

# 第七节 matlab的程序设计

—— matlab语言称为第四代编程语言，程序简洁、可读性很强而且调试十分容易。  
是matlab重要组成部分。





# m文件简介

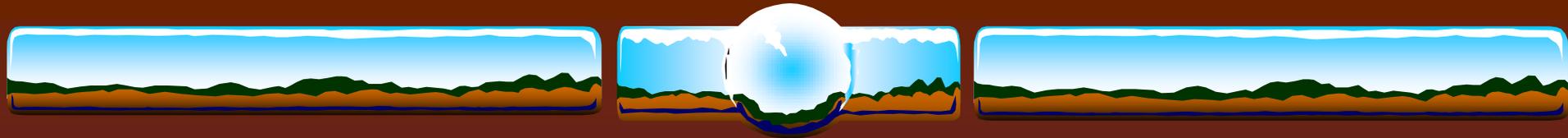
- ❖ 用户如想灵活应用matlab去解决实际问题，充分调用matlab的科学技术资源，就需要编辑m文件
- ❖ 包含matlab语言代码的文件称为m文件，其扩展名为m。
- ❖ 编辑m文件可使用各种文本编辑器。



# 一、matlab的工作模式

## ❖ 指令驱动模式

- 通常matlab以指令驱动模式工作，即在matlab窗口下当用户输入单行指令时，matlab立即处理这条指令，并显示结果，这就是matlab命令行方式。
- 命令行操作时，matlab窗口只允许一次执行一行上的一个或几个语句。



例如:

$x1=0:10$

$x1 =$

0 1 2 3 4 5 6 7 8 9 10

$x2=0:3:11$

$x2 =$

0 3 6 9

$x3=11.5:-3:0$

$x3 =$

11.5000 8.5000 5.5000 2.5000



$x1=0:10,x2=0:3:11,x3=11.5:-3:0$

$x1 =$

0 1 2 3 4 5 6 7 8 9 10

$x2 =$

0 3 6 9

$x3 =$

11.5000 8.5000 5.5000 2.5000

- ❖ 在matlab窗口输入数据和命令进行计算时，当处理复杂问题和大量数据时是不方便的。因此应编辑m文件。

- 
- 命令行方式程序可读性差，而且不能存储，对于复杂的问题，应编写成能存储的程序文件。

## ❖ m文件模式

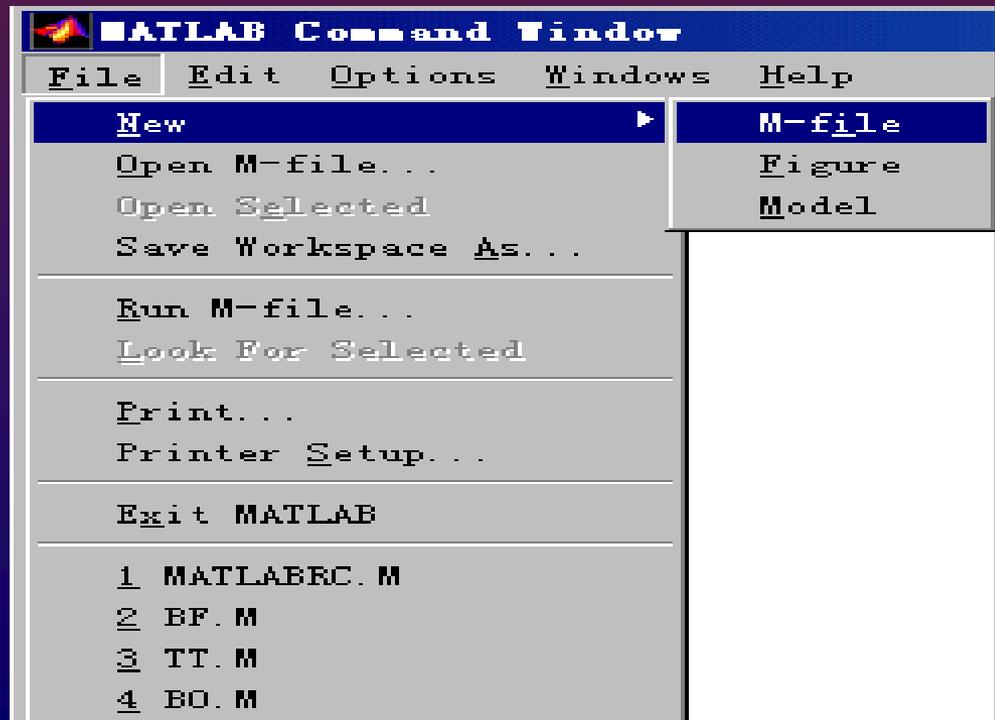
- 将matlab语句构成的程序存储成以m为扩展名的文件，然后再执行该程序文件，这种工作模式称为程序文件模式。
- 程序文件不能在指令窗口下建立，因为指令窗口只允许一次执行一行上的一个或几个语句。

## 二、m文件的创建

❖ m文件的类型是普通的文本文件，我们可以使用系统认可的文本文件编辑器来建立m文件。如dos下的edit， windows的记事本和word等。

❖ 具体的创建方法：

1. 在matlab命令窗口点击file 菜单  
new → m-file



无标题 - 记事本

文件(F) 编辑(E) 搜索(S) 帮助(H)

```
t=0:2*pi/180:2*pi;  
y=sin(t);plot(t,y)
```

另存为

保存在(I): 桌面

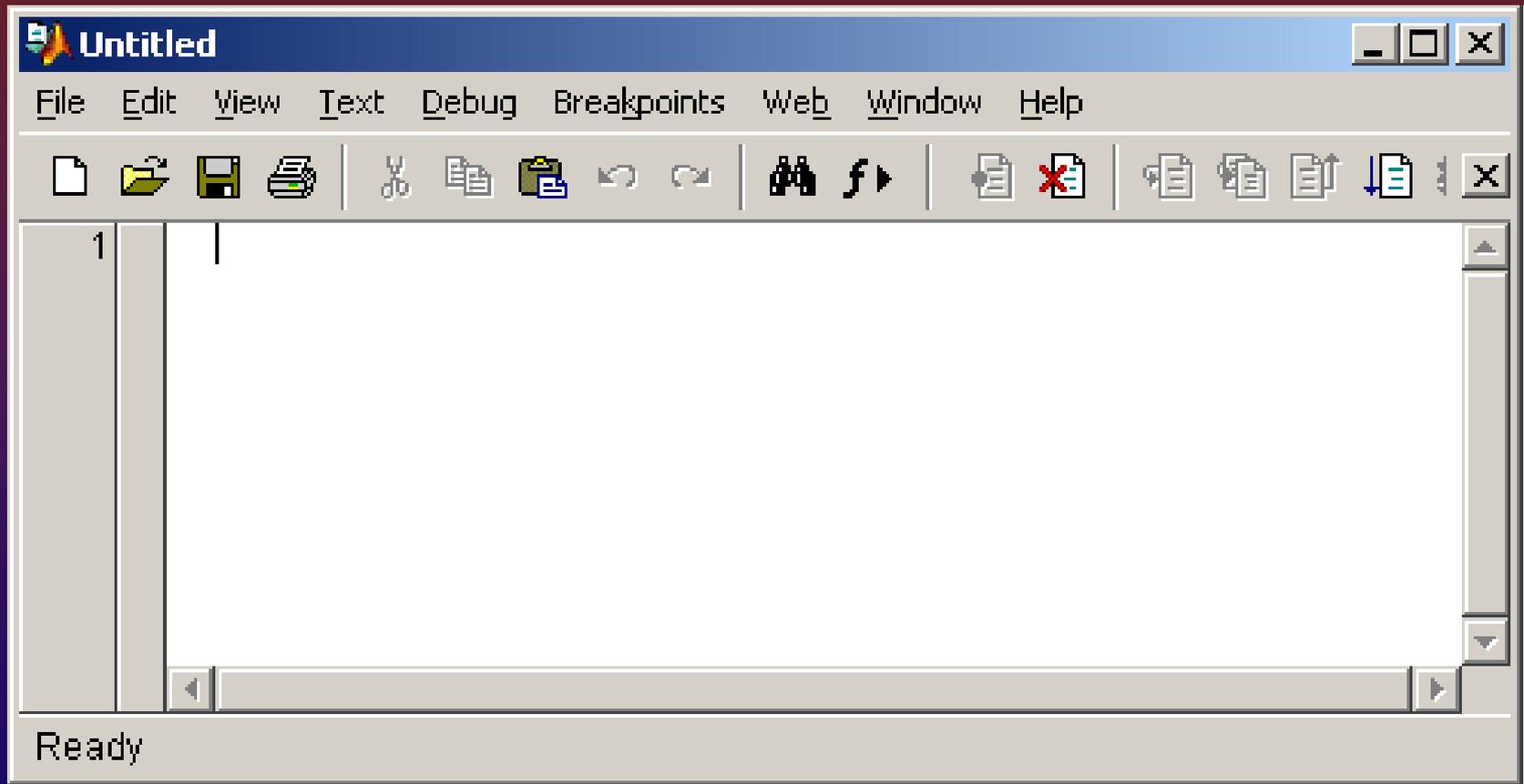
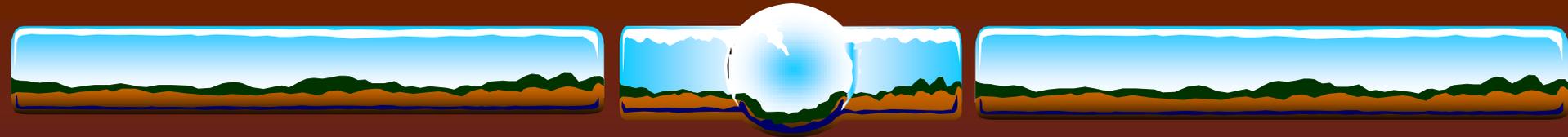
- 我的电脑
- 我的文档
- 网上邻居
- 我的公文包

文件名(N): aa.m

保存类型(T): 文本文档

保存(S)

取消





2. 点击file — Save Workspace As

将工作空间中的内容存入文件。

3. !edit —— 用dos 下edit 编辑器编制m文件。

4. edit —— 用matlab自带的编辑器编制m文件。

5. word —— 用word编辑m文件。

### ❖ m文件编辑器的设定

m文件编辑器可自己设定，设定方法是matlab命令窗口 options 菜单下点击 Edit Preference 选定m文件编辑器。



m文件编辑器缺省设置是

c:\windows\notepad.exe

还可设成: c:\windows\write.exe

c:\windows\command\edit.com

c:\program file\office\winword.exe

注意: 所创建的m文件必须纳入matlab搜索路径后, 才能在matlab命令窗口运行。

## ❖ matlab搜索路径的设置

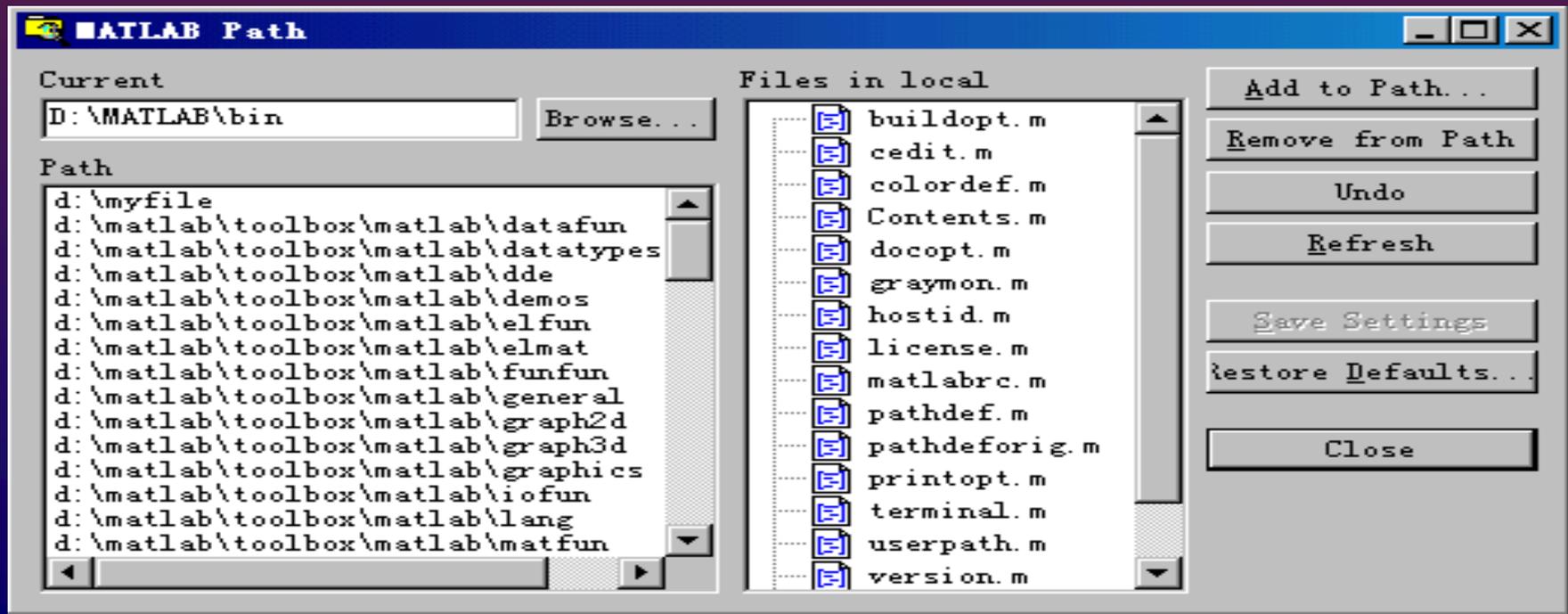
matlab 缺省路径是c:\matlab\bin

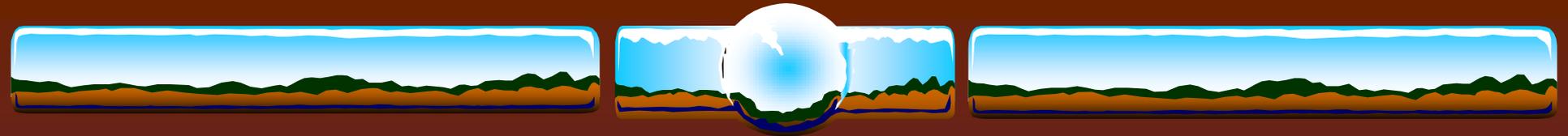
1. cd c:\myfile

2. path(path,'c:\myfile')

3. 打开matlabrc.m文件，将c:\myfile加上即可。

4. 用pathtool路径浏览器设定当前目录。





## 三、matlab文件的类型

### 1. 数据文件 .mat

- ❖ mat文件是matlab以标准二进制格式保存的数据文件，可将工作空间中有用的数据变量保存下来。
- ❖ mat文件的生成和调用是由函数save和load完成的。

例如：`load gatlin`  
`image(X)`

colormap(gray); axis equal; axis('off')





## 2. m文件

m文件的语法类似于c语言，但又有其自身特点。它只是一个简单的ASCII码文本文件，执行程序时逐行解释运行程序，**matlab**是解释性的编程语言。

m文件有两类①独立的m文件 — 称命令文件  
②可调用m文件 — 称函数文件

### (1). 命令文件 — 简单的m文件

命令文件实际上是一串指令的集合，与在命令窗口逐行执行文件中的所有指令，其结果是一样的。没有输入输出参数。



❖ 命令文件包括两部分：注释文件和程序文件

`%RANK`      Number of linearly independent rows or columns.

`% K = RANK(X)` is the number of singular values of `X`

`%` that are larger than `MAX(SIZE(X)) * NORM(X) * EPS`.

`% K = RANK(X,tol)` is the number of singular values of `X` that  
`%` are larger than `tol`.

`% Copyright (c) 1984-94 by The MathWorks, Inc.`

```
s = svd(x);  
if (nargin == 1)  
    tol = max(size(x)) * max(s) * eps;  
end  
r = sum(s > tol);
```

程序部分



## (2).matlab内置函数文件

matlab自定义的函数文件称内置函数文件

调用内置函数的方法：使用函数名并给出相应的入口、出口参数即可。

例如：sin.m函数——用type sin查不到。

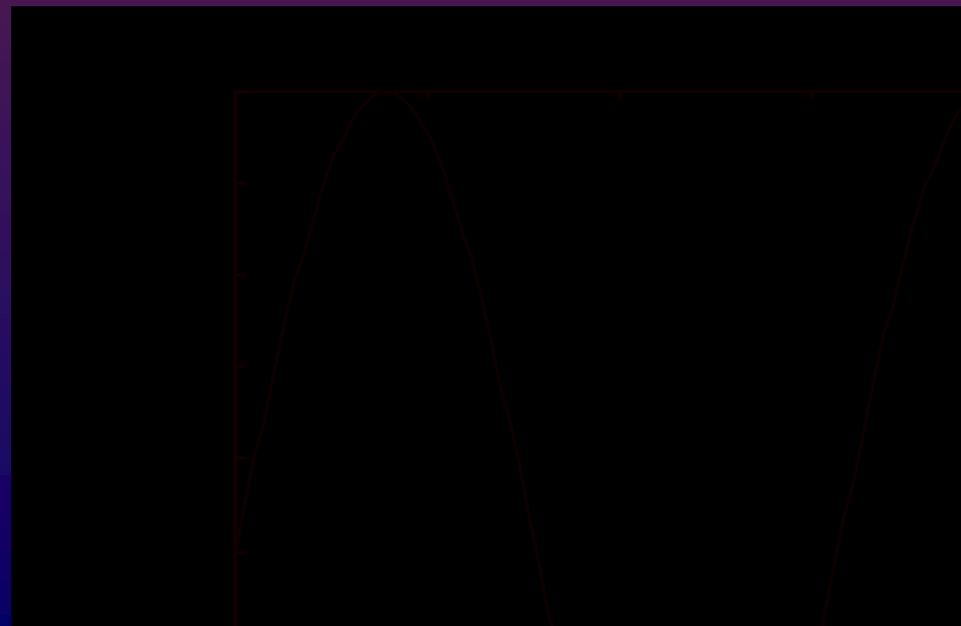
调用格式： $y=\sin(2*\underline{x})$

实际应用中：

```
x=0:2*pi/180:2*pi;
```

```
y=sin(2*x)
```

```
plot(x,y)
```





又如  $\text{sqrt}(x)$  —— 求 $x$ 的平方根

$\text{abs}(x)$  —— 求 $x$ 的绝对值

$\text{det}(a)$  —— 求行列式的值

等等。 $\text{matlab}$ 拥有大量的内置数学函数，供我们在数值运算和符号运算中调用。

(3). 函数 $m$ 文件——需要输入变量，返回输出变量

- ❖  $\text{matlab}$ 用户可以根据需要编辑自己的 $m$ 文件，它们可以像库函数一样方便的调用，从而极大地扩展了 $\text{matlab}$  的能力。

- 
- ❖ 对于某一类特殊问题，如创建了许多m函数文件，则可形成新的工具箱。
  - ❖ 这种用matlab语言创建定义新的matlab函数的功能，正体现了matlab语言强大的扩展功能。



## 函数m文件的格式:

`function` 返回变量=函数名 (输入变量)

注释说明语句段

程序语句段

## 特定规则:

1. 函数m文件第一行必须以单词`function`作为引导词, 必须遵循如下形式:

`function` <因变量>=<函数名>(<自变量>)

2. m文件的文件名必须是<函数名> .m。
3. 程序中的变量均为局部变量, 不保存在工作空间中。其变量只在函数运行期间有效。



#### (4). 系统文件——s函数

用于描述系统运动的专用函数，是特殊的 m 文件。

- s 函数有3类：
- ① 由simulink结构图自动创建
  - ② 可用c语言程序设计mex文件直接定义
  - ③ 用函数m文件编写

s函数一旦创建，即可在框图中使用，也可在文件中调用。



## S函数的调用格式



S函数与函数m文件类似，只是输入、输出变量是限定的。



## (4). 函数句柄

函数句柄是matlab6特有的语言结构，其优点：

- ❖ 方便地实现函数间互相调用
- ❖ 兼容函数加载的所有方式
- ❖ 拓宽子函数包括局部函数的使用范围
- ❖ 提高函数调用的可靠性
- ❖ 减少程序设计中的冗余
- ❖ 提高重复执行的效率
- ❖ 数组、结构数组、细胞型数组结合定义数据



## 函数句柄的创建和显示

定义函数句柄的定义只需在提示符@后添加相应函数的函数名

```
f_h=@plot
```

```
f_h =
```

```
@plot
```

函数句柄的内容通过functions显示

```
functions(f_h)
```

```
ans =
```

```
function: 'plot'
```

```
type: 'overloaded'
```

```
file: 'MATLAB built-in function'
```

```
methods: [1x1 struct]
```



# matlab的函数类型

simple —— 未加载的内部函数、m文件

overloaded —— 加载的内部函数、m文件

subfunction —— 子函数

constructor —— matlab类的创建函数

private —— 局部函数



# 函数句柄的调用和操作

可通过函数`feval`进行函数句柄的调用

`feval(函数句柄)`

`f_h1=@peaks`

`feval(f_h1)`

`feval(函数句柄, 参数列表)`

`f_h=@plot`

`feval(f_h,0:10,0:10)`



## 函数句柄与函数名字符串转换

func2str —— 函数名字符串转换函数句柄

```
funh=str2func('det')
```

```
funh =
```

```
@det
```

str2func —— 函数句柄转换函数名字符串

```
func2str(funh)
```

```
ans =
```

```
det
```



## 四、matlab的程序结构

matlab语言的程序结构与其它高级语言是一致的，分为顺序结构，循环结构，分支结构。

- (1).顺序结构 —— 依次顺序执行程序的各条语句
- (2).循环结构 —— 被重复执行的一组语句，循环是计算机解决问题的主要手段。

循环语句主要有：

① for — end



语法为:

```
for i = 表达式 (表达式为一个向量)
    可执行语句
end
```

例: 利用for循环求 $1! + 2! + 3! + \dots + 5!$  的值

```
sum=0;
for i=1:5
    pdr=1;
    for k=1:i
        pdr=pdr*k;
    end
    sum=sum+pdr;
end
```

```
sum =
    153
```



## ② while — end 循环

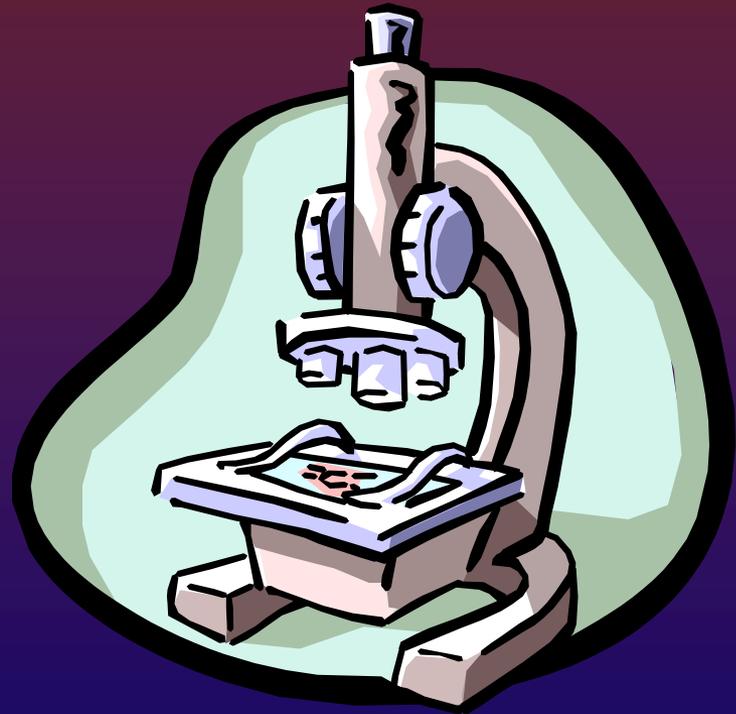
while循环将循环体中的语句循环执行不定次数。

语法为：**while** 表达式  
循环体语句  
**end**

表达式一般是由逻辑运算和关系运算以及一般运算组成的，以判断循环的进行和停止；只要表达式的值非0，继续循环；直到表达式值为0，循环停止。

例：用while循环求1~100间整数的和

```
sum=0;
i=1;
while i<=100
sum=sum+i;
i=i+1;
end
sum
sum =
    5050
```





(3). 分支结构——根据一定条件来执行的各条语句。

if — else — end语句有3种形式

① if 表达式  
    执行语句  
    end

② if 表达式 —— 是  
    语句1  
    else —— 否  
    语句2  
    end



③ if 表达式1 —— 多分支

语句1

elseif 表达式2

语句2

elseif 表达式3

语句3

⋮

else

语句n

end

这种结构实现了一种多路选择，比较复杂。可替代一般高级语言中的switch-case-end语句



❖ 在条件表达式中，通常都是由

关系操作符  $>$ ,  $<$ ,  $<=$ ,  $>=$ ,  $=$ ,  $\sim$  等

逻辑操作符  $\&$ ,  $|$ ,  $\sim$  等

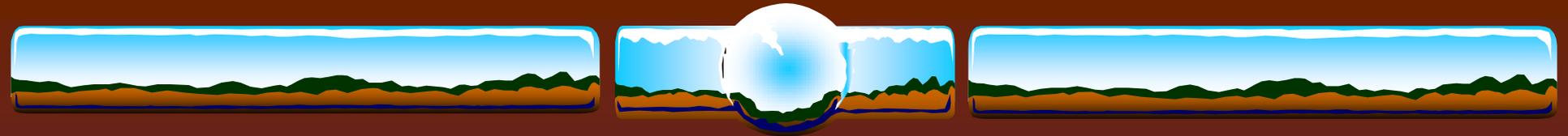
逻辑函数: `isequal` —— 若是相等则为真

`isempty` —— 若是空矩阵则为真

`isstr` —— 若是字符串则为真

例：编写m函数，计算函数值

$$\begin{cases} x & (x < 1) \\ 2x-1 & (1 < x < 10) \\ 3x-11 & (x \geq 10) \end{cases}$$



## 五、matlab程序流的控制

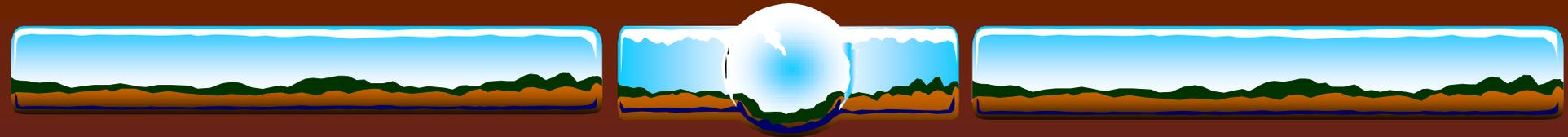
matlab的控制语句同c语言有相似之处,但没有c语言复杂、灵活和多变。因而语法比较简单,容易掌握。

❖ **echo 指令**— 用来控制m文件在执行过程中是否显示

**echo on** — 打开所有命令文件的显示方式

**echo off** — 关闭所有命令文件的显示方式

**echo** — 在以上两者间切换



- ❖ echo对于命令文件函数文件有所不同，命令文件用法简单，函数对所有命令起作用；函数文件用法较复杂，具体请help查询。

echo file on — 打开file函数文件的显示方式

echo file off — 关闭file函数文件的显示方式

echo file — 切换file函数文件的显示方式

echo on all — 打开所有函数文件的显示方式

echo off all — 关闭所有函数文件的显示方式



❖ **input** —— 提示用户从键盘输入数值、字符串、表达式。

例： `n=input('How many apples= ')`

How many apples= 5

n =

5

❖ **pause** —— 暂停，等待用户响应

可令m文件执行暂停，并在继续执行前按任一个键。



❖ keyboard —— 与input类似，功能更强

keyboard 调用键盘与调用m文件一样，当keyboard在m文件中出现时，它停止文件的执行并控制键盘，用于修改程序。

❖ break —— 中断

break 中断for, while循环语句的执行，在嵌套循环结构中，break进从最里层循环推出。

❖ menu —— 产生一个拥护可选择输入的菜单

格式：k=menu(‘标题’， ‘菜单1’， ‘菜单2’ ...)



## 六、文件操作

matlab文件输入、输出（I/O）函数允许把其它格式的数据文件直接读进matlab，它是以c语言文件I/O函数为基础的。

### 1. 打开和关闭文件（fopen, fclose）

格式：`fid=fopen('数据文件', 'r')`

'r' —— 读

'w' —— 写

'a' —— 附加

'rt' —— 读写



❖ 文件一旦打开，即可进行读写操作。

`sta=fopen(fid)` —— 关闭上述打开文件

`sta=fopen(all)` —— 关闭所有打开的文件

❖ 与`save,load`类似，是matlab专用数据文件读写工具（\*.mat）。

## 2. 读写二进制数据文件

`fread, fwrite` —— 可读写数据文件，文本文件。

例：`fid=fopen('user.dat','r')`

`a=fread(fid,50)`

`sta=fopen(fid)`



## 七、应用程序接口

应用程序接口——是matlaab的附加组件。是一个由相关函数组成的接口函数库，可实现与外部程序的交互。

- ❖ **MEX文件**：是matlab环境下调用C、FORTRAN语言编写的应用程序模块。
- ❖ **matlab 计算引擎**：在使用C、FORTRAN编程时可调用matlab函数库进行计算。
- ❖ **MAT文件**：实现matlab与C、FORTRAN语言程序间的数据交换



## 八、matlab编译

matlab编译的原因：

1. 提高运算速度
2. 隐藏特定算法
3. 可产生独立的m文件

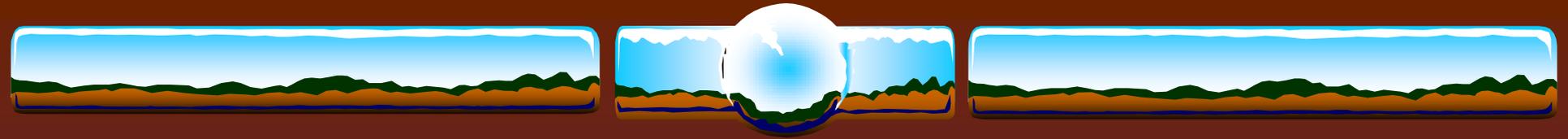
matlab编译函数： `mcc`

`MCC [-options] fun [fun2 ...] [mexfile1 ...] [mlibfile1 ...]`



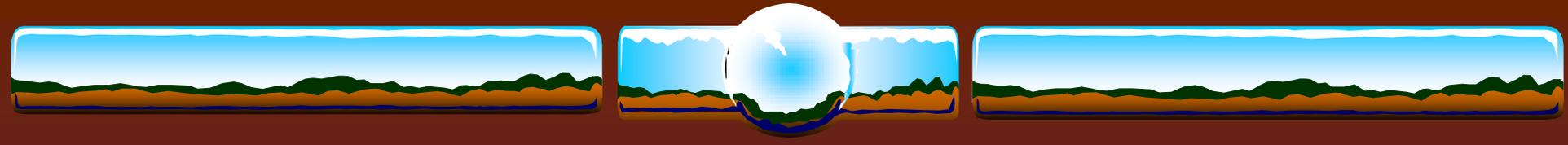
# 小 结

- ❖ 本节主要介绍了matlab语言程序设计基础。matlab 编程简单，容易掌握。
- ❖ 通过学习，应能编写简单程序



# 结束语

- ❖ 本课程基于matlab语言基础，望大家以后进一步学习，方能编写出高质量m文件，以适应高科技的需要。
- ❖ 望你们今后能成为matlab高手。



再

见