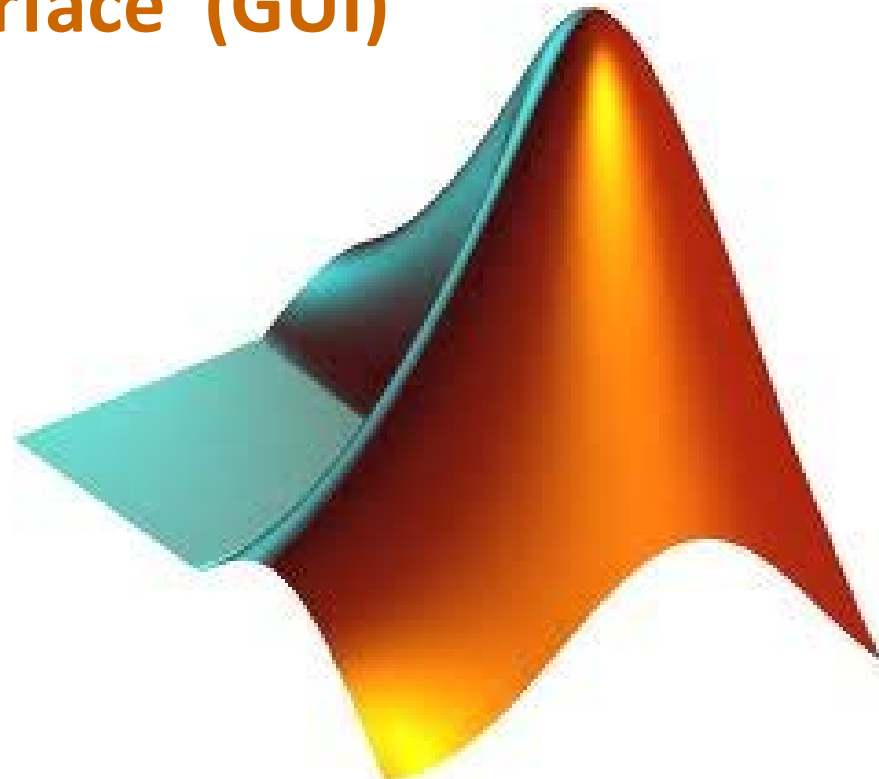


Matlab

Kap. IV

- **Modifying Displays**
- **Graphical User Interface (GUI)**



Manipulating Plots/Images,...

- Handles are used to manipulate plots (all objects)

```
im = imread('pout.tif');
```

```
h = imshow(im);
```

```
get(h)
```

% Shows all properties of object with handle h

```
xrange = get(h, 'Xdata')
```

% Gets the data range w/r x-axis

```
yrange = get(h, 'Ydata')
```

% Gets the data range w/r y-axis

```
set(h, 'Xdata', [1,200])
```

% Sets the x range from Xmin=1 to Xmax=200

```
figure
```

```
H = plot( sin([0:0.1:2*pi]) )
```

```
get(H)
```

% Shows all properties of plot h

```
set(H, 'color', [1,0,0])
```

% Sets the color of the line to red

```
set(H,'Marker','x', 'MarkerSize',30)
```

% Sets a cross of width 30 at each data point

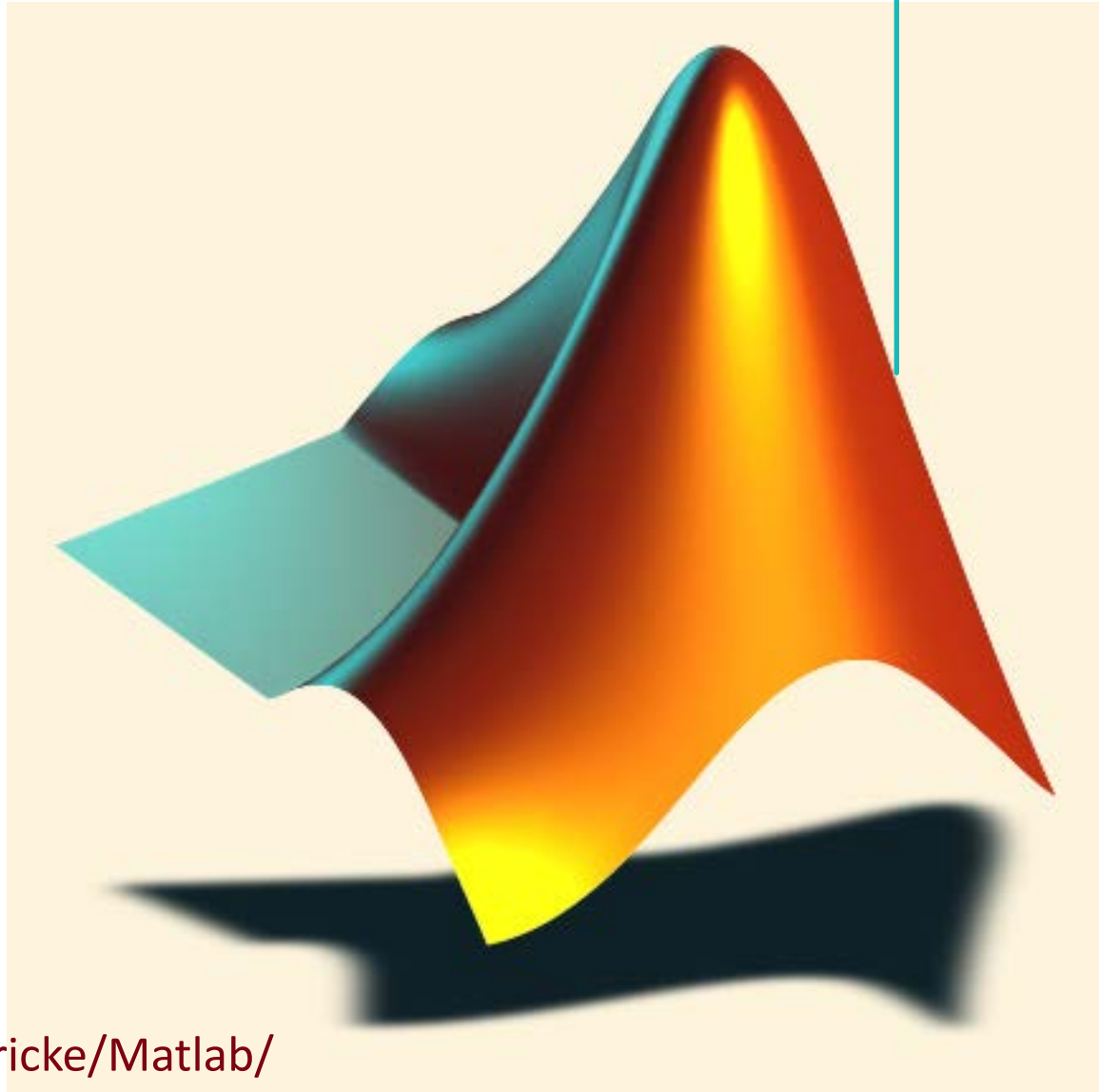
- Use **inspect(handle)** to view and manipulate (most) properties

Building GUI's with MATLAB

GUI's

An example can be found in
our download page:

<http://www.pci.tu-bs.de/aggericke/Matlab/>



Create a figure

In Matlab a window is called a figure. Let's create one with some userdefined properties.

Syntax: **figure('PropertyName',propertyvalue,...)**

```
hFig = figure( 'Units','pixels',... % Other units are inches, normalized,...  
              'MenuBar','none',...  
              'Toolbar','none',...  
              'Position',[100,300,500,400],... % [lower left x, lower left y, width, height]  
              'Visible','on');
```

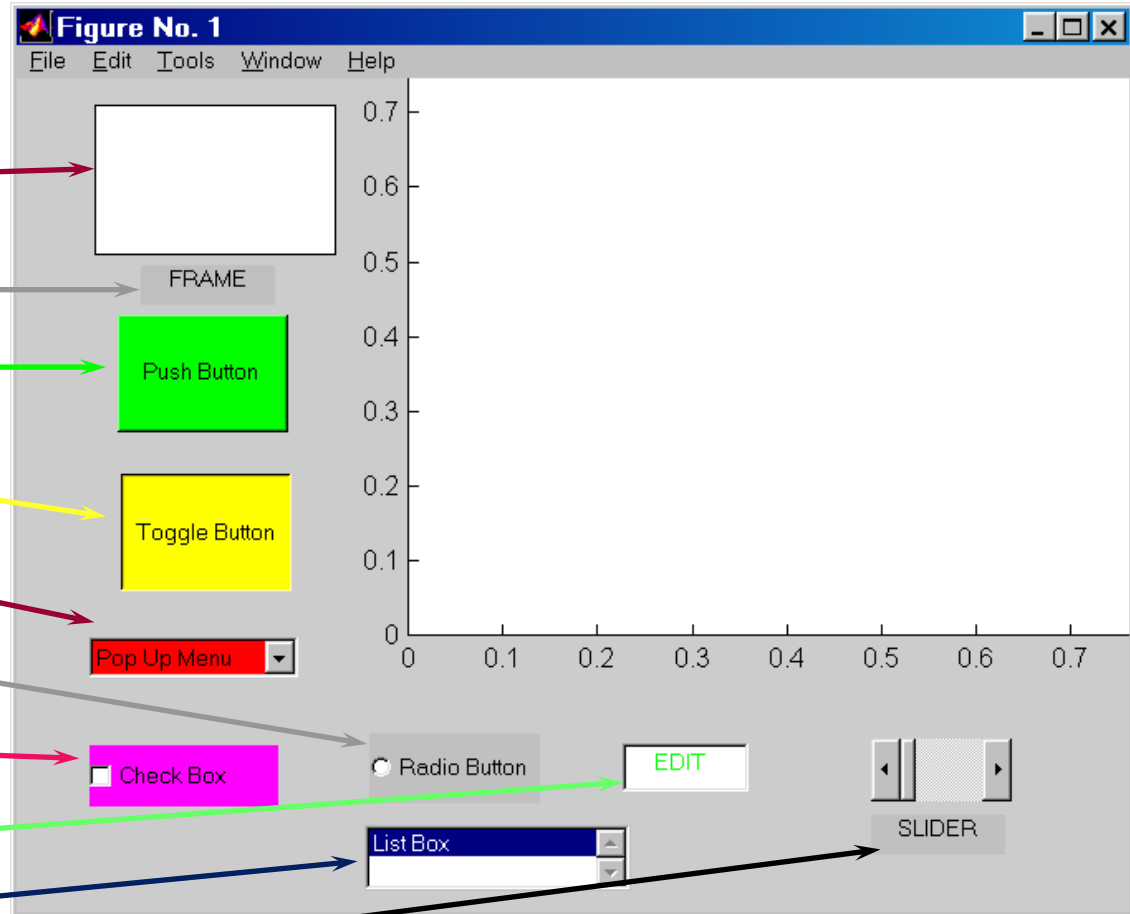
Use **inspect(hFig)** or **get(hFig)** to see most properties.

With **set(hFig, 'PropertyName',propertyvalue,...)** you can set the properties:

```
set(hFig, 'Name', 'Check and radio'); % Setting figure name
```

The 10 styles of Matlab Uicontrol objects

- Frame.
- Static Text
- Push Button.
- Toggle Button.
- Pop-up Menu.
- Radio Button.
- Check Box.
- Editable Text.
- List Box.
- Slider.



The syntax of all Uicontrols is
handle = uicontrol(parent,'Name',Value,...)

handle = uicontrol(parent,'Name',Value,...)

,Name' ,Value'

,Style' 'frame'

Frame.

'text'

Static Text

'pushbutton'

Push Button.

'togglebutton'

Toggle Button.

'popupmenu'

Pop-up Menu.

'radiobutton'

Radio Button.

'checkbox'

Check Box.

'edit'

Editable Text.

'listbox'

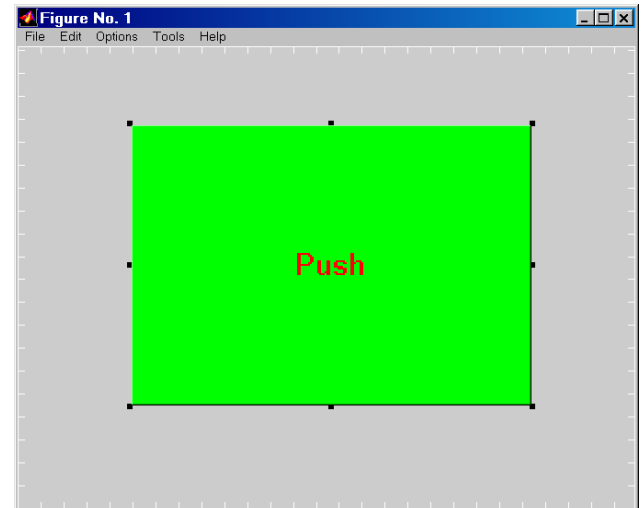
List Box.

'slider'

Slider.

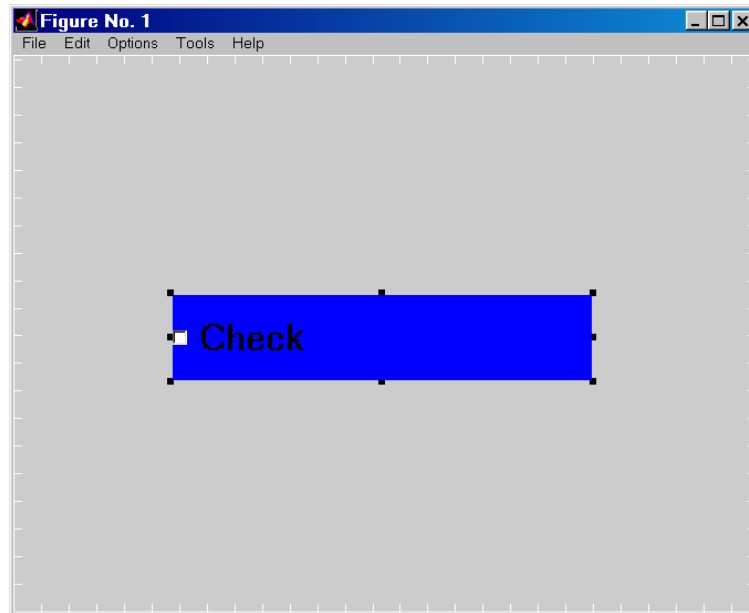
Push/Toggle Buttons

- The push button is widely prevalent uicontrol style that is used primarily to indicate that a desired action should immediately take place.
- The toggle button looks just like a push button, except there is no intermediate state. Rather, the button will remain in its selected or not selected state after the user clicks on it.



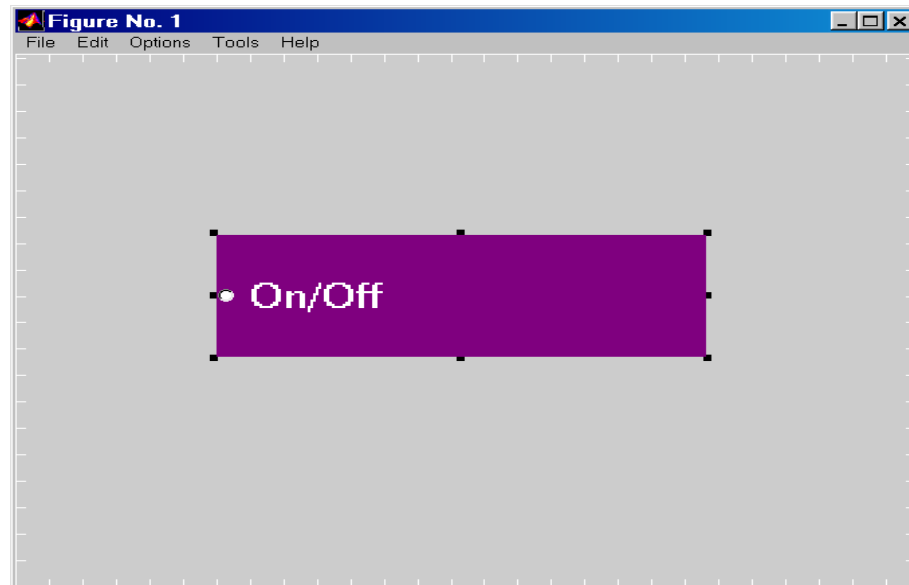
Check Box

- Useful for representing two states of an option that you may want to provide (usually as on and off).
- In its 'off' state the check box will consist of an empty or unfilled square. In the 'on' state, the check box's square will contain a 'V' sign.



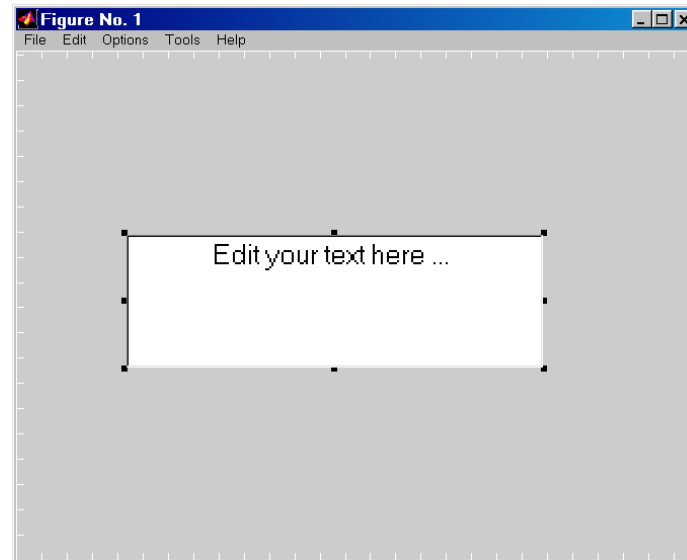
Radio Button

- Similar to the check box in that there are two states associated with each other.
- Usually two or more radio buttons are linked together as a group. They are linked in the sense that only one of the buttons will be in its selected state.



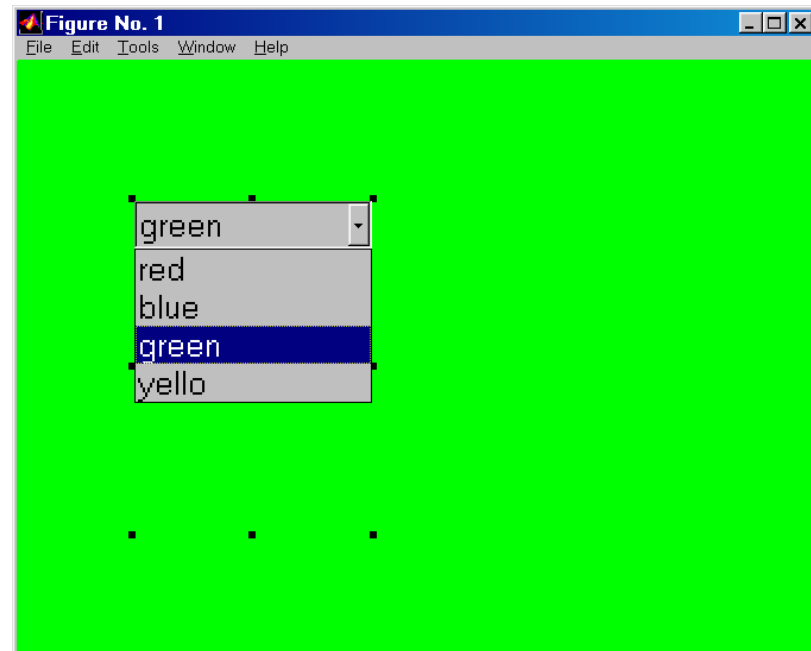
Editable Text

- Used in situations that require the user to enter strings or characters or numbers. The strings, in turn, are used by the application for which the interface has been built.
- Clicking anywhere within this object will change the mouse from a pointer to a text insertion indicator.



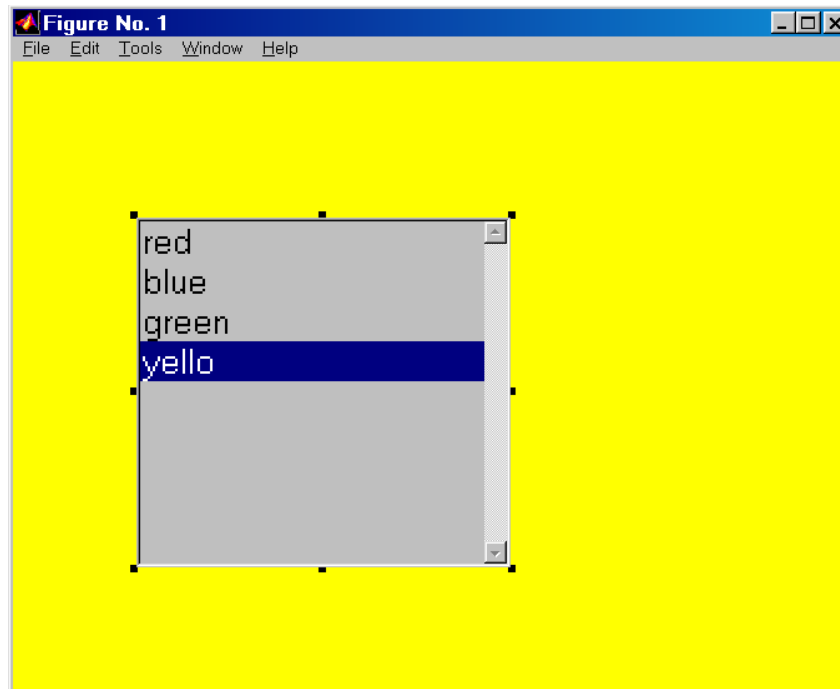
Pop-up Menus

- Used in situations where multiple choices need to be available to the user.
- When the user clicks and holds the mouse button anywhere within the object, a list of choices appear.



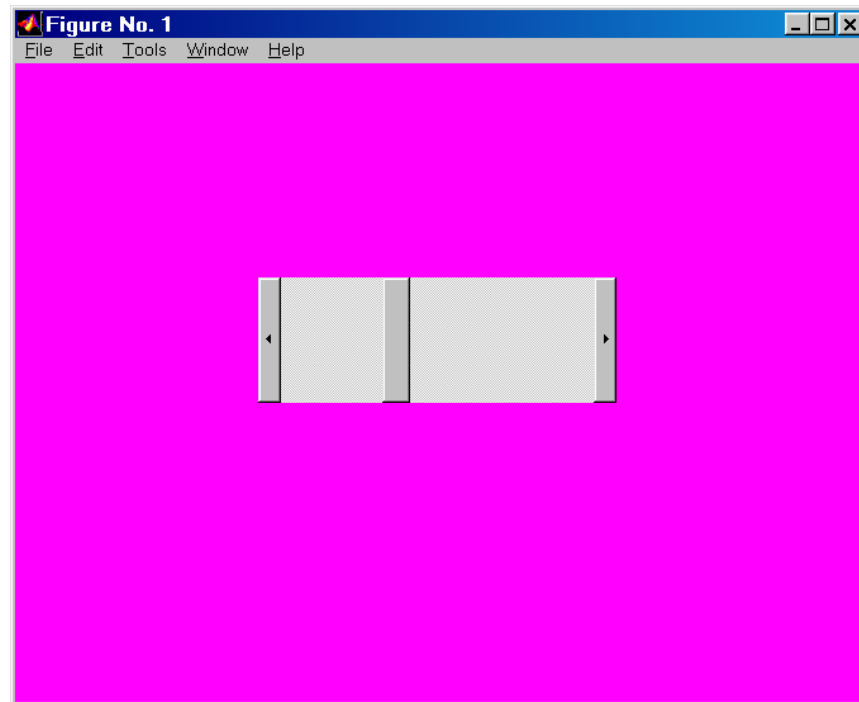
List Boxes

- Very similar to pop-up menus.
- The main difference with a list box is that you can make the set of options visible to the user at all times.



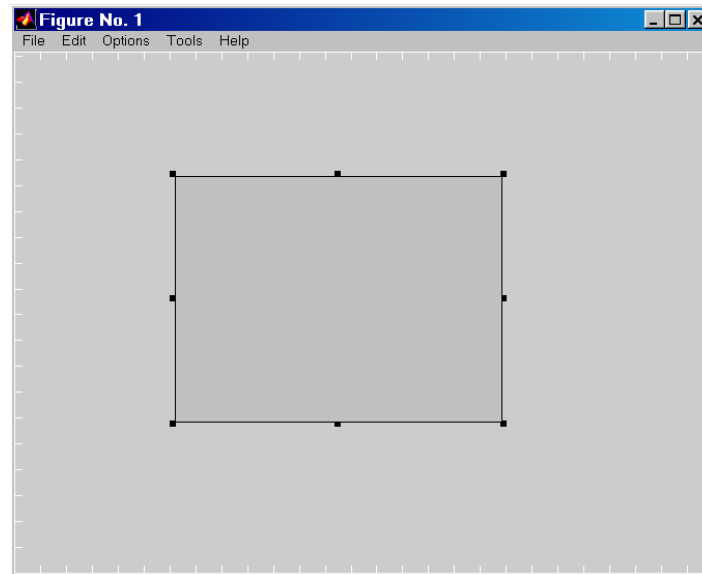
Sliders

- Useful in representing a fixed range of values from which to choose.
- The slider has no way of explicitly indicating the numeric value that the slider represents. Therefore, it is recommended that an editable text or static text style uicontrol accompanies the slider.



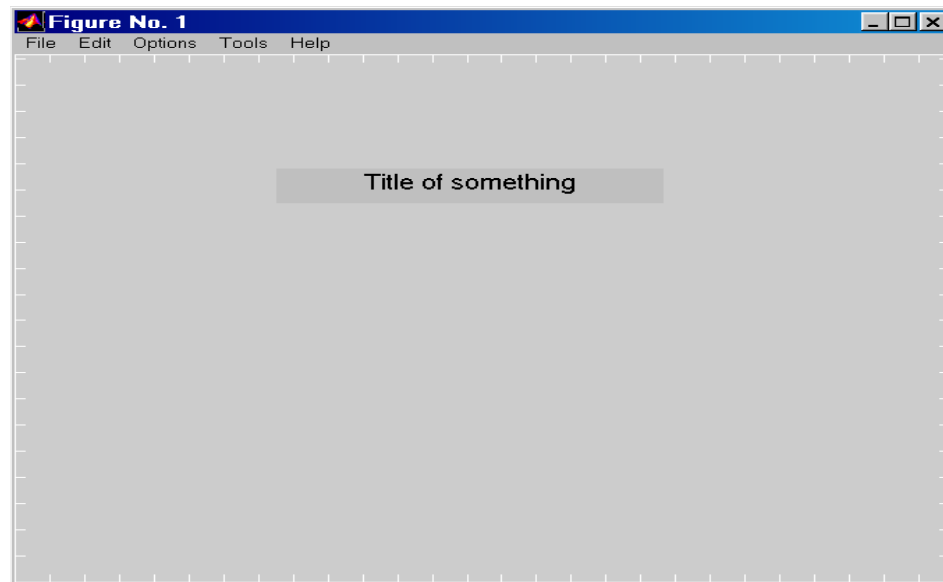
Frames

- Provide a solid background that helps blend a set of uicontrols into one complete and cohesive interface.
- Used as an effective method of organizing the GUI in a logical and intuitive fashion.



Static Text

- Available for creating labels, status messages or other information pertinent to the user.
- Static text does not perform any action if the user clicks on any part of the object. In addition , the user can not edit the information that is displayed.



What is a Callback?

- **A callback is a function which is executed by activating a graphics object:**
e.g. CreateFcn, CallBack, ButtonDownFcn, DeleteFcn....

Callbacks that Track User Actions

- **ButtonDownFcn**
 - When clicking the mouse button while the pointer is over the object.
- **WindowButtonDownFcn**
 - When clicking the mouse button down within the figure boundaries.
- **WindowButtonUpFcn**
 - When clicking the mouse button up within the figure boundaries.
- **WindowButtonMotionFcn**
 - When the mouse pointer moves within the figure boundaries.
- **WindowKeyPressFcn**
 - Last key pressed, when the figure is active.
- **WindowKeyReleaseFcn**
 - key is released.
- **WindowScrollWheelFcn**
 - The scroll wheel is turned
- **CreateFcn**
 - When creating an object.
- **CloseRequestFcn**
 - When closing a figure.
- **DeleteFcn**
 - When deleting an object.
- **ResizeFcn**
 - When resizing the figure.

Some useful instructions.

- **gcf** % Get current figure
- **gca** % Get current axes
- **gco** % Get current object
- **figure(*figure_handle*)** % Make the figure current (or create one).
- **axes(*axes_handle*)** % Make the axes current (or create one).
- **Object_Handle = gcbo** % Get the object that makes the callback.
- **findobj('PropertyName',PropertyValue,...)**
- **findobj(Object_Handle,...)** % Finds objects with specified properties
- **findall(...)** % Like findobj, but also finds hidden obj.

Setting Push Button

```
hPush = uicontrol(hFig, 'Style', 'pushbutton');           % oder 'togglebutton'

set(hPush,...
    'Position', [10,15,90,22],...                         % Position in Pixeln
    'String', 'Drück mich',...                             % Text auf dem Button
    'FontWeight', 'bold',...                               % Fettdruck
    'TooltipString','Führt die Funktion Fct. aus',...      % Erscheint beim Verharren
                                                         mit der Maus auf Button
    'Callback','disp("Thank you for clicking me.")') % Ausführen der Fct. Beim
                                                         Klicken auf Button

set(hPush, 'ForegroundColor', [0,0,0]);                   % Textfarbe (schwarz)
set(hPush, 'BackgroundColor', [0.0,1.0,0.0]);             % Hintergrundfarbe (grün)
```

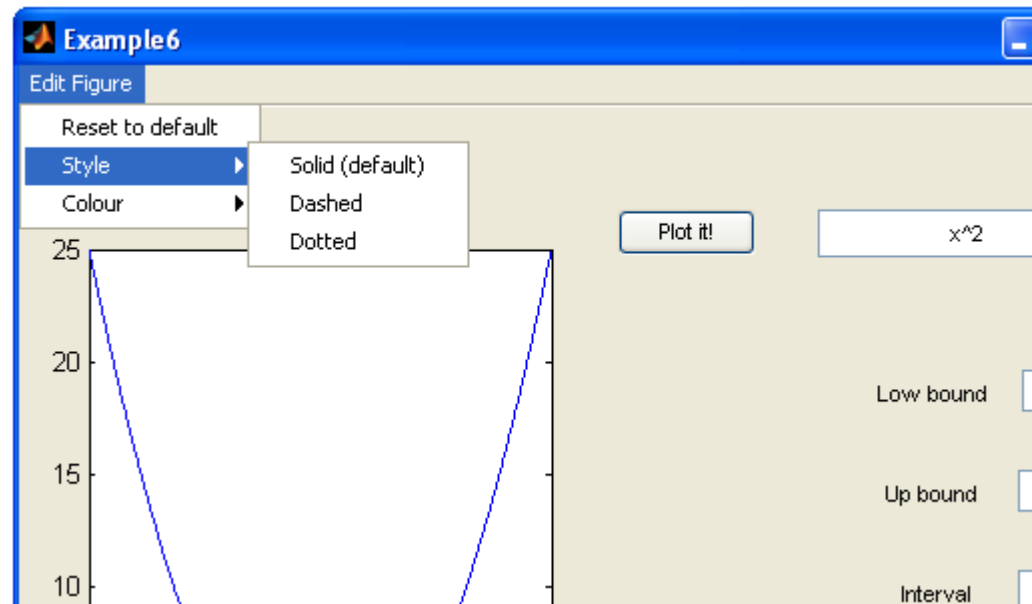
Work with file MeinErstesGUI.m

UI menus and Context menus

Additional menus that help enhance the GUI

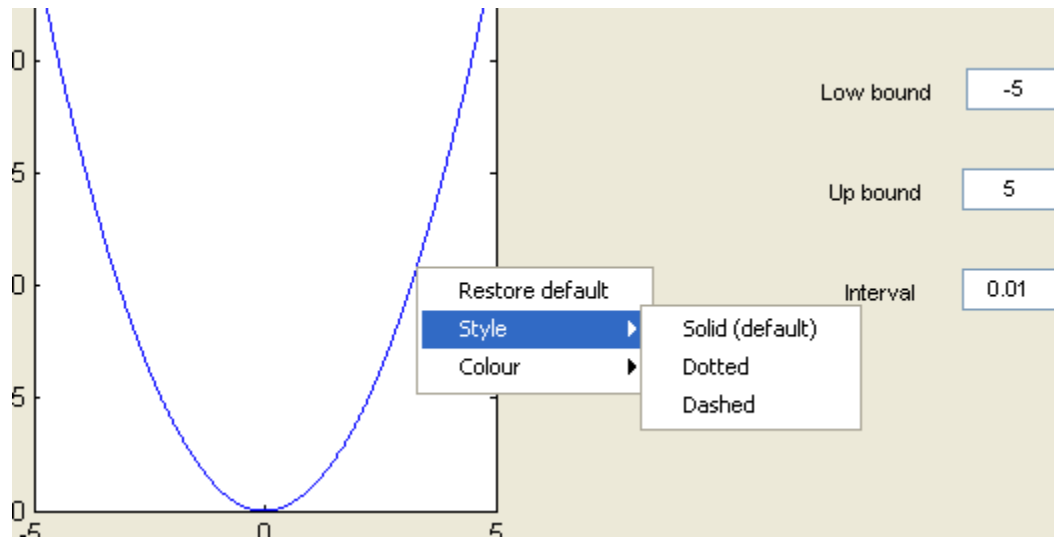
A **UI menu** creates a hierarchy of menus and submenus that are displayed in the figure window's menu bar.

Program the Callbacks as normal. The Callbacks will be executed when the user clicks on the option.



UI Context menu

A **UI context menu** appears when the user right-clicks a graphics object



Work with file MeinErstesGUI.m