

Matlab GUI 零基础学员快速入门

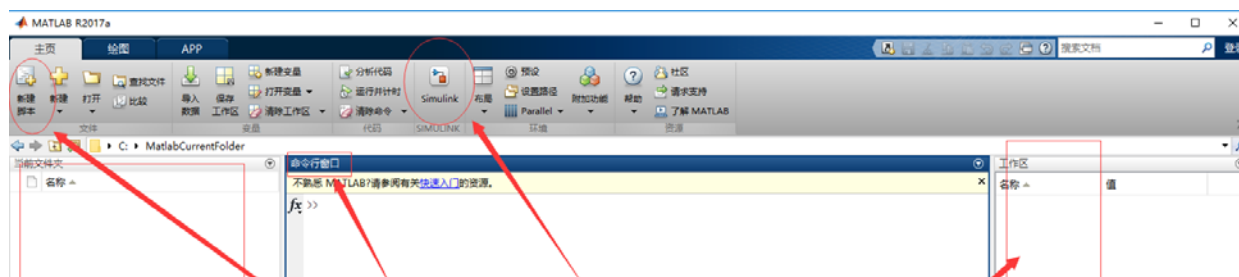
Matlab广泛地应用于数学分析、计算、自动控制、系统仿真、数字信号处理、图像处理、数理统计、人工智能、通信工程和金融系统等领域。本活动针对零基础学员，结合个人经验，使Matlab新手少走弯路、快速入门（相应的代码用斜体表示）。具体内容如下：

- Matlab 基础知识
- Matlab 绘图
- Matlab GUI 编程
- 相关知识介绍
- 交流讨论

1. Matlab 基础知识

1.1 Matlab操作环境

Matlab为用户提供了简洁易用的界面操作环境，主界面上显示的常用主要功能模块包括：新建脚本、当前工作目录、命令行窗口、工作空间、Simulink工具等。如下图所示。



示。

1.2 常用命令介绍

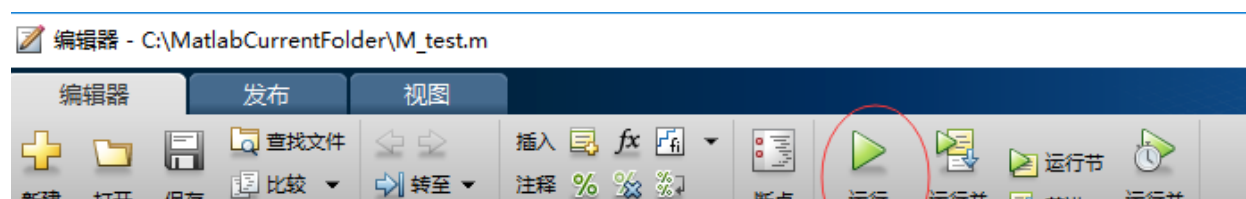
Matlab提供了一些简单的命令，平时经常用到。总结在下表中。

- `clc` 清空命令行窗口
- `clear` 清除工作空间所有变量
- `clear all` 清除工作空间所有变量和函数
- `close all` 关闭图形窗口
- `save name` 保存工作空间变量到文件name.mat中
- `save name x y` 保存工作空间变量x,y到文件name.mat中
- `load name` 载入name.mat文件中的所有变量到工作空间
- `load name x y` 载入name.mat文件中的的变量x,y到工作空间
- “*向上箭头*” 调用上一次的命
- “*向下箭头*” 调用下一行的命令
- `Tab` 函数自动补全

注1.2：了解常用命令即可，不必死记硬背。使用时，可百度查询相应的命令或者查询帮助文档。

1.3 M文件编辑器

代码通常可写在Matlab的M文件编辑器中，这样便于调试代码。使用时可通过快捷键ctrl+n或者点击Matlab主界面的“新建脚本”来创建空白的M文件。然后可在M文件中编写代码。此外，笔者个人习惯上在M文件开头写上`clc; clear all; close all;`等命令，如下图所示。

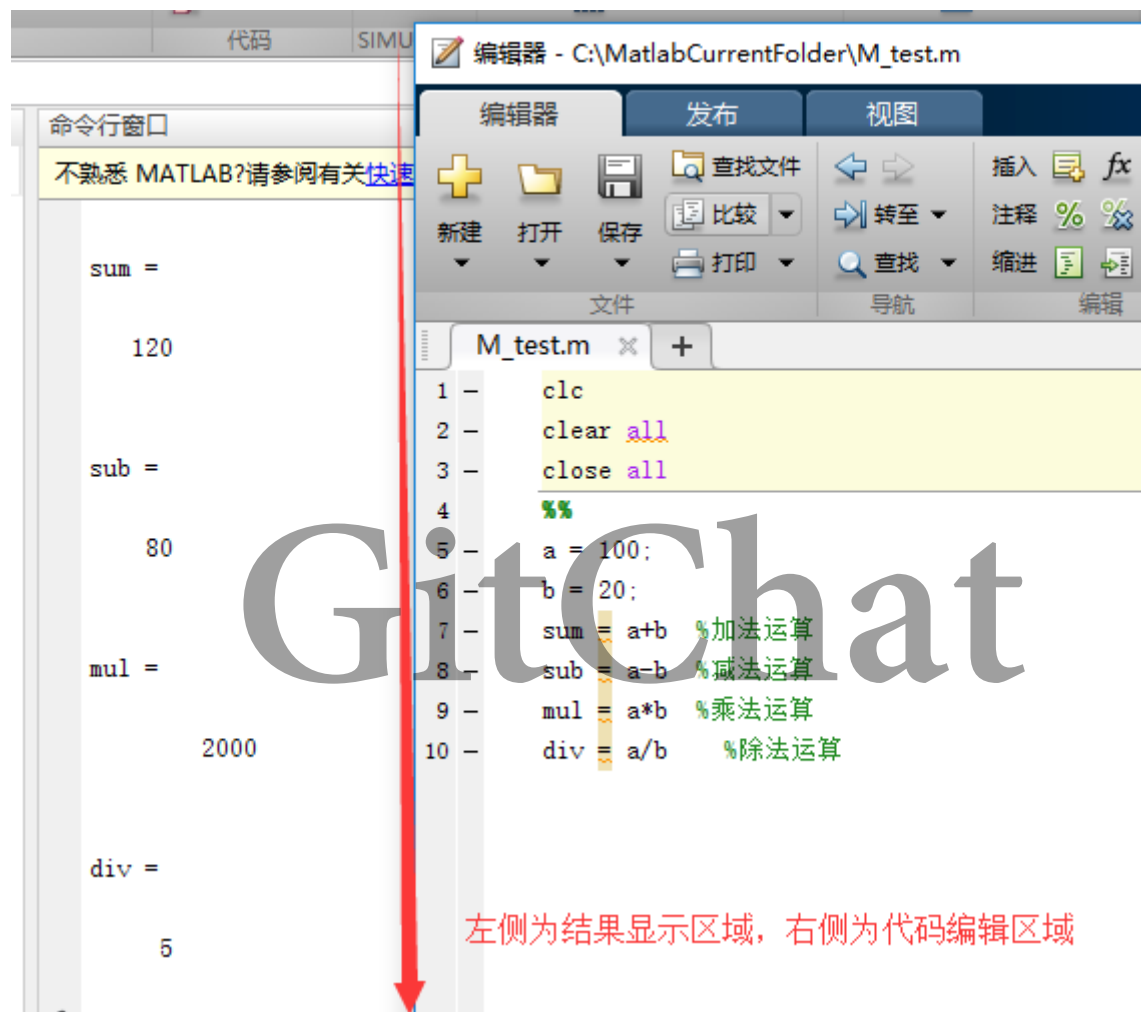


1.4 Matlab使用帮助

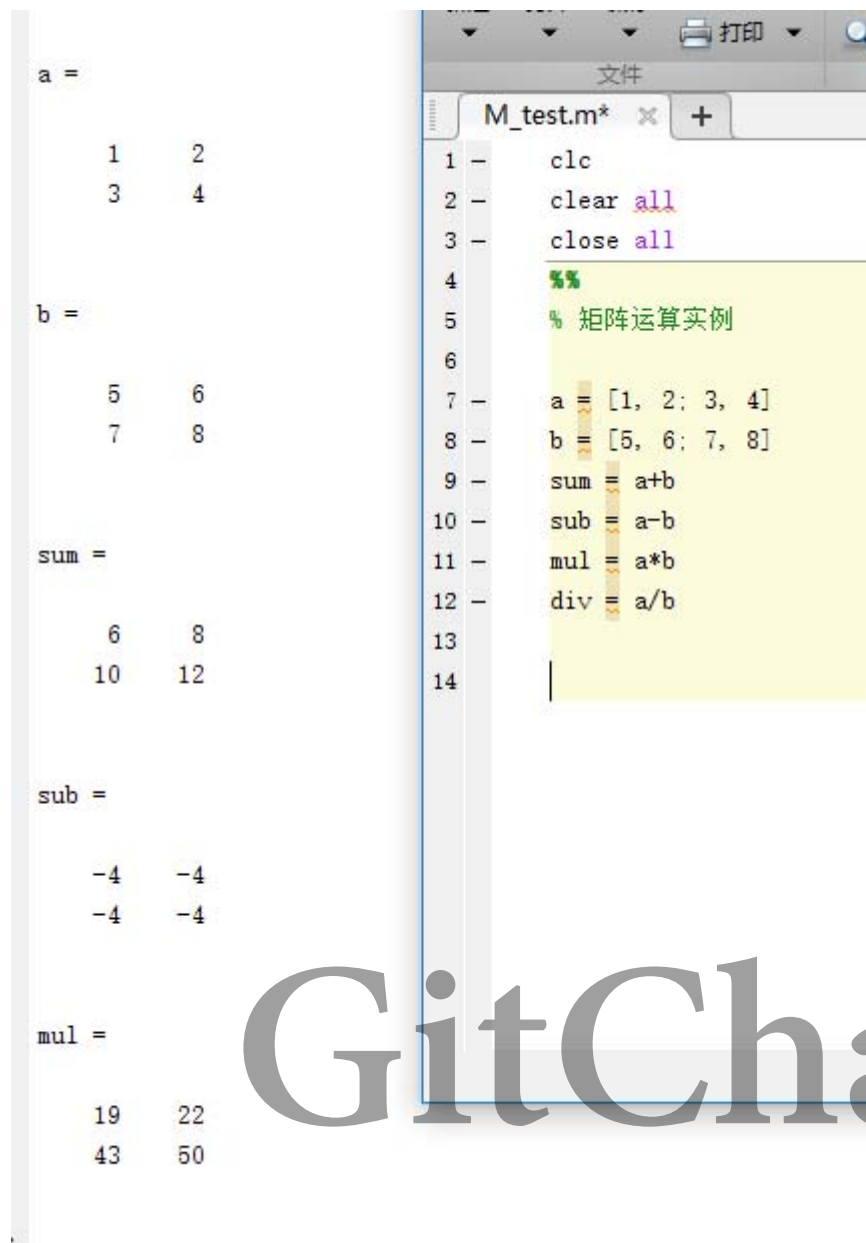
1. 可通过help命令查询相关函数使用说明
2. 可在互联网查询相关函数使用说明，并往往可以查到多个实例，便于初学者使用。

1.5 Matlab数值运算

1) 加减乘除实例



可以看到左侧为结果 / 右侧为代码编辑区域 / 运行过程和结果 / 右侧为代码编辑区域



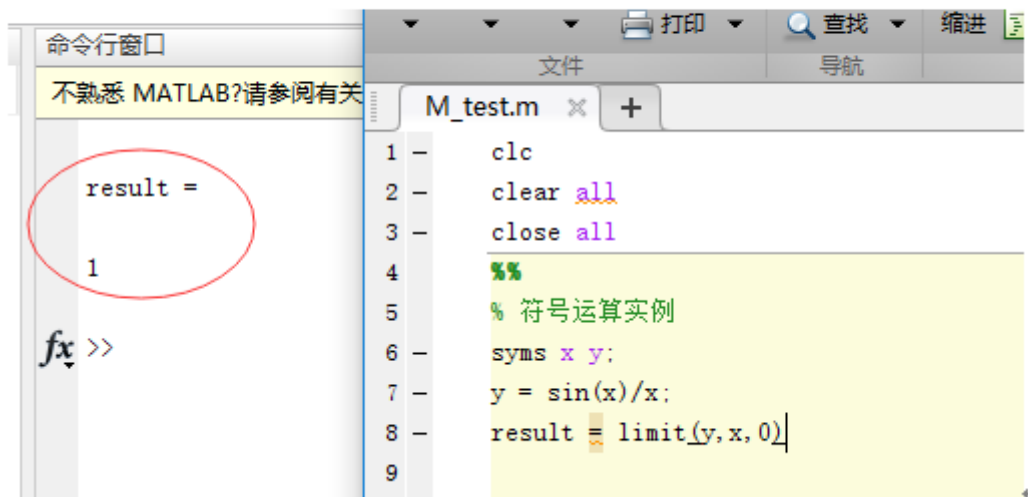
GitChat

1.6 Matlab符号运算

Matlab符号运算种类很多，支持常用的代数运算、积分、微分、极限等。

1、符号实例

求解 $\lim_{x \rightarrow 0} \frac{\sin(x)}{x}$



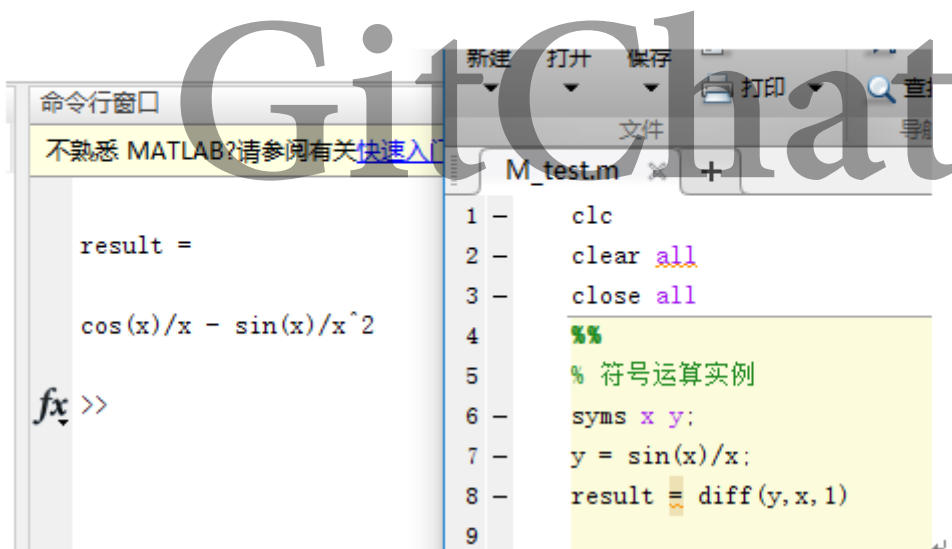
```
命令窗口
不熟悉 MATLAB? 请参阅有关快速入门
result =
1
fx >>

M_test.m
1 - clc
2 - clear all
3 - close all
4 - %%
5 - % 符号运算实例
6 - syms x y;
7 - y = sin(x)/x;
8 - result = limit(y,x,0)
9 -
```

可以看到，结果为：1。

2) 微分实例

求 $\left(\frac{\sin(x)}{x}\right)'$



```
命令窗口
不熟悉 MATLAB? 请参阅有关快速入门
result =
cos(x)/x - sin(x)/x^2
fx >>

M_test.m
1 - clc
2 - clear all
3 - close all
4 - %%
5 - % 符号运算实例
6 - syms x y;
7 - y = sin(x)/x;
8 - result = diff(y,x,1)
9 -
```

求解 $\int x e^x dx$

```
不熟悉 MATLAB?请参阅有关快速入门
M_test.m
1 - clc
2 - clear all
3 - close all
4 - %%
5 - % 符号运算实例
6 - syms x y;
7 - y = x*exp(x);
8 - result = int(y)
```

结果为: $e^x(x-1)$

此外，Matlab还支持其他一些常用的符号计算。相应的函数如下：

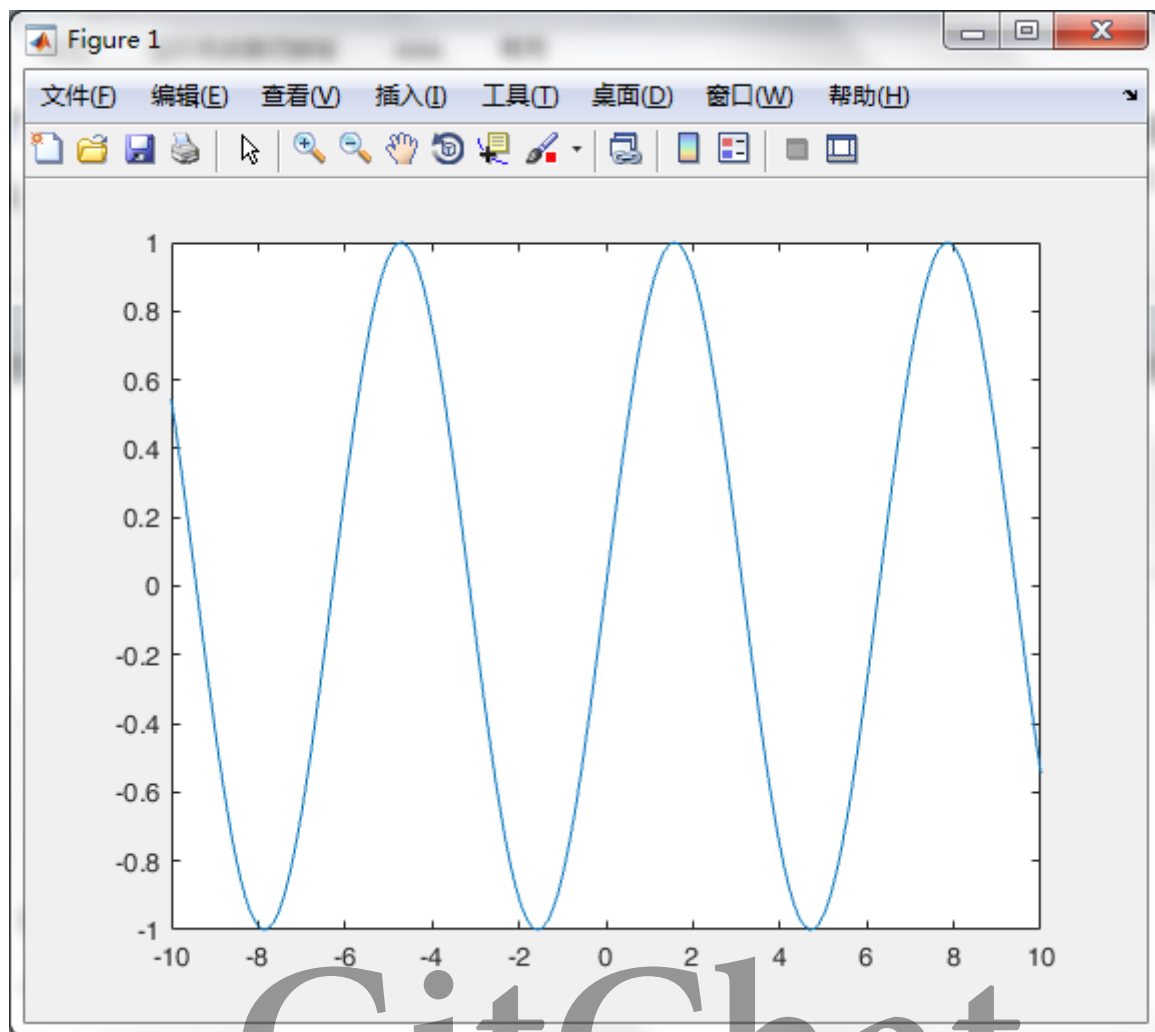
- 合并表达式 *collect*
- 展开表达式 *expand*
- 因式分解 *factor*
- 嵌套表达式 *horner*
- 计算最小分母公因式 *Numden*
- 求反函数 *finverse*

2. Matlab 绘图

2.1 绘二维图形

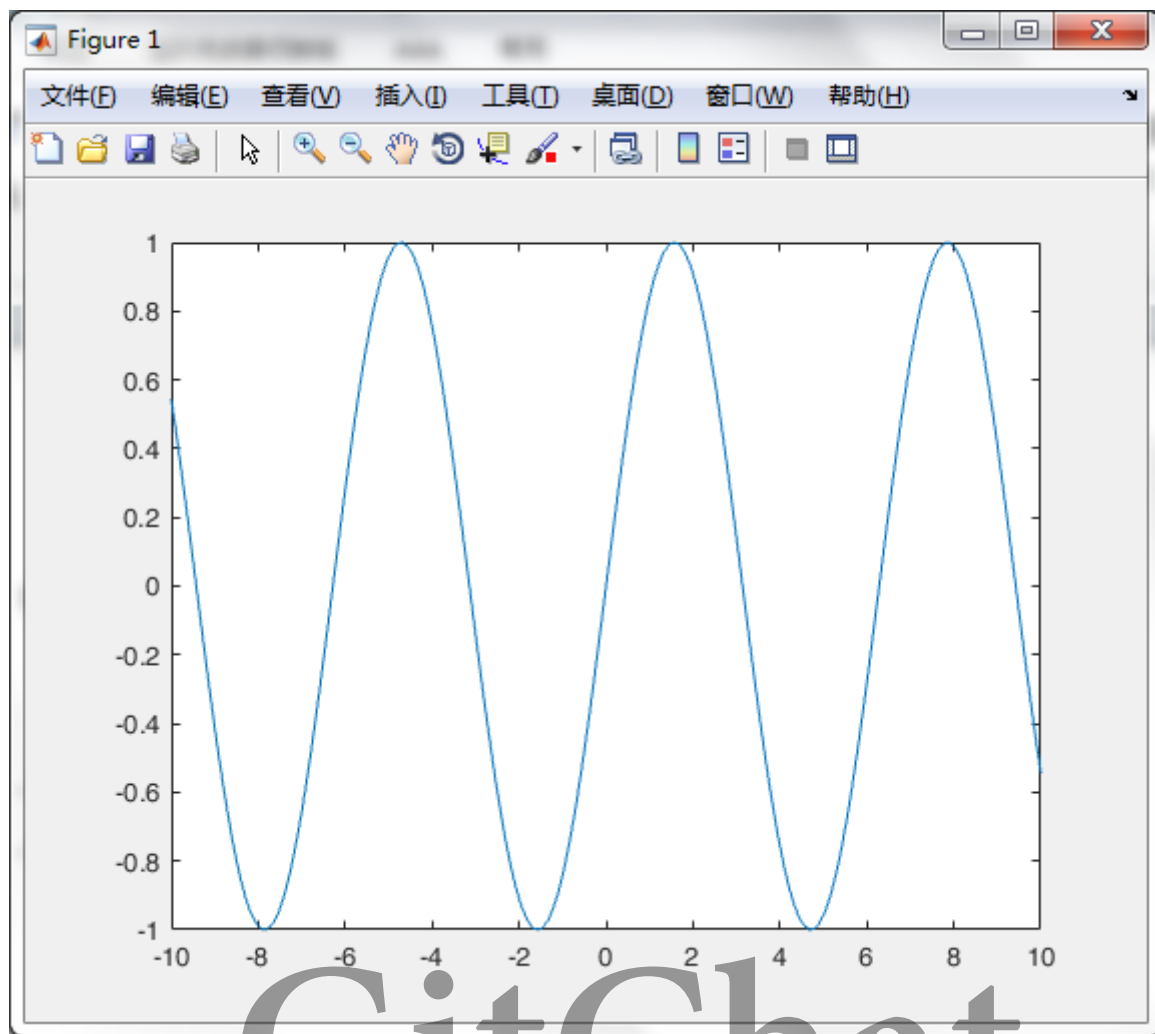
1) 实例1：绘制正弦曲线（使用的函数：*plot*）。代码如下：

```
clc
```



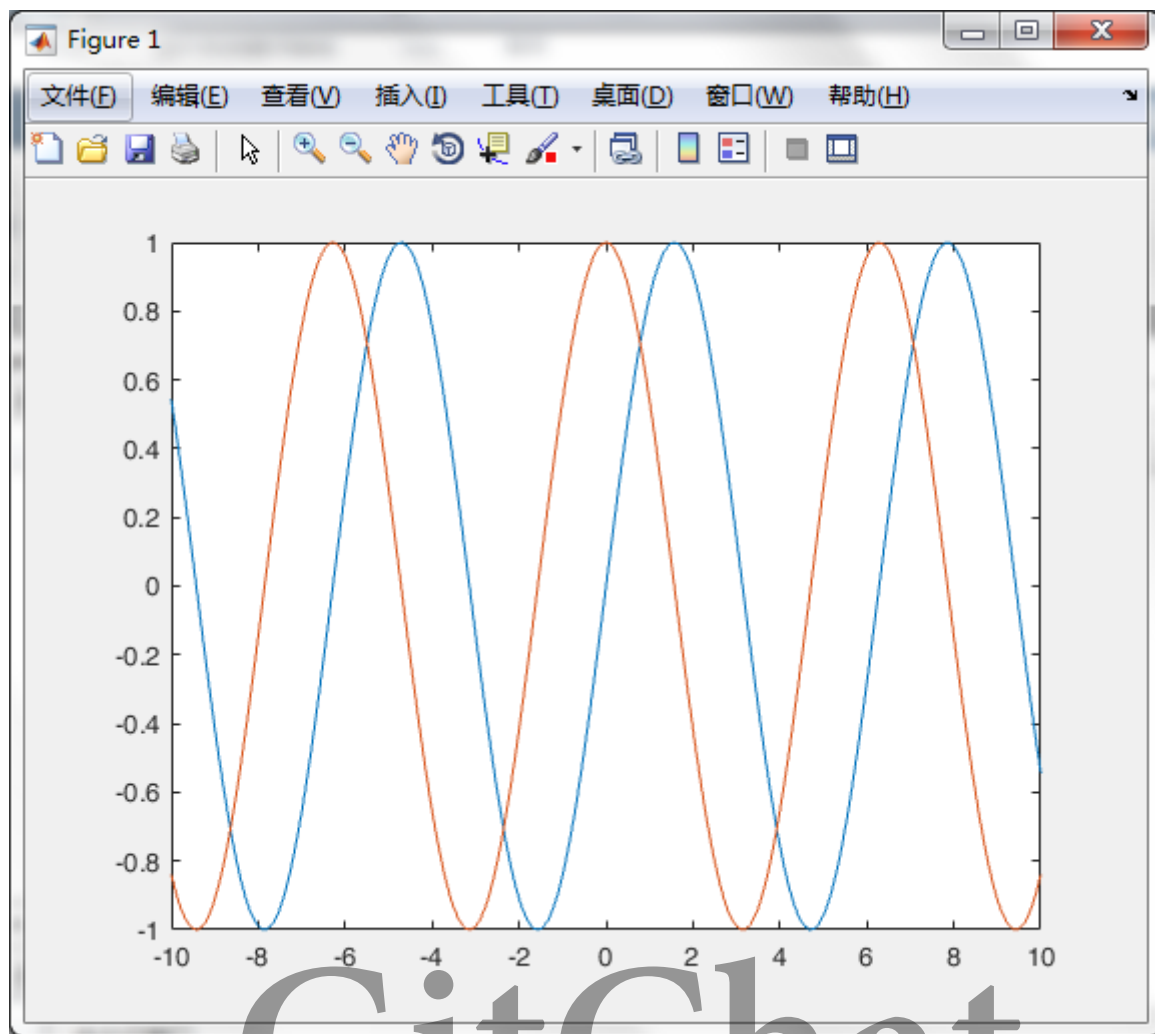
2) 实例2：绘制余弦曲线。代码如下：

```
clc :  
clear all  
close all  
%%  
x = -10:0.1:10;  
y = sin(x);  
figure(1)
```



3) 实例3：同时绘制多个图形（使用函数`hold on`）绘制正余弦曲线，代码如下：

```
clc
clear all
close all
%%
x = -10:0.1:10;
y1 = sin(x);
y2 = cos(x);
```

2.2 设置曲线属性

- - 实线
- : 虚线
- -. 点划线
- - 双划线
- *b* 蓝色
- *g* 绿色
- *r* 红色

- d 菱形
- h 六角星
- o 空心圆圈
- p 五角星
- s 方块符
- x 差字符号

1) 实例说明代码如下：

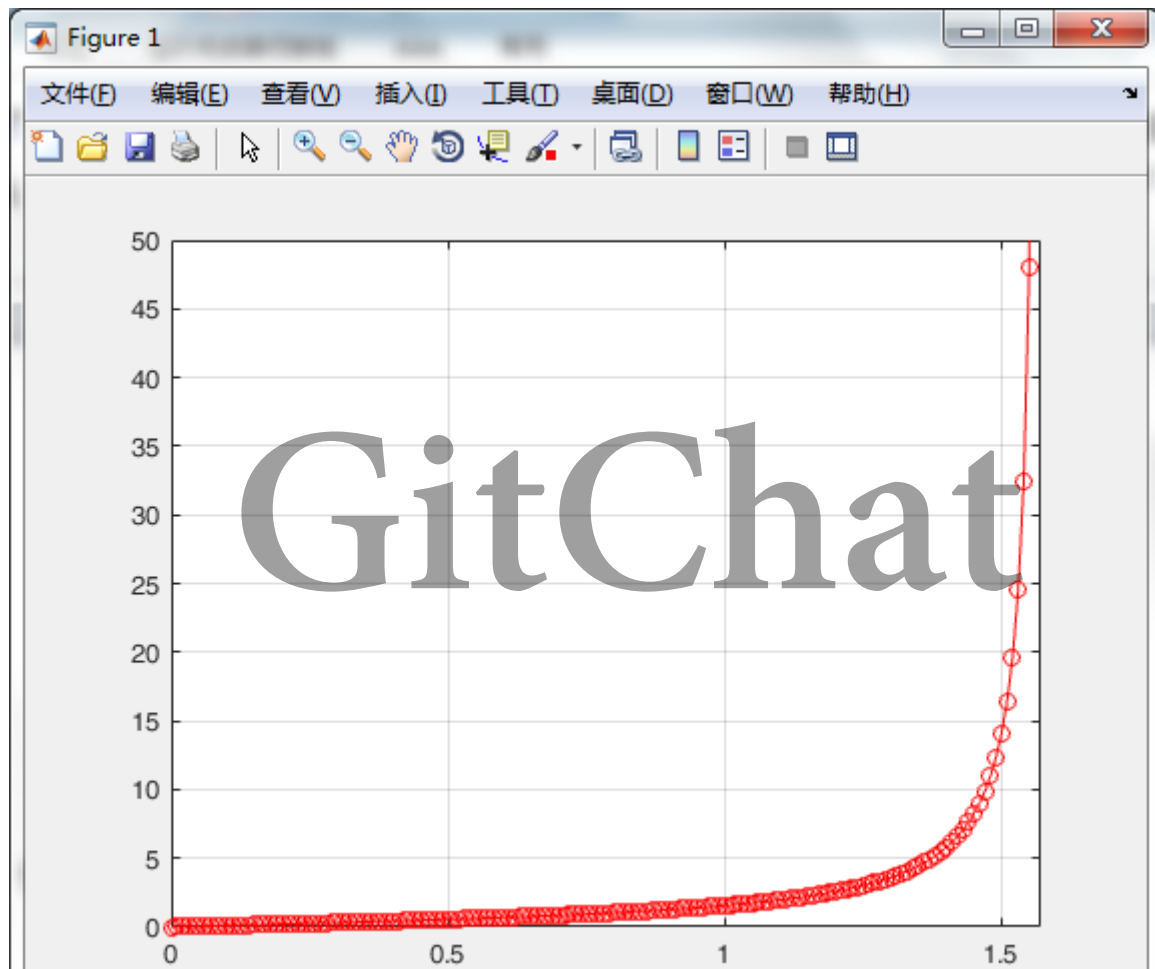
```
clc
clear all
close all
%%
x = -6*pi:0.1:6*pi;
y = sin(x);
figure(1)
plot(x,y,'r-','linewidth',2);
```

图形如下：



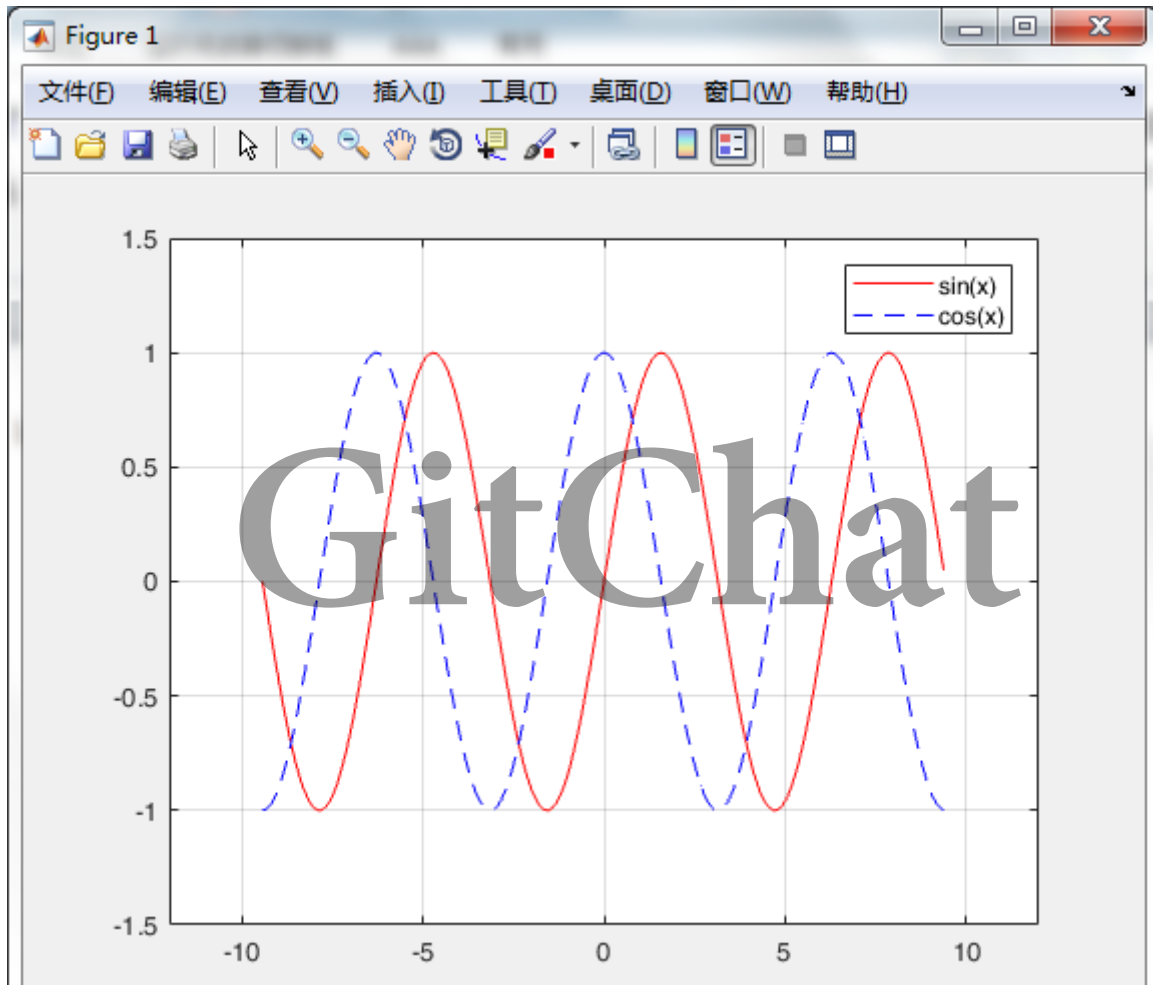
```
clc
clear all
close all
%%
x = 0:0.01:pi/2;
plot(x,tan(x),'-ro');
axis([0 pi/2 0 50])
grid on
```

相应的图形如下：



```
y2 = cos(x);  
figure(1)  
plot(x,y1,'r-');  
hold on  
plot(x,y2,'b-');  
legend('sin(x)','cos(x)');  
axis([-12 12 -1.5 1.5])  
grid on
```

相应的图形如下：



Matlab绘图命令非常丰富，未能一一详细列举，如有用到可上网查询。

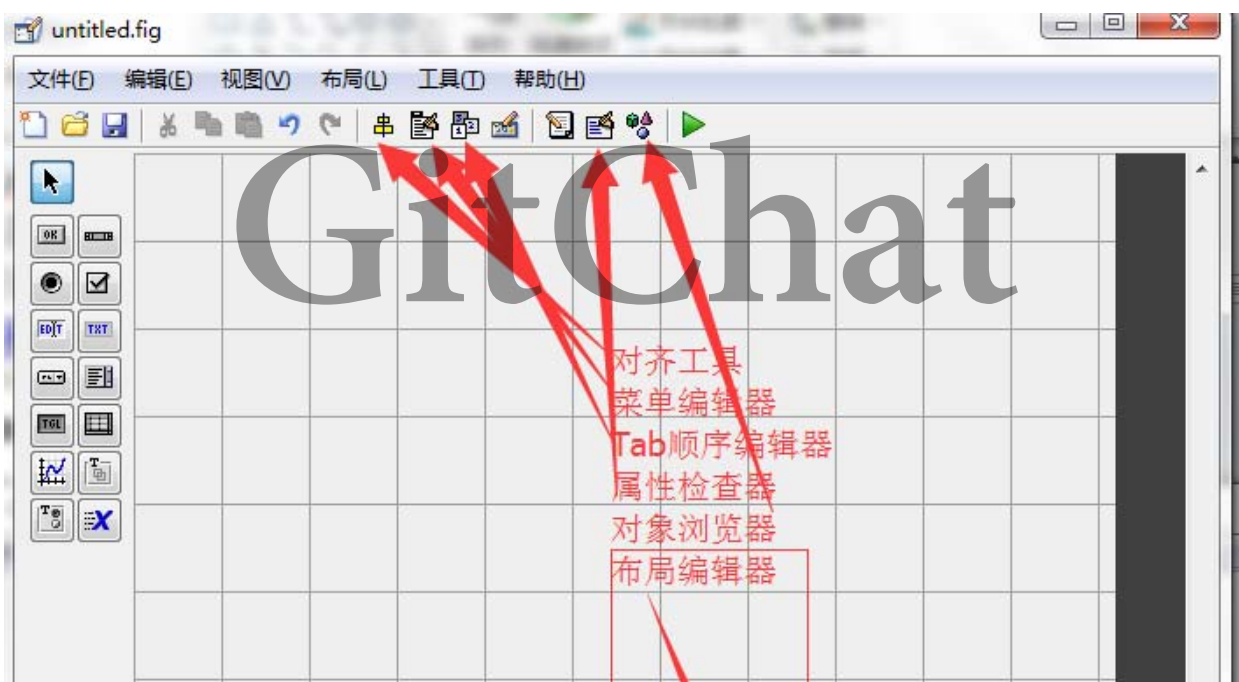
3. Matlab GUI编程

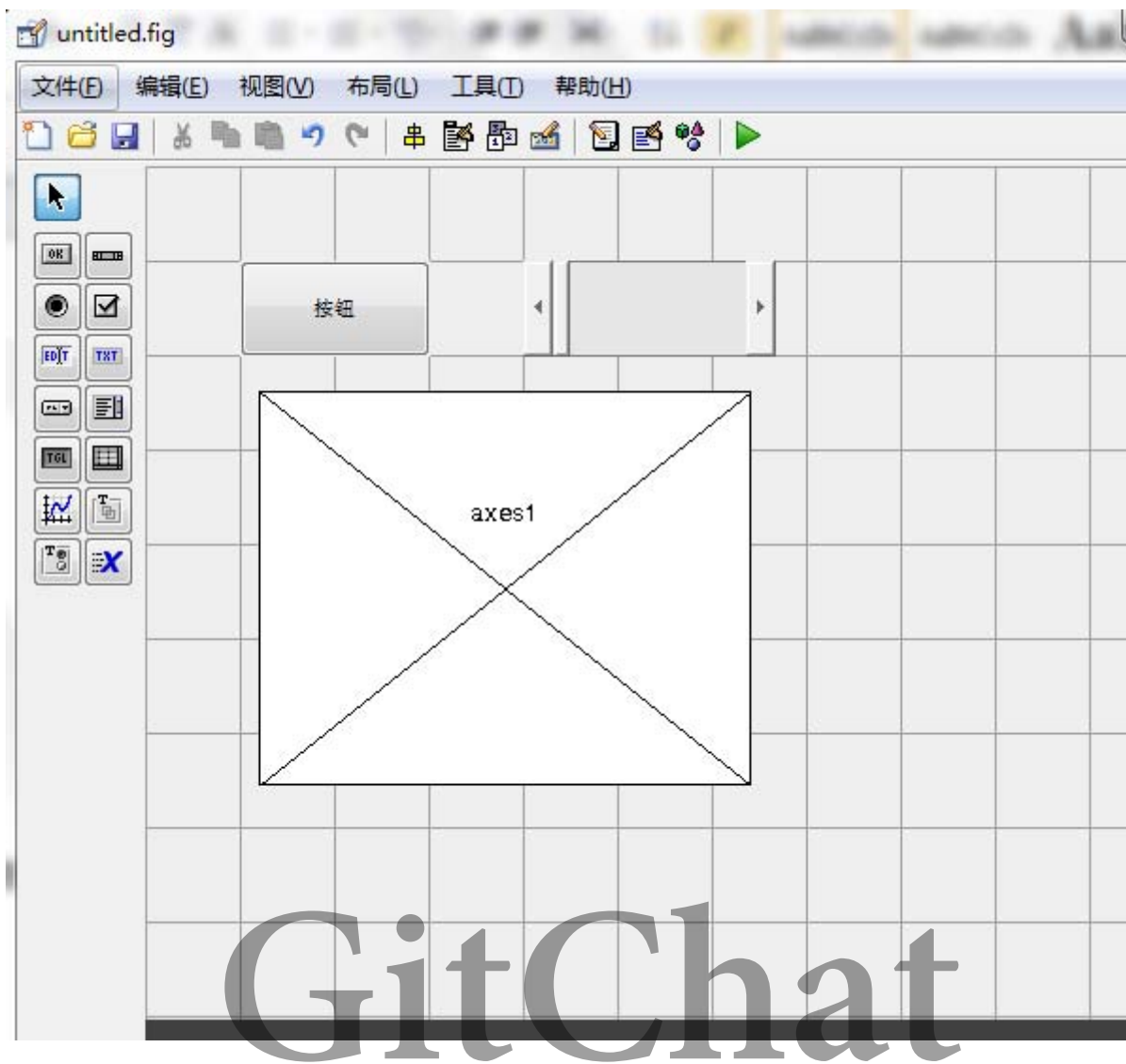
图形用户界面(GUI)是为方便用户和计算机信息交流而设计的计算机程序，它是由窗口、菜单、图标、光标、按键、对话框、文本框等组成的用户界面。它让用户定制与Matlab的交互方式，而命令窗口不是唯一与Matlab的交互方式。

3.1 图形用户界面开发环境(GUIDE)

Matlab提供了一套可视化的创建图形窗口的工具，使用图形用户界面开发环境可方便地创建GUI应用程序，它可以根据用户设计的GUI布局，自动生成M文件的框架，用户使用这一框架编制自己的应用程序。

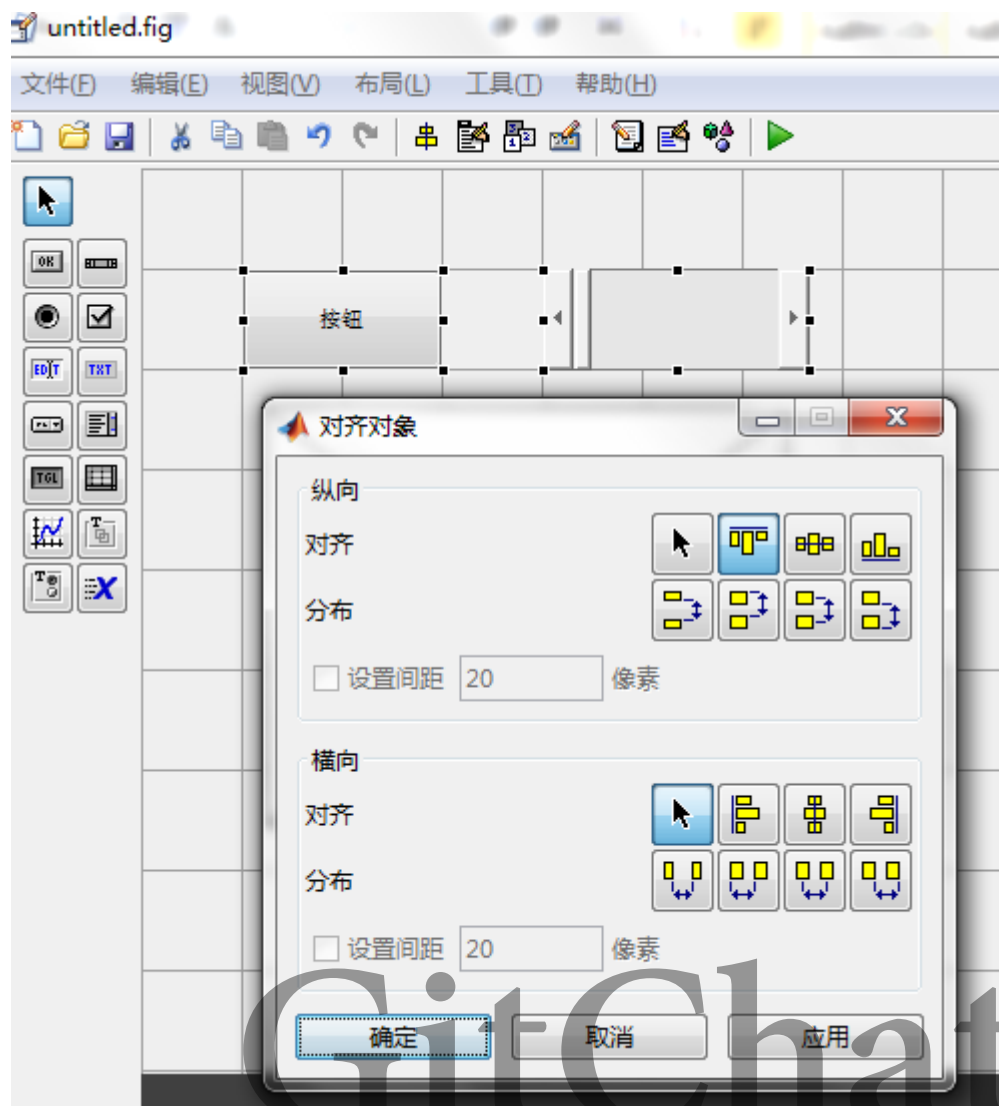
同时，Matlab提供了一套可视化的创建图形用户接口（GUI）的工具，如下图所示：





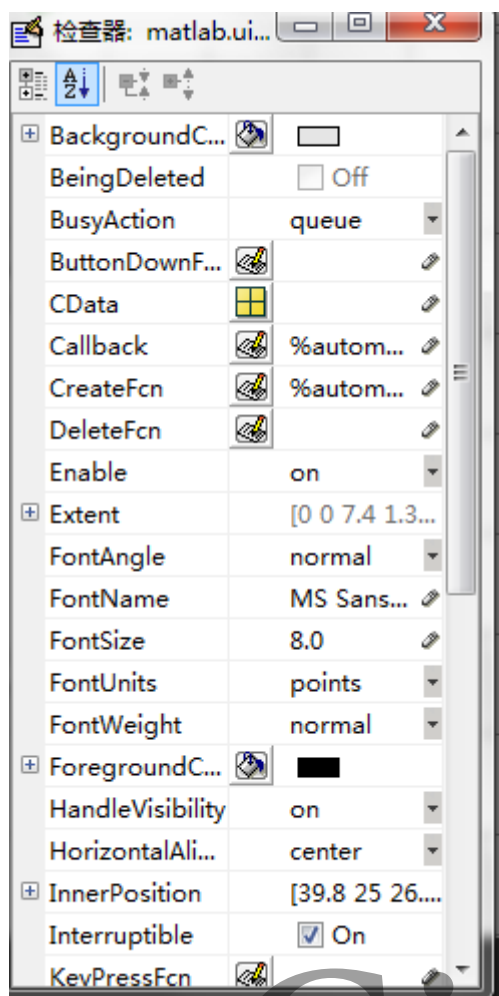
布局编辑器用于从控件选择板上选择控件对象并拖拽到布局区合适的位置去。在命令窗口输入GUIDE命令即可打开空白的布局编辑器图形用户界面。

2) 位置调整工具(Alignment tool)如下：



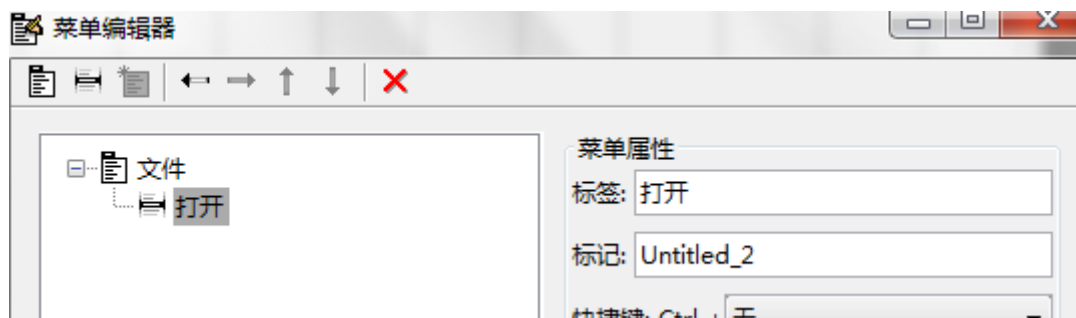
利用该工具，可对GUI对象设计区的多个对象的位置对齐调整。

3) 属性查看器



利用对象属性查看器，可以查看每个对象的属性值，也可以修改、设置对象的属性值。

4) 菜单编辑器(Menu Editor)如下



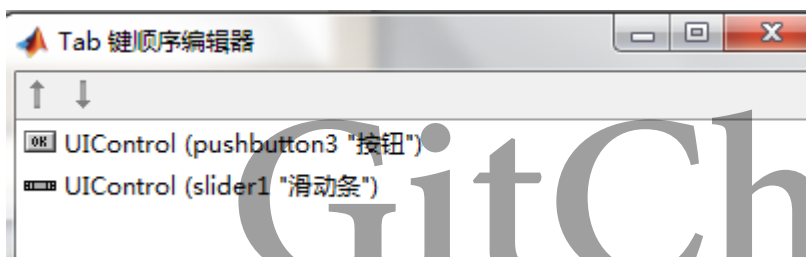
菜单编辑器包括菜单的设计和编辑，菜单编辑器有八个快捷键，可以利用它们任意添加或删除菜单，可以设置菜单项的属性，包括名称(Label)、标识(Tag)、选择是否显示分隔线(Separator above this item)、是否在菜单前加上选中标记(Item is checked)、调用函数(Callback)。利用菜单编辑器，可以创建、设置、修改下拉式菜单和快捷菜单。

5) 对象浏览器(Object Browsers)



利用对象浏览器，可以查看当前设计阶段的各个句柄图形对象。可以在对象浏览器中选一个或多个控件来打开该控件的属性编辑器。

6) Tab顺序编辑器 (Tab Order Editor)



3.2 控件对象及属性

Matlab中的控件大致可分为两种，一种为动作控件，鼠标点击这些控件时会产生相应的响应。一种为静态控件，是一种不产生响应的控件，如文本框等。常用的控件属性及其描述如下：

- BackgroundColor：取值为颜色的预定义字符或RGB数值；缺省值为浅灰色；
- Callback：取值为字符串，可以包含M文件名称、小段MATLAB语句、当前窗口的

- Value：取值可以是矢量，也可以是数值，其含义及解释依赖于控件对象的类型；
- FontAngle：取值为normal（正体，缺省值），italic（斜体），oblique（方头）；
- FontName：取值为控件标题等字体的字库名；
- FontSize：取值为数值；
- FontUnits：取值为points（缺省值），normalized, inches, centimeters或pixels；
- FontWeight：取值为normal（缺省值），light，demi和bold，定义字符的粗细；
- HorizontalAligment：取值为left，center（缺省值）或right，定义控件对象标题等的对齐方式。

3.3 GUI编程思路介绍

1）（**控件属性设置查看**）往布局编辑器添加一个控件，并双击，会出现一个界面。界面左侧是该控件的属性，包括“BackgroundColr”、“BeingDeleted”、“BusyAction”等，界面右侧是与每个属性相对应的属性值。

2）（**控件名字**）每个对象都有一个“名字”，即Tag属性，之所以称之为名字，是因为我们在使用这个对象的时候是通过这个Tag来找到该对象的。即通过一个非常重要的get()函数来实现。

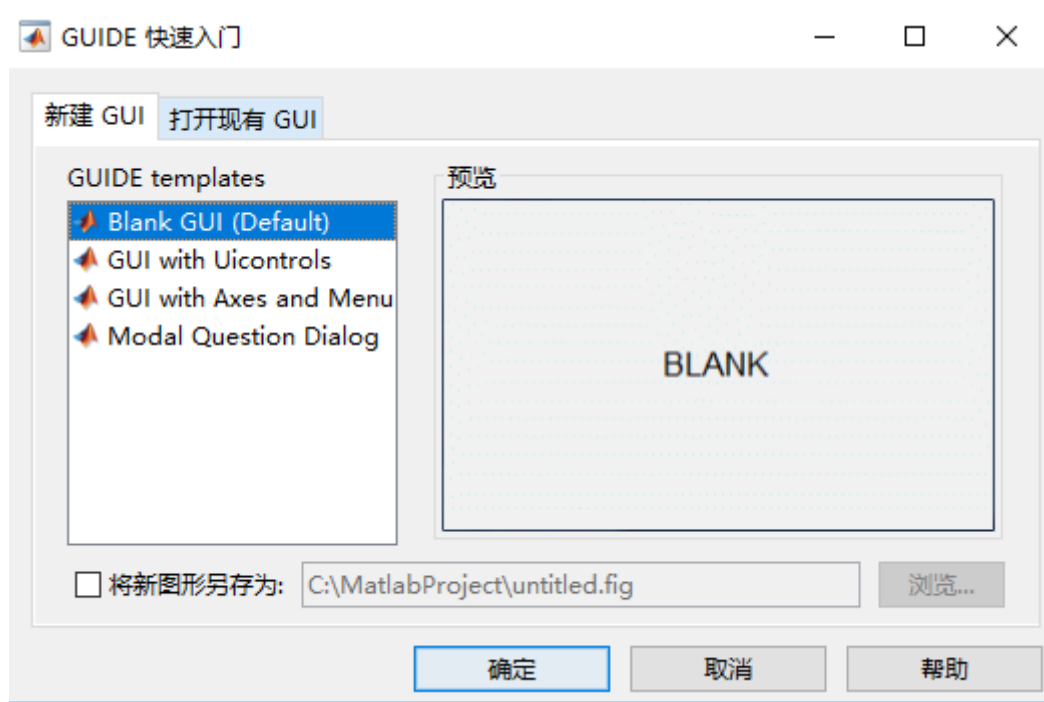
3）（**get(), set()函数**）通过对象的名字（Tag属性）找到该对象以后，对该对象的操作是通过set()函数来实现。

4）（**hObject, handles**）用户按下某个按键，必然是想触发某个响应，即调用callbackfunction（回调函数）来执行这个响应。在这个过程中，数据通过hObject和handles.对象名来传递。

3.4实例演示（视频讲解）

1）两个数相加

命令行窗口输入guide，弹出如下窗口：



选择第一个空白GUI，点“确定”按钮。弹出如下未命名GUI界面：



点击“文件”——“另存为”，保存当前GUI界面，命名为GUI_test：

拖拽三个“可编辑文本”控件，两个“静态文本”控件，一个“按钮”控件，并在适当位置摆放好（可使用对齐工具进行控件间对齐），效果如下：

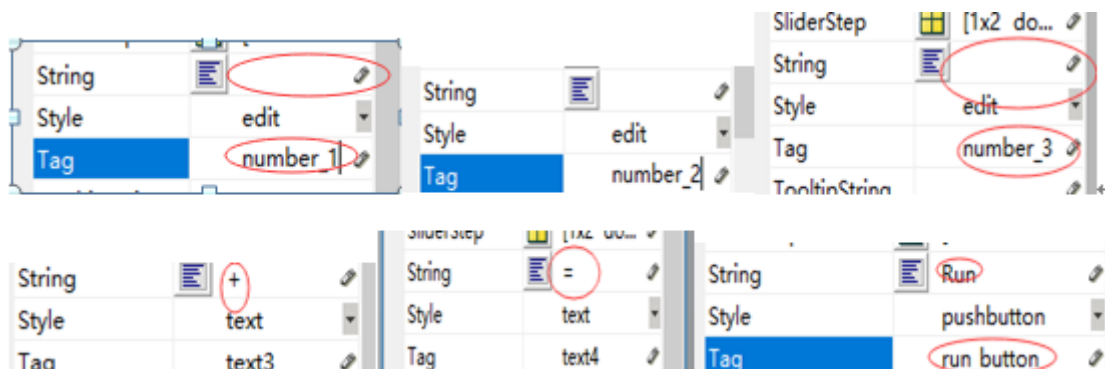


双击每个控件，对其对象属性（Tag）进行设置，属性分别设置为三个“可编辑文本”控件：string属性的属性值均为“空”；tag属性的属性值分别为number_1, number_2, number_3（对tag属性进行设置，相当于给每个控件起个不同的名字，便于后面对每个控件进行操作）。

两个“静态文本”控件：string属性的属性值分别为“+”和“=”；tag属性的属性值默认即可（因为此例中我们不需要对这两个控件的属性进行操作）。

一个“按钮”空间：string属性的属性值为“Run”；tag属性的属性值为“run_button”。

设置后的效果如下图：

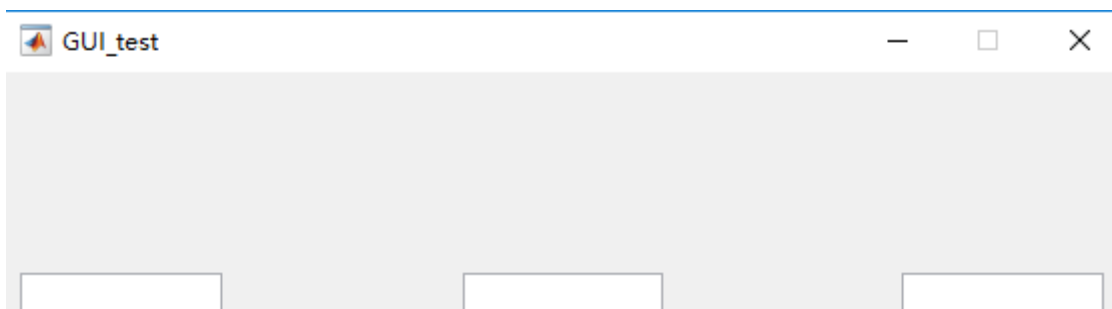


下面进行代码编辑。注意get() set() hObject, handles的使用，充分理解数据如何获得和在空间之间进行传递。代码如下：

```
data_1_str = get(handles.number_1, 'string');  
data_1_num = str2num(data_1_str);  
data_2_string = get(handles.number_2, 'string');  
  
data_2_num = str2num(data_2_string);  
sum_num = data_1_num + data_2_num;  
  
sum_str = num2str(sum_num);  
set(handles.number_3, 'string', sum_str);
```

```
function run_button_Callback(hObject, eventdata, handles)  
data_1_str = get(handles.number_1, 'string');  
data_1_num = str2num(data_1_str);  
  
data_2_string = get(handles.number_2, 'string');  
data_2_num = str2num(data_2_string);  
  
sum_num = data_1_num + data_2_num;  
sum_str = num2str(sum_num);  
  
set(handles.number_3, 'string', sum_str);
```

效果图如下：



- 1) 对话框
- 2) 串口
- 3) 文件和IO
- 4) 推荐书籍 《MATLAB GUI 设计学习手记 第2版 罗华飞》

4. 相关知识介绍

Matlab应用领域非常广泛，包括数值分析、数值和符号计算、工程与科学绘图、控制系统的设计与仿真、数字图像处理技术、数字信号处理技术、MATLAB在通讯系统设计与仿真的应用、MATLAB在通讯系统设计与仿真的应用、通讯系统设计与仿真、财务与金融工程、管理与调度优化计算（运筹学）等各个领域，我们不可能对每个领域的都很擅长，因此结合自身研究方向，选择合适的参考资料进行学习。

这里给出一个比较好的学习网站（不仅限于Matlab）：www.youtube.com，Google和百度里也有非常多的学习资料。

学习建议：针对Matlab GUI，理解所列举的核心思想；针对自己的研究课题，以任务为导向，有针对性学习某方向的知识；借助互联网，搜索相关资料。

GitChat