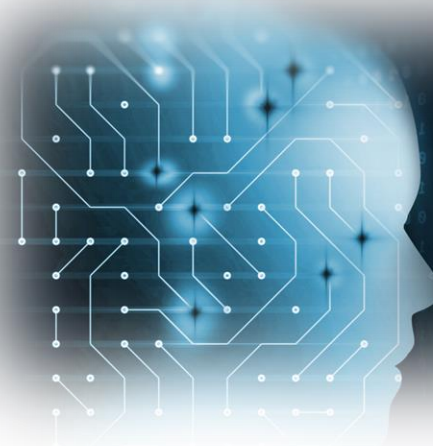
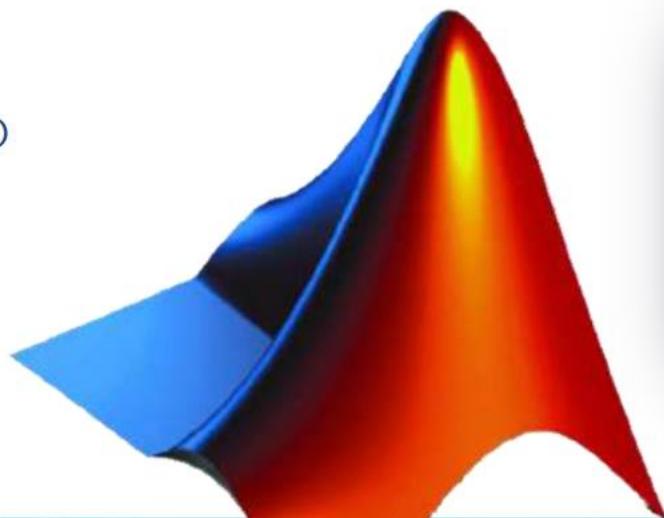




Programming Way to AI

心理学编程

MATLAB®



陈骥
心理与行为科学系

ji.chen@zju.edu.cn



浙江大学
ZHEJIANG UNIVERSITY

数据可视化

» 数据可视化：二维图和三维图

- MATLAB可绘制的图片类别

线图	散点图和气泡图	数据分布图	离散数据图	地理图	极坐标图
plot 	scatter 	histogram 	bar 	geoplot 	polarplot 
plot3 	scatter3 	histogram2 	barh 	geoscatter 	polarhistogram 
stairs 	bubblechart 	pie 	bar3 	geobubble 	polarscatter 
errorbar 	bubblechart3 	pie3 	bar3h 		polarbubblechart 
area 	swarmchart 	scatterhistogram 	pareto 		compass 
stackedplot 	swarmchart3 	swarmchart 	stem 		ezpolar 

» 数据可视化：二维图和三维图

- MATLAB可绘制的图片类别

等高线图	向量场	曲面图和网格图	三维可视化	动画	图像
<code>contour</code> 	<code>quiver</code> 	<code>surf</code> 	<code>streamline</code> 	<code>animatedline</code> 	<code>image</code> 
<code>contourf</code> 	<code>quiver3</code> 	<code>surfc</code> 	<code>streamslice</code> 	<code>comet</code> 	<code>imagesc</code> 
<code>contour3</code> 	<code>feather</code> 	<code>surf1</code> 	<code>streamparticles</code> 	<code>comet3</code> 	
<code>contourslice</code> 		<code>ribbon</code> 	<code>streamribbon</code> 		
<code>fcontour</code> 		<code>pcolor</code> 	<code>streamtube</code> 		

» 数据可视化：二维线图

- 以创建二维线图为例：绘制基础图像
 - 使用 `plot` 函数创建二维线图。例如，绘制从 0 到 2π 之间的正弦函数值

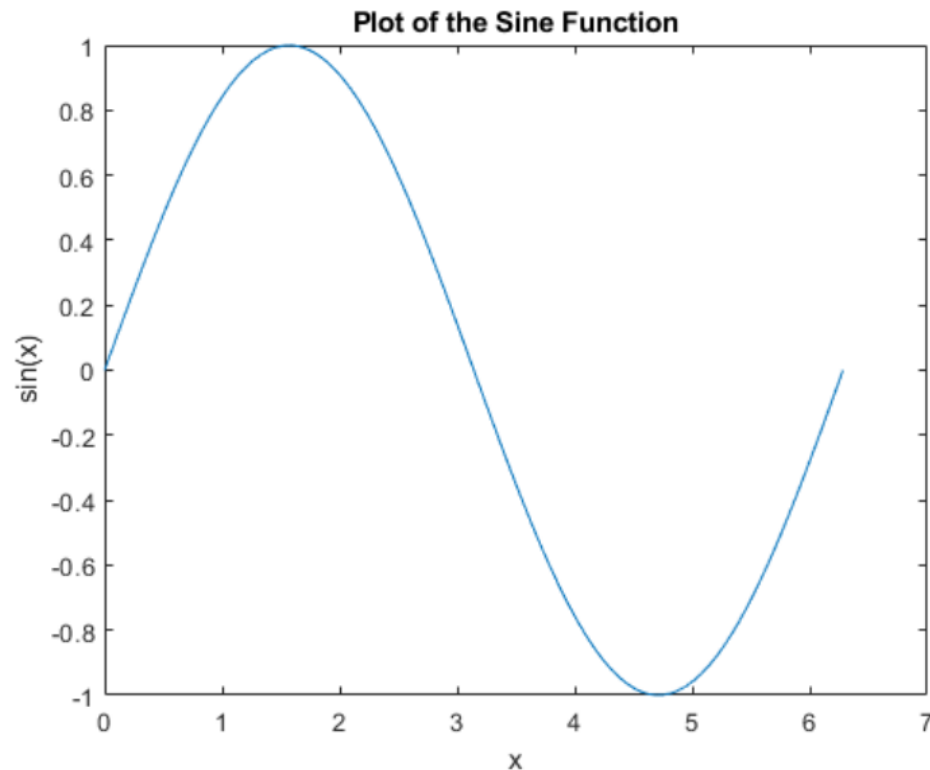
```
x = linspace(0,2*pi,100);  
y = sin(x);  
plot(x,y)
```

`linspace`函数（获取等间隔的点）

第一个参数0是下界，第二个参数 2π 是上界，最后一个参数表示等间隔点的个数100

- 标记坐标区并添加标题。

```
xlabel('x')  
ylabel('sin(x)')  
title('Plot of the Sine Function')
```



» 数据可视化：二维线图

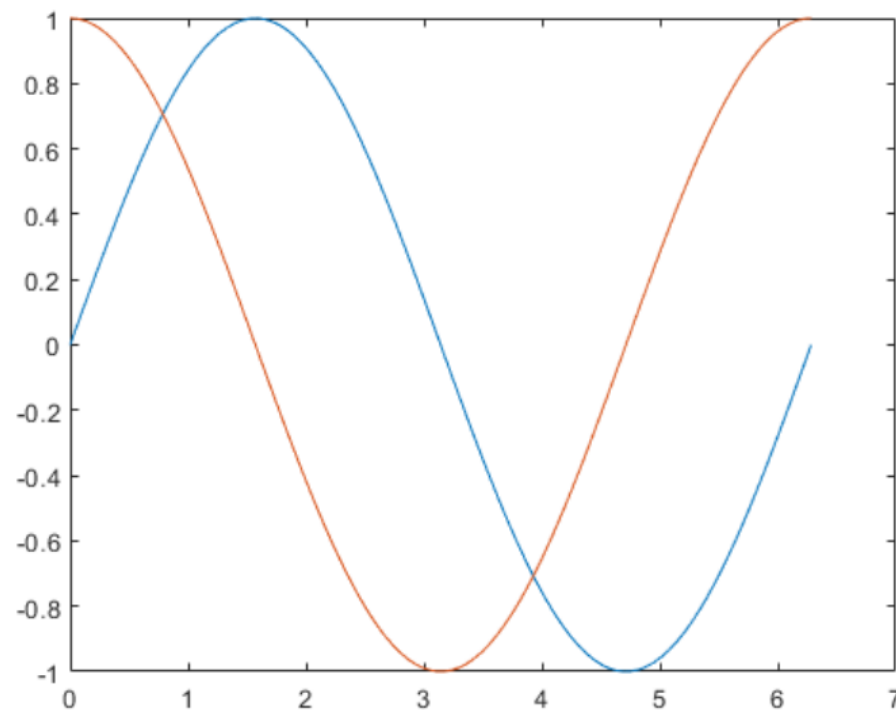
- 以创建二维线图为例：绘制基础图像

- 绘制多个线条

使用hold on/off在原有图片上添加
线条

```
figure  
x = linspace(0,2*pi,100);  
y = sin(x);  
plot(x,y)
```

```
hold on  
y2 = cos(x);  
plot(x,y2)  
hold off
```

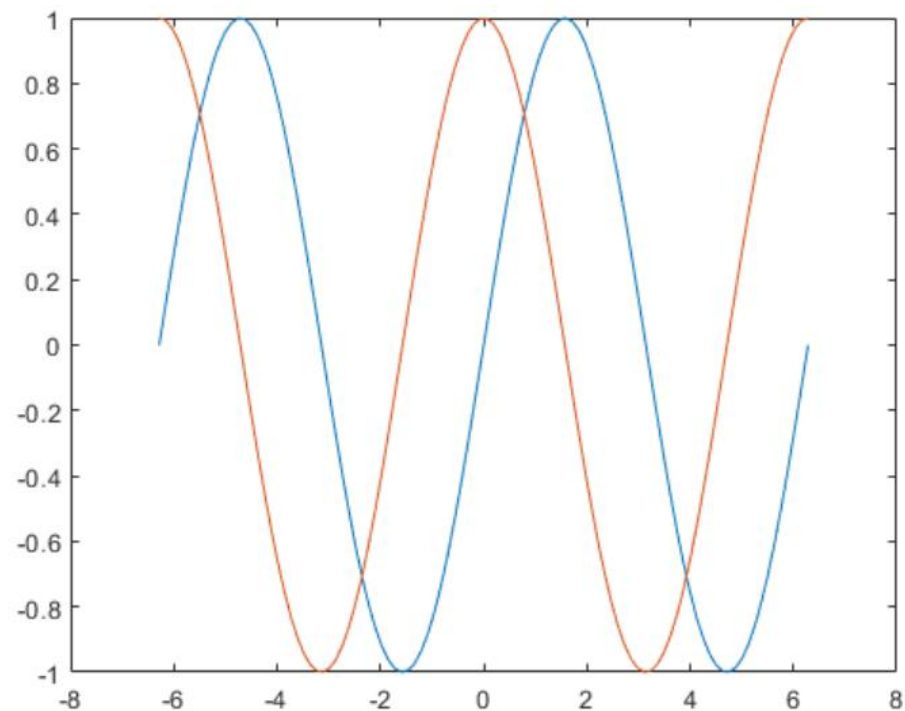


» 数据可视化：二维线图

- 以创建二维线图为例：绘制基础图像
 - 绘制多个线条

一次性绘制多个线条

```
x = linspace(-2*pi,2*pi);  
y1 = sin(x);  
y2 = cos(x);  
  
figure  
plot(x,y1,x,y2)
```



» 数据可视化: Plot function 二维线图





```
plot(x,y, 'line specifiers', 'PropertyName', PropertyValue)
```

- 更改线型和标记
 - Line style、 color and marker (if markers are desired)

Line style	说明	表示的线条
' - '	实线	
' - - '	虚线	
' : '	点线	
' - . '	点划线	

Marker	说明	生成的标记
' o '	圆圈	
' + '	加号	
' * '	星号	
' . '	点	
' x '	叉号	

颜色名称	短名称	RGB 三元组	外观
'red'	'r'	[1 0 0]	
'green'	'g'	[0 1 0]	
'blue'	'b'	[0 0 1]	
'cyan'	'c'	[0 1 1]	

'magenta'	'm'	[1 0 1]	
'yellow'	'y'	[1 1 0]	
'black'	'k'	[0 0 0]	
'white'	'w'	[1 1 1]	

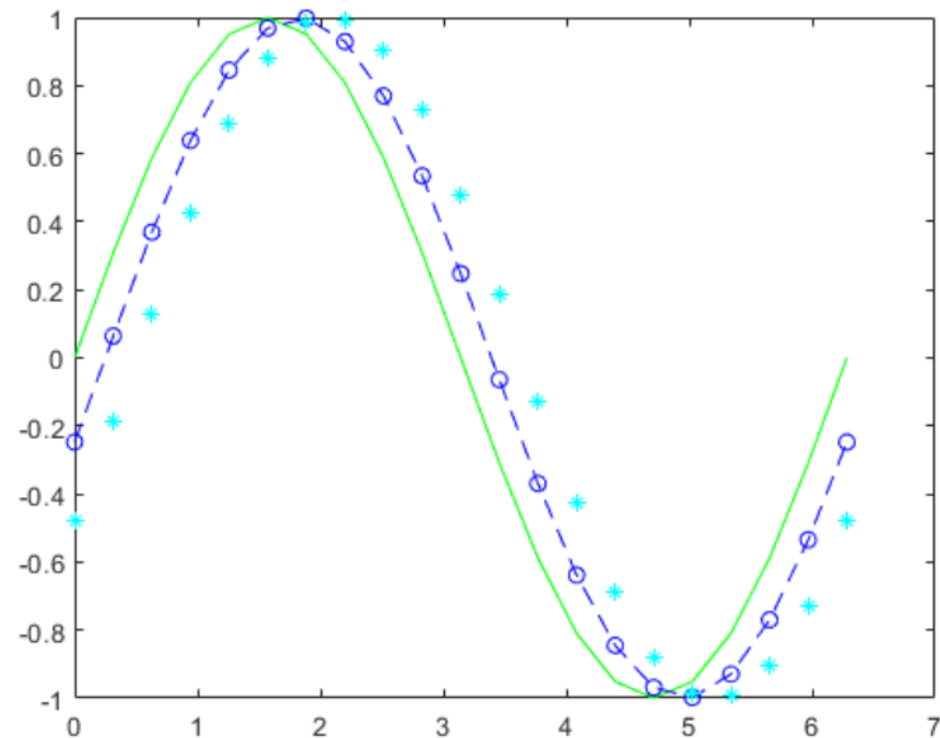
» 数据可视化：二维线图

- 以创建二维线图为例：更改线型和标记

- 更改线条和标记的外观

通过plot 函数包含的可选线条设定进行更改

```
x = 0:pi/10:2*pi;  
y1 = sin(x);  
y2 = sin(x-0.25);  
y3 = sin(x-0.5);  
  
figure  
plot(x,y1,'g', x,y2,'b--o', x,y3,'c*')
```



» 数据可视化

Plot function 二维线图

```
plot(x,y, 'line specifiers', 'PropertyName', PropertyValue)
```

- 更改线条和标记的外观

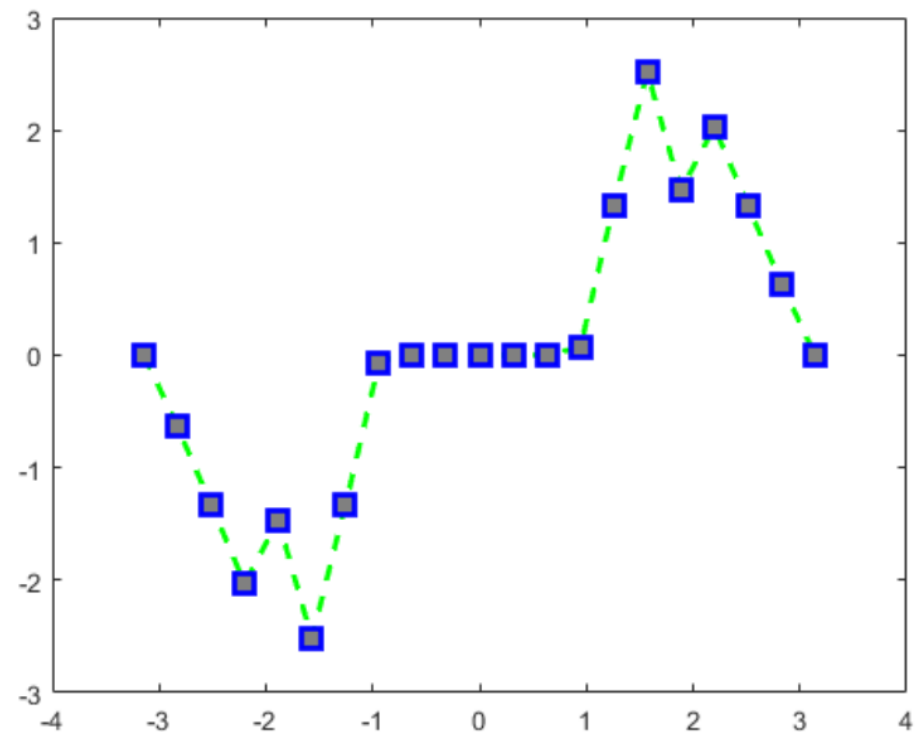
名称-值参数

MarkerEdgeColor	标记轮廓颜色	RGB三元组、十六进制颜色代码、颜色名称或短名称
MarkerFaceColor	标记填充颜色	
LineWidth	线条宽度	0.5（默认） 正值
MarkerIndices	要显示标记的数据点的索引	1:length(YData)（默认） 正整数向量 正整数标量
MarkerSize	标记大小	6（默认） 正值

» 数据可视化：二维线图

- 以创建二维线图为例：更改线型和标记
 - 更改线条和标记的外观
- 通过**名称-值参数**，在调用 plot 函数时设定更改

```
plot(x,y,'--gs',...  
     'LineWidth',2,...  
     'MarkerSize',10,...  
     'MarkerEdgeColor','b',...  
     'MarkerFaceColor',[0.5,0.5,0.5])
```

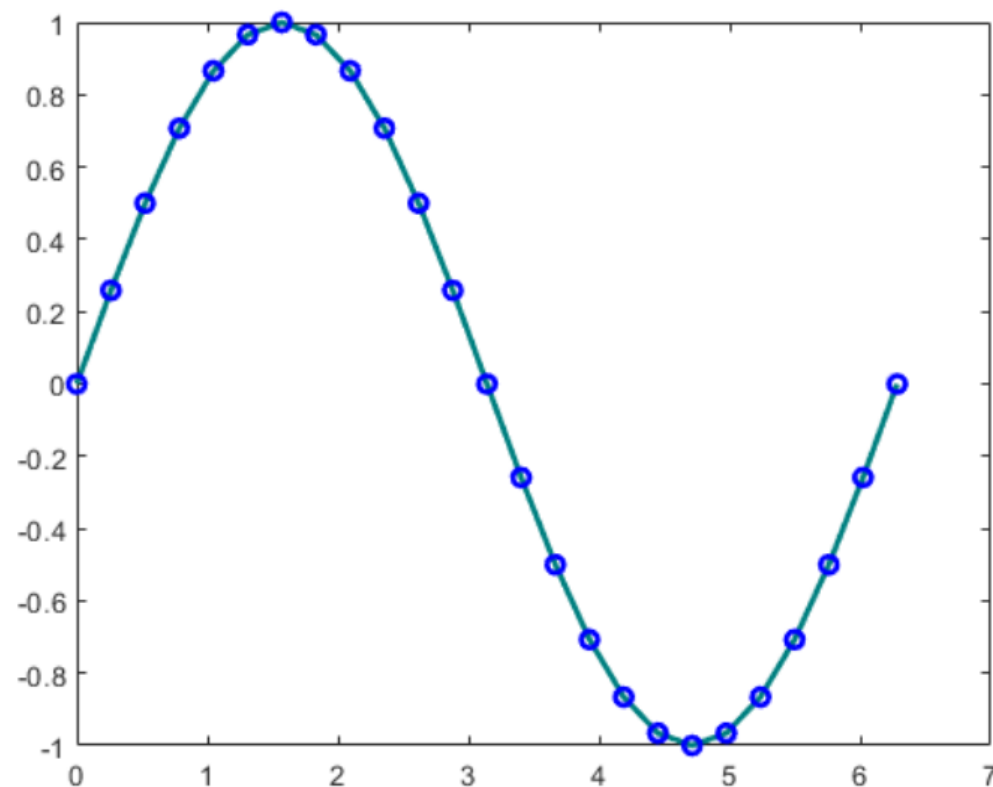


» 数据可视化：二维线图

- 以创建二维线图为例：
 - 更改线条和标记的外观

将创建的 Line 对象赋给变量 ln，使用圆点表示法访问各属性。

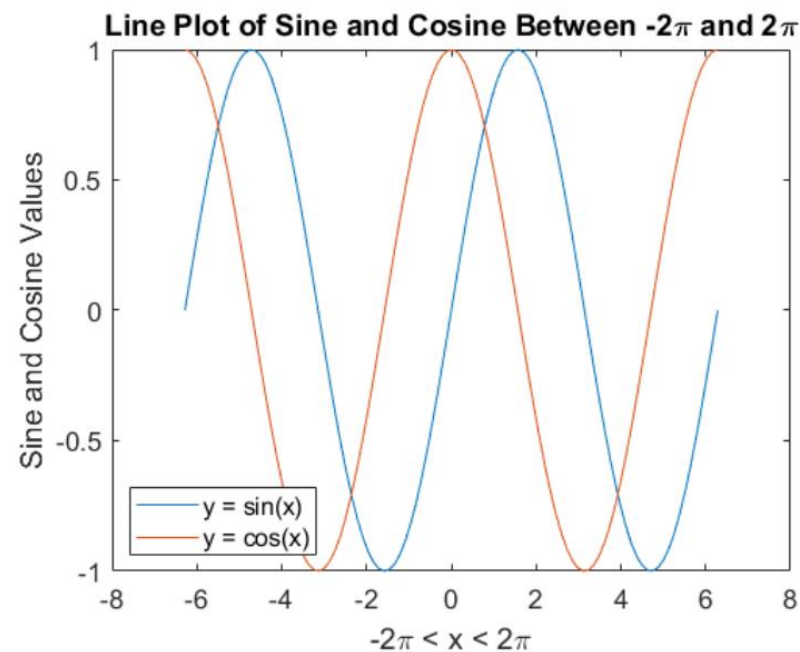
```
x = linspace(0,2*pi,25); y = sin(x);  
ln = plot(x,y)  
ln.LineWidth = 2;  
ln.Color = [0 0.5 0.5];  
ln.Marker = 'o';  
ln.MarkerEdgeColor = 'b';
```



» 数据可视化：二维线图

- 以创建二维线图为例：添加格式和注释
 - 直接使用函数命令添加：
 - 标题(title)、坐标轴标签(xlabel, ylabel)、图例(legend)、文本 (text)、注释 (annotation)
 - 使用gca获取当前图的坐标区或图，赋予axes对象，然后使用圆点表示法设置图像属性

```
ax = gca;  
ax.FontSize = 13;
```



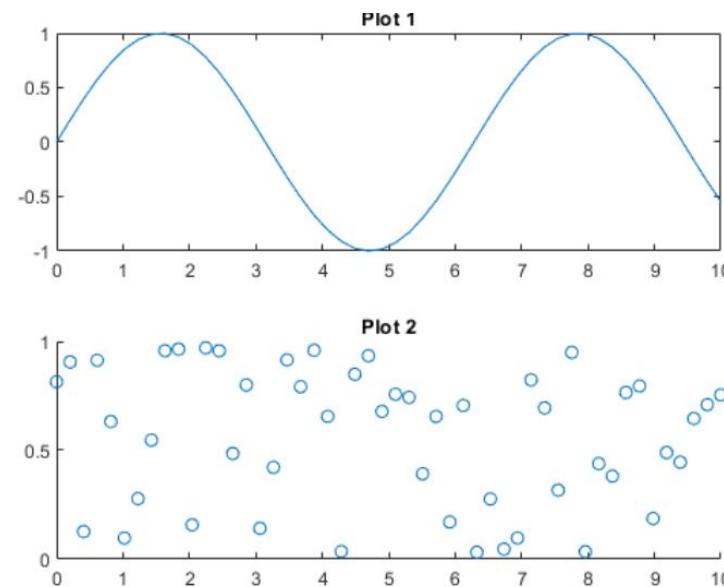
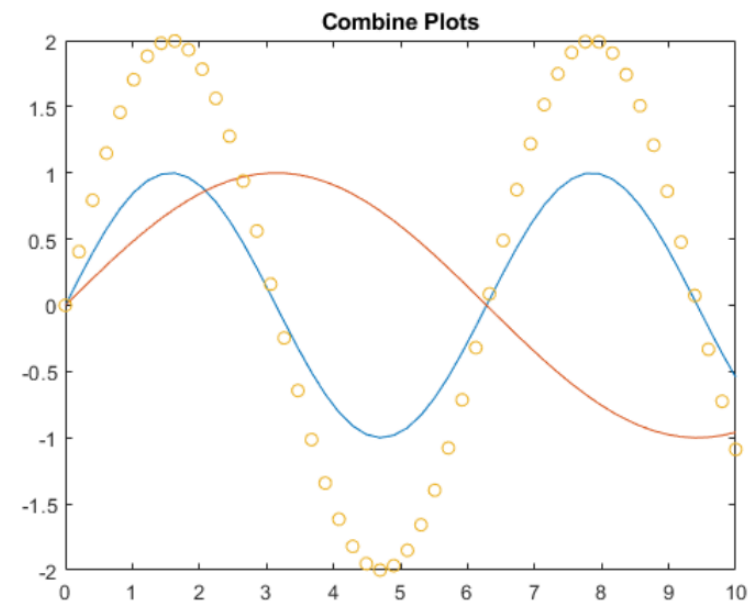
» 数据可视化：二维线图

- 以创建二维线图为例：添加格式和注释
 - axes属性
 - 字体: FontName, FontWeight, FontSize, FontAngle, TitleFontWeight
 - 刻度: XTick, YTick, XTickLabel, TickDir, TickLength
 - 标尺: XLim, YLim, XAxis, XAxisLocation
 - 网格: XGid, Layer, GridColor, GridAlpha
 - 标签: Title, Subtitle, Legend
 - 多个绘图: ColorOrder, LineStyleOrder
 - 颜色图和透明图: Colormap, ColorScale
 -

» 数据可视化：合并绘图

- 同一坐标区中合并绘图
 - hold on/off
- 在图窗中显示多个坐标区
 - tiledlayout(m,n)

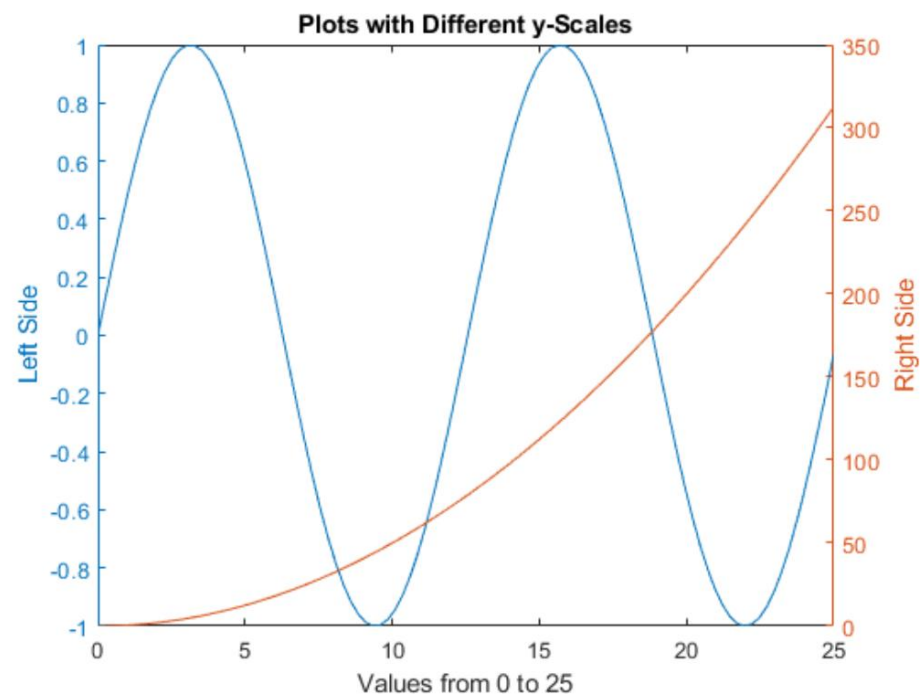
```
x = linspace(0,10,50);  
y1 = sin(x);  
y2 = rand(50,1);  
tiledlayout(2,1)  
% Top plot  
nexttile  
plot(x,y1)  
title('Plot 1')  
% Bottom plot  
nexttile  
scatter(x,y2)  
title('Plot 2')
```



» 数据可视化：创建包含双 y 轴的图

- 使用 `yyaxis` 函数创建左侧和右侧带有 y 轴的图表 `hold on/off`

```
x = linspace(0,25);  
y = sin(x/2);  
  
yyaxis left  
plot(x,y);  
r = x.^2/2;  
title('Plots with Different y-Scales')  
xlabel('Values from 0 to 25')  
ylabel('Left Side')  
  
yyaxis right  
plot(x,r);  
ylabel('Right Side')
```





浙江大学
ZHEJIANG UNIVERSITY

脚本和函数

Matlab命令的执行方式

交互式命令执行方式（命令窗口）

逐条输入，逐条执行，操作简单、直观，但速度慢，执行过程不能保留。

M文件的程序执行方式

将命令编成程序存储在一个文件中（M文件），依次运行文件中的命令，可以重复进行。

Matlab程序设计有传统高级语言的特征，又有自己独特的特点，可以利用数据结构的特点，使程序结构简单，编程效率高。

脚本

```
b = 5;  
h = 3;  
a = 0.5*(b.*h)
```

文件名(N): triarea.m

保存类型(T): MATLAB 代码文件 (*.m)

triarea

脚本文件只是将一系列相关的代码结合封装

函数

```
function a = triarea(b,h)  
a = 0.5*(b.*h);  
end
```

保存

```
a1 = triarea(1,5)  
a2 = triarea(2,10)  
a3 = triarea(3,6)
```

函数文件是在脚本文件的基础之上多添加了一行函数定义行

函数文件是以函数声明行 “function...”作为开始的，其实质就是用户往MATLAB函数库里边添加了子函数

函数文件的基本格式

```
function [outputArg1,outputArg2] = name(inputArg1,inputArg2)
%name 此处是有关此函数的摘要
% 此处显示变量的详细说明
% 记录修改版本日期等信息
if nargin < 2 % nargin返回输入参数格式
    inputArg2 = 2;
end
outputArg1 = inputArg1 + inputArg2;
outputArg2 = inputArg2 * inputArg1;
end
```

示例

```
function [out1,out2] = partsum(in1,in2)
if nargin > 1
    out1 = in1+in2;
else
    out1 = in1;
end
out2 = in1;
end
```

*输入参数将按顺序进入函数，推荐使用如上所示 if *nargin* 结构管理默认参数
一般来说，函数的输入和输出只能低于规定数量而不能高于，比如

```
>> [a,b] = partsum(1)
[c] = partsum(1,2)
```

能够正常得到结果

a	1
b	1
c	3

```
>> [d,e,f] = partsum(1,2)
[g,h] = partsum(1,2,3)
```

而

错误使用 partsum
输出参数太多。

就会报错

错误使用 partsum
输入参数太多。

M文件：脚本和函数

脚本文件没有输入参数，也不返回输出参数；

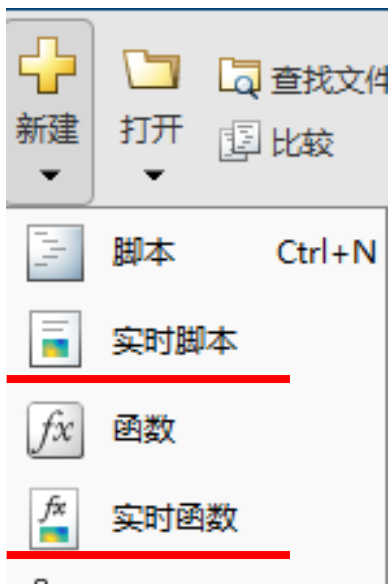
脚本文件中的变量都是全局变量，对工作空间中的变量进行操作，文件中所有命令的执行结果也返回工作区中；

脚本文件可以直接运行

函数文件可以带输入参数，也可以返回输出参数

函数文件中定义的变量为局部变量，除非使用global命令，否则当函数文件执行完毕时，这些变量也被清除

函数文件不能直接运行，要以函数调用的方式来调用它



当切换当前窗口时，脚本/函数文件就会保存，所以调试或新写文件时建议使用实时脚本/实时函数或另存为备份。



浙江大学
ZHEJIANG UNIVERSITY

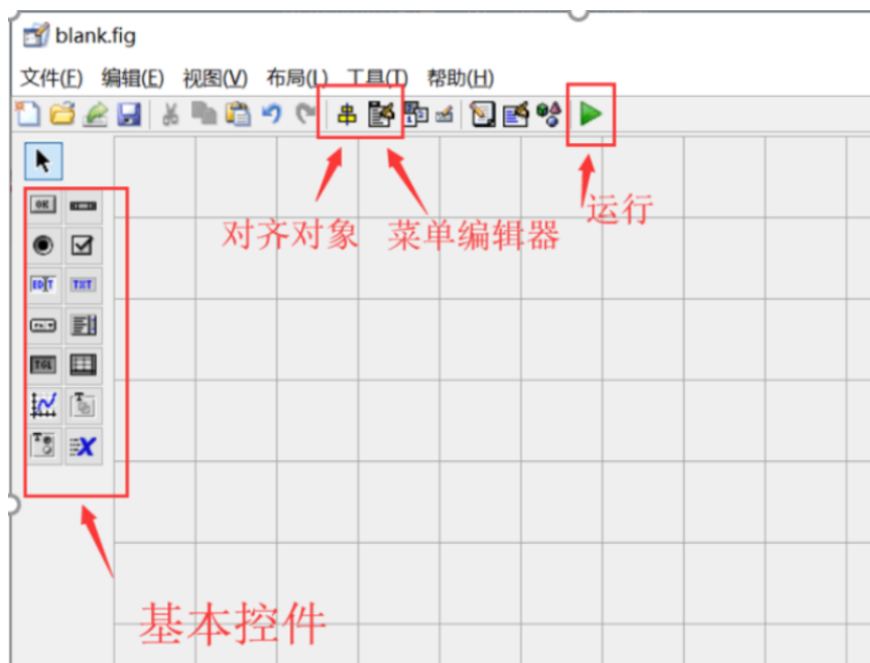
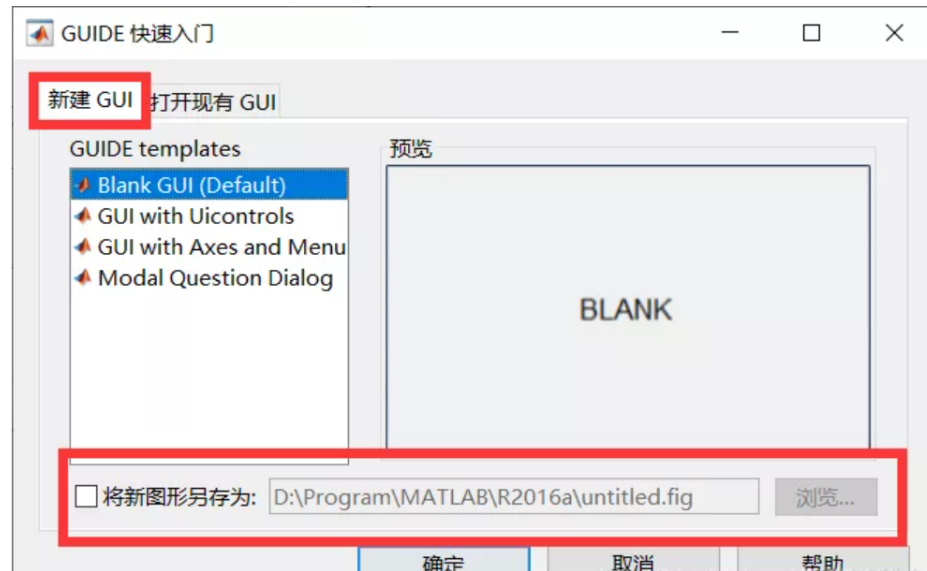
GUI界面设计



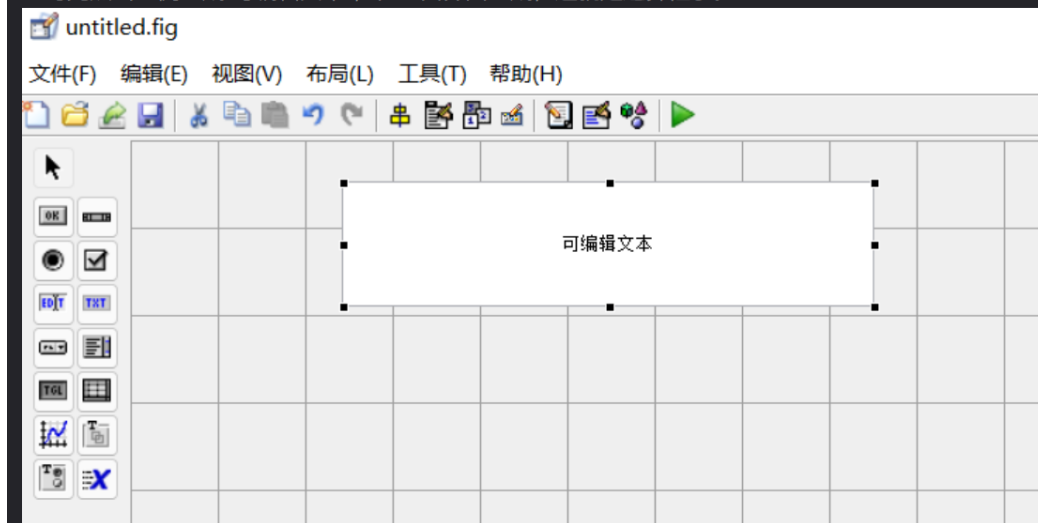
1 命令行窗口输入>>guide



弹出GUIDE快速入门窗口，选择新建GUI(或打开现有的)，将保存路径前面的勾打上，然后点击浏览，选择想要保存的路径



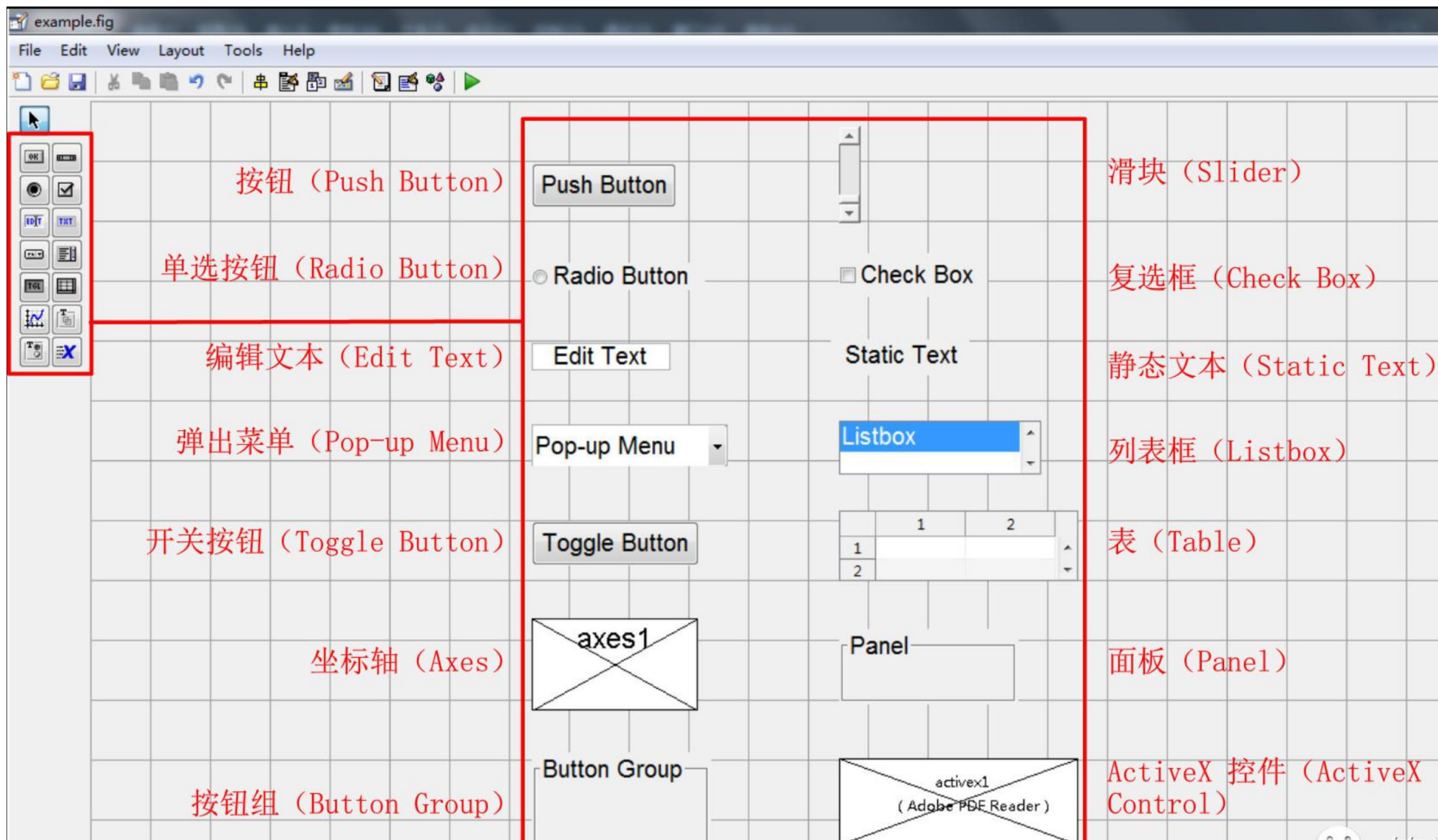
此时先点击左侧生成“可编辑文本”，在空白界面生成，这就是运算框了。





GUI控件介绍

选择 (Select)、按钮 (Push Button)、滑块 (Slider)、单选按钮 (Radio Button)、复选框 (Check Box)、编辑文本 (Edit Text)、静态文本 (Static Text)、弹出菜单 (Pop-up Menu)、列表框 (Listbox)、开关按钮 (Toggle Button)、表 (Table)、坐标轴 (Axes)、面板 (Panel)、按钮组 (Button Group)、ActiveX 控件 (ActiveX Control)



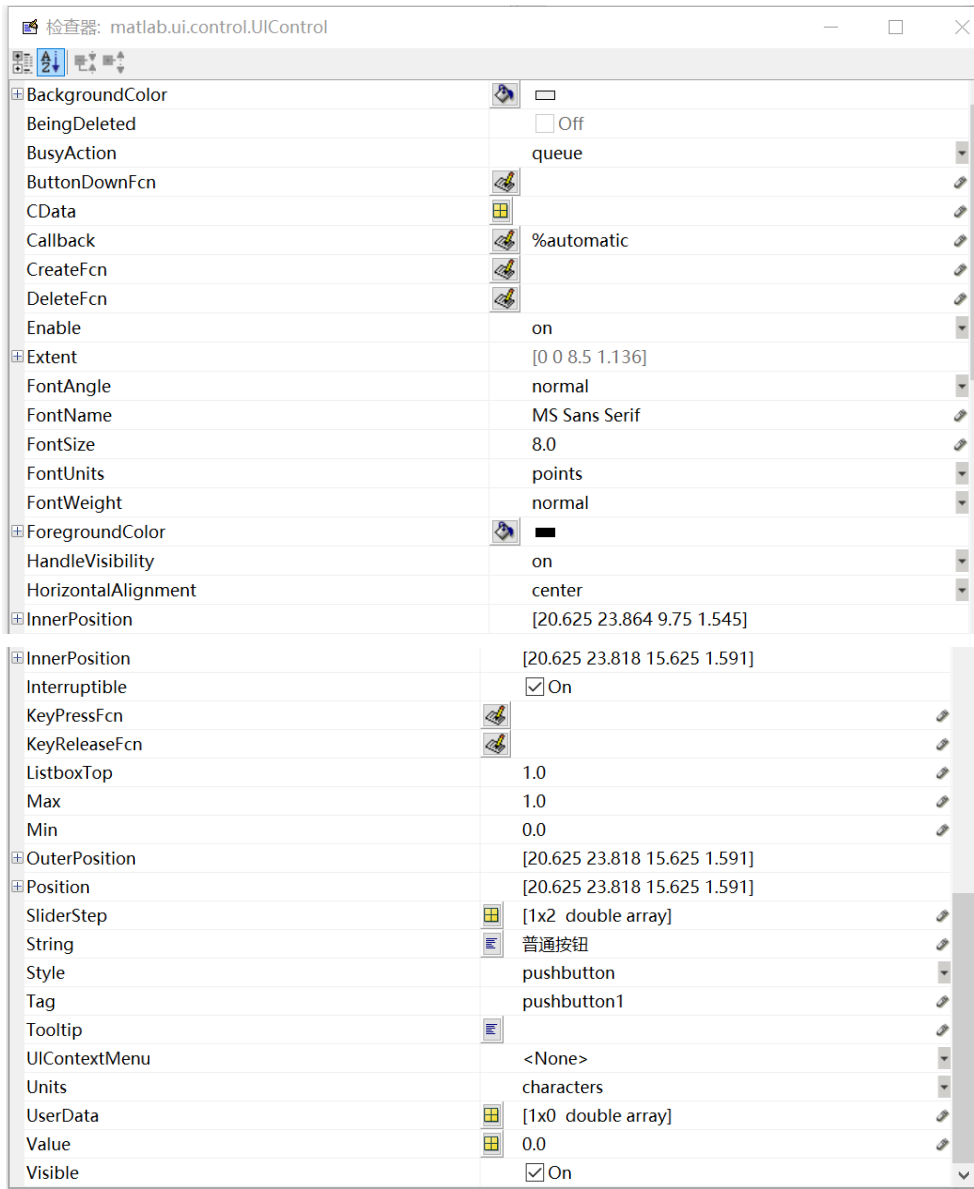


在GUI设计面板创建控件有两种方法：

- ①用鼠标选中左侧列表中的控件，按住鼠标左键拖动控件至GUI设计面板，即可创建控件；
- ②双击左侧列表控件，然后在GUI设计面板单击鼠标，即可创建该控件

在GUI设计面板创建控件后，双击控件或在右键菜单中选择“Property Inspector”，可以打开该控件的属性面板。重要属性：

1. 背景色 (BackgroundColor)：设置控件底色。
2. 激活 (Enable)：设置控件是否激活。不激活，则显示为灰色，无法对其进行操作。
3. 字体角度 (FontAngle)：设置字体角度。
4. 字体类型 (FontName)：设置字体类型。
5. 字体大小 (FontSize)：设置字体大小。
6. 字体单位(FontUnits)：设置字体单位。
7. 字体粗细 (FontWeight)：设置字体粗细。
8. 前景色 (ForefroundColor)：设置控件上字符颜色。
9. 位置 (Position)：设置控件位置和大小。
10. 控件上字符 (String)：设置控件上显示字符。
11. 类型 (Style)：设置控件类型。
12. 句柄 (Tag)：设置控件句柄名称。这个名称是之后回调函数里该控件的引用名称。
13. 单位 (Units)：设置控件单位，与是否缩放有关。
14. 值 (Value)：控件选中状态或内容数字索引，代表控件是否被选中或者选中内容条目的索引号。
15. 可见性 (Visible)：设置控件是否可见。



1.按钮 (Push Button)：用来响应用户的触发指令，特别是最终的执行命令。

2.滑块 (Slider)：滑块控件主要是作为进度条使用。

3.单选按钮 (Radio Button)：一般2个以上和按钮组组合使用，只选其一。

4.复选框 (Check Box)：复选框常用于需要多选的情况。

5.编辑文本 (Edit Text)：用于获取用户输入或某些需要反复编辑的情况。

6.静态文本 (Static Text)：作为标签使用或只需要编辑一次的情况。

7.弹出菜单 (Pop-up Menu)：用于需要预设多个选项菜单的情况。

8.列表框 (Listbox)：列表框一般作为列表使用。

9.切换按钮 (Toggle Button)：一般用作状态指示，选中会变色。

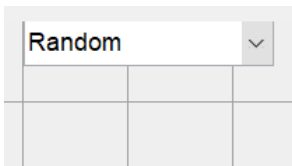
10.表 (Table)：表一般作为表格使用。

11.坐标轴 (Axes)：坐标轴一般用于显示图像。

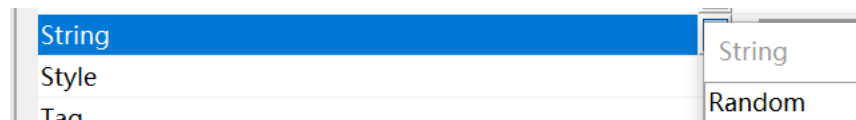
12.面板 (Panel)：面板主要用于布局设计。

13.按钮组 (Button Group)：按钮组一般和单选按钮配合使用。

14.ActiveX 控件 (ActiveX Control)：调用外部Windows控件。

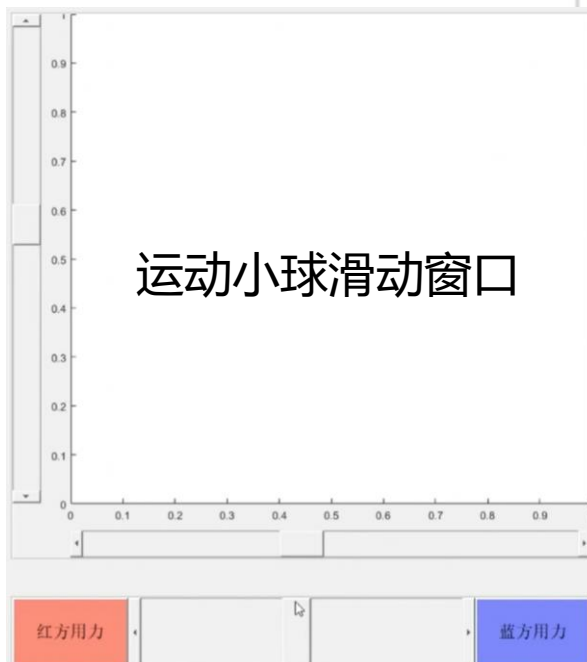


弹出式菜单显示具体选项



回调函数索引到值，赋予变量与选项对应的值

```
val=get(handles.popupmenu2,'Value');  
switch val  
    case 1  
        InitMeth=0;  
    case 2  
        InitMeth=1;  
    case 3  
        InitMeth=2;  
    case 4  
        InitMeth=3;  
    case 5  
        InitMeth=4;  
    case 6  
        InitMeth=[];
```



function GUI7()

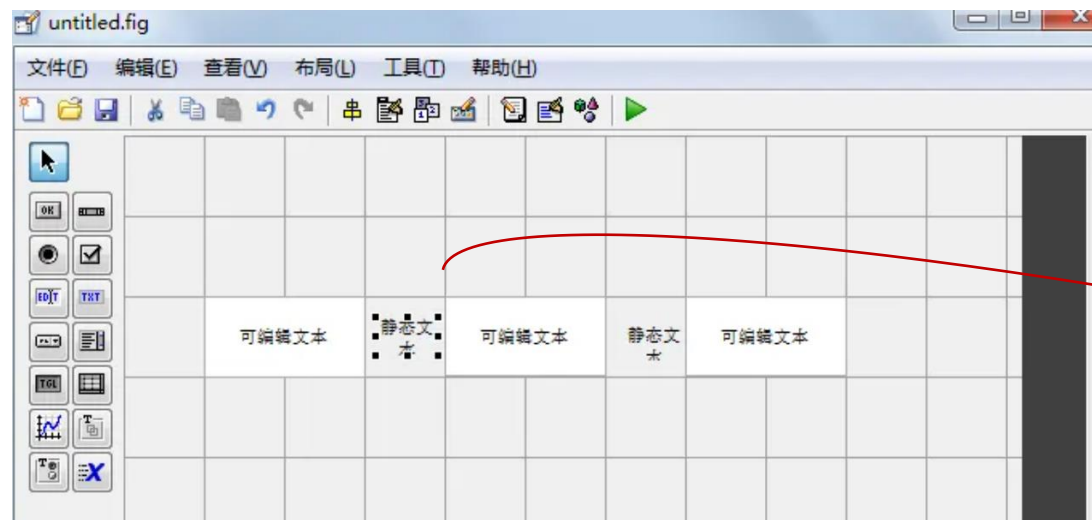
```
%% 搭框架  
Fig = figure('Position',[800,100,800,900],'menu','none');  
% 面板  
Pnl1 = uipanel(Fig,'Position',[0.05,0.2,0.9,0.75]);  
Pnl2 = uipanel(Fig,'Position',[0.05,0.05,0.9,0.1]);  
% 绘图窗  
Axes = axes(Pnl1,'Position',[0.1,0.1,0.9,0.9]);  
Lbx = uicontrol(Pnl1,'style','slider',...  
    'String',[],'FontSize',12,...  
    'Unit','normalized','Position',[0.1,0,0.9,0.05],...  
    'Max',1,'Min',-1,...  
    'Callback',@Move);  
Lby = uicontrol(Pnl1,'style','slider',...  
    'String',[],'FontSize',12,...  
    'Unit','normalized','Position',[0,0.1,0.05,0.9],...  
    'Max',1,'Min',-1,...  
    'Callback',@Move);  
  
% 操作窗  
uicontrol(Pnl2,'style','pushbutton',...  
    'String','红方用力','FontSize',16,'BackgroundColor',[1,0.5,0.5],...  
    'Unit','normalized','Position',[0,0,0.2,1]);  
uicontrol(Pnl2,'style','pushbutton',...  
    'String','蓝方用力','FontSize',16,'BackgroundColor',[0.5,0.5,1],...  
    'Unit','normalized','Position',[0.8,0,0.2,1]);
```

参考教程：

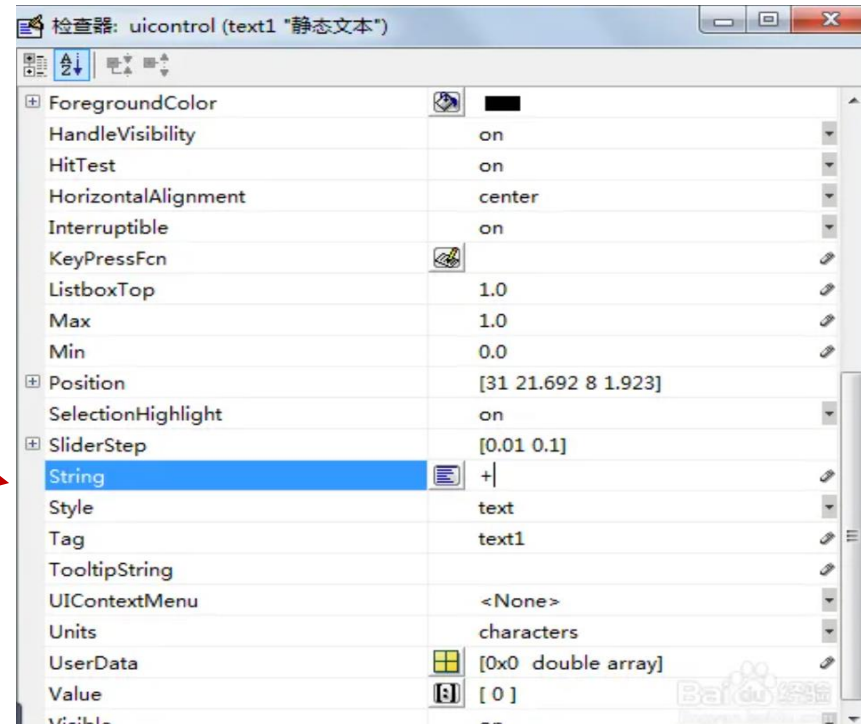
https://www.bilibili.com/video/BV1S3411K7p5/?vd_source=9252e477d4cbe64b84e61b86fe7cb031



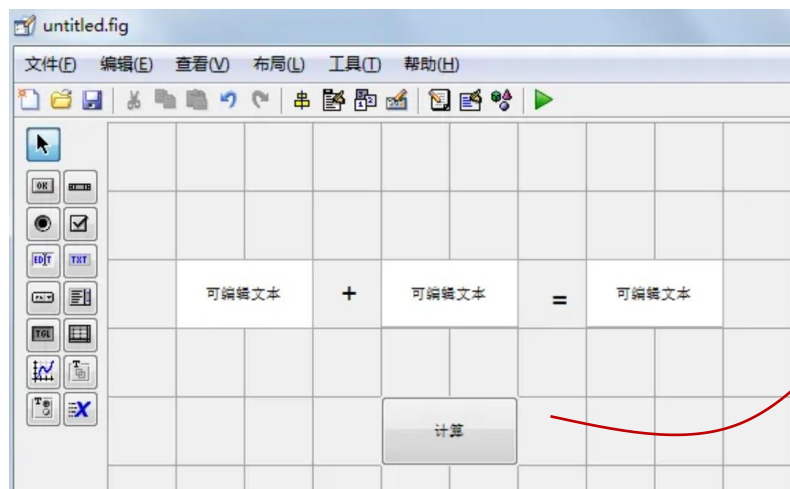
编写简易计算器：



双击静态文本，将String分别改为 “+” 和 “=”



从左边拖动的方式，添加一个按钮，并双击这个按钮，将String改为 “计算”

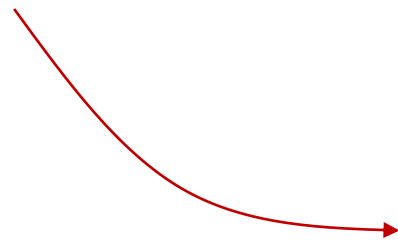
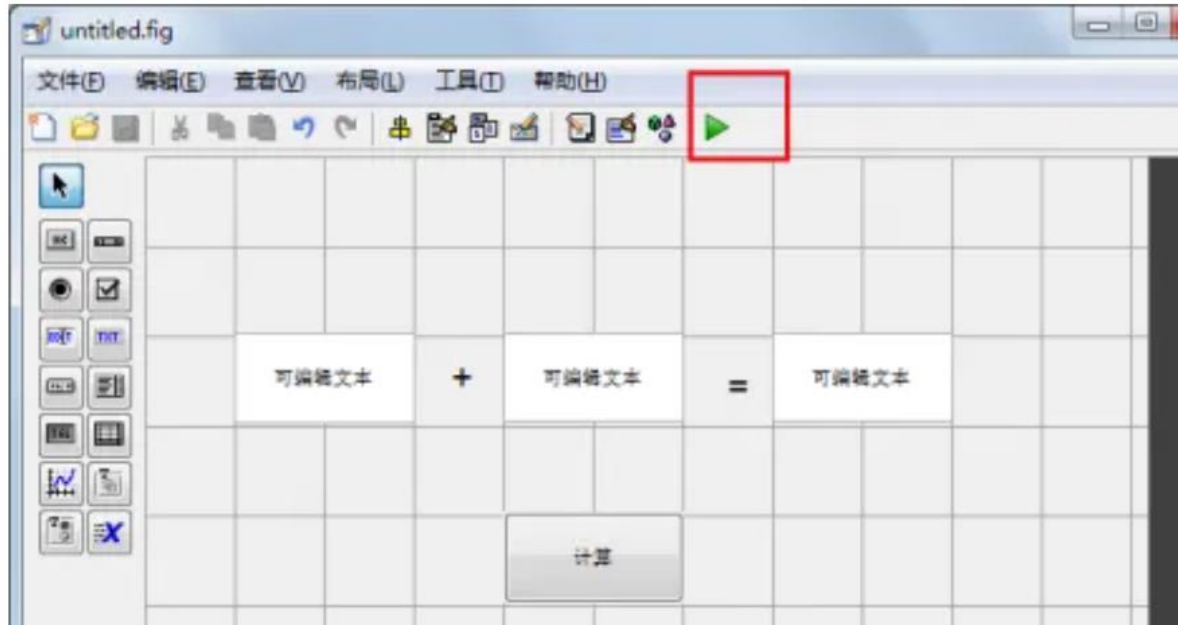


在按钮“计算”上右键，选择“查看回调” -> Callback，然后添加如下代码，用于进行加法计算

```
% --- Executes on button press in pushbutton1.  
function pushbutton1_Callback(hObject, eventdata, handles)  
% hObject    handle to pushbutton1 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
a = str2double(get(handles.edit1,'string'));  
b = str2double(get(handles.edit2,'string'));  
c = a + b;  
set(handles.edit3,'String',num2str(c));  
a = str2double(get(handles.edit1,'string'));  
b = str2double(get(handles.edit2,'string'));  
c = a + b;  
set(handles.edit3,'String',num2str(c));
```



点击图形界面上绿色三角，打开图形界面形式的加法计算器





浙江大学
ZHEJIANG UNIVERSITY

编程规范概要

编程语言规范

使用简单易懂的变量名：weight_lst、height_matrix、RIASEC_data

变量名间减少混淆：除非作为中间变量临时使用并确保通过clear命令即时销毁，
不要在同一文件中使用point和Point或者points作为变量名

不要使用保留字：严禁将sum、mean等函数名用作变量名

```
for i = 1:10 %循环
    k = k + i;
end
```

控制好缩进：在循环结构和分支结构时多缩进一个tab键

初始化变量：在使用临时变量时总是先清空原来的变量,如 RI = []; count = 0;

空行：在逻辑自然中断的部分使用空行分割代码，使结构明晰。

注释：对变量意义、整个代码块的行为添加足量的注释

空格：在等号等运算符前后加上空格，使代码美观。

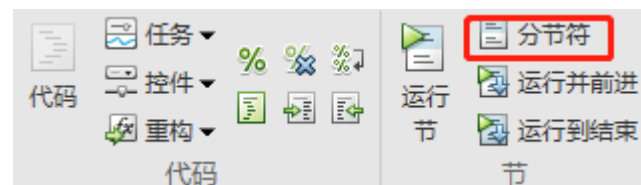
代码分节

非实时文件



非实时文件中也可以手动用两个注释符号%%实现分节

实时文件



分节的代码能够逐节运行，一般来说，一个维护良好的代码大多含有如下节：

- 清除节：clc;clear;close all等命令组成，清理工作区、命令窗格、绘图窗格节约内存
- 定义节：包括全局变量的声明及参数值的设定
- 运行节：实际需要执行运算的命令主体部分
- 绘图节：可视化结果
- 局部函数节：包含脚本中使用的一些局部函数

➤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；紫色：属性值的设定；蓝色：循环和条件控制等流程

➤在运行程序之前，必须设置好MATLAB的工作路径，使得所要运行的程序及运行程序所需要的其他文件处在当前目录之下，只有这样，才可以使程序得以正常运行。否则可能导致无法读取某些系统文件或数据，从而程序无法执行。

➤通过cd指令在命令窗口中可以更改、显示当前工作路径

示例：一个良好分节的matlab脚本程序

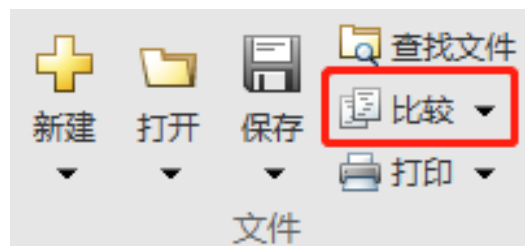
```
%%
clear;clc;close all;
%%
syms x;
k = 1;
%%
]for i = 1:10 %循环
    k = k + i;
end
f = x * innerfun(k);
%%
fplot(f)
%%
]function [a]=innerfun(i)
    a = i + 3;
end
```

当前选中的节
会被高亮

示例：一个良好分节的matlab实时脚本

1	clear;clc;close all;
2	syms x;
3	实时函数允许在代码注释之外提供额外的文本说明和注解。
4	<input type="button" value="运行"/> %实时脚本还允许使用控件添加额外的控制
5	for i = 1:10 %循环
6	f = innerfun(i) * x;
7	end
8	fplot(f)
9	function [a]=innerfun(i)
10	a = i + 3;
	end

比较



通过交换和合并能够方便的做出正确的版本



比较用于版本管理：方便比较备份文件和修改后文件的差异
也可比较文件夹，查看文件夹中文件的不同

也可以使用visdiff函数进行文件比较

old.m

E:\360MoveData\Users\Acedia\Desktop\old.m
Thu Sep 22 15:46:56 CST 2022

new.m

E:\360MoveData\Users\Acedia\Desktop\new.m
Thu Sep 22 15:47:06 CST 2022

修改、增加、删除将以不同颜色显示

已发现 2 处差异。使用工具条按钮可以导航到这些差异。

```
1 a = 1;  
2 b = 2;  
3 c = 3;  
4 d = 4;  
5 e = 5;  
-
```

```
. a = 1;  
. b = 2;  
x c = 3 + 1;  
<  
. e = 5;  
> f = 6;  
1  
2  
3  
-  
4  
5
```

匹配的行数: 3

左侧文件中不匹配的行数: 2

右侧文件中不匹配的行数: 2

Quiz: 服务器上传了lengthoffline.m 和 lengthoffline2.m
请尝试比较这两个文件。

Find函数复习题 - 查找特定元素

1. >> x=[1 1 2 3 5 8 13]

2.

3. x =

4.

5. 1 1 2 3 5 8 13

6.

7. >> find(x>=5)

8.

9. ans =

10.

11.

?

find结果是索引，不是元素值
索引最小是1

基本选择与循环结构复习题 - 矩阵赋值

遍历矩阵并为每个元素指定一个新值。

对主对角线赋值 2，
对相邻对角线赋值 -1，
对其他位置赋值 0。

例如：2 -1 0

-1 2 -1

0 -1 2

请为右边这段代码改错。

```
1. for c = 1:ncols
```

```
2.     for r = 1:nrows
```

```
3.
```

```
4.         if r = c
```

```
5.             A(r,c) = 2;
```

```
6.         else if abs(r-c) = 1
```

```
7.             A(r,c) = -1;
```

```
8.         else
```

```
9.             A(r,c) = 0;
```

```
10.        end
```

```
11.
```

```
12. end
```

1. 逻辑表达式中应使用双等号判断

2. 应连在一起，写作elseif；分开表示在else中再加入条件判断

3. 少了一个end

一个for对应一个end

If elseif else体结束后只加一个end

基本选择与循环结构复习题 - 正确代码

```
1. for c = 1:ncols
2.     for r = 1:nrows
3.
4.         if r == c
5.             A(r,c) = 2;
6.         elseif abs(r-c) == 1
7.             A(r,c) = -1;
8.         else
9.             A(r,c) = 0;
10.        end
11.
12.    end
13. end
```

基本选择与循环结构复习 - switch语句

```
1. grade = 'B';
```

```
2.
```

```
3.
```

```
4.
```

```
5.
```

```
6.
```

```
7.
```

```
8.
```

```
9.
```

```
10
```

```
11
```

```
12
```

```
13
```

```
14
```

```
15. end
```

填空:

当输入的grade为A时，输出Excellent!
为B、C时，输出Well done
为D时，输出You passed
为F时输出Better try again
其余成绩输出Invalid grade

```
switch grade
```

```
case 'A'
```

```
    fprintf('Excellent!\n' );
```

```
case {'B' , 'C'}
```

```
    fprintf('Well done\n' );
```

```
case 'D'
```

```
    fprintf('You passed\n' );
```

```
case 'F'
```


```
    fprintf('Better try again\n' );
```

```
otherwise
```

```
    fprintf('Invalid grade\n' );
```

```
end
```

使用元胞来实现多重选择



- Quiz 1:
- 对datatest.csv的数据进行操作
- 使用任意的循环结构(for/while)和选择结构(if-else/switch-case), 遍历所有被试的年龄age, 当年龄为30以下, 输出'young' (使用disp()或者fprintf()函数输出), 当年龄为31-60时, 输出'not old', 61以上输出'old'(可以认为年龄没有超过150的)


```
for i = 1:5000
    switch age(i)
        case mat2cell(1:30,1,ones(1,30))
            fprintf('young' );
        case mat2cell(31:60,1,ones(1,30))
            fprintf('not old' );
        case mat2cell(61:150,1,ones(1,90))
            fprintf('old' );
    end
```

← 这里硬要用switch-case结构非常不方便

```
if age(i)>=1 && age(i)<=30
    fprintf('young' );
elseif age(i)>=31 && age(i)<=60
    fprintf('not old' );
else
    fprintf('old' );
end
end
```

← If-else结构就很顺利； 注意逻辑and要用两个&符号

基本选择与循环结构复习题扩充 - while语句

```
1. limit = 0.8;
2. s = 0;
3.
4. while 1
5.     tmp = rand; % 用 tmp
6.     if tmp > limit
7.         continue
8.     end
9.     s = s + tmp;
10. end
```

输出结果是？怎么改正？

continue跳过该循环次后进入下一循环，
应改为break跳出整个循环

- Quiz 2:
- 对datatest.csv的数据进行操作
- 找出测试当年参加过总统选举(voted栏数据 = 1)的人，求他们每个人TIPI量表的总分
- 使用find函数找到voted = 1的人的索引，再写循环，对TIPI总分对应的索引处求和

```
data = readtable('datatest.csv');  
TIPI = table2array(data(:,52:61)); %获取TIPI数据  
voted = table2array(data(:,87)); %获取voted数据  
ind = find(voted == 1); %获取索引， 列向量  
for i = ind' %对索引中的被试循环求和， 索引转换成行向量  
    result(i) = sum(TIPI(i,:));  
end
```

- Quiz 3:
- 创建一个1*2的图窗，设置大标题为“函数图像”
- 设置窗口大小，使宽为900，高为200
- 在第一个子图中绘制sin(x)和cos(x)的图像，x轴范围[0,4pi]
- 设置子图标题为“正弦函数余弦函数图像”
- 其中，sin(x)的线条为红色，cos(x)的线条为蓝色
- X轴的刻度值以pi/2为最小单位，y轴刻度值以0.2为最小单位
- （提示使用转义字符串“0.5\pi”在刻度区显示符号 π ）
- 向子图添加图例legend，显示在子图外侧右上角
- 在第二个子图中画出椭圆 $\frac{a^2}{16} + \frac{b^2}{9} = 1$ ，椭圆与坐标轴相切
- 设置子图标题为“椭圆图像”

```
figure
set(gcf,'position',[300 200 900 200])
sgtitle("函数图像")
syms x
subplot(1,2,1)
fplot(sin(x),[0,4*pi],'-r')
hold on
fplot(cos(x),[0,4*pi],'-b')
xticks([0:(1/2)*pi:4*pi])
xticklabels({'0','0.5\pi','1\pi','1.5\pi','2\pi','2.5\pi','3\pi','3.5\pi','4\pi'})
yticks([-1:0.2:1])
legend('Location','northeastoutside')
title("正弦函数余弦函数图像")
subplot(1,2,2)
ezplot('x^2/16 + y^2/9 = 1')
axis([-4 4 -3 3])
title("椭圆图像")
```

- Quiz 4:
- 在服务器下载SimpleCal简单计算器app
- 读懂代码，向代码中添加乘方运算

```

elseif btn == '+' || btn == '-' || btn == 'x' || btn == '÷' || btn == '^'
    ch = str(end);
    if ch == '+' || ch == '-' || ch == 'x' || ch == '÷' || ch == '^'
        str(end) = btn;
    elseif ch == '.'
        str(end) = [];
        str = [str, btn];
    else
        str = [str, btn];
    end
    app.Label.Text = str;
% 等于
elseif btn == '='
    ch = str(end);
    if ch == '+' || ch == '-' || ch == 'x' || ch == '÷' || ch == '=' || ch == '^'
        str(end) = [];
    end
    str0 = str;
    str = strrep(str, 'x', '*');
    str = strrep(str, '÷', '/');
    result = eval(str);
    app.Label.Text = [str0, '=', num2str(result)];
% 小数点
elseif btn == '.'
    ch = str(end);
    if ch == '+' || ch == '-' || ch == 'x' || ch == '÷' || ch == '^'
        str = [str, '0.'];
    elseif ch == '=' || ch == '.'
    else
        % 找最后一个点的位置
        pos = find(str=='.', 1, 'last');
        if ~isempty(pos) && ~isnan(str2double(str(pos+1:end)))

    else

```

添加完按钮后，向代码条件中加入四个并列的或在将按钮回调函数绑定到同一函数上即可
可供思考：如何添加开n次方的功能？

