

# 《数学实验》第4讲

## 主要内容:

分支结构语句: if, switch

循环结构语句: for, while

控制语句: continue, break

编写函数: 函数语法基础、主函数与子函数

inline函数和匿名函数

## 结构化程序设计的三种语句结构：

- 顺序结构
- 分支结构
- 循环结构

## 要求：

- 在应用中根据逻辑计算的需要使用这三种结构
- 关键理解语句的执行流程

## 理解三种控制结构：

分析下面任务的执行流程，指出分别用哪种结构来描述更合理？

- 家庭准备午餐的步骤
- 教师根据学生成绩对学生进行分类
- 教师批阅一个班学生的试卷

## 以上任务分别适合用：

顺序结构、分支结构、循环结构来描述。

(1) 有一个选择分支的一般形式:

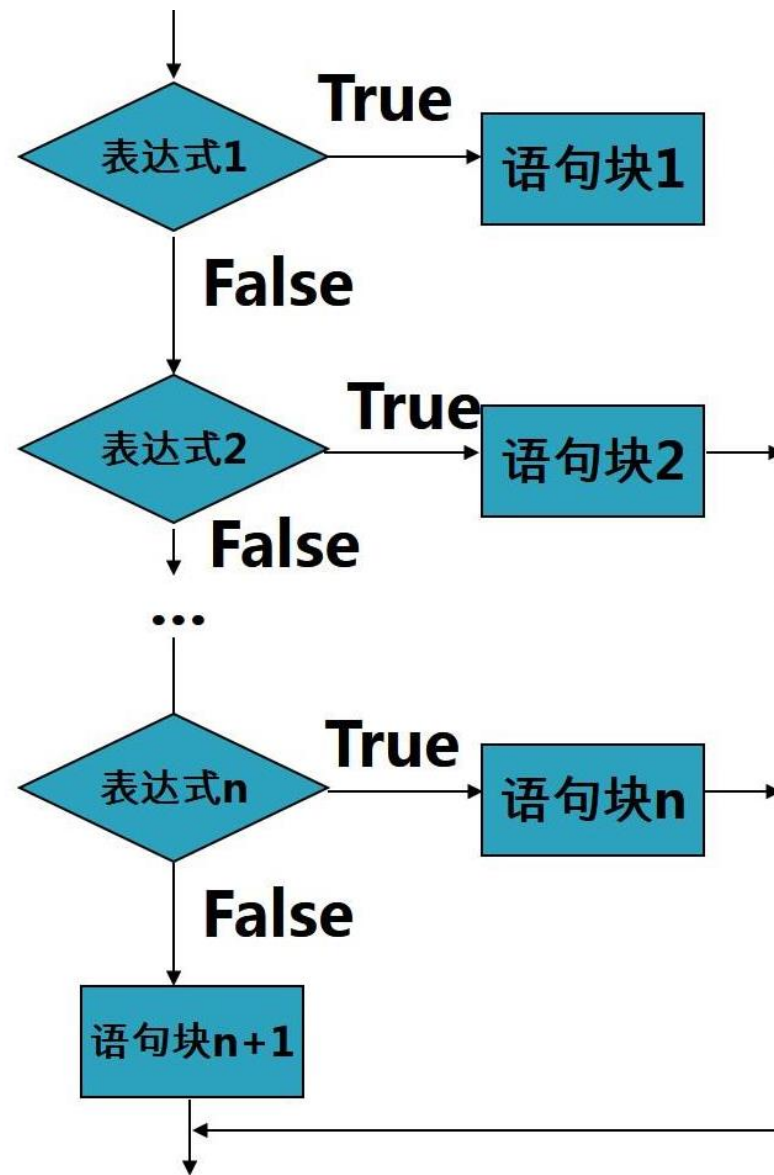
```
if 逻辑表达式,  
    语句块
```

```
end
```

如果在“逻辑表达式”里的所有元素**为真**, 就执行if和end语句之间的“语句块”

(2) 有两个或更多的选择分支一般形式

```
if 逻辑表达式1  
    语句块1  
elseif 逻辑表达式2  
    语句块2  
elseif .....  
    .....  
else  
    语句块  
end
```



**例1. 输入一个姓名, 如果姓名第1个字母是'a'-'h',则输出Hello <姓名>; 否则输出'Welcome, <姓名>'**

```
name=input('input one name:', 's')
if name(1)>='a' & name(1)<='h',
    disp(['Hello ', name]) % 拼接字符数组
else
    disp(['Welcome, ', name]) % 拼接字符数组
end
```

**运行输出:**

```
input one name: Li San
name = Li San
Welcome, Li San
```

**例2.** 请对输入的成绩给出等级描述: [90,100] 优异,[80,90) 优秀,[60,80) 中等, [0,60) 未及格.

```
s = input('请输入成绩')
if s >= 90
    disp('成绩优异')
elseif s >= 80 & s < 90
    disp('成绩优秀')
elseif s >= 60 & s < 80
    disp('成绩中等')
else
    disp('未及格')
end
```

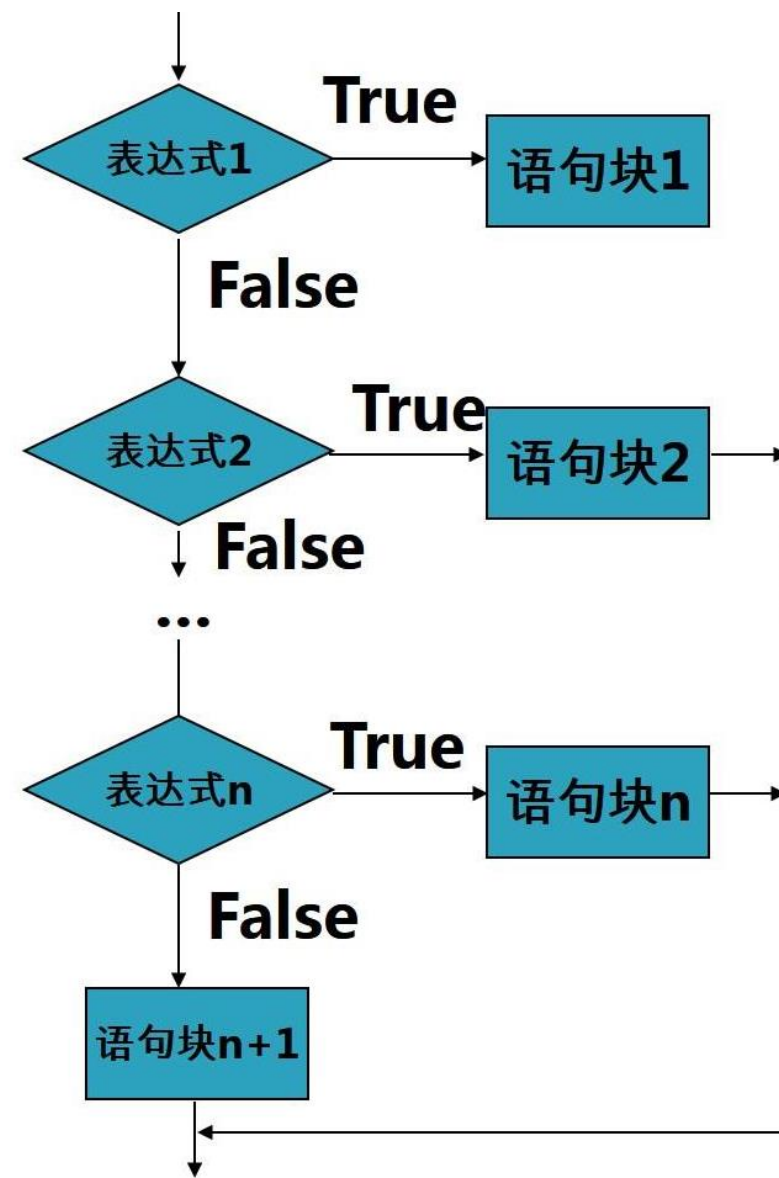
或重新改写:

```
s = input('请输入成绩')
if s >= 90
    disp('成绩优异')
elseif s >= 80
    disp('成绩优秀')
elseif s >= 60
    disp('成绩中等')
else
    disp('未及格')
end
```

多分支语句, 基本用法:

```
switch switch_expr  
  case case_expr1,  
    语句块1  
  case case_expr2,  
    语句块2  
  case {case_expr3, case_expr4, ...}  
    语句块2  
  otherwise, %本子句为可选的  
    语句块3  
end
```

switch\_expr: 一般为标量或字符串.



分析下列语句功能:

```
name='abc';
```

```
switch name(1) % 根据name第1个字符串进行不同处理
```

```
    case {'a','b','c','d','e','f','g','h'},
```

```
        disp(['Hello, ', name])
```

```
    otherwise,
```

```
        disp(['Welcome, ', name])
```

```
end
```

此段程序根据name字符数组第1个字符进行处理。

## ❖ 实践题(5 min)

- 请对输入的成绩判断等级  
([90,100],[80,89],[70,79],[60,69],[0,59])
- 用switch语句实现 (请自己定义等级的文本)

- 思路: 将分数除以10取整, 再利用分支语句(switch语句)计算. (注: 脚本文件编写)

```
grade = input('请输入成绩');  
grade = fix(grade/10) %取整  
switch grade  
    case {9,10},    disp('成绩优异')  
    case {8},      disp('成绩优秀')  
    case {6,7},    disp('成绩中等')  
    otherwise,     disp('没有及格')  
end
```



在什么情况下设计程序用循环结构?

• 举例:

- 1. 模拟超市一年的运行情况—超市每天的运行流程基本一致
- 2. 找出矩阵满足条件的行向量—对矩阵各行向量的处理相同

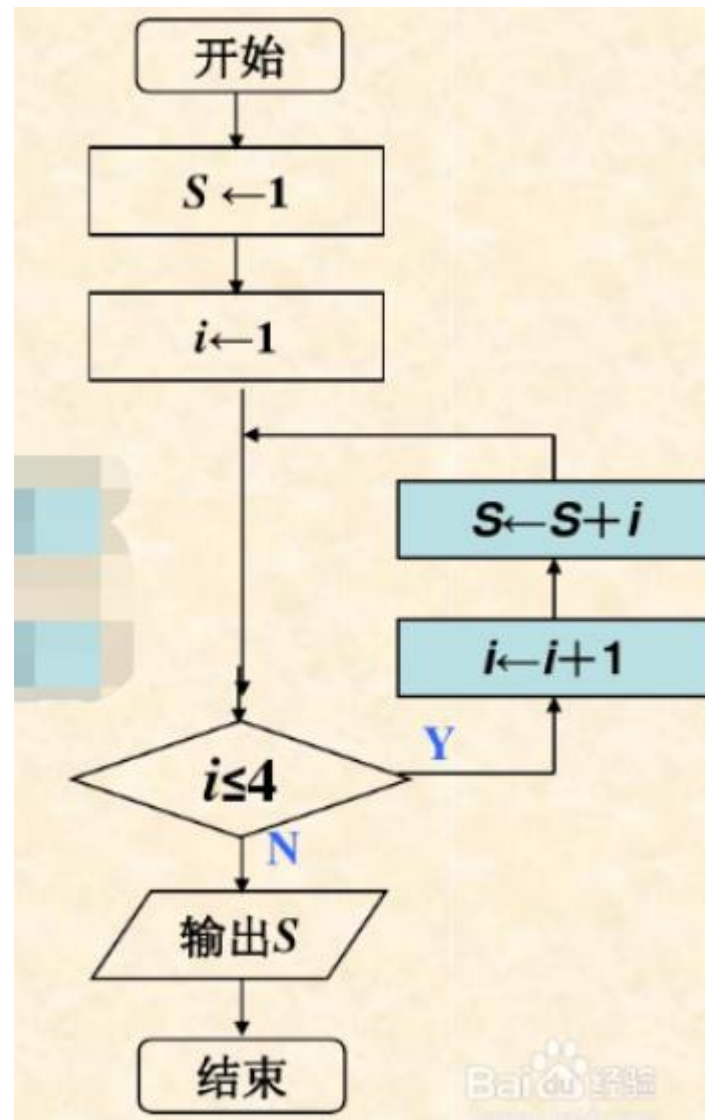
预先设定循环次数, 重复执行语句块

在什么情况下设计程序用循环结构?

• 示例:

表示循环计算,

$$S=1+2+3+4+5$$



## 常用格式:

```
for 变量=初值:步长:终值  
    语句块  
end
```

## 运行流程示例1

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

**注:** 完成以上语句功能一般用下列语句: **`s=sum(1:100)`**

**例1.** 给定一个整数 $n=10$ , 然后输出如下形式的字符串

- line 1
- line 2
- ...
- line 10

```
n=10;  
for i=1:n  
    s=sprintf('line %d',i);  
    disp(s)  
end
```

## ❖ 实践题(5 min)

**例.** 编程计算斐波那契数列中的元素, 然后输出 $F_{20}$ .

$$\begin{cases} F_n = F_{n-1} + F_{n-2}, n \geq 3 \\ F_1 = 1, F_2 = 1 \end{cases}$$

```
n=input('input n:=');% 输入总项数
f(1)=1; % 初始化前2项
f(2)=1;
for k=3:n % 依次计算后面n-2项
    f(k)=f(k-1)+f(k-2); % 按递推方程计算
end
f % 显示数组元素
```

例3. 构造一个5行4列的矩阵m, 元素

$$m_{ij} = 100i + j$$

- 思路: 用嵌套的for循环语句.

```
for i=1:5 % 遍历行
    for j=1:4 % 遍历列
        m(i,j)=100*i+j;
    end
end
```

分析下列代码含义, 指出两个循环语句的差异:

```
for i= 1:length(x)
    %循环体
```

```
end
```

```
for i=x
    %循环体
```

```
end
```

关键差异:

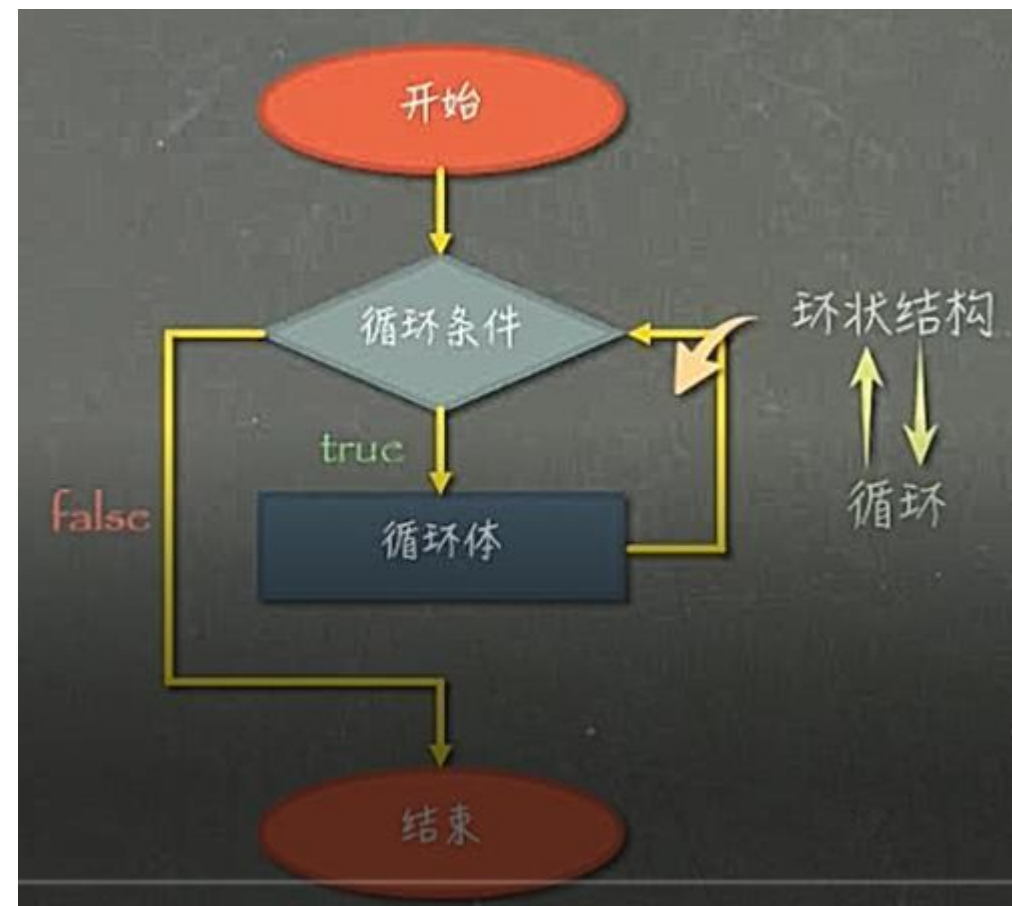
**i=1:length(x)**

**i=x**

•注: 循环变量i的取值有差异

- 循环次数已知，常用for循环语句
- 循环次数不确定，常用while循环语句
- 一般语法：  
    while 逻辑表达式  
        循环语句块  
    End
- 当“逻辑表达式”里的所有元素为真，就执行while和end语句之间的“语句块”。

## • 一种循环语句示意图



**例1.** 用while语句输出如下50行字符串

- iteration 1
- iteration 2
- ...
- iteration 50

注意实现细节:  
循环变量的使用  
循环变量的初始化  
循环体变量的更新

**编程实现1:**

```
c=0;  
while c<50,  
    c=c+1;  
disp(sprintf('iteration %d',c))  
end
```

**编程实现2:**

```
c=1;  
while c<=50,  
    disp(sprintf('iteration %d',c))  
    c=c+1;  
End
```

**例2.** 找出 $x^2-2y=1$  ( $x,y\geq 1, y\leq 1000$ ) 所有正整数解, 并将所有正整数解保存到矩阵M中 (每行存储一个解), 最后输出解的个数及所有的解.

**伪算法:**

- $y=1; M=[];$
- **while**  $y\leq 1000$
- 根据 $y$ 计算 $x$ ;
- **if**  $x$ 为整数, 则
- 将 $x,y$ 保存到矩阵M.
- **end**
- $y=y+1$
- **end**

**程序实现:**

```
M = []; y = 1;
while y <= 1000, % 遍历y的范围
    x = sqrt(1+2*y);
    if fix(x) == x, % 判断x是否为整数
        M = [ M; x y]; % 保存找到的一个解
    end
    y = y + 1;
end
disp(sprintf('解的个数=%d', size(M, 1)))
disp('所有解:')
disp(M)
```

**结果: 找到21个解.**

- continue通常用于for, while循环语句中,与if语句一起使用.  
continue **跳过**本次循环, 继续执行下一次循环
- 典型使用格式:

```
for x=array (或 while 条件表达式) ,  
    语句块1  
    if 条件表达式,  
        continue,  
    end  
    语句块2  
end
```

- break **中止**本次循环, 跳出最内层循环
- 典型使用格式:

```
for x=array (或 while 条件表达式) ,  
    语句块1  
    if 条件表达式,  
        break,  
    end  
    语句块2  
end
```



- **例3.** 请反序输出字符串中的字母。如“abcde”反序后为“edcba”.
- 用循环完成字符串反序处理

```
str=input('请输入字符串： ','s')
tmpstr=str;
n = length(str);
i=1;
while i<=n
    str(n-i+1)=tmpstr(i);
    i=i+1;
end
```

注：本问题可调用函数`flip1r(str)`完成.

# 思考题（课下复习）：



**例1.** 设银行年利率为4.25%。将10000元钱存入银行，问多少年后存单价值连本带利翻一番？ $a_k = (1 + 4.25\%)a_{k-1}$ ,  $a_0 = 10000$ .

**分析：**

$r = 0.0425$

$a_1 = (1 + r) a_0$  %a1 第1年后本金+利息之和

$a_2 = (1 + r) a_1$  %a2 第2年后本金+利息之和

$a_3 = (1 + r) a_2$  %a3 第3年后本金+利息之和

**编写程序：**

```
a0=10000;
```

```
a=a0; r=0.0425;
```

```
for i=1:50, % 假设50年内可以翻番，最多循环50次
```

```
    a = a*(1+r); % 计算第i年末存单价值
```

```
    if a >= 2*a0,
```

```
        disp(sprintf('存了%d年终于翻番了', i))
```

```
        break; %结束迭代
```

```
    end
```

```
end
```

**思考：**如果要存储每年存单的价值，如何修改程序？

**提示：**使用数组存储。

**基于循环结构的穷举法！！**

## 程序文件类型

- 程序文件存储格式：纯文本文件
  - 可用Matlab程序编辑器、记事本、写字板等软件编辑。
- 程序文件类型：1. 脚本文件；2. 函数文件

### 示例：

```
A = rand(3);  
[v,d] = eig(A);  
if max(diag(d))>1  
    disp('max eigenvalue >= 1')  
else  
    disp('max eigenvalue <1 ')  
end
```

图1. 脚本程序

```
function r = fun(x)  
    r = x^2 +1;
```

图2. 函数程序

## 运行程序文件的方法

### (一) 脚本程序文件

假设编写的脚本程序文件名为myfile.m。运行方法：

#### 1) 命令窗口输入程序文件名（不含文件类型扩展名）

```
>> myfile
```

#### 2) 在程序文件中调用 ... ..

```
myfile
```

```
... ..
```

#### 3) 用MATLAB程序编辑器打开程序文件，按F5键运行

### (二) 函数程序文件

函数程序文件与脚本程序文件的运行方法类似，应根据需要在调用时设置好该函数的输入参数、输出参数。

## 运行程序文件的典型错误处理

### 1. 常见运行错误：“Undefined function or variable”

示例：假定编写了一个程序myprg.m，保存在文件夹“E:\”中；

在Matlab命令窗口输入程序名myprg提示出错

```
>> myprg
```

??? Undefined function or variable 'myprg'.

原因： myprg不在Matlab的搜索路径中。

解决办法：将文件夹“E:\”添加到Matlab搜索路径中。

### 2. 文件内添加路径：

addpath E:\mywork      % 以命令方式添加

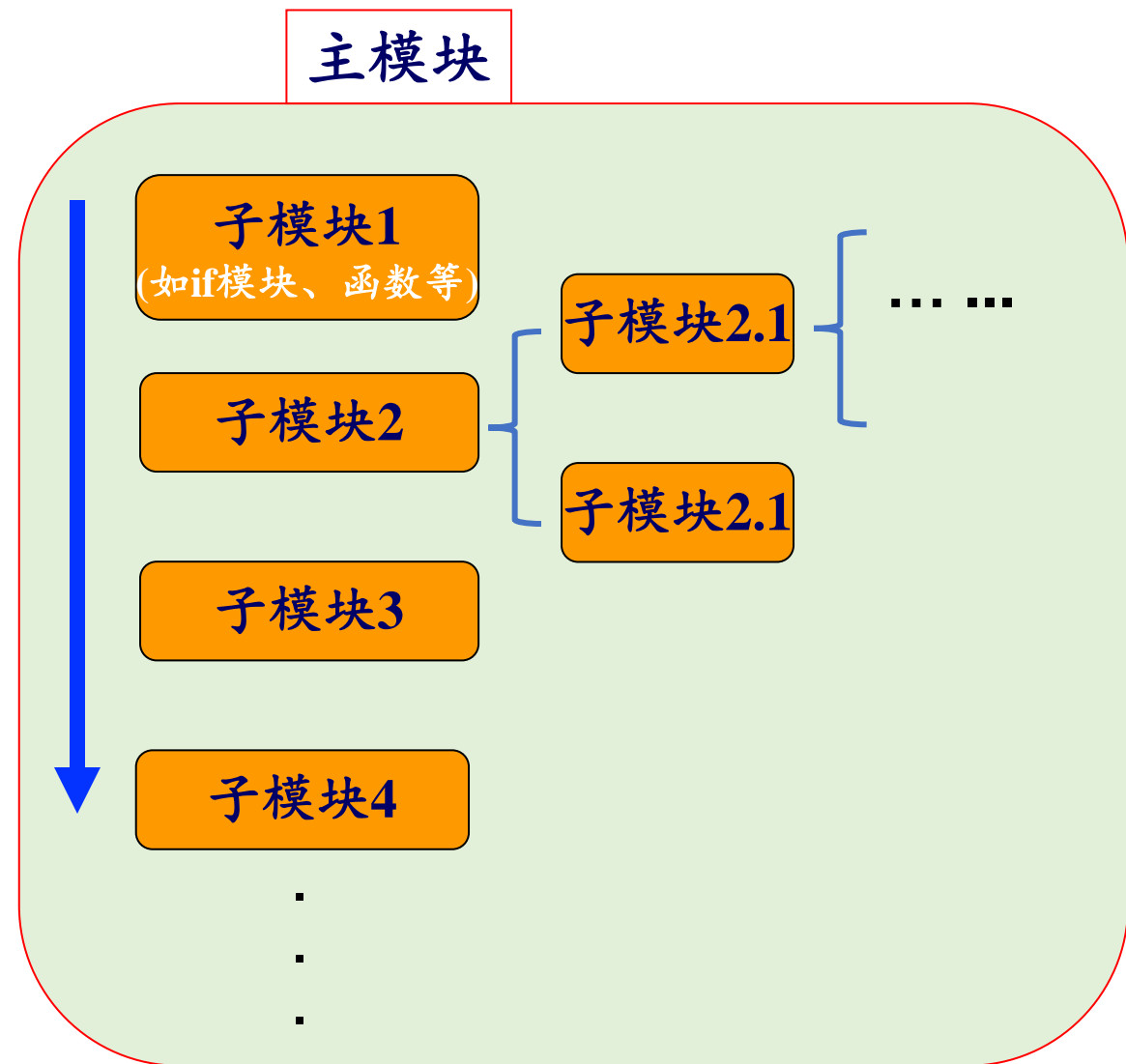
addpath('E:\mywork')    % 以函数调用方式添加

## 程序模块化设计与函数编程

使用MATLAB语言进行结构化程序设计的一般步骤：

1. 将功能划分为多个模块；
2. 为每个模块编写一个函数文件；
3. 为整个功能实现编写一个程序

(可以使用Matlab函数文件或脚本文件)。



- 函数 (function) 语法是Matlab语言的重要组成部分，是程序模块化设计的需要。
- 函数一般都有输入参数、输出参数，并完成一定的功能。
- 函数名命名规范参考变量名命名规范
  - (1) 必须以字母开头
  - (2) 区分大小写
  - (3) 函数名可由字母、数字、下划线组成

`function` [输出参数列表]=函数名(输入参数列表)  
参数用逗号分隔

例如：编写函数文件`funname.m`，该文件第一行如下：

```
function [r1, r2, r3]=funname(a1, a2, a3, a4)
```

4个输入参数，3个输出参数

**注意：**1. 保存函数的文件名一般与程序中定义的函数名相同；否则，调用该函数时以保存函数的文件名作为函数名来调用。

2. 函数名要有一定实际含义，便于记忆，同时避免与系统内部函数名相同。可以通过“`help 函数名`”检查该函数是否为系统函数。

• 无输入参数、输出参数的函数定义  
**示例：**

1. 无输入参数示例：

```
function [r1, r2, r3]=funname
```

2. 无输出参数示例：

```
function funname(a1, a2, a3, a4)
```

3. 无输入参数、无输出参数示例：

```
function funname
```



**例1.** 编写一个函数求任意两个自然数之间（包含两个自然数）所有自然数的和。

**分析与设计：**

1. 输入参数2个，输出参数1个（返回求和结果）假设不限制两个输入参数的顺序：

$\text{fun}(a, b), \text{fun}(b, a)$

2. 计算 $a$ 与 $b$ 之间自然数的和

$$a \leq b: \frac{(a+b)(b-a+1)}{2}$$

$$a \geq b: \frac{(a+b)(a-b+1)}{2}$$

3. 函数名这里取为sumbetween

程序sumbetween.m

```
function r=sumbetween(a,b)

%sumbetween.m 任意自然数a和b之间（含a和b）所有整数的和

if a <= b,
    r = (b-a+1) * (a+b) / 2;
else
    r = (a-b+1) * (a+b) / 2;
end
```

调用函数sumbetween

```
>>sumbetween(1,100)
ans =
    5050
```

系统变量nargin,nargout和语句return, pause

- nargin, nargout 分别表示函数调用时的输入参数个数，输出参数个数。
- return 返回调用函数
- pause 暂停运行，按任意键执行
- pause(n) 暂停n秒，如pause(0.5)

**举例：**编写函数文件myfun.m

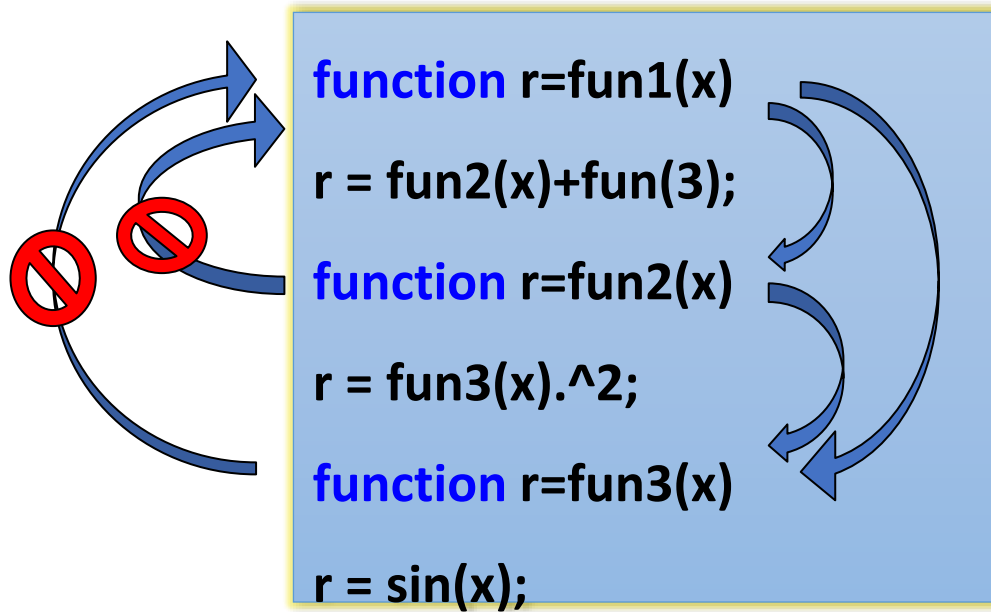
```
function [r, v]=myfun(x)
r=x.^2;      %计算第1个输出参数
if nargin>=2,
    v = 2*x;%计算第2个输出参数
end
```

## 主函数与子函数应用场景说明

- 一个模块可能需要划分为多个子模块。如果每个模块都编写为一个单独的函数文件，则函数文件会较多。
- MATLAB语言允许编写一个函数文件，可以定义多个函数。
- 在这种情形下，第一个函数实现主模块，后续函数为主模块的子模块的编程实现。

### 函数文件内调用规则：

- 主函数可以调用子函数
- 子函数可以调用子函数
- 子函数不能调用主函数
- 子函数可以调用其他程序文件



例1. myfun.m (主函数： myfun; 子函数1个： mysum)

```
function s=myfun           %文件名myfun.m
num=input('输入一个正整数: ')
s=mysum(num)
disp(sprintf('从1到%d的所有整数的和为%d', num, s))

function s=mysum(num)
%求1到num所有整数相加之和
s=sum(1:num);
```

运行示例:

```
>> myfun
```

输入一个正整数: 100

```
num =
```

```
    100
```

从1到100的所有整数的和为5050

## 例2. 编写函数计算分段函数

$$f(x) = \begin{cases} x^2 + x - 6, & x < 0 \\ x^2 - 5x + 6, & 0 \leq x < 10 \\ x^2 - x - 1, & x \geq 10 \end{cases} \quad \text{在多个点的函数值。}$$

分析：

1. 设计主函数 `function vals=calfun(x)`

向量x存储自变量的取值

主函数通过遍历输入向量x所有点计算出  
函数值向量vals

2. 编写子函数getval计算一个点的函数值

`function r=getval(x)`

x为标量

```
function vals=calfun(x)
vals=zeros(size(x)); %初始化
for i=1:length(x),
    vals(i)=getval(x(i));
end

function r=getval(x)
if x<0,
    r = x^2+x-6;
elseif x<10,
    r = x^2-5*x+6;
else
    r = x^2-x-1;
end
```

命令窗口运行演示

```
>> calfun(linspace(-10,20,5))
ans =
    84.0000    -2.2500     6.0000
   142.7500   379.0000
```

## inline函数

### 用法:

- `inline(expr)` 根据expr建立内联函数, 函数自变量符号根据表达式自动确定;
- `inline(expr, arg1, arg2, ...)` 定义时指定自变量符号.
- `inline(expr, N)` 自变量符号为  $x, P1, P2, \dots, P_N$

### 示例:

```
f = inline('2*x.*x-x+sin(exp(x))');  
x = linspace(-10,10,100);  
plot(x,f(x))
```

## inline函数示例:

- `f1 = inline('t^2')`      % 系统自动确定参数
- `f2 = inline('x*y*z')` % 系统自动确定参数
- `f3 = inline('x*y*z','x','y','z')` % 指定参数
- `f4 = inline('x^P1+x^P2',2)`      % 指定参数为x, P1, P2

## 命令窗口运行演示

```
>>f = inline('x.^2+y.^3') , v=f(1:3,5:7)
f =
    内联函数:
    f(x,y) = x.^2+y.^3
v =
    126    220    352
```

**例1.** 创建inline函数计算函数  $y = \sin x + \cos x + e^x \sin x$

在数组  $-2\pi:0.01:2\pi$  中各点的函数值.

## 编写程序

```
f=inline('sin(x)+cos(x)+exp(x).*sin(x)','x')  
x=-2*pi:0.01:2*pi;  
value=f(x) % 存储函数值
```

**注意:**

- 使用inline函数创建初等函数的程序很简洁;
- 要计算多个点的函数值时, 注意四种数组运算符的使用: `.*`, `./`, `.\`, `.^` (乘、左除、右除、幂运算).



定义匿名函数使用@符号: @(参数列表)(函数表达式)

示例: 定义匿名函数  $z=x^2$

## 编写程序

```
>> f=@(x) (x.^2)
f =
    @(x) (x.^2)
```

说明: 该函数可计算输入数组每个元素的平方, 并返回与输入参数同维数的数组。

## 调用示例

```
>> val=f(1:5)
val =
     1     4     9    16    25
```

**例2.** 请定义匿名函数  $f$ , 并计算下列分段函数在点  $x=-2:0.1:2, y=-2:0.1:2$  处的函数值, 调用方式为  $f(x, y)$ 。

$$f(x, y) = \begin{cases} x + y, & x^2 + y^2 \leq 1 \\ x - y, & x^2 + y^2 > 1 \end{cases}$$

## 编写程序

```
f=@(x,y) ((x.^2+y.^2<=1) .* (x+y) + ((x.^2+y.^2)>1) .  
* (x-y));  
x=-2:0.1:2;  
y=-2:0.1:2;  
v=f(x,y);
```

**练习. 判断下列程序正误，如果有错误请改正。**

**编写匿名函数计算 $z=x^2+y^2$ 在多个点的函数值。**

**编写程序**

```
f=@(x)(x^2+y^2)
val=f(1:5,11:15)
```

**错误之处：**缺少参数y;  
使用的^运算符不支持数组运算

**分析：**定义f的运算符^表示输入参数只应为标量或者方阵。  
如果输入参数为标量，则计算标量的平方，  
如果输入参数x,y为方阵，则^为矩阵的幂。

**改写：**如果要计算多个点的函数值，要用.^运算符。因此，仅需将程序的第一句修改为：

```
f=@(x,y)(x.^2+y.^2)
```

# 学到了什么？



分支结构语句: if, switch

循环结构语句: for, while

控制语句: continue, break

编写函数: 函数语法基础、主函数与子函数

inline函数和匿名函数