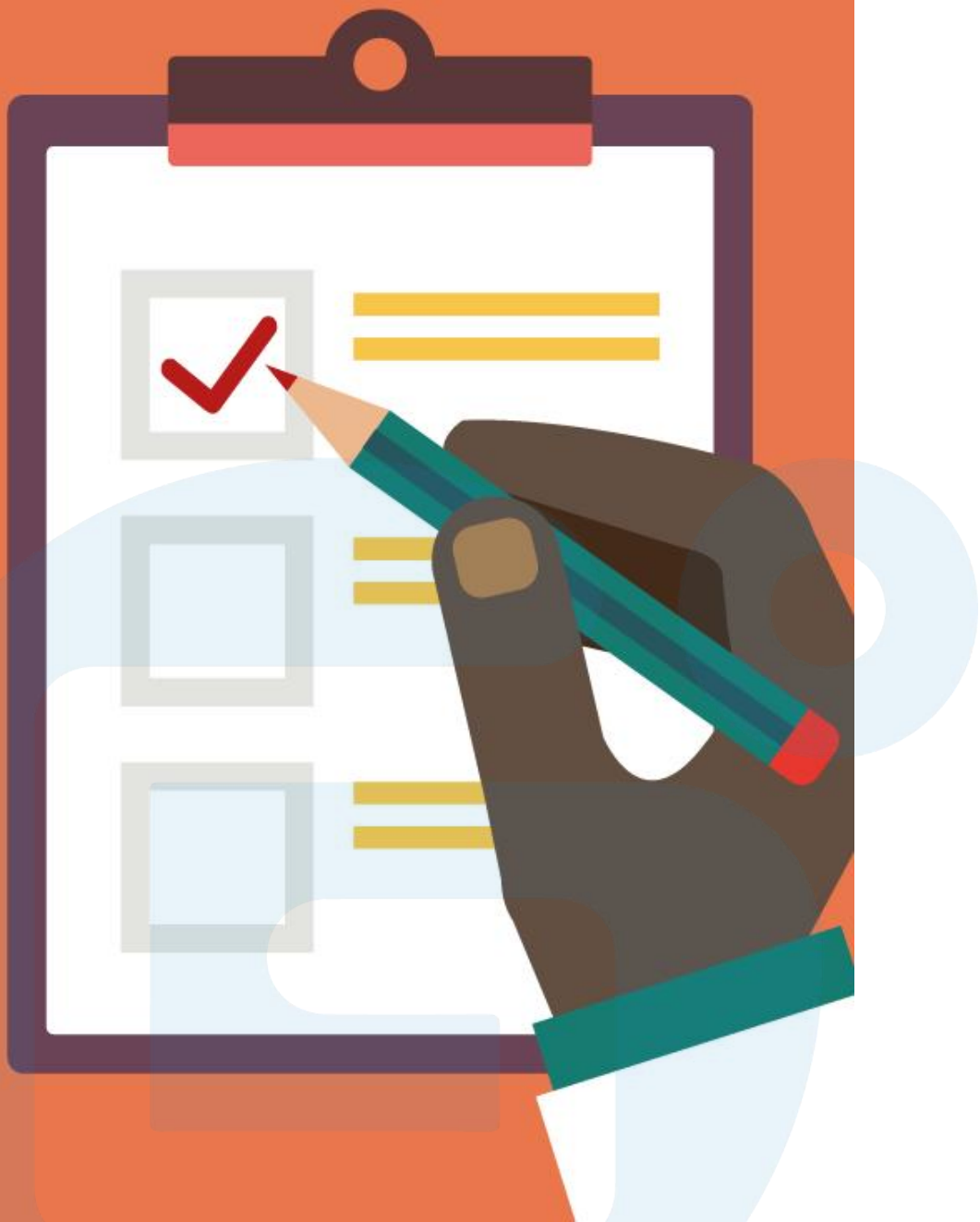


Talk is cheap, show me the code

第三课：Matlab语言编程基础

Matlab入门课程系列



OUTLINE

- Matlab脚本编辑器
 - Matlab变量
 - Matlab控制流
 - Matlab脚本
 - Matlab函数
 - Matlab调试
 - Matlab性能优化
-



一 Matlab脚本编辑器



Matlab脚本编辑器

- .M文件编辑器是一个文本编辑器，用于替代命令行命令，可以完成更加复杂的语言命令。
- 可以用edit命令或者图标来创建新的M文件。
- 打开已有M文件的方法是edit filename，或者直接在目录树里双击文件。

数据格式	说 明
whatdirname	列出当前文件夹下所有的 MATLAB 文件。如果给定 dirname，就列出目录 dirname 下的文件
dirname	dir 列出一个目录或子目录中的所有文件。这个命令可以用不同的路径名和程序单
Dir	列出一个目录或子目录中的所有文件。这个命令可以用不同的路径名和程序单
ls	以不同的输出格式列出文件
Delete filename	删除文件 filename
cd	改变当前文件夹
Type filename	显示文件 filename 的内容。如果没有指定扩展名，则 MATLAB 就读 filename.m
Edit file	打开一个编辑器。如果给定 file，那么这个文件就在编辑器中打开
copyfile(file1, file2)	file1 复制到 file2。有关错误处理可参见 helpcopyfile
Which filename	显示由 filename 指定的函数的搜索路径
path	显示 MATLAB 的目录搜索路径。如果给出带自变量的命令，就改变搜索路径。输入 helppath 可以获得更多的信息



二 Matlab变量



- **变量**——程序里定义的变量是用于保存和处理数据的，其本质是一段可以操作的内存空间，变量名是内存地址的符号化表达。
- Matlab作为一种高级语言，无需事先声明即可赋值，matlab会根据赋值的情况给变量自动分配一个类型。

变量命名规则

- 变量名和函数名对字母的大小写敏感，因此 x 和 X 是两个不同的变量；sin 是 MATLAB 定义的正弦函数，而 SIN 不是。
- 变量名必须以字母开头，其后可以是任意字母或下画线，但是不能有空格、中文或标点，例如_xy、a.b 均为不合法的变量名，而 classNum_x 是一个合法的变量名。
- 不能使用 MATLAB 的关键字作为变量名。避免使用函数名作为变量名。如果变量采用函数名，则该函数失效，例如设置变量名为“if”、“end”等。
- 变量名最多可包含 63 个字符，从第 64 个字符开始之后的字符将被忽略。为了程序可读性及维护方便，变量名一般代表一定的含义。

- 变量分为局部变量、全局变量和永久变量。
 - 局部变量只作用于函数体内，与工作区分割，函数调用完毕后，局部变量会被删除，不保存。
 - 全局变量用global关键字声明，写在函数体开头位置。使用全局变量可以减少数据传递次数，但不可滥用，容易造成程序结构混乱。
 - 永久变量(又叫静态变量)用persistent关键字声明，只能在M文件里定义和使用，只允许声明它的函数存取，当函数退出时，也不会从matlab内存里清除。在函数中声明的变量，当函数调用完之后就会释放。如果想保留这个变量的值（供该函数下一次调用），可以把这个变量声明为永久变量。永久变量不能在声明的时候赋值，而且只能在function里声明,且只有这个function才能认识它。
-

- 特殊变量

特殊变量	描 述
ans	系统默认的用作保存运算结果的变量名
pi	圆周率
eps	机器零阈值，MATLAB 中的最小数
inf	表示无穷大
NaN 或 nan	表示不定数
i 或 j	虚数

nargin	函数的输入参数个数
nargout	函数的输出参数个数
realmin	可用的最小正实数
realmax	可用的最大正实数
bitmax	可用的最大正整数（以双精度格式存储）
varargin	可变的函数输入参数个数
varargout	可变的函数输出参数个数
beep	使计算机发出“嘟嘟”声音

- 关键字 iskeyword, 流程控制变量, 属于保留变量名。

```
>> iskeyword
```

```
ans =
```

```
20×1 cell 数组
```

```
{'break' }
```

```
{'case' }
```

```
{'catch' }
```

```
{'classdef' }
```

```
{'continue' }
```

```
{'else' }
```

```
{'elseif' }
```

```
{'end' }
```

```
{'for' }
```

```
{'function' }
```

```
{'global' }
```

```
{'if' }
```

```
{'otherwise' }
```

```
{'parfor' }
```

```
{'persistent' }
```

```
{'return' }
```

```
{'spmd' }
```

```
{'switch' }
```

```
{'try' }
```

```
{'while' }
```



三 Matlab控制流



- 控制流包括顺序控制、if-else-end分支结构、switch-case结构、try-catch结构、for循环结构和while循环结构。这6种结构算法的使用是各家编程语言所通用的，整体上大同小异。

顺序控制

从上到下依次执行

if-else-end语句

```
1 - clc;clear all;  
2 - a=input(' a=' );  
3 - if a>10  
4 -     b=77;  
5 - elseif a<0  
6 -     b=-77;  
7 - else  
8 -     b=0  
9 - end  
10 - b
```

命令窗口

```
a=-50  
b =  
-77
```

```
clc;clear all;  
a=input(' a=' );  
if a>10  
    b=77;  
elseif a<0  
    b=-77;  
else  
    b=0;  
end  
b
```

行窗口

```
a=5  
b =  
0  
>> |
```

switch-case

switch后面可以跟标量，或者字符串

```
clc;clear all;  
a=input('a=');  
switch a  
    case 1  
        b=11;  
    case 2  
        b=12;  
    case 3  
        b=13;  
    case 4  
        b=14;  
    otherwise  
        b=0;  
end  
b
```

```
a=1  
b =  
    11
```

```
Untitled.m x +  
1 - clc;clear all;  
2 - a=input('a=');  
3 - switch a  
4 -     case 'allen'  
5 -         b=11;  
6 -     case 'bob'  
7 -         b=12;  
8 -     case 3  
9 -         b=13;  
0 -     case 4  
1 -         b=14;  
2 -     otherwise  
3 -         b=0;  
4 - end  
5 - b
```

命令窗口

```
a='bob'  
b =  
    12
```

if 语句	switch 语句
比较复杂，特别是嵌套使用的 if 语句	可读性强，容易理解
要调用 strcmp 函数比较不同长度的字符串	可比较不同长度的字符串
可检测相等和不相等	仅检测相等

try-catch

Matlab在执行try命令发生错误时候，才会执行catch命令。本结构只能提供两组命令，当执行try命令出错时，可以用lasterr给出一个错误原因，如果catch也出错，则终止该结构。

```
1 -   clc;clear all;
2 -   num=6;
3 -   mat=magic(4)
4 -   try
5 -       mat_num=mat(num,:)
6 -   catch
7 -       mat_end=mat(end,:)
8 -   end
9 -   lasterr
```

命令行窗口

```
mat =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1

mat_end =
     4    14    15     1

ans =
'位置 1 处的索引超出数组边界(不能超出 4)。'
```

for循环

for循环用于执行已知循环次数的循环过程。可以嵌套。

```
1 -   clc;clear all;
2 -   num=100;
3 -   sum=0;
4 -   for i=1:num
5 -       sum=sum+i;
6 -   end
7 -   sum
8
```

命令行窗口

```
sum =
    5050
```

```
1 -   clc;clear all;
2 -   for i=1:4
3 -       for j=1:4
4 -           array(i,j)=(i+j)*(-1)^(i+j);
5 -       end
6 -   end
7 -   array
```

命令行窗口

```
array =
     2    -3     4    -5
    -3     4    -5     6
     4    -5     6    -7
    -5     6    -7     8
```

while循环

while循环又叫**条件循环**，可用于执行未知循环次数的循环过程。也可以嵌套。

```
1 -   clc;clear all;
2 -   i=1;
3 -   while i<5
4 -       j=1;
5 -       while j<5
6 -           array(i,j)=(i+j)*(-1)^(i+j);
7 -           j=j+1;
8 -       end
9 -       i=i+1;
10 -   end
11 -   array
12 -
```

命令行窗口

```
array =
     2     -3     4     -5
    -3     4    -5     6
     4    -5     6    -7
    -5     6    -7     8
```

其他 指令

return

在函数里插入return指令，可以强制matlab结束函数执行，把控制权交回主函数。

input和keyboard(调试模式)

(1) input 指令将 MATLAB 的“控制权”暂时交给用户，用户通过键盘输入数值、字符串或表达式等，并按 Enter 键将输入内容传递到工作区，同时把“控制权”交还给 MATLAB。其常用的句法格式如下：

- `Value = input('message');` 将用户输入的内容赋值给变量 Value。
- `Value = input('message', 's');` 将用户输入的内容以字符串的形式赋值给变量 Value。

说明：

- ① 指令中的“message”是显示在屏幕上的字符串。
- ② 对于上面第一种调用格式，用户可以输入数值、字符串等各种形式的数据。
- ③ 对于上面第二种调用格式，用户无论输入什么内容，均以字符串的形式赋值给变量。

(2) 当执行遇到 keyboard 指令时，MATLAB 将“控制权”暂时交给键盘，用户可以通过键盘输入各种合法的 MATLAB 指令。只有当用户输入完成，并输入 return 指令后，“控制权”才交还给 MATLAB。

input 指令和 keyboard 指令的不同之处在于：keyboard 指令允许输入任意多个 MATLAB 指令，而 input 指令只允许用户输入赋值给变量的“值”，即数组、字符串或元胞数组等。

return

在函数里插入return指令，可以强制matlab结束函数执行，把控制权交回主函数。

break

break是退出该层循环，进入下一个循环。

pause

pause，暂停程序执行，除非按下任意按键，pause(n)，暂停n秒。

continue

continue退出循环的本轮计算，之后执行下一轮循环。

Matlab控制流

```
1 -   clc;clear all;
2 -   a(1)=1;
3 -   a(2)=1;
4 -   n=1000;
5 -   for i=3:n
6 -       a(i)=a(i-1)+a(i-2);
7 -       if a(i)>999
8 -           [i,a(i)]
9 -           break
10 -       end
11 -   end
```

命令行窗口

```
ans =
    17    1597
```

```
1 -   clc;clear all;
2 -   a(1)=1;
3 -   a(2)=1;
4 -   n=1000;
5 -   i=3;
6 -   a(i)=a(i-1)+a(i-2);
7 -   while a(i)<1000
8 -       i=i+1;
9 -       a(i)=a(i-1)+a(i-2);
10 -   end
11 -   [i,a(i)]
```

命令行窗口

```
ans =
    17    1597
```

```
1 -   clc;clear all;
2 -   a(1)=1;
3 -   a(2)=1;
4 -   n=1000;
5 -   i=3;j=1;
6 -   a(i)=a(i-1)+a(i-2);
7 -   while a(i)<1000
8 -       i=i+1;
9 -       a(i)=a(i-1)+a(i-2);
10 -       if i<10
11 -           continue;
12 -       else
13 -           j=j+1;
14 -       end
15 -   end
16 -   [i,a(i),j]
```

命令行窗口

```
ans =
    17    1597     9
```

error和warning

常见的错误和警告指令如下

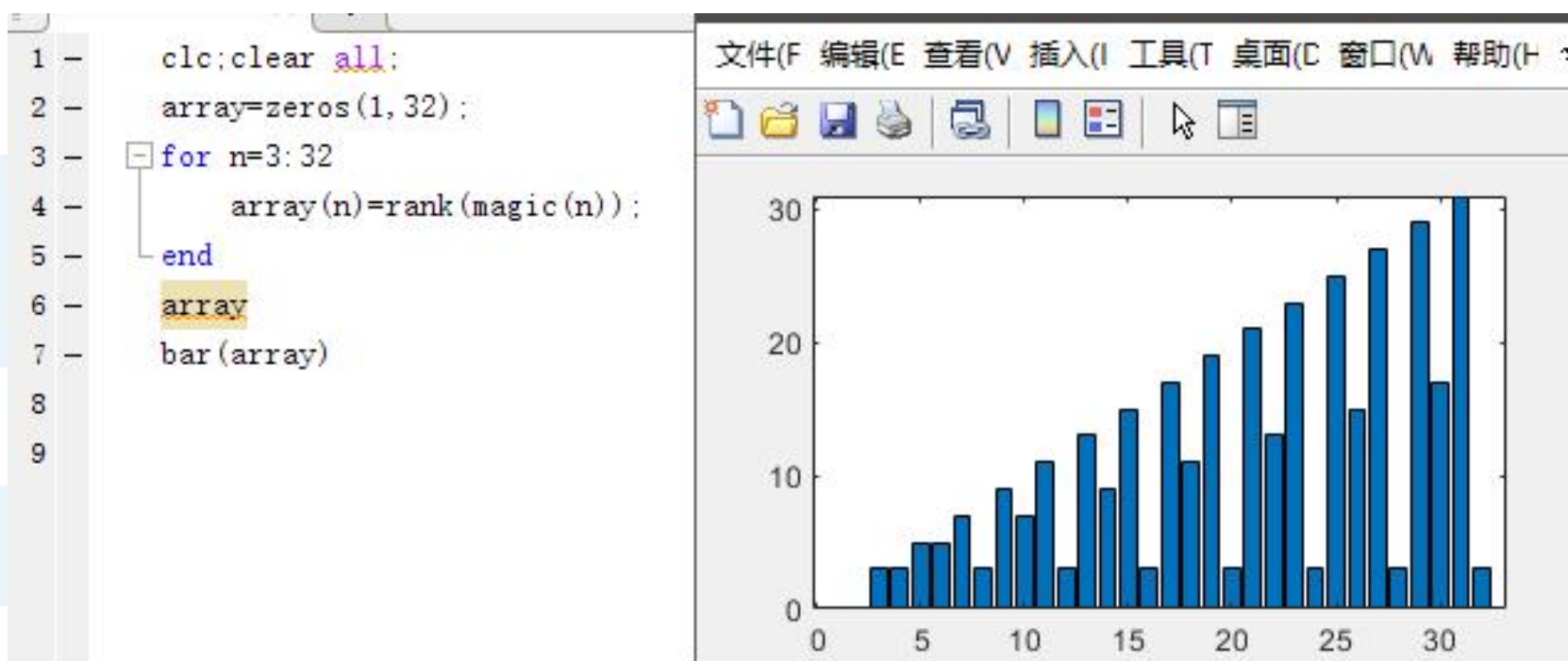
- `error('message')`: 显示出错信息 `message`, 终止程序。
 - `errortrap`: 错误发生后, 控制程序继续执行与否的开关。
 - `lasterr`: 显示 MATLAB 系统判断的最新出错原因, 并终止程序。
 - `warning('message')`: 显示警告信息 `message`, 继续运行程序。
 - `lastwarn`: 显示 MATLAB 系统给出的最新警告程序, 并继续运行。
-



四 Matlab脚本



脚本是一连串的代码，但是可以反复修改，点击运行后即运行于命令行窗口内。一般是.M文件。



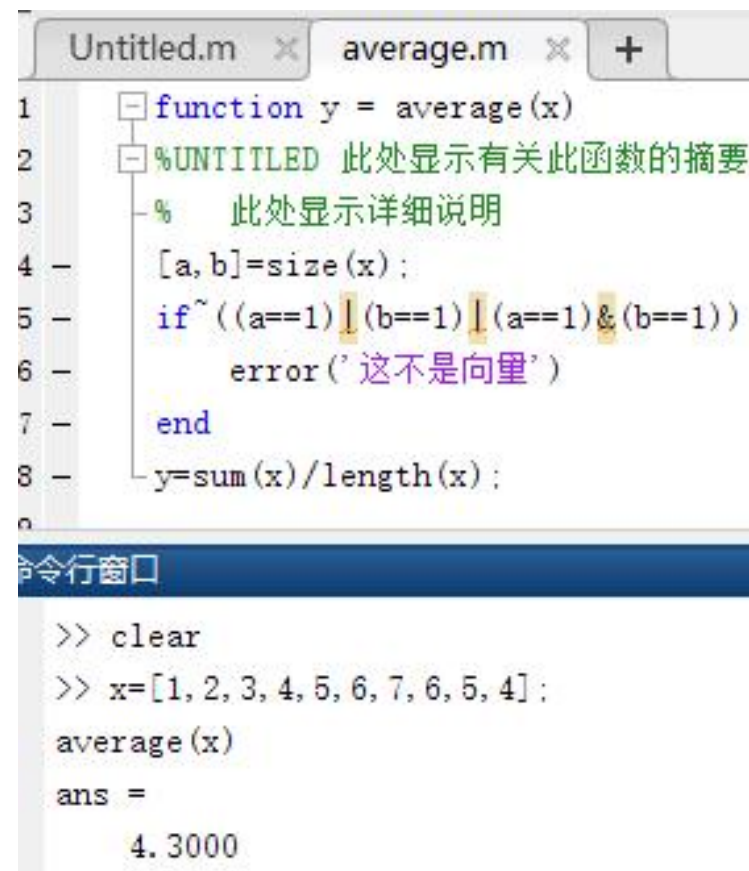


五 Matlab函数





函数也是一连串的代码，通常也是.M文件，但是第一条可执行语句是以function开始的。函数可以理解为一个黑箱，一些数据送入，经过函数处理后输出一些数据，函数内部的变量一般是局部变量，执行完函数后会自动清理，不留在工作区里。toolbox实际上就是由一系列函数文件组成的。

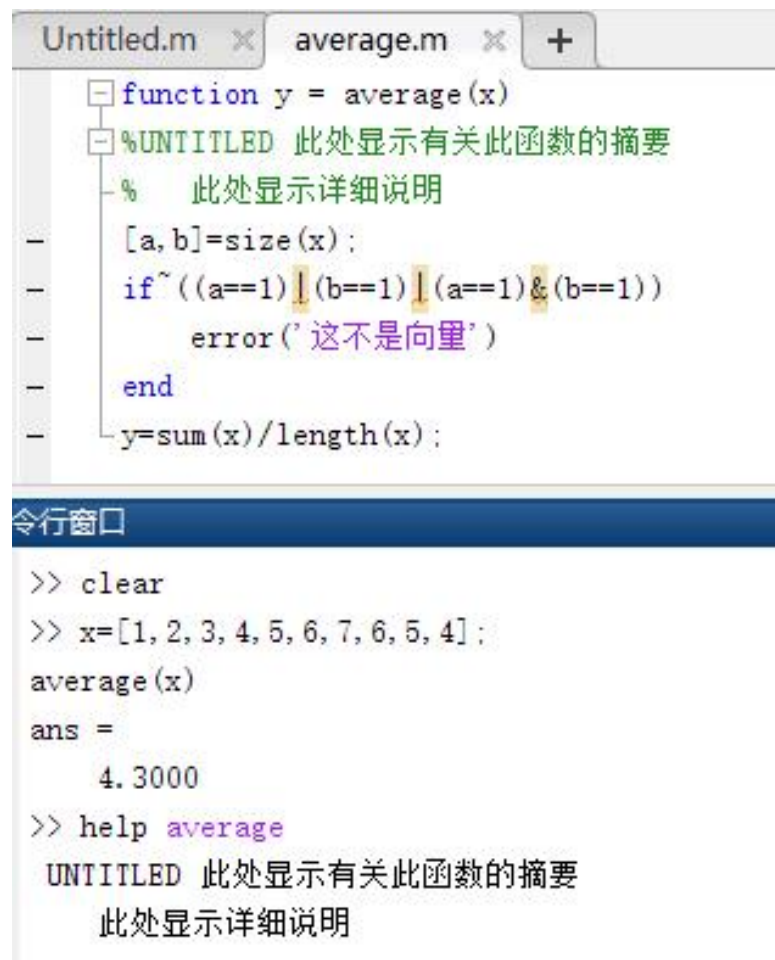


```
Untitled.m x average.m x +
1 function y = average(x)
2 %UNTITLED 此处显示有关此函数的摘要
3 % 此处显示详细说明
4 [a,b]=size(x);
5 if ~(a==1)|(b==1)|(a==1)&(b==1)
6     error('这不是向量')
7 end
8 y=sum(x)/length(x);
9
命令窗口
>> clear
>> x=[1,2,3,4,5,6,7,6,5,4];
average(x)
ans =
    4.3000
```

函数特点

(1) 函数定义行：第一行，由function引导，包含函数名，输入参数和输出参数，函数名必须与文件名相同(如果不同，则以文件名为准)，可以带有多个输入和输出参数列表，也可以没有输出参数。

(2) 第一行下面的注释用于解释该函数的功能，可以被help函数调用。

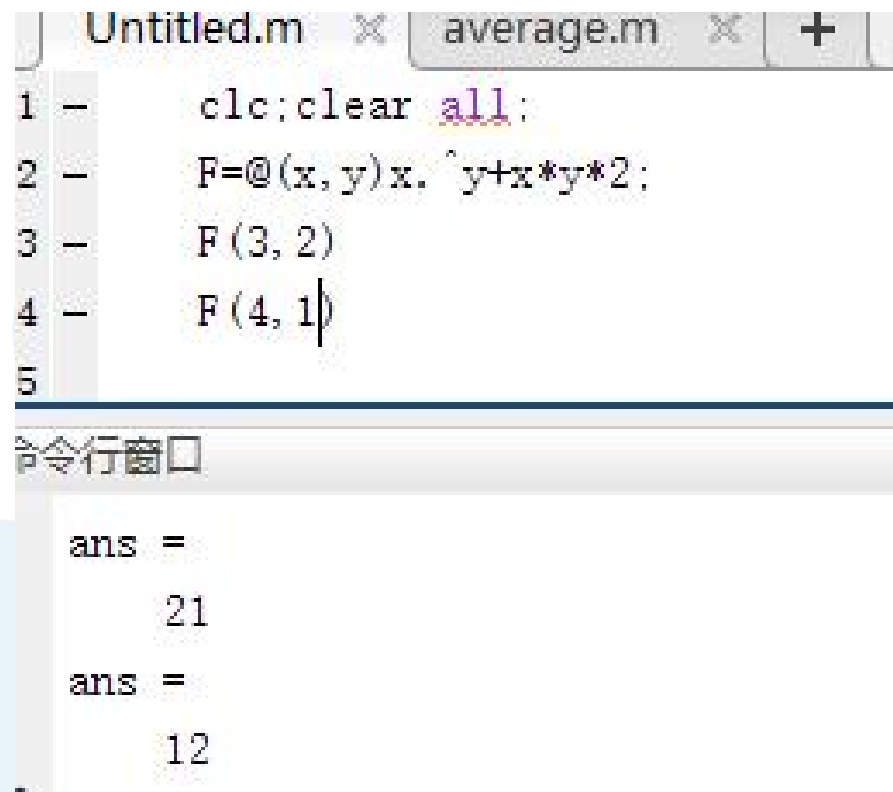


```
Untitled.m x average.m x +
function y = average(x)
%UNTITLED 此处显示有关此函数的摘要
% 此处显示详细说明
[a,b]=size(x);
if ~(a==1)|(b==1)|(a==1)&(b==1)
    error('这不是向量')
end
y=sum(x)/length(x);

命令行窗口
>> clear
>> x=[1,2,3,4,5,6,7,6,5,4];
average(x)
ans =
    4.3000
>> help average
UNTITLED 此处显示有关此函数的摘要
此处显示详细说明
```


匿名函数

对于一些需要复用的简单表达式，没有必要另外新建一个文件，可以利用@生成匿名函数



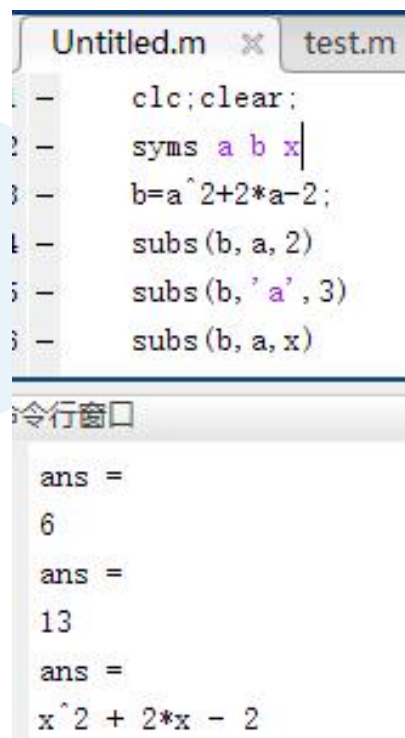
```
Untitled.m  average.m  +  
1 -      clc;clear all;  
2 -      F=@(x,y)x.^y+x*y*2;  
3 -      F(3,2)  
4 -      F(4,1)  
5  
命令窗口  
ans =  
    21  
ans =  
    12
```

The screenshot shows a MATLAB script in the editor window. The script defines an anonymous function `F` and evaluates it at two points. The command window shows the results of these evaluations.

Input	Output (ans)
<code>F(3,2)</code>	21
<code>F(4,1)</code>	12

syms预定义函数+subs替代函数

类似匿名函数，也不用重新生成m文件。但需要预先定义未知量，而且由于生成符号变量会调用了符号运算内核，计算速度会降低。



```
Untitled.m x test.m
-   clc;clear;
-   syms a b x
-   b=a^2+2*a-2;
-   subs(b,a,2)
-   subs(b,'a',3)
-   subs(b,a,x)

命令窗口
ans =
6
ans =
13
ans =
x^2 + 2*x - 2
```

eval函数和feval函数

符号型和字符型的函数表达式如果要进行求值，则需要利用eval和feval函数。eval用于将字符串转化成普通指令，feval用于调用函数（比起函数名直接调用，其优势在于调用函数函数，即泛函）。

```
Untitled.m x test.m
- clc;clear;
- syms a b x y z
- b=a^2+2*a-2;
- y=2/x+3;
- z=b+y;
- a=2;x=3;
- eval(z)
```

```
ans =
    9.6667
```

```
Untitled.m x
- clc;clear
- a='b=1';
- eval(a)
```

```
b =
    1
```

```
Untitled.m x test.m x +
- clc;clear;
- NameSource=[10 20 30];
- for i=1:3
-     Name=['VarName',num2str(i)];
-     eval([Name,'=NameSource(i)'])
- end
```

```
VarName1 =
    10
VarName2 =
    20
VarName3 =
    30
```

```
Untitled.m x test.m x
2 - y=@(a,b)a^3+4*b-12;
3 - y(2,3)
4 - feval(y,3,2)
```

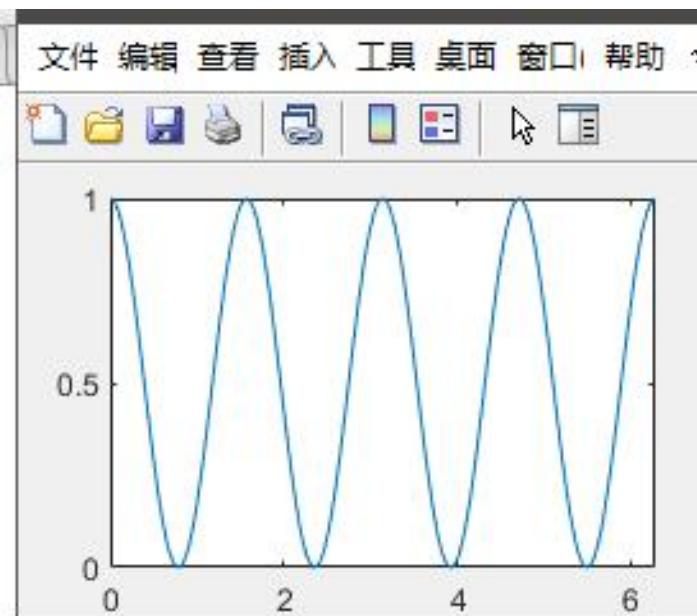
```
ans =
    8
ans =
    23
```

子函数

写在函数文件里的其它函数，只要不是第一个函数，其它一律是子函数，子函数只能被主函数和其它子函数调用。

```
Untitled.m x test.m x +
function F=test(n)
-   A=1;w=2;phi=pi/2;
-   signal=createsig(A,w,phi);
-   F=signal.^n;
function signal=createsig(A,w,phi)
-   x=0:pi/100:pi*2;
-   signal=A*sin(w*x+phi)
```

```
Untitled.m x test.m x
1 -   clc;clear all;
2 -   test(2);
3 -   x=0:pi/100:pi*2;
4 -   plot(x,ans)
5
6
7
8
```





六 Matlab调试



调试，是指处理程序bug的过程。主要包括直接调试法和工具调试法。

常见错误

语法错误（程序会给出提示）

逻辑错误（不易发现）

异常

对于简单的程序(例如只有单一模块组成的程序),直接调试法是一种简便快捷的方法。直接调试法的基本方法大致有如下几种:

(1) 通过分析,将重点怀疑语句后的分号删除,将结果显示出来,然后与预期值比较,从而快速地判断程序执行到该处时是否发生了错误。

(2) 在适当的位置添加输出变量值的语句。

(3) 在程序的适当位置添加 `keyboard` 命令。当程序执行到该处时将暂停,并显示提示符,用户可以查看或变更工作区中显示的各个变量的值。在提示符后输入 `return` 指令可以继续执行原文件。

(4) 调试程序时,可以利用注释符号“`%`”屏蔽函数声明行,并定义输入变量的值,以脚本 M 文件的方式执行程序,可以方便地查看中间变量,从而有利于找出相应的错误之处。

```
>> help debug
```

debug List MATLAB debugging functions

dbstop	- Set breakpoint.
dbclear	- Remove breakpoint.
dbcont	- Resume execution.
dbdown	- Change local workspace context.
dbmex	- Enable MEX-file debugging.
dbstack	- List who called whom.
dbstatus	- List all breakpoints.
dbstep	- Execute one or more lines.
dbtype	- List file with line numbers.
dbup	- Change local workspace context.
dbquit	- Quit debug mode.

When a breakpoint is hit, MATLAB goes into debug mode, the debugger window becomes active, and the prompt changes to a K>>. Any MATLAB command is allowed at the prompt.

To resume program execution, use DBCONT or DBSTEP.

To exit from the debugger use DBQUIT.





七 Matlab性能优化





尽可能地使用 load 函数及 save 函数，而不是文件 I/O 操作函数。

避免更改变量的数据类型或维数。如果确实需要这么做，可以预先创建一个新的变量。

尽可能地使用实数运算。因此，对于复数的运算可以转换为多个实数运算，由此提升效率。

尽可能地避免对实数和复数的相互赋值。

在进行逻辑运算时，采用&&等短路逻辑运算具有更高的效率。

代码向量化。这是因为 MATLAB 执行循环的效率比较低，因此，将诸如 for 循环和 while 循环转化为矩阵的按位运算，可以提高计算效率。

对不可避免且耗时很长的循环操作，可以尝试在 MEX 文件内实现。

在进行 for、while 等循环前，对于循环过程中不断变化的变量应预先分配足够大的数组，从而避免 MATLAB 频繁地进行变量数组重生成操作，提高运算速度。



下节课预习



下节课将就matlab的绘图功能进行讲解。matlab的一大关键特点就是它方便快捷的绘图功能。



感谢参与 下堂课见