的一般格式

输出实参列表=函数名 (输入实参列表)

- ◆ 函数调用时，**实参的顺序**应与函数定义时的**形参的顺序**一致。
- ◆ 函数可以**嵌套调用**，即一个函数可以被其它函数调用，甚至可以被它自身调用，此时称为**递归调用**。
- ◆ 函数所传递的参数具有可调性，Matlab 用两个永久变量 **nargin** 和 **nargout** 分别记录调用该函数时的输入实参和输出实参的个数。

# 递归函数举例

例3：利用函数的递归调用计算  $n!$

$$n! = \begin{cases} 1, & n = 1 \\ n \cdot (n-1)!, & n > 1 \end{cases}$$

```
% 函数文件 factor.m
function f=factor(n)
if (n<=1)
    f=1;
else
    f=n*factor(n-1);
end
```

```
% main.m
% 计算 s=1!+2!+3!+4!+5!
%
clear;
s=0;
for i=1:5
    s=s+factor(i);
end
fprintf(' s=%g \n',s)
```



# 函数文件编程示例

**例4** 编写M文件，求方程  $ax^2 + bx + c = 0$  的根（a、b、c自取）

- 新建一个 M 文件 `gen.m`，内容如下：

```
function [x1,x2]=gen(a,b,c)
    d=b*b-4*a*c;
    x=[(-b+sqrt(d))/(2*a), (-b-sqrt(d))/(2*a)];
    disp('函数的根为: ')
    disp(['x1=', num2str(x(1)), 'x2=', num2str(x(2))])
```

- 在命令窗口中输入 `gen(1, 2, 1)`，即可调用该 M 文件中的函数 `gen`。

**注意：**

- (1) 函数名必须与文件名相同
- (2) 调用时需给出相应的输入参数

# 函数文件编程示例



**例5** 比较下列两个程序，注意命令文件（以dd1命名）与函数文件（以dd2命名）的区别与联系.

```
x=input('输入初值x=');  
n=input('输入迭代次数=');  
y(1)=x ;  
for k=1:n  
    x=1/(x+1); y(k)=x;  
end  
y
```

```
function y=dd2(x,n)  
y(1)=x ;  
for k=1:n  
    x=1/(x+1); y(k)=x;  
end
```



# 函数文件编程示例

## 例6 建立函数

$$f(x) = \frac{5x-6}{(x-3)^2+2} + \frac{9x}{x^3+4}$$

解：程序如下：

```
function y=f(x)
```

$$y = (5 * x - 6) / ((x - 3) * (x - 3) + 2) + 9 * x / (x * x * x + 4)$$

以文件名f.m保存。

# 命令文件和函数文件的区别

- 命令文件没有输入参数，也不返回输出参数；函数文件可以带输入参数，也可以返回输出参数。
- 命令文件对工作空间中的变量进行操作，文件中所有命令的执行结果也返回工作空间中；函数文件中定义的变量为局部变量，当函数文件执行完毕时，这些变量也被清除。
- 命令文件可以直接运行；函数文件不能直接运行，要以函数调用的方式来调用它。



## 1.3.2 程序控制结构

□ 程序控制结构有三种：

**顺序结构、循环结构和选择结构**

任何复杂的程序都由这三种基本结构组成。

Matlab提供了实现控制结构的语句，利用这些语句可以编写解决实际问题的程序。

- ◆ **顺序结构：** 数据输入输出（input、disp、fprintf）
- ◆ **选择结构：** if—else-end语句、switch-end语句
- ◆ **循环结构：** for -end循环、while-end 循环



# 顺序结构

## ◆ 顺序结构

- 按排列顺序依次执行各条语句，直到程序的最后
- 这是最简单的一种程序结构，一般涉及数据的输入输出、数据的计算或处理等

### ● 数据的输入： `input`

```
A=input(out,in)
```

- 其中 `out` 为提示信息，通常为字符串，`in` 可以缺省
- 该命令运行后，要求用户在命令窗口输入 `A` 的值  
(输入的值可以是数、数学表达式或字符串)

# input

例

```
A=input('Please input A: ')
```

例

```
name=input('What's your name?','s')
```

- `input`应用时，注意：
  - 输入字符串时必须带单引号
  - 单引号的输出：两个连续的单引号
  - 若输入的是数、数学表达式，则in不能出现

# disp

- 数据的输出: `disp`

```
disp(X)
```

- 输出变量 `x` 的**值**, `x` 可以是数值矩阵或字符串
- 一次只能输出一个变量

例

```
>> A='Hello, Tom!';  
>> disp(A)
```

```
>> B=[1 2 3; 4 5 6; 7 8 9];  
>> disp(B)
```

# fprintf

## ◆ 数据的输出: **fprintf**

按指定的格式将变量的值输出到指定的文件

**fprintf(fid, format, variables)**

- **fid** 为文件句柄，若缺省，则将变量的值输出到屏幕上
- **format** 用来指定数据输出时采用的格式，常见的有

- **%e** (采用科学计算形式)
- **%f** (采用浮点数形式)
- **%g** (由系统自动选取上述两种格式之一)
- **%s** (输出字符串)

- **format** 中还可以使用一些特殊格式，如：

**\n** (换行)   **\t** (制表符)   **\b** (退格)   **\\** (反斜杆)   **%%** (百分号)

# 数据输出 fprintf

例:

```
>> a='Hello' ; b=2.4 ; c=100*pi ;  
>> fprintf('a=%s,b=%f,c=%e\n',a,b,c)  
a=Hello,    b=2.400000,    c=3.141593e+002
```

- **format** 中的输出格式要与输出变量一一对应
- 可以没有输出变量

例:

```
>> fprintf('Today is the 77 anniversary of the Mukden  
Incident\n')
```



# 选择结构

**选择结构** 是根据给定的条件成立或不成立，分别执行不同的语句。

Matlab 用于实现选择结构的语句有

**if -end语句**

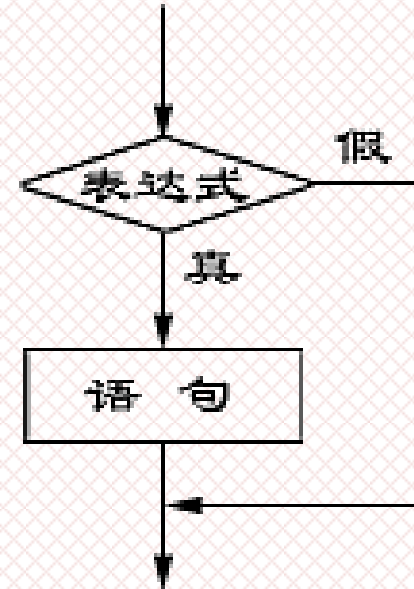
和

**switch-end 语句**

# if-end条件语句

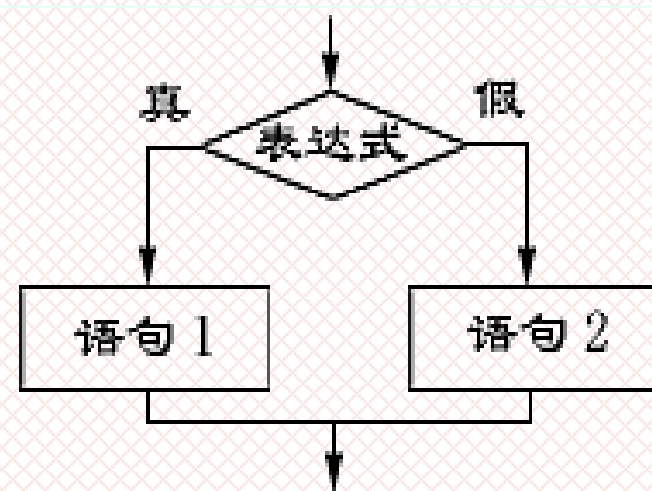
- 单分支结构

```
if 条件表达式  
    语句组  
end
```



- 双分支结构

```
If 条件表达式  
    语句组1  
else  
    语句组2  
end
```



# if-end 条件语句

## ● 多分支结构

**if** 条件表达式1

语句组1

**elseif** 条件表达式 2

语句组2

...

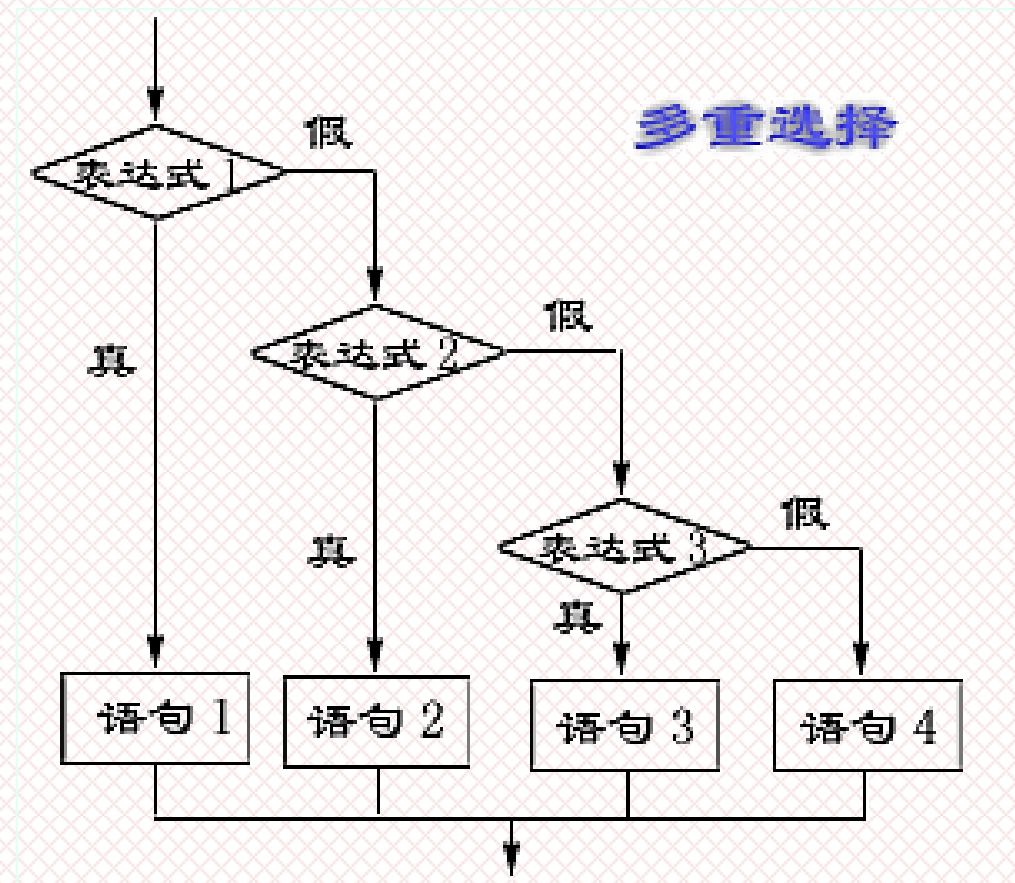
**elseif** 条件表达式 m

语句组m

**else**

语句组

**end**





# if-end编程示例

例6 控制函数

$$y = \begin{cases} a+1, & n=0 \\ a(1+n), & n=1 \\ a+n, & n=2 \\ a & n>2 \end{cases}$$

```
function y=control(n)
a=20;
if n==0
    y=a+1;
elseif n==1
    y=a*(1+n);
elseif n==2
    y=a+n;
else
    y=a;
end
```

# if-end编程示例

**例7** 一个三位整数，其各位数字的立方和等于该数本身，称该数为水仙花数。求全部水仙花数。

**解：**程序如下

```
A=[];  
for m=100:999  
    m1=fix(m/100);           %求m的百位数字  
    m2=rem(fix(m/10),10);    %求m的十位数字  
    m3=rem(m,10);           %求m的个位数字  
    if m==m1*m1*m1+m2*m2*m2+m3*m3*m3  
        A=[A,m];  
    end  
end  
disp(A)
```

# 编制M函数文件，实现不等长多项式的加法运算。

## 例8 多项式加法

%实现不等长多项式加法运算

```
function y=polyadd(x1,x2)
```

```
n1=length(x1);
```

```
n2=length(x2);
```

```
if n1>n2
```

```
    x2=[zeros(1,n1-n2),x2];
```

```
else
```

```
    x1=[zeros(1,n2-n1),x1];
```

```
end
```

```
y=x1+x2;
```

%x1和x2为两个不等长多项式系数向量

%测试x1的长度

%测试x2的长度

%对较短的向量进行补零处理

```
>> x1=[1 2]; x2=[5 6 7 8]; y=polyadd(x1,x2)
```

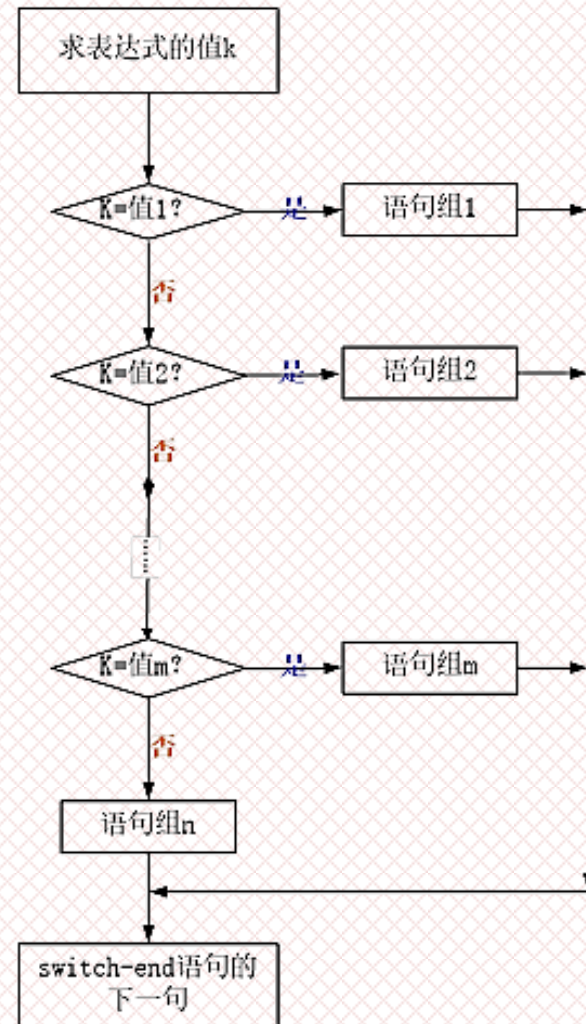
```
y =
```

```
    5     6     8    10
```

# switch-end 语句

- 根据表达式的不同取值，分别执行不同的语句

```
switch 表达式  
  case 值1  
    语句组1  
  case 值2  
    语句组2  
  ...  
  case 值m  
    语句组m  
  otherwise  
    语句组n  
end
```



# switch-end 语句

## ● 关于 switch-end 语句的几点注解：

- Matlab 首先计算 **表达式** 的值，然后将它依次与各个 **case** 指令后的检测值进行比较，当比较结果为真时，就执行相应的语句组，然后跳出 **switch-end** 结构
- 如果所有的比较结果都为假，则执行 **otherwise** 后面的语句组，然后跳出 **switch-end** 结构
- **otherwise** 指令可以不出现
- **switch** 后面的表达式的值可以是一个标量或字符串
- **case** 指令后的检测值超过一个时，应用 { } 括起来

# switch-end编程示例

## 例9

```
method=input('请输入方法名：')  
switch method  
    case {'linear','bilinear'}  
        disp('Method is linear')  
    case 'cubic'  
        disp('Method is cubic')  
    case 'nearest'  
        disp('Method is nearest')  
    otherwise  
        disp('Unknown method.')  
end
```

# switch-end编程示例

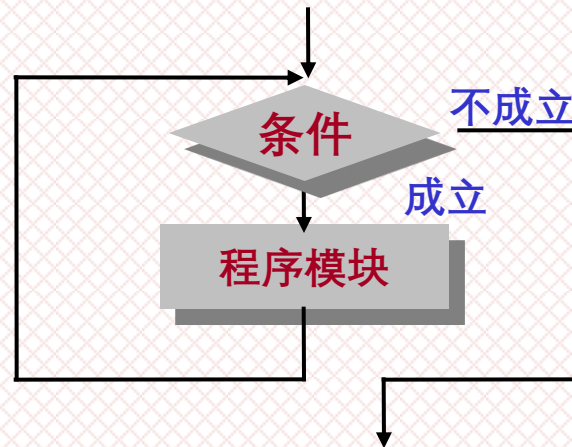
**例10** 用switch-end语句编一程序，对于给定的一个百分制成绩，输出相应的五分制成绩，

设： 90分以上为‘A’，  
80 ~ 89分为‘B’，  
70 ~ 79分为‘C’，  
60 ~ 69分为‘D’，  
60分以下为‘E’。

```
function sctole(score)
    grade=fix(score/10);
    switch grade
        case {10,9}
            grade='A'
        case 8
            grade='B'
        case 7
            grade='C'
        case 6
            grade='D'
        otherwise
            grade='E'
    end
```

# 循环结构

**循环结构** 是按照给定的条件，重复执行指定的语句。  
.Matlab 用于实现循环结构的语句有 **for -end语句** 和 **while -end语句**





# for-end 循环

循环变量

```
for var = expr
```

循环体

行向量或矩阵

```
end
```

例11 使用for-end结构计算 $1+2+3+\dots+100$

```
sum=0;  
for i=1:100  
    sum=sum+i;  
end  
sum
```

# for-end编程示例

**例12** 已知  $y = 1 + \frac{1}{3^2} + \frac{1}{5^2} + \cdots + \frac{1}{(2n-1)^2}$ ，当  $n=100$  时，求  $y$  的值

```
y=0;  
n=100;  
for k=1:n  
    y=y+1/(2*k-1)^2;  
end
```

```
y=0;  
n=input('n=');  
for k=1:2:2*n-1  
    y=y+1/(k*k);  
end
```

```
n=input('n=');  
k=1:2:2*n-1;  
y=sum(1./(k.*k))
```



# for-end 循环

- 循环语句可以嵌套使用
- 不能在 `for-end` 循环体内改变循环变量的值
- 为了提高代码的运行效率，应尽可能提高代码的向量化程度，避免 `for-end` 循环的使用
- 如果预先就知道循环的次数，则可以采用 `for-end` 循环；否则，如果预先无法确定循环的次数，则可以使用 `while-end` 循环。

# while-end循环

```
while (条件表达式)  
    循环体  
end
```

例13 使用while-end结构计算 $1+2+3+\dots+100$

```
s=0;  
i=1;  
while i<=100  
    s=s+i;  
    i=i+1;  
end  
s
```

```
s=0;  
for i=1:100  
    s=s+i;  
end  
s
```

# while循环举例

例14 Fibonacci数组的元素满足Fibonacci 规则:

$$a_{k+2} = a_k + a_{k+1} \text{ 且 } a_1 = a_2 = 1。$$

现要求该数组中第一个大于10000的元素。

```
a(1)=1;a(2)=1;i=3;  
a(i)=a(i-1)+a(i-2);  
while a(i)<=10000  
    i=i+1;  
    a(i)=a(i-1)+a(i-2);  
  
end  
i,a(i)
```

## 例15

编制程序实现运算： $s = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{100}$

%此程序文件名为example3\_5\_1.m

clear %清变量

s=0;n=1; %为s和n置初值

while n<=100

s=s+1/n;

n=n+1;

end

s %显示最后的求和结果

```
>> example3_5_1
```

```
s =
```

```
5.1874
```



## 例16：用循环求解

$$\sum_{i=1}^{100} i$$

```
s=0;  
for i=1:100  
    s=s+i;  
end
```

```
s=0;  
i=1;  
while (i<=100)  
    s=s+i;  
    i=i+1;  
end
```

```
sum(1:100)  
ans =  
    5050
```



例17：用循环求解求最小的  $m$ ，使得  $\sum_{i=1}^m i > 10000$

```
s=0; m=0;  
while (s<=10000)  
    m=m+1;  
    s=s+m;  
end  
s,m % 求出的 m 即是所求
```

```
ans =  
    10011    141
```





例18: 求  $\sum_{i=1}^{100000} \left( \frac{1}{2^i} + \frac{1}{3^i} \right)$

```
tic
s=0;
for i=1:100000
    s=s+1/2^i+1/3^i;
end
toc

elapsed_time =
    1.1820
```

```
tic
i=1:100000;
s=sum(1./2.^i+1./3.^i);
toc
%向量化所需时间少

elapsed_time =
    0.3010
```



# 其它流控制语句

## ◆ **break** 和 **continue**

- **break** 语句用于**终止循环**的执行，即跳出最内层循环
- **continue** 语句用于结束**本次循环**，进行下一次循环
- **break** 和 **continue** 一般与 **if** 语句配合使用

## ◆ **return**

- **return** 语句用于**退出**正在运行的脚本或函数，通常用在函数文件中。

# continue语句

例**19** 把**100**到**120**之间的能被**7**整除的整数输出。

```
for i=100:120  
    if rem(i,7)~=0  
        continue  
    end  
    i  
end
```

# break语句

例**20** 输出**100**到**120**之间第一个能被**7**整除的整数

```
for i=100:120  
    if rem(i,7)~=0  
        continue  
    end  
    i  
    break  
end
```



例21：用循环求解求最大的  $m$ ，使得  $\sum_{i=1}^m i > 10000$

```
s=0;  
for i=1:10000  
    s=s+i;  
    if s>10000  
        break  
    end  
end
```

```
i  
i =  
141
```

# return语句

```
function output = fact(n)  
if n == 1  
    output = 1;  
    return;  
end  
output = n*fact(n-1);
```

# pause语句

## ◆ 程序的暂停： **pause**

**pause** 或 **pause (n)**

其中 **n** 是延迟时间，以秒为单位；  
若缺省，则将暂停程序，直到用户按任意键后继续

- **pause off** 屏蔽程序中所有 **pause** 的作用
- **pause on** 打开 **pause** 的作用

若想强行终止程序的运行，可以使用 **Ctrl+c**



## 1.3.3 编程实例

实例1 计算分段函数  $y = \begin{cases} \cos(x+1) + \sqrt{x^2+1} & (x=10) \\ x\sqrt{x+\sqrt{x}} & (x \neq 10) \end{cases}$  的值.

```
x=input('请输入x的值:');  
if x==10  
    y=cos(x+1)+sqrt(x*x+1);  
else  
    y=x*sqrt(x+sqrt(x));  
end  
y
```





**实例2** 建立一个函数文件将变量**a,b**的值互换，然后在命令窗口调用该函数文件.

```
function [a,b]=exch(a,b)  
c=a;a=b;b=c;
```

然后在MATLAB的命令窗口调用该函数文件：

```
clear;  
x=1:10;  
y=[11,12,13,14;15,16,17,18];  
[x,y]=exch(x,y)
```

## 实例3 将百分制的成绩转换为五级制的成绩输出

```
clear
n=input('输入n= ');
if n>=90
    r='A'
elseif n>=80
    r='B'
elseif n>=70
    r='C'
elseif n>=60
    r='D'
else
    r='E'
end
```

```
clear
n=input('输入n= ');
switch n
case n>=90
    r='A'
case n>=80
    r='B'
case n>=70
    r='C'
case n>=60
    r='D'
otherwise
    r='E'
end
```

实例4 输入一个字符，若为大写字母，则输出其后继字符，若为小写字母，则输出其前导字符，若为数字字符则输出其对应的数值，若为其他字符则原样输出。

```
c=input('请输入一个字符','s');  
if c>='A' & c<='Z'  
    disp(setstr(abs(c)+1));  
elseif c>='a' & c<='z'  
    disp(setstr(abs(c)-1));  
elseif c>='0' & c<='9'  
    disp(abs(c)-abs('0'));  
else  
    disp(c);  
end
```

## 实例5. 编制程序，判断输入数据的奇偶性。

### (1) 只考虑输入数字的情况

```
%该程序文件名为f3.m  
clear  
n=input(' 请输入数据 ');  
if rem(n,2)==0  
    A=' 偶数';  
else  
    A=' 奇数';  
end  
disp(A)
```

```
>> f3  
请输入数据 23  
奇数  
>> f3  
请输入数据 78  
偶数  
>>
```

## 实例5. 编制程序，判断输入数据的奇偶性。

### (2) 考虑输入是空格或直接按回车键的情况

```
%该程序文件名为f4.m  
clear  
n=input(' 请输入数据 ');  
if isempty(n)==1  
    A=' 没有输入数据' ;  
elseif rem(n, 2)==0  
    A=' 偶数' ;  
else  
    A=' 奇数' ;  
end  
disp(A)
```

```
>> f4  
请输入数据    35  
奇数  
>> f4  
请输入数据    46  
偶数  
>> f4  
请输入数据  
没有输入数据  
>>
```

## 实例6. 编制程序，寻找输入数组中的最大数。

```
% f5.m完成寻找最大数
clear
x=input(' 请输入数据x=[x1, x2, ...]=' );
n=length(x);
max=x(1);
for i=2:n
    if max<=x(i)
        max=x(i);
    end
end
disp(max)
```

```
>> f5
请输入数据x=[x1, x2, ...]=[9 -4 0 3 10 2]
10
```



实例7 已知5个学生4门功课的成绩，求每名学生的总成绩.

```
s=0;  
a=[65,76,56,78;98,83,74,85;76,67,78,79;98,58,42,73;67,89,  
   76,87];  
for k=1:4  
    s=s+a(:,k);  
end  
disp(s);
```

## 实例8 求某自然数范围内的全部素数.

```
m=input('m=');  
p=2:m;  
for i=2:sqrt(m)  
    n=find(rem(p,i)==0&p~=i);  
    p(n)=[];  
end  
p
```



## 实例9 求 $1! + 2! + 3! + \dots + n!$ 的值

```
%递归调用函数factor.m:  
function f=factor(n)  
if n<=1  
    f=1;  
else  
    f=factor(n-1)*n;  
end
```

```
s=0;  
for k=1:n  
    t=factor(k);  
    s=s+t;  
end
```

实例10. 数论中的一个有趣问题：任取一个正整数，如果是偶数，用 2 除，如果是奇数，用 3 乘再加 1，反复这个过程，直到所得到的数为 1。

问：是否存在使该过程永不中止的整数

```
while 1
    n=input('Please enter n(nonpositive quit):');
    m=n;
    if n<=0, break; end
    while n>1
        if rem(n,2)==0
            n=n/2;
        else
            n=3*n+1;
        end
        fprintf('\n n=%d',n);
    end
    fprintf('\n %d is not we need! continue ... \n',m);
end
```

# 注意的问题

- (1) 各种程序结构的格式.
- (2) `A=input('请输入A的值:')`
- (3) 变量、函数及M文件的命名规则，不可使用关键字像  
`if,while,for,end,sum`  
及内部函数名等命名，M文件名不可与其中的变量重名.
- (4) 学会使用Matlab的帮助系统.



# 注意的问题

## ❑ 程序出错主要为两类：

- ✓ 1) 格式错误，如缺 ‘ ( ’ 或 ‘ ) ’ 等，在运行时可检测出大多数该类错误，并指出错在哪一行。
- ✓ 2) 算法错误，逻辑上的错误，不易查找，遇到此类错误时需耐心。



# 实验题1

- (1) 从键盘输入一个数，将它反向输出，例如输入693，输出为396
- (2) 输入一个百分制成绩，要求输出成绩等级A，B，C，D，E其中90-100为A，80-89为B，70-79为C，60-69为D，60以下为E。
  - 1) 分别用if语句和switch语句实现
  - 2) 输入百分制成绩后要判断成绩的合理性，对不合理的成绩应输出出错信息



## 实验题2

(1). 编程计算  $y = \sum_{n=-10}^{10} 2^n = 2^{-10} + 2^{-9} + \cdots + 2^9 + 2^{10}$

(2). 编程求满足  $\sum_{i=1}^m 2^i > 10000$  最小的m。



## 实验题3

(1) 编写一个函数，计算下面函数的值，给出标量x的值，调用该函数后，返回y的值。

$$y(x) = \begin{cases} \sin x, & x \leq 0 \\ x, & 0 < x \leq 3 \\ -x + 6, & x > 3 \end{cases}$$

(2) 编写一个函数求向量x中元素的平均值、最大值、最小值、均方根值，用下面数据测试你写的函数：

(1) `x=sin(0:0.01:6*pi)`

(2) `x=rand(1,200)`

# Q & A

- 有什么问题吗？

