

GUI Design
- GUIDE and Callbacks

Prof. Byoungjo CHOI

Embedded Systems Engineering Dept.

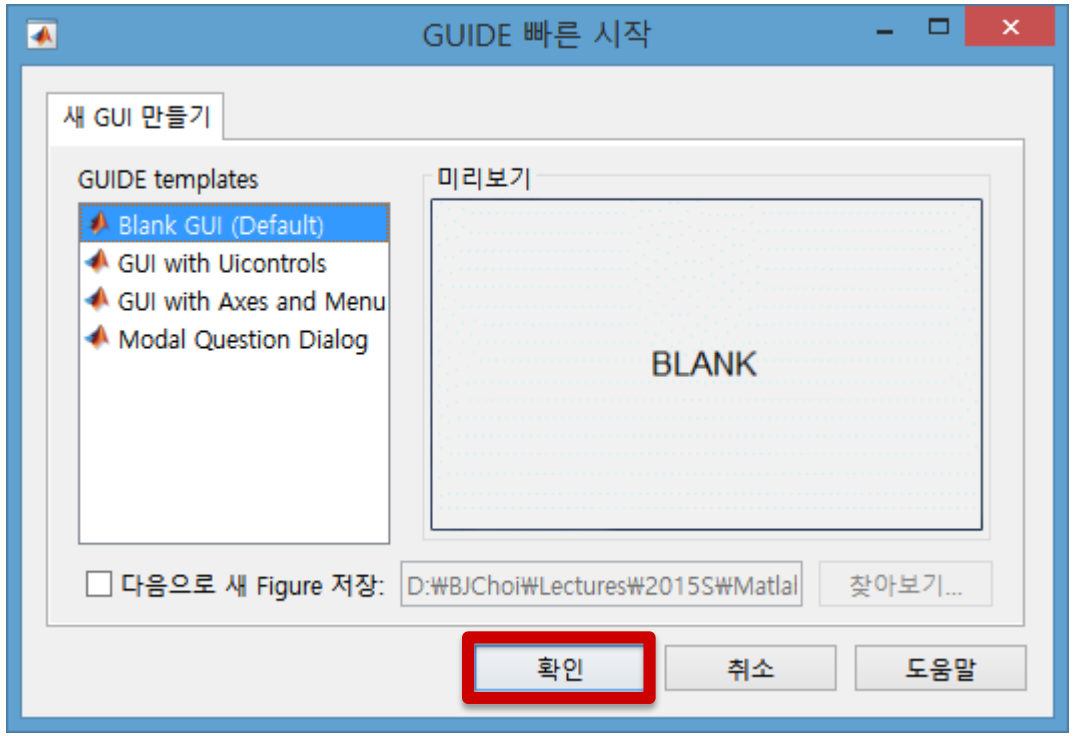
Incheon National University

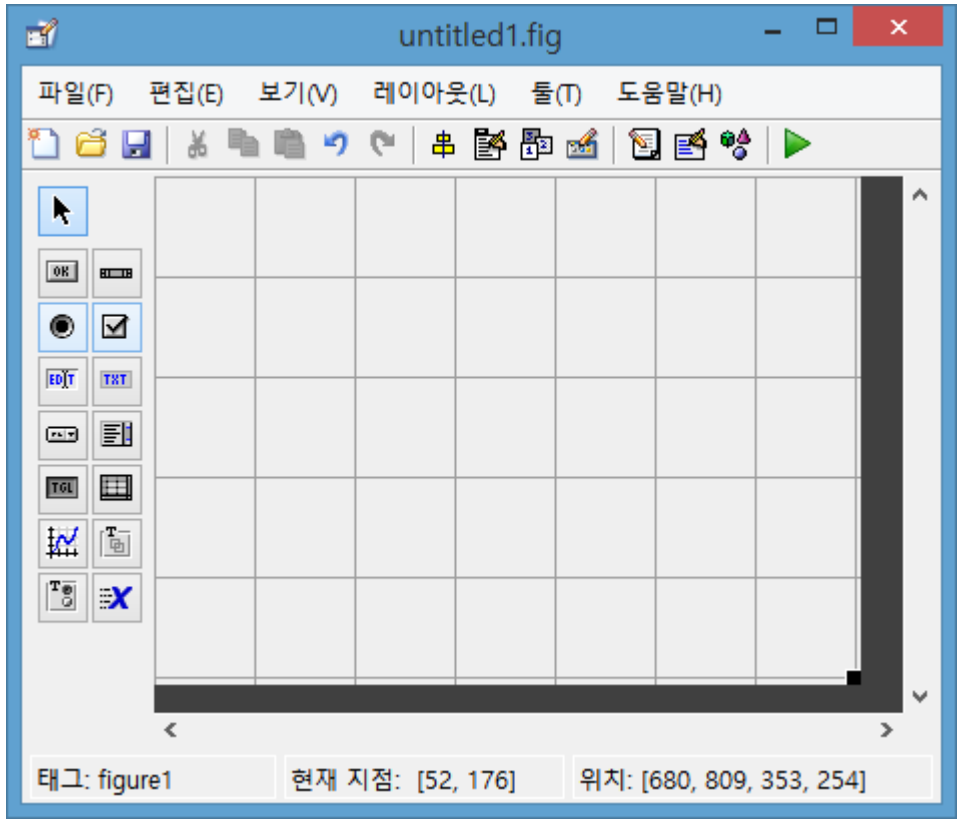
You will be able to

- Use GUIDE to design the layout of your GUI,
- Identify the properties of some GUI controls,
- Write callback functions for interactive GUI controls,
- Get / Set the properties of some GUI controls in your callback functions, and
- Manage your user-data for GUI.

GUIDE

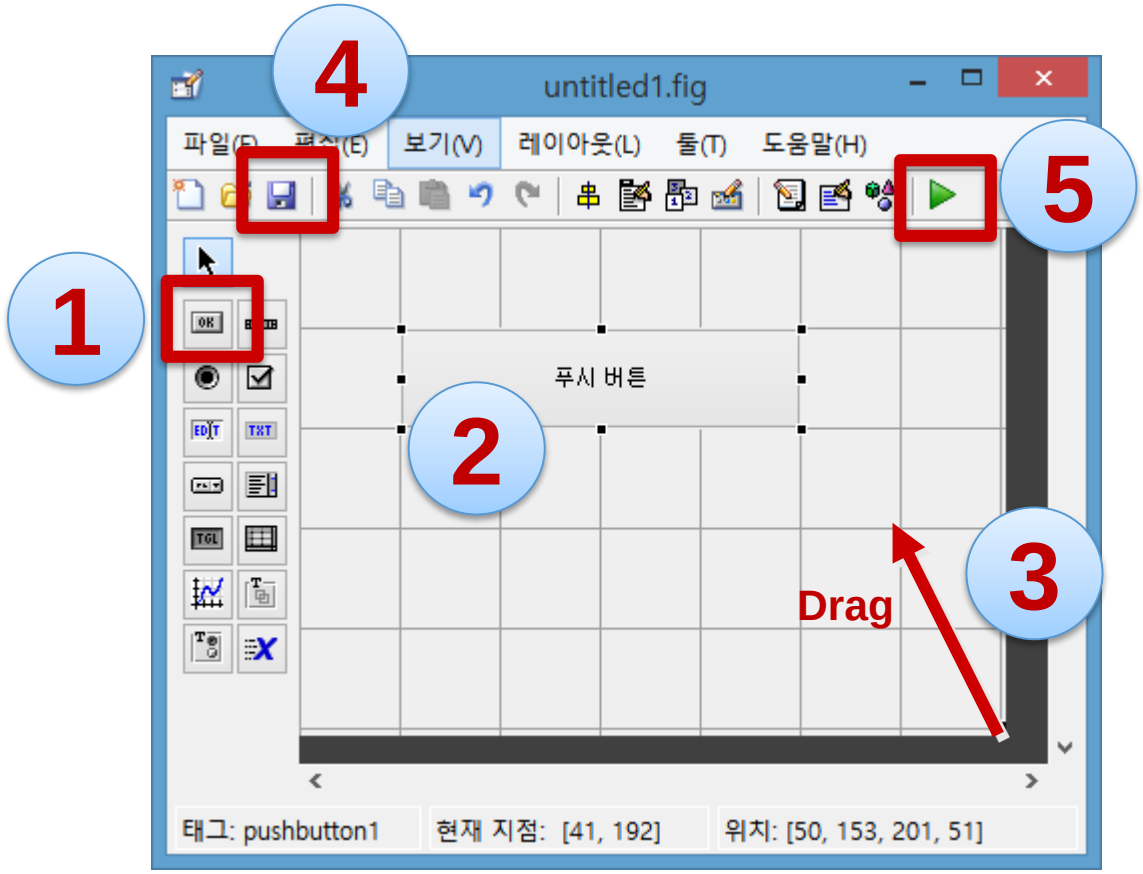
- GUIDE: GUI Design Environment
- `>> guide`

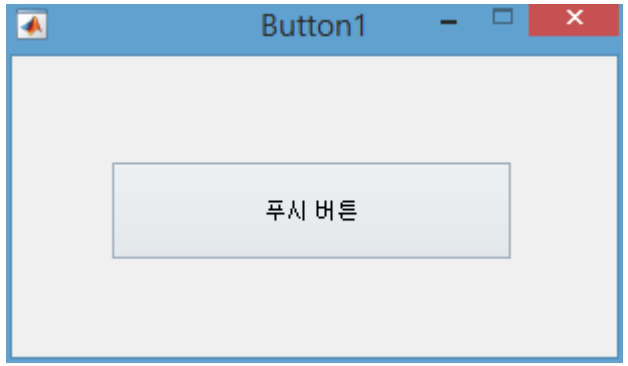




Push Button

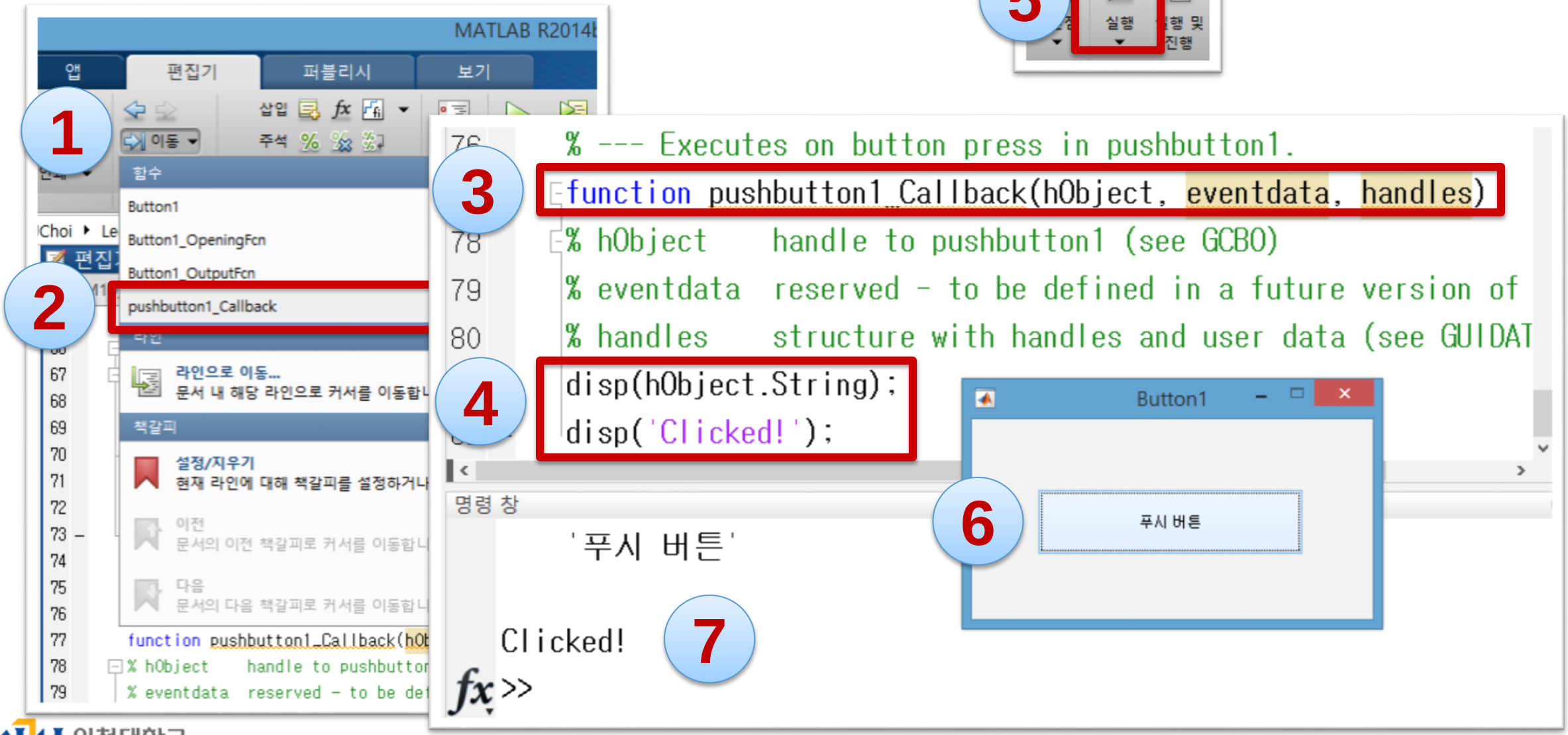
- A GUI with a Push Button





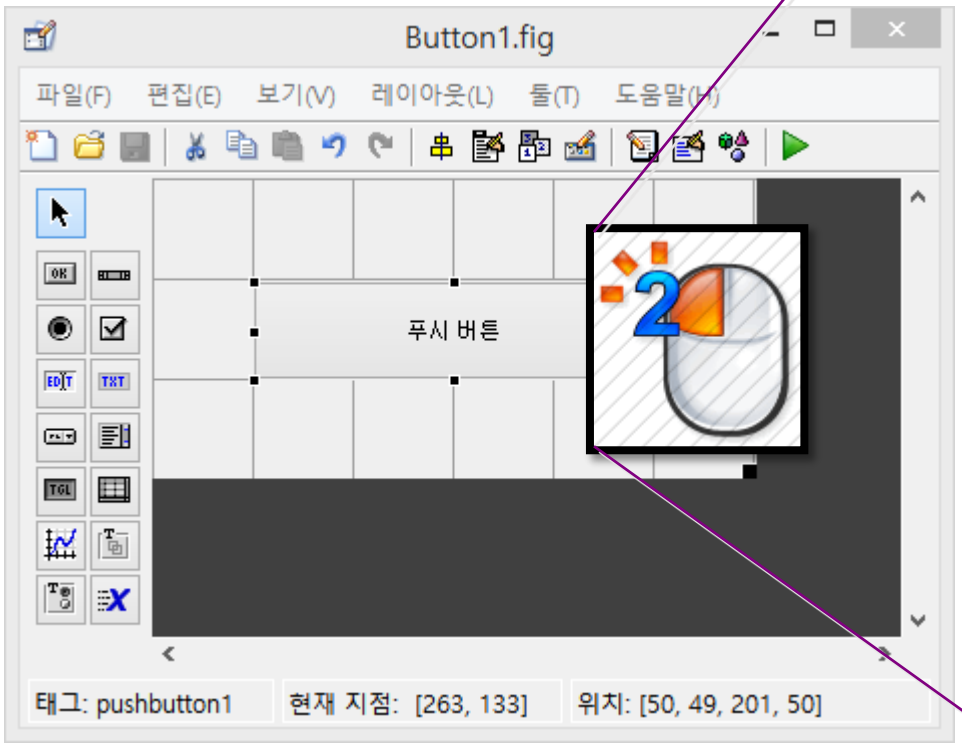
- 1. Select [Push Button].
- 2. Place it by Selecting a Region.
- 3. Resize the GUI Canvas.
- 4. Save as ‘Button1.fig’.
  - Button1.m is generated automatically.
- 5. Run ‘Button1’ GUI.
  - Click the button.

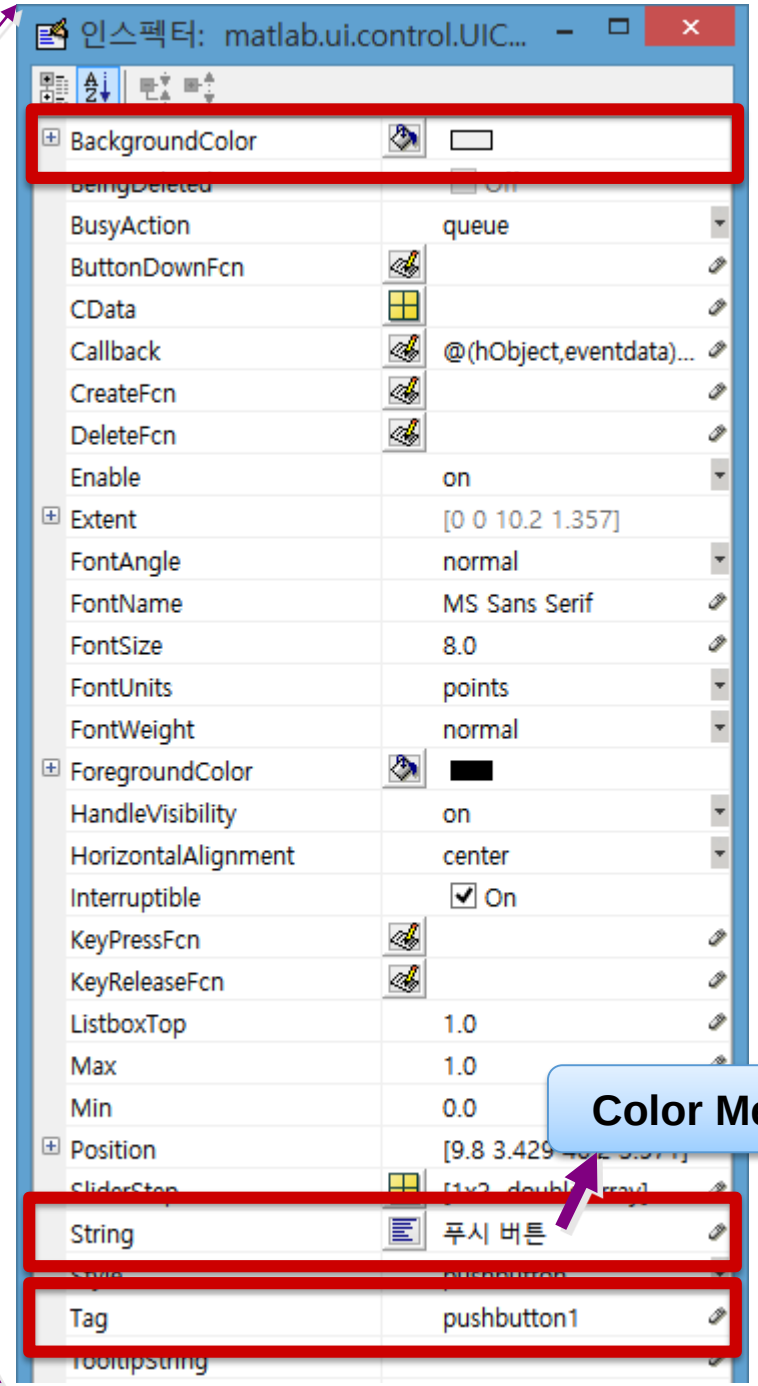
Push Button - Callback



Push Button - Properties

- Property Inspector
  - Double click the bush button at GUIDE.





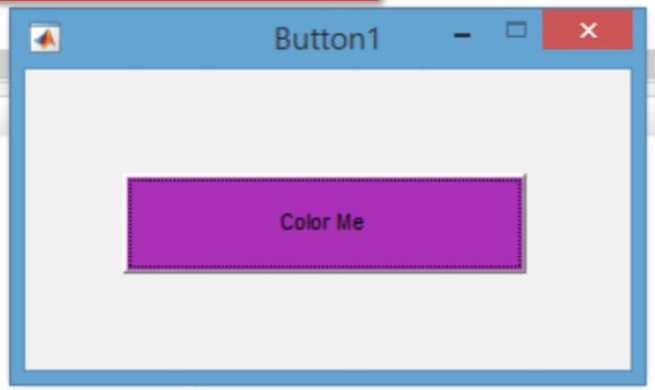
Push Button – Setting Properties

- Change properties at Callback function.

```
77    function pushbutton1_Callback(hObject, eventdata, handles)
78    % hObject    handle to pushbutton1 (see GCBO)
79    % eventdata  reserved - to be defined in a future version
80    % handles    structure with handles and user data (see GUI
81 —   disp('Clicked!');
82 —   c = rand(1,3);
83 —   set( hObject, 'BackgroundColor', c );
84
```

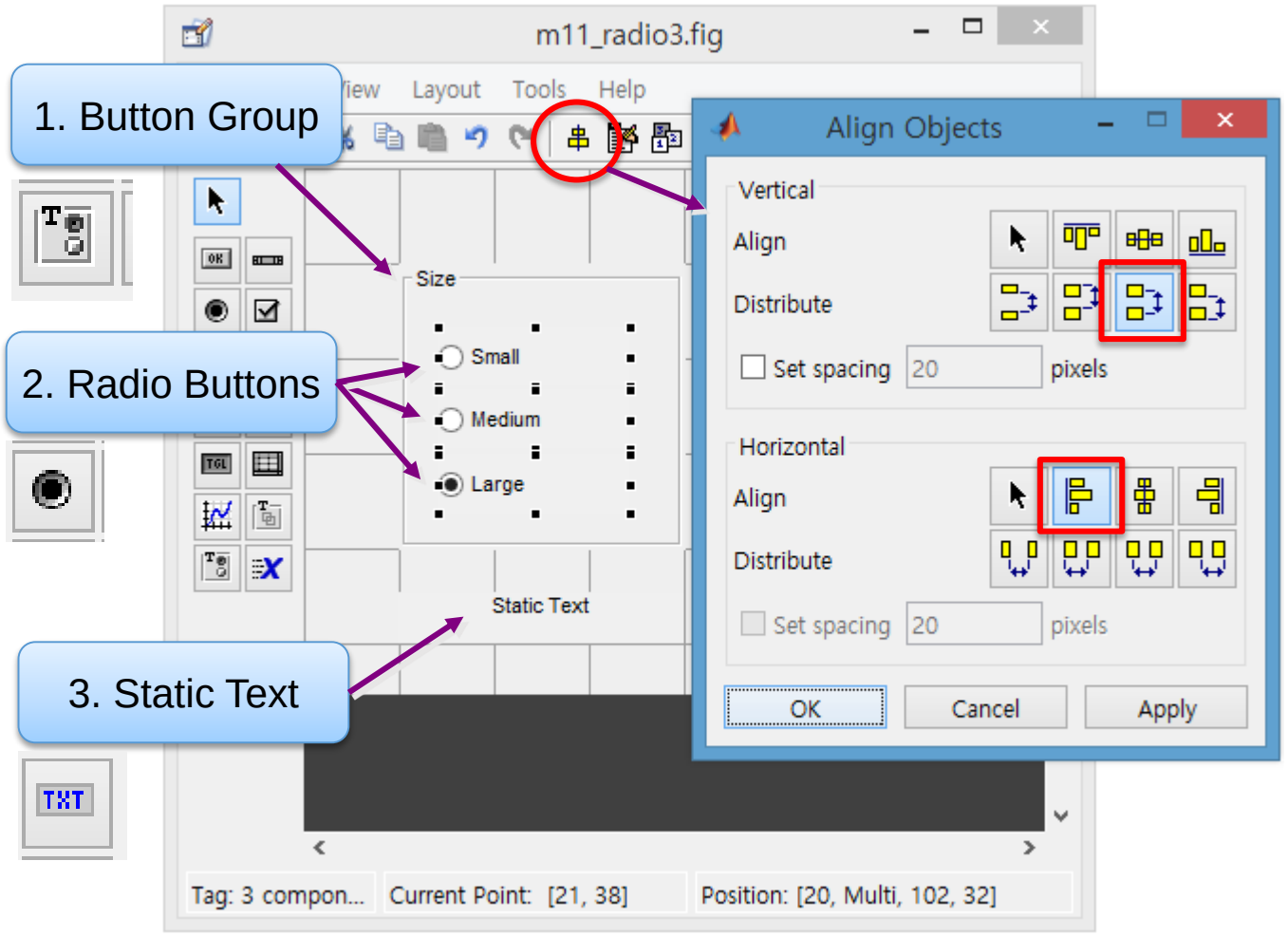
명령 창

```
Clicked!
Clicked!
Clicked!
Clicked!
fx >>
```



Radio Button Group

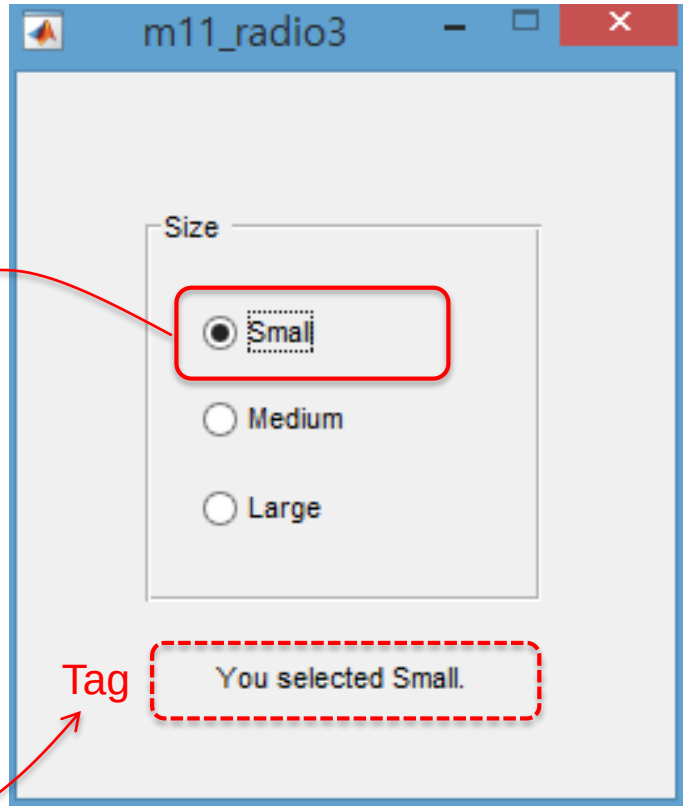
- Place a Button Group first.
- Then, a Radio Button on the Button Group.
- Use Ctrl+D to duplicate the radio buttons.
- Use Align Tool to place radio buttons evenly.



Radio Button Group - Callback

- Called when the selection changes.

```
75    % --- Executes when selected object is changed in uil
76    function uibuttongroup1_SelectionChangedFcn(hObject,
77    % hObject    handle to the selected object in uibutt
78    % eventdata  reserved - to be defined in a future ve
79    % handles    structure with handles and user data (s
80 —  disp(hObject.String);
81 —  set( handles.text2, 'String', ...
82        sprintf('You selected %s.', hObject.String ));
```



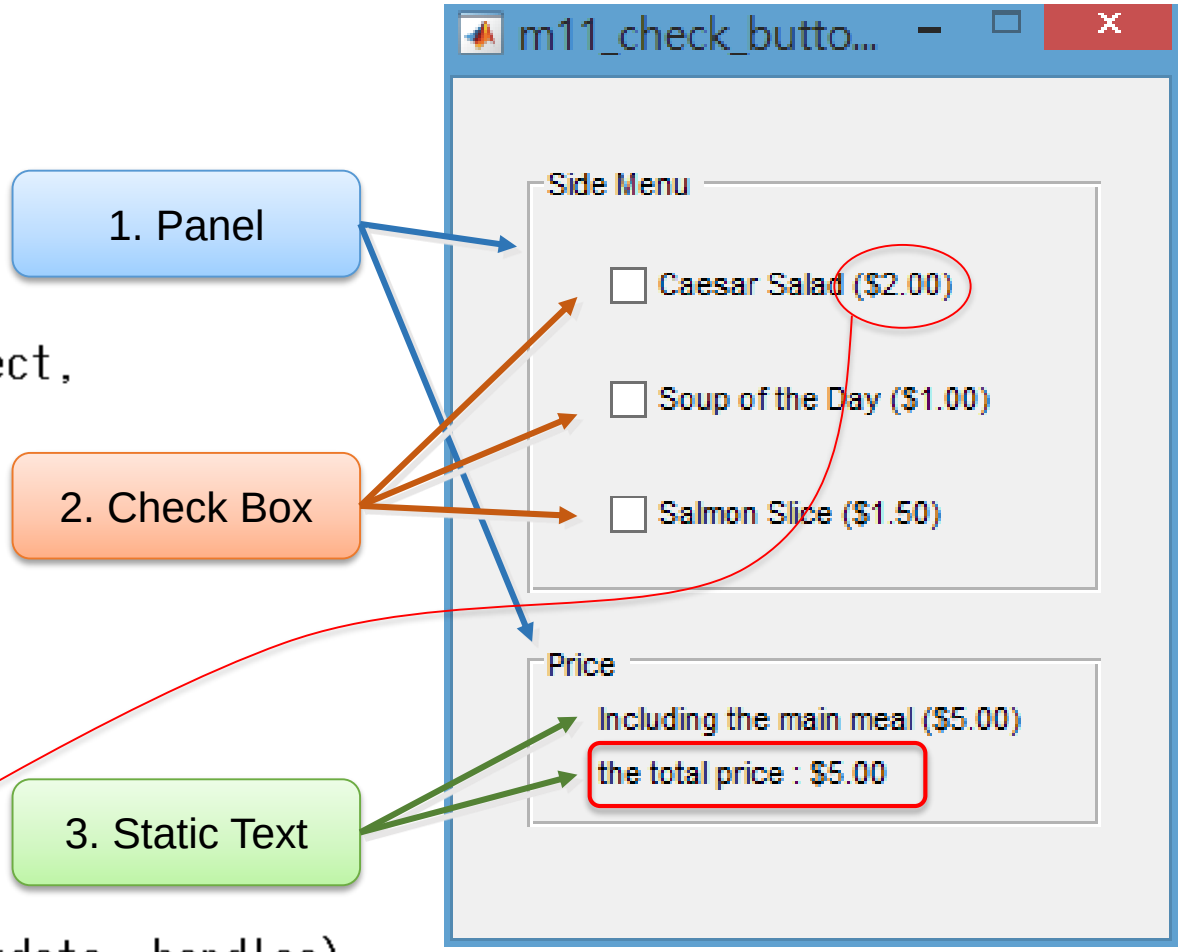
Check Boxes

- Initialize the price at the opening function.

```
function m11_check_buttons_OpeningFcn(hObject,
handles.output = hObject;
handles.price  = 5;
guidata(hObject, handles);
```

- Each check box callback update the price.

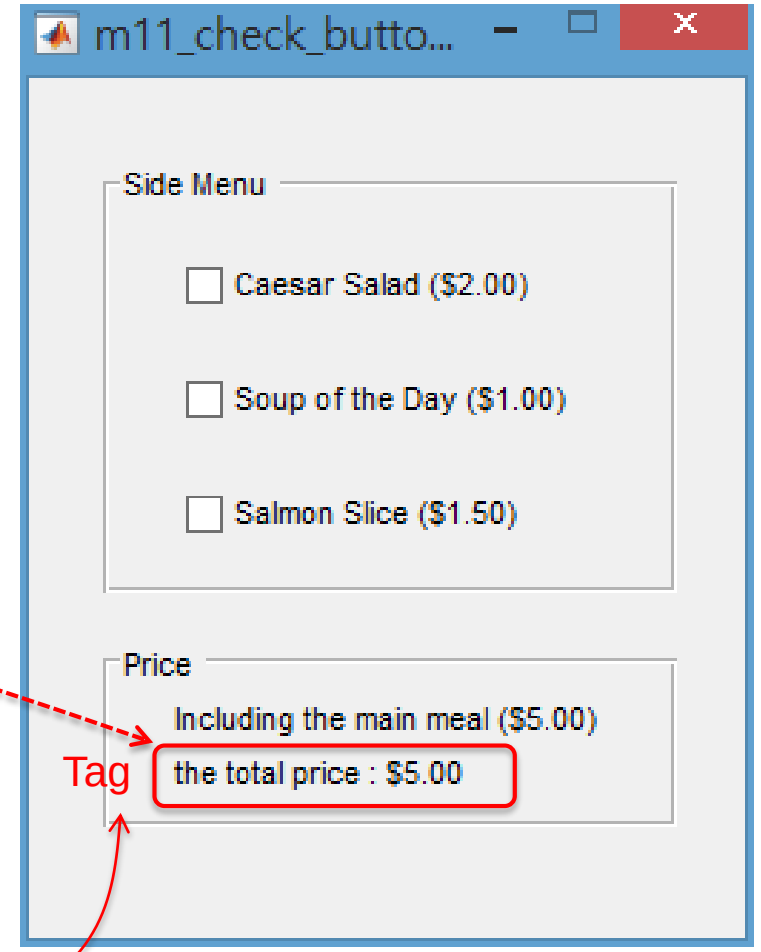
```
function checkbox1_Callback(hObject, eventdata, handles)
update_price(hObject, handles, 2);
```



Check Boxes – Local Function

- Update the price.

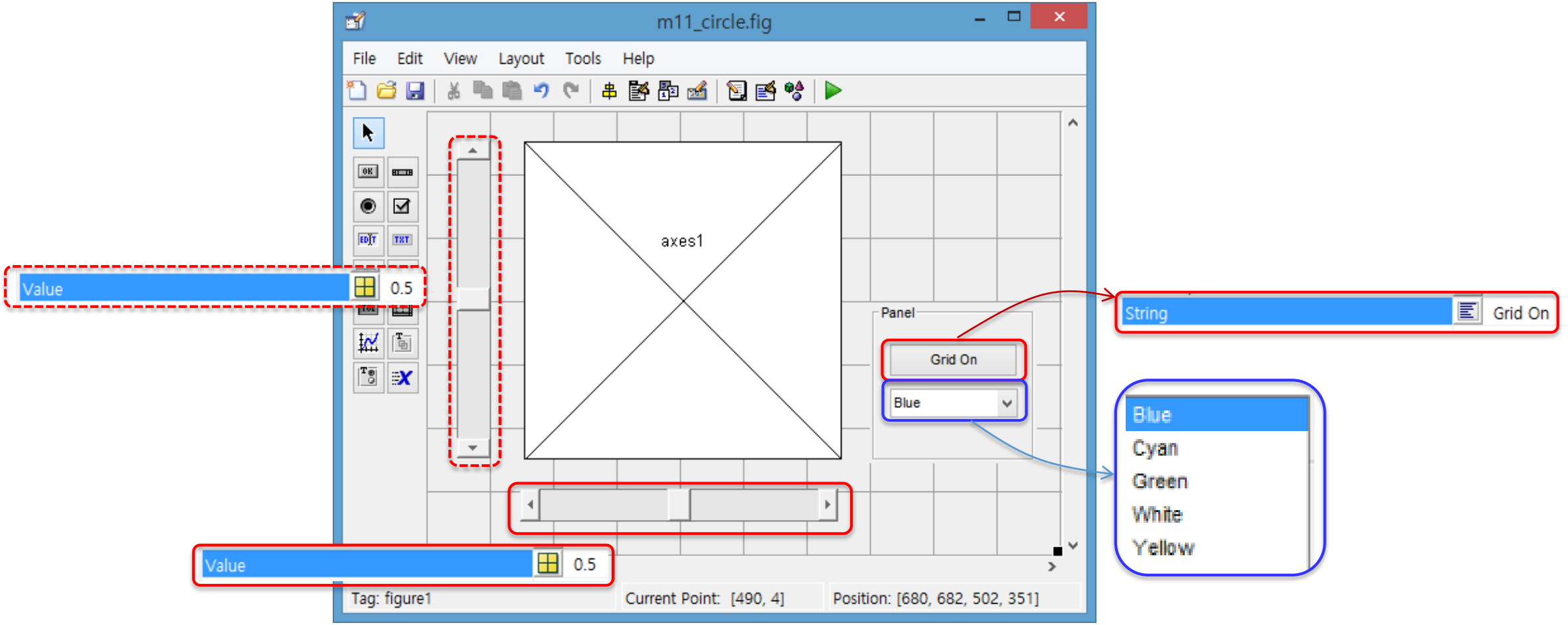
```
function update_price(hObject, handles, price )

if get(hObject,'Value') == 1
    handles.price = handles.price + price;
else
    handles.price = handles.price - price;
end
set( handles.text3, 'String', ...
    sprintf('the total price : $%.2f', handles.price));
guidata(hObject, handles);
```



Required to update the internal handle structure when some values in handles are changed.

Axes, Sliders, and Others

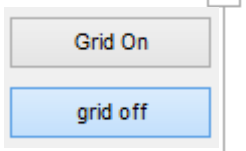
- Circle control – Width, Height, Grid, and Color



Axes, Sliders, and Others

- Callbacks

```
function m11_circle_OpeningFcn(hObj
handles.output = hObject;
draw_cricle(handles)
guidata(hObject, handles);

function slider1_Callback(hObject, even
draw_cricle( handles );

function togglebutton1_Callback(hObject,
grids = {'grid on', 'grid off'};
gi    = get(hObject,'Value');
set( hObject, 'String', grids{gi+1} );
draw_cricle( handles );

function popupmenu1_Callback(hObject, eve
draw_cricle( handles );
```

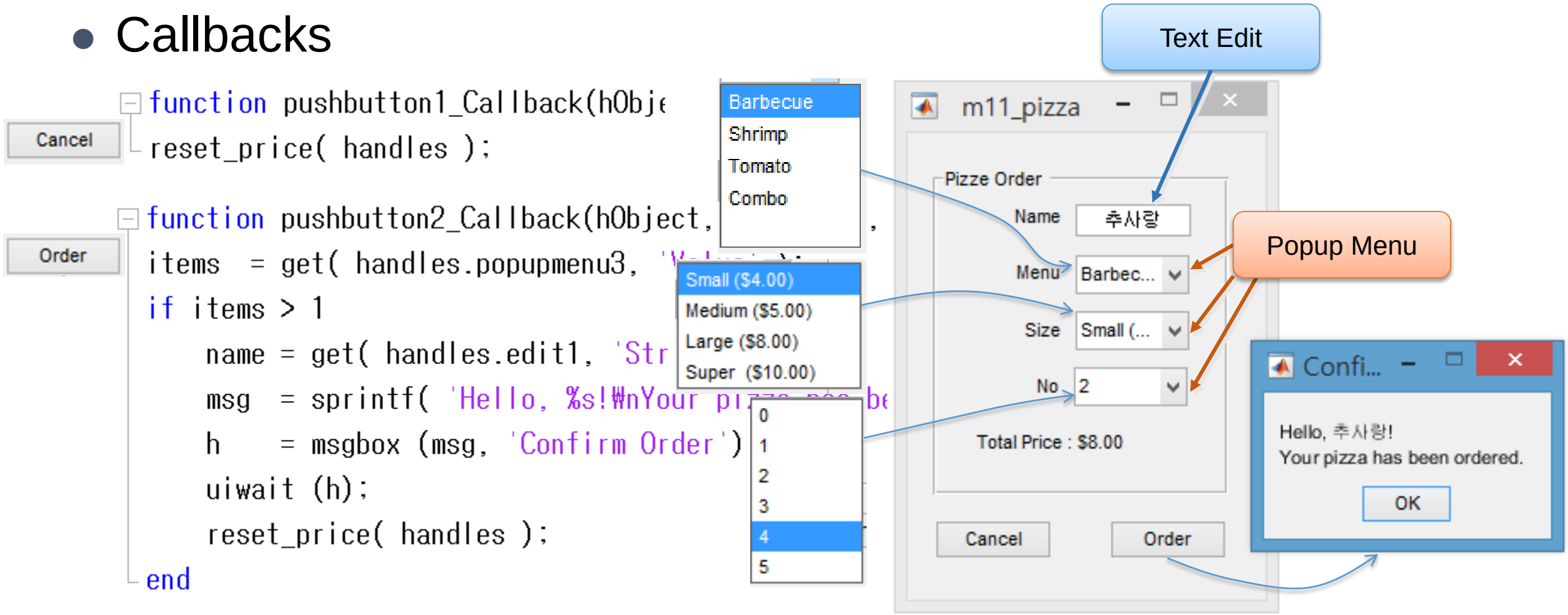



```
function draw_cricle( handles )

sx = get( handles.slider2, 'Value' );
sy = get( handles.slider1, 'Value' );

ci = get( handles.popupmenu1, 'Value' );
colors = 'bcgwy'; c = colors(ci);

gi = get( handles.togglebutton1, 'Value' );
grids  ={'on', 'off'}; g = grids{gi+1};

t  = linspace(0, 2*pi, 100);
x  = sx * cos(t); y = sy * sin(t);

if gi==0, cla; end;
patch(x, y, c);
axis([-1 1 -1 1]); axis square; grid(g);
```

Order Pizza

- Callbacks

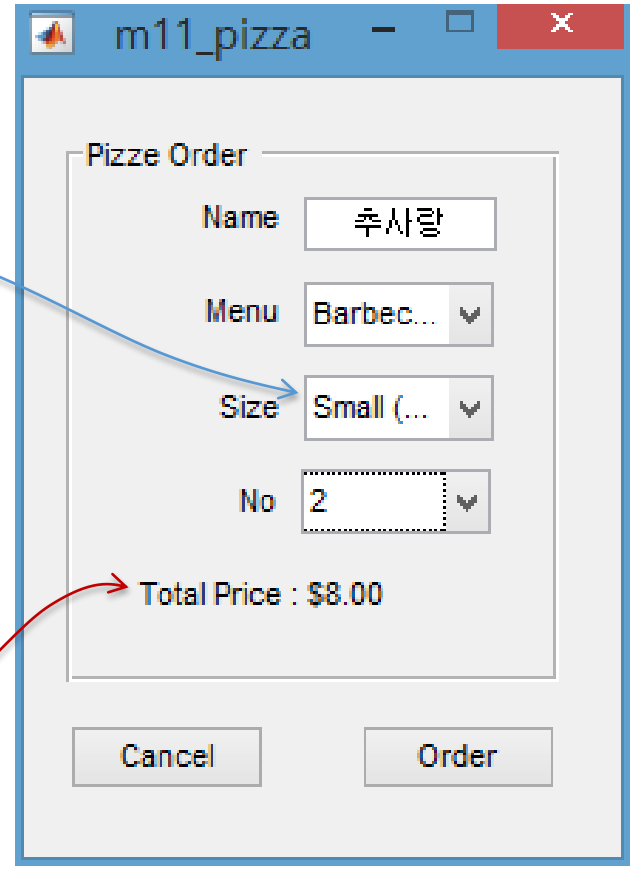
```
function pushbutton1_Callback(hObje
reset_price( handles );

function pushbutton2_Callback(hObject,
items  = get( handles.popupmenu3,
if items > 1
    name = get( handles.edit1, 'Str
    msg  = sprintf( 'Hello, %s!₩nYour pi
    h    = msgbox (msg, 'Confirm Order')
    uiwait (h);
    reset_price( handles );
end
```



Order Pizza

- Callbacks

```
function popupmenu2_Callback(hObject, eventdata, handles)
update_price( handles );

function update_price( handles )
prices = [4, 5, 8, 10];
s      = get(handles.popupmenu2,'Value');
price  = prices(s);
items  = cellstr(get(handles.popupmenu3,'String'));
n      = str2double(items{get(handles.popupmenu3,'Value')});
set( handles.text5, 'String', ...
    sprintf('Total Price : $%.2f', n * price ) );

function reset_price( handles )
set( handles.popupmenu2, 'Value', 1 );
set( handles.popupmenu3, 'Value', 1 );
update_price(handles);
```



Summary

- GUI : Graphical User Interface
- GUIDE: GUI Development Environment
- GUI Controls
  - Buttons: Push Button, Check Box, Toggle Button, Radio Button
  - Text: Edit Text, Static Text,
  - Choices: Popup Menu, List Box
  - Sliders and Axes / Button Group and Panel
- Callbacks
  - `value = get( handles.uicontrol,'Property');`
  - `set( handles.uicontrol, 'Property', Value);`

Learning Resources

- http://kr.mathworks.com/discovery/matlab-gui.html
- http://kr.mathworks.com/support/2014b/matlab/8.4/demos/creating-a-gui-with-guide.html?refresh=true
- http://blogs.mathworks.com/videos/category/gui-or-guide/