



# 数学软件 Matlab

---

## —— 编程基础（脚本文件）

# 本讲主要内容

---

- M 文件
- 基本运算：算术、关系、逻辑
- 简单的输入输出
- 控制结构：顺序、选择、循环

# Matlab 编程

---

## ● Matlab 编程

- Matlab 作为一种广泛应用于科学计算的工具软件，不仅具有强大的数值计算能力和丰富的绘图功能，同时也可以与 C、FORTRAN 等高级语言一样进行程序设计
- 利用 Matlab 的程序控制功能，将相关 Matlab 命令编成程序存储在一个文件中（M 文件），然后在命令窗口中运行该文件，Matlab 就会自动依次执行文件中的命令，直到全部命令执行完毕
- 在 Matlab 程序设计中，要充分利用 Matlab 数据结构的特点，提高编程效率

# M 文件

---

- M 文件介绍

- 用 Matlab 语言编写的程序称为 **M 文件**
- M 文件以 **.m** 为扩展名
- 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

# M 文件

- M 文件的创建的与编辑

M 文件是文本文件，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 → Script / Function )

- 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

# 编程示例

**例：用 mesh 绘制半径为 3 的球**

● 命令行方式：

```
u=[0:pi/60:2*pi];  
v=[0:pi/60:pi];  
R=3;  
[U,V]=meshgrid(u,v);  
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：新建一个 **M 文件**，将上面的内容输入到该文件中，然后在命令窗口中输入该文件名，即可执行

**注：M 文件的命名规则与变量相同！**

# 编程示例

**例：**编写M文件，将华氏温度转化为摄氏温度：

$$c = \frac{5}{9}(f - 32)$$

- 新建一个脚本文件 **f2c.m**，内容如下：

```
clear; % 清除当前工作空间中的所有变量
f=input('Please input f:');
c=5*(f-32)/9;
fprintf('c=%g\n',c);
```

- 在命令窗口中输入 **f2c**，即可执行该 M 文件

- 脚本文件运行所产生的变量都驻留在 Matlab 当前工作空间
- 百分号是**注释符**
- M文件必须在**当前工作目录**下才能运行

# 本讲主要内容

---

- M 文件
- 基本运算：算术（略）、关系、逻辑
- 简单的输入输出
- 控制结构：顺序、选择、循环



# 关系运算

- 关系运算：比较大小

>	大于	>=	大于等于	==	等于
<	小于	<=	小于等于	~=	不等于

- 比较大小，如果结论是 **真** 则返回 **1**，否则返回 **0**
- 注意 **==** 与 **=** 的区别

- 可以对两个**同样大小**的数组进行比较运算，即对应分量进行比较，结果为一个同样大小的0-1数组
- 也可以对一个**数组**和一个**标量**进行比较运算，此时标量和数组中的每一个元素极限比较，结果也是一个0-1数组

# 关系运算举例

---

例:

$$2+2==4$$

$$x=\pi^2 < 8$$

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

$$B = \begin{bmatrix} 3 & 1 & 0 \\ 2 & 4 & 6 \end{bmatrix}$$

$$A \geq B$$

$$B < 4$$

# 逻辑运算

## ● 逻辑运算

&	逻辑与
	逻辑或
~	逻辑非

and(x,y)	与
or(x,y)	或
xor(x,y)	异或
not(x)	非
&&	与（短格式）
	或（短格式）

注意 &, | 与 &&, || 的区别

例:

```
x=5;  
3<x<8          % ERROR  
3<x & x<8
```

# 逻辑运算

- 运算法则

运算对象		与	或	非	异或
A	B	A&B	A B	~A	Xor(A,B)
0	0	0	0	1	0
0	1	0	1	1	1
1	0	0	1	0	1
1	1	1	1	0	0

在 Matlab 中，0 表示“假”，非零表示“真”

# 逻辑运算

- 逻辑运算函数：**all**、**any**

**all(x)** % 逻辑与的推广

如果向量 **x** 中所有元素都非零，则返回 **1**，否则返回 **0**

**any(x)** % 逻辑或的推广

如果向量 **x** 中存在非零元素，则返回 **1**，否则返回 **0**

若 **x** 为矩阵，则 **any** 和 **all** 按列运算，返回一个 **0-1** 向量

例：

```
A=[61 83 58 94 75]; % 学生成绩
```

```
all(A>=60) % 全部及格否？
```

```
any(A>=90) % 有没有90分以上的？
```

# 一些测试函数

## ● 测试函数

<code>isfinite(x)</code>	若 <code>x</code> 为有限值，返回真值 <b>1</b>
<code>isinf(x)</code>	若 <code>x</code> 为无穷大，返回真值 <b>1</b>
<code>isnan(x)</code>	若 <code>x</code> 为不定值，返回真值 <b>1</b>
<code>isreal(x)</code>	若 <code>x</code> 无虚部，返回真值 <b>1</b>
<code>isstr(x)</code>	若 <code>x</code> 为一个字符串，返回真值 <b>1</b>
<code>isempty(x)</code>	若 <code>x</code> 为空，返回真值 <b>1</b>
<code>isprime(x)</code>	判断 <code>x</code> 是否为素数

例:

```
isfinite(5)
```

```
isinf(5)
```

# 运算优先级

小括号	
幂，点幂	
正号，负号，逻辑非	
乘，除，点乘，点除	
加，减	
冒号运算	
关系运算	
&	
&&	

# 本讲主要内容

---

- M 文件
- 基本运算：算术、关系、逻辑
- 简单的输入输出
- 控制结构：顺序、选择、循环



# input

---

- 数据的输入：input

```
x=input(提示信息)
```

- 其中 提示信息 为字符串，
- 该命令要求用户输入  $x$  的值 (可以是数或字符串)

例：

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

例：

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

- 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

# 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的值

例：

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

```
str='Hello, Tom!';  
disp(str)
```

# fprintf

## ● 数据的格式化输出

**fprintf(fid, format, 变量列表)**

- 按指定的格式将变量的值输出到屏幕或指定的文件中
- **fid** 为文件句柄，若缺省，则输出到屏幕上
- **format** 用来指定数据输出时采用的格式，包含：  
普通字符串、格式字符串、转义字符

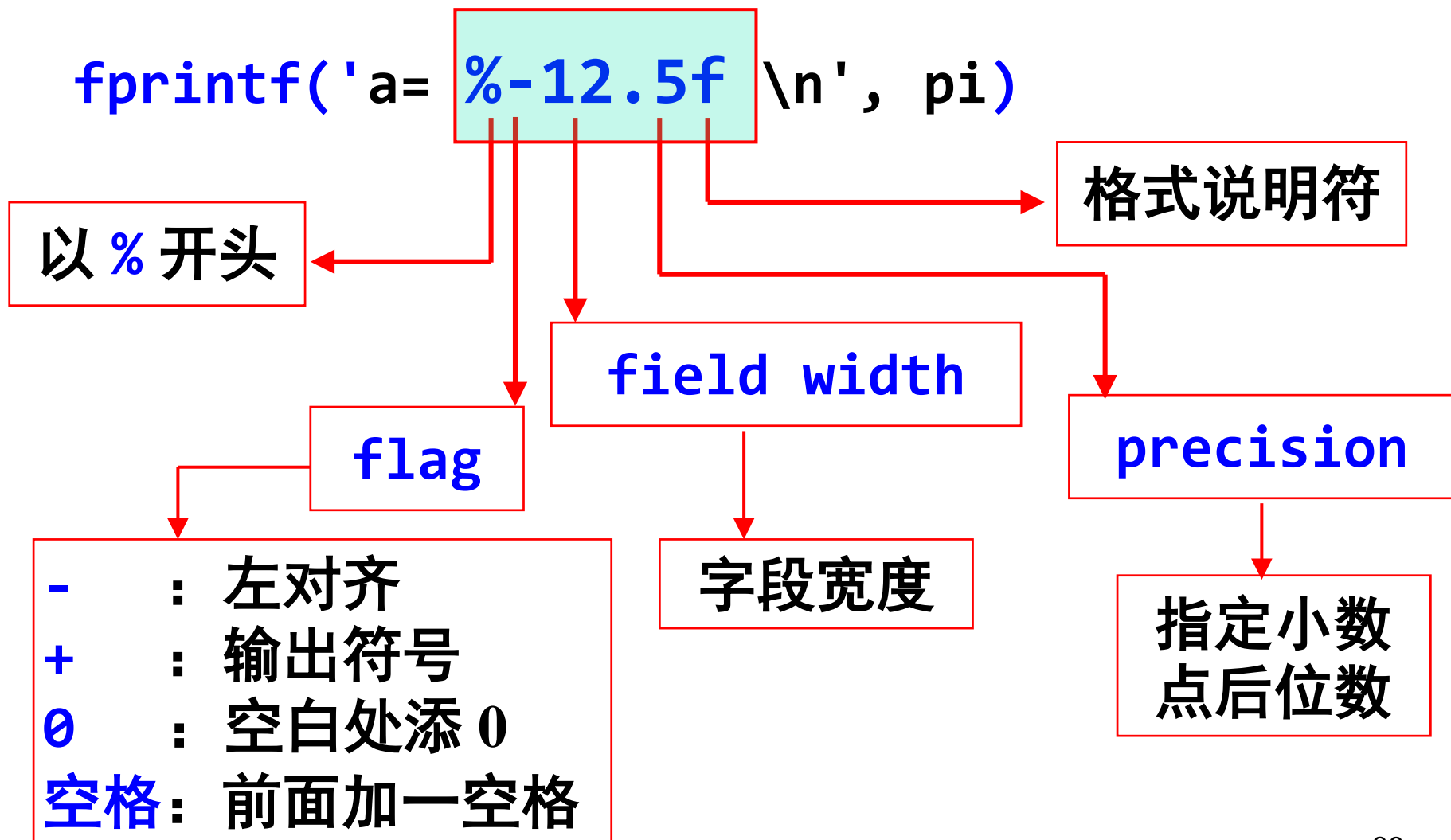
**格式字符串**：以 % 开头，包括：

- **flags** (可选)
- **Width and precision fields** (域宽和精度, 可选)
- **Conversion character** (格式说明符, 必须)

# fprintf 举例

## ● 格式字符串示例

```
fprintf('a= %-12.5f \n', pi)
```



# fprintf

## ● 常用的格式说明符

<b>c</b>	字符型	<b>g</b>	浮点数（自动）
<b>d</b>	十进制整数	<b>o</b>	八进制
<b>e</b>	浮点数（科学计数法）	<b>s</b>	字符串
<b>f</b>	浮点数（小数形式）	<b>x/X</b>	十六进制

## ● 常用的转义字符

<b>\b</b>	退后一格	<b>\t</b>	水平制表符
<b>\f</b>	换页	<b>\\</b>	反斜杠
<b>\n</b>	换行	<b>''</b>	单引号（两个连续的单引号）
<b>\r</b>	回车	<b>%%</b>	百分号（两个连续的百分号）

# fprintf 举例

例:

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

- 格式字符串与输出变量要一一对应!

例:

```
fprintf('pi=%12.5f=\n', pi)  
fprintf('pi=%-12.5f=\n', pi)  
fprintf('pi=%012.5f=\n', pi)
```

例:

```
fprintf('Today is Tuesday!\n')
```

- 可以只输出一个字符串

# 本讲主要内容

---

- M 文件
- 基本运算：算术、关系、逻辑
- 简单的输入输出
- 控制结构：顺序、选择、循环

# M文件控制流

---

- **程序控制结构有三种：**

  - 顺序结构、选择结构和循环结构**

  - 任何复杂的程序都由这三种基本结构组成

- **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



# 选择结构

---

- 选择结构

- 根据给定的条件成立或不成立，分别执行不同的语句

- 选择结构的实现

- `if` 语句
- `switch` 语句

# if 条件语句

---

- 单分支 **if** 语句

```
if 条件表达式  
    语句组  
end
```

例：

```
if score>=60  
    mark='passed';  
end
```

- 双分支 **if** 语句

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

例：

```
if a>b  
    x=a;  
else  
    x=b;  
end
```

# if 条件语句

- 多分支 **if** 语句

```
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
```

## 例：计算一个数符号

```
if x>0
    sign_of_x=1;
elseif x<0
    sign_of_x=-1;
elseif x==0
    sign_of_x=0;
else
    disp('I dont know!')
end
```

# switch 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0
  case 表达式1
    语句组1
  case 表达式2
    语句组2
    ...
  case 表达式m
    语句组m
  otherwise
    语句组
end
```

- 先计算表达式0的值，然后将它依次与各个 case 指令后表达式的值进行比较，当两者一样时，就执行相应语句组，然后跳出 switch 结构
- switch 后面的表达式0的值可以是一个数或字符串
- otherwise 指令可以不出现
- 如果所有的比较都不匹配时，执行 otherwise 后面的语句组

# switch 举例

例:

```
score=input('请输入你的得分 (A~E) : ');
switch score
    case 'A'
        disp('90-100');
    case 'B'
        disp('80-89');
    case 'C'
        disp('70-79');
    case 'D'
        disp('60-69');
    otherwise
        disp('Sorry!');
end
```

# 循环结构

---

- 循环结构

- 根据给定的条件，重复执行指定的语句

- 循环结构的实现

- `for` 语句
- `while` 语句

# for 循环

---

- for 循环

```
for 循环变量 = 取值列表  
    循环体  
end
```

- 取值列表 通常是一个向量
- 将取值列表中的值依次赋给循环变量，直到全部取完，循环结束

- 不要在循环体内改变循环变量的值
- 为提高代码的运行效率，应尽可能提高代码的向量化程度

# for 循环

**例：** 已知  $y = 1 + \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{2n-1}$ ，当  $n=100$  时，求  $y$  的值

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

**例：** 计算  $s = 1 + 3 + 10 - 28 + 30 + 50 - 12 - 8$

```
s=0;
x=[1, 3, 10, -28, 30, 50, -12, -8];
for k=x    % 将 x 的分量依次赋值给循环变量
    s=s+k;
end
```

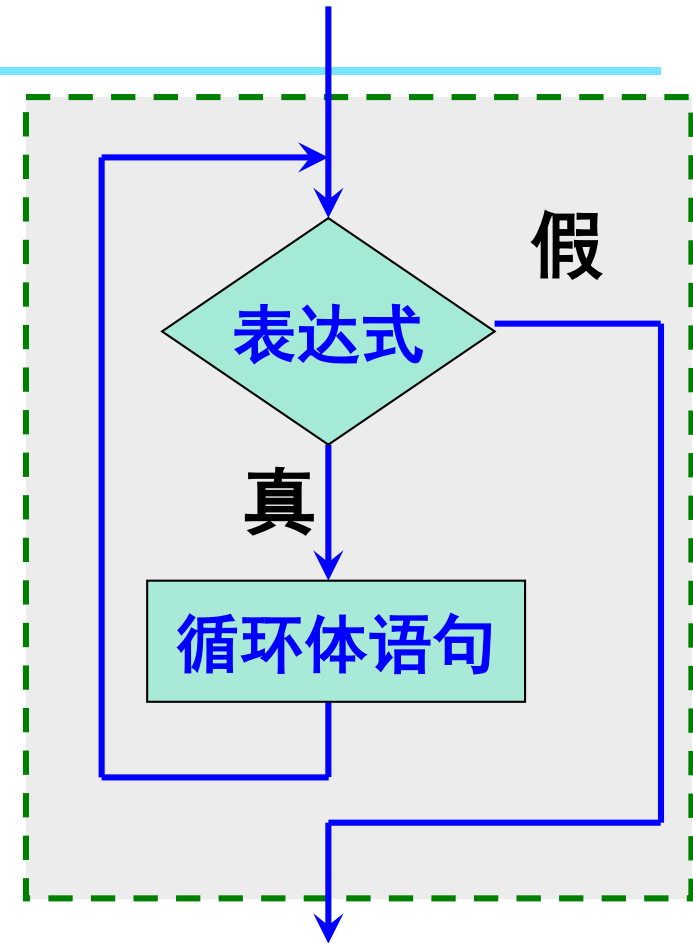


# while 循环

## ● while 循环

```
while 条件表达式  
    循环体语句  
end
```

- 当条件表达式的值为真（非 0）时，执行循环体语句，直到条件表达式的值是假为止



- 循环语句可以嵌套使用
- 如果预先知道循环的次数，可采用 for 循环  
如果预先无法确定循环次数，则可使用 while 循环

# while 循环

**例：**数论中的一个有趣问题： $3n+1$  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？

```
n=input('请输入一个正整数 n: ');
x=n;
while x>1
    if mod(x,2)==0
        x=x/2;
    else
        x=3*x+1;
    end
    fprintf(' x=%d \n', x);
end
```

ex4while01.m

# while编程示例

## 例：计算 Matlab 中 eps 值

```
k=0;  EPS=1;
while (1+EPS) > 1
    EPS = EPS/2;
    k = k+1;
end
```

ex4while02.m

- 这个例子给出了计算 `eps` 的一种方法。
- 这里我们用大写 `EPS`，因此系统中 `eps` 的值不会被覆盖。当 `EPS=eps` 时，条件仍成立；而当 `EPS=eps/2` 时，条件不成立，退出循环，所以最后应有 `EPS=eps/2`。 `k` 用来记录循环次数。

# 循环的终止

---

## ● 循环的中断

### break

- 跳出循环体，终止循环
- break 只能跳出一层循环

### continue

- 结束本轮循环，执行下一轮循环

break 和 continue 通常需要与 if 语句配合使用

若想强行终止运行中的程序，可以使用 **Ctrl+c**

# pause

## ● 程序暂停

`pause`      % 暂停程序的运行，按任意键后继续  
`pause(x)`    % 暂停程序 `x` 秒

- 这里的 `x` 可以是小数

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

## ● 返回/退出

`return`

- **退出**正在运行的脚本或函数，通常用在函数文件中

# 编程示例

## 例：猜数游戏

首先由计算机随机产生一个  $[1, 100]$  之间的一个整数，然后由用户猜测所产生的这个数。

根据用户猜测的情况给出不同的提示，如果猜测的数大于产生的数，则显示 **Higher**，小于则显示 **Lower**，等于则显示 **You won!** 同时退出游戏。用户最多有 **7** 次机会

ex4for.m

`randi(T,m,n)`

`randi(T,n)`

产生  $1 \sim T$  间满足均匀分布的随机**整数**矩阵

$m=n$  时简写为 `randi(T,n)`

# 上机作业

## 1、已知程序：

```
x=1; while (x+x>x), x=x/2; end
```

试说明程序运行结束后，最后得到的  $x$  和倒数第二个  $x$  值分别为多少？  
若将程序改为下面的语句，结果又如何？

```
x=1; while (x+x>x), x=2*x; end
```

(在程序中将最后两个值输出来，程序取名 m51.m 中)

## 2、已知 $\sin(x)$ 的幂级数展开公式为

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \dots$$

试利用该公式计算  $\sin(\pi/2)$  和  $\sin(31\pi/2)$  的值（直到级数某项的绝对值小于  $10^{-16}$  为止），并与 Matlab 自带的  $\sin$  函数比较，误差分别多大？为什么？  
(程序取名 m52.m)

# 上机作业

---

3、用 while 循环实现猜数游戏

(程序取名 m53.m)

4、编写程序，用穷举法找出 1~100间的所有素数

(程序取名 m54.m，要求使用 while 循环，不能使用 isprime 函数)

5、编写程序，实现下面的功能

从键盘输入一个大于 1 的整数，然后求出不超过这个整数的最大素数。

(要求使用 for 循环和 if 语句，可以使用 isprime 函数)

(程序取名 m55.m)



# 上机作业

## 6、按下面的要求编写程序

学生的成绩有两种表示方法：等级制和百分制。对应关系如下：

等级	A	B	C	D	E
分数	90--100	80--89	70--79	60--69	0--59

请编写程序，实现它们之间的互换，即

(a) 要求从键盘输入一个分数，然后输出其对应的等级，要求用 if 语句实现，程序取名为：m56.m

(b) 从键盘输入一个等级，然后输出其对应的分数区间，要求用 switch 语句实现，程序取名为：m57.m

注：输入时要有提示，如“请输入成绩（百分制）：”

# 上机要求

---

## □ 上机要求

- 将所有文件作为附件，通过 foxmail 以邮件形式发给 `mhjs@system.mail`
- 邮件主题为：**机号-学号-姓名**，其中机号为 **两位数**
- 三个字段之间用英文状态下的减号连接
- 每个 M 文件的第一行添加一条注解语句：  
**% 机号-学号-姓名**