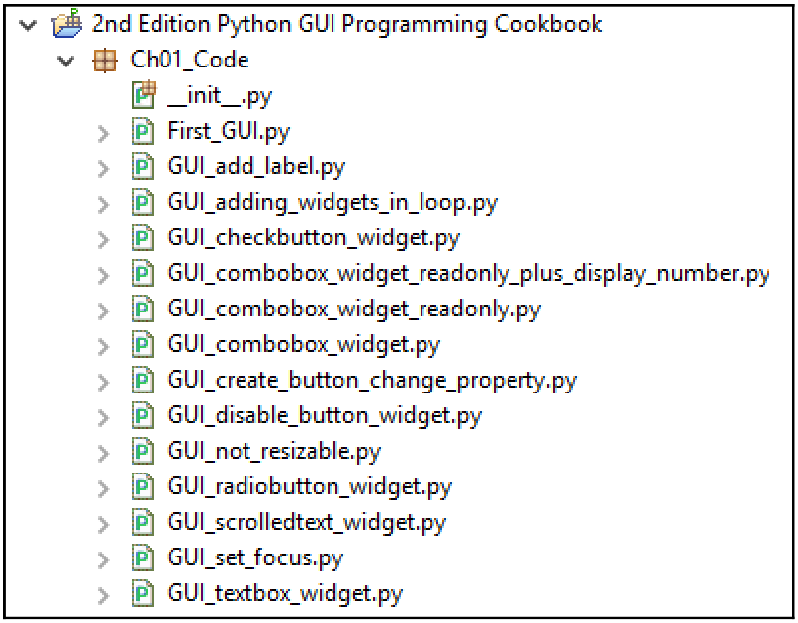
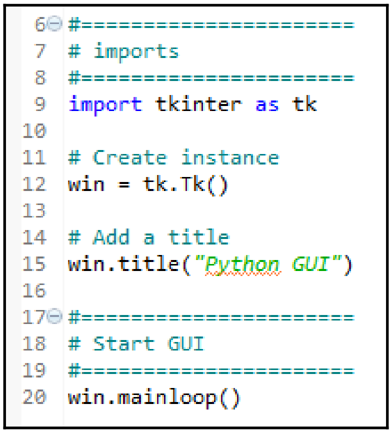
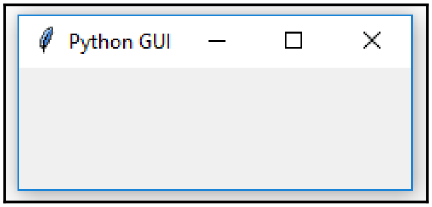
Chapter 1: Creating the GUI Form and Adding Widgets

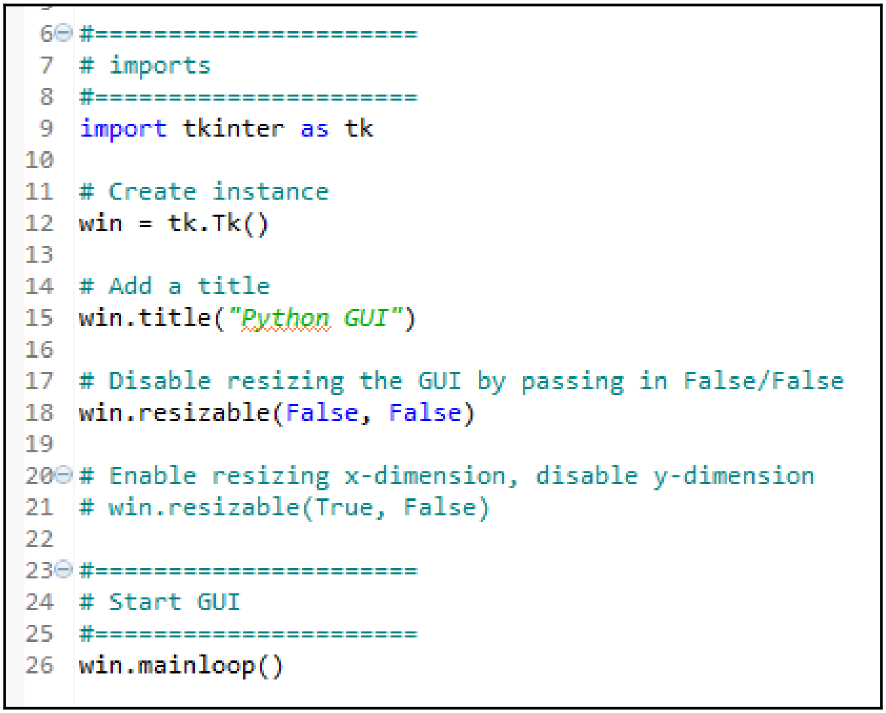


- 2nd Edition Python GUI Programming Cookbook
  - Ch01_Code
    - __init__.py
    - First_GUI.py
    - GUI_add_label.py
    - GUI_adding_widgets_in_loop.py
    - GUI_checkbutton_widget.py
    - GUI_combobox_widget_readonly_plus_display_number.py
    - GUI_combobox_widget_readonly.py
    - GUI_combobox_widget.py
    - GUI_create_button_change_property.py
    - GUI_disable_button_widget.py
    - GUI_not_resizable.py
    - GUI_radiobutton_widget.py
    - GUI_scrolledtext_widget.py
    - GUI_set_focus.py
    - GUI_textbox_widget.py

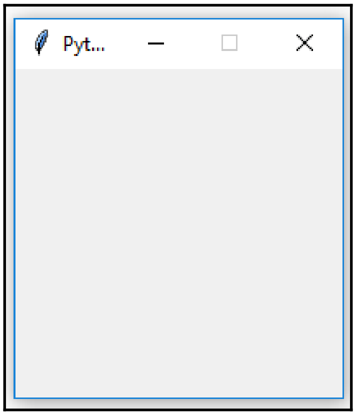


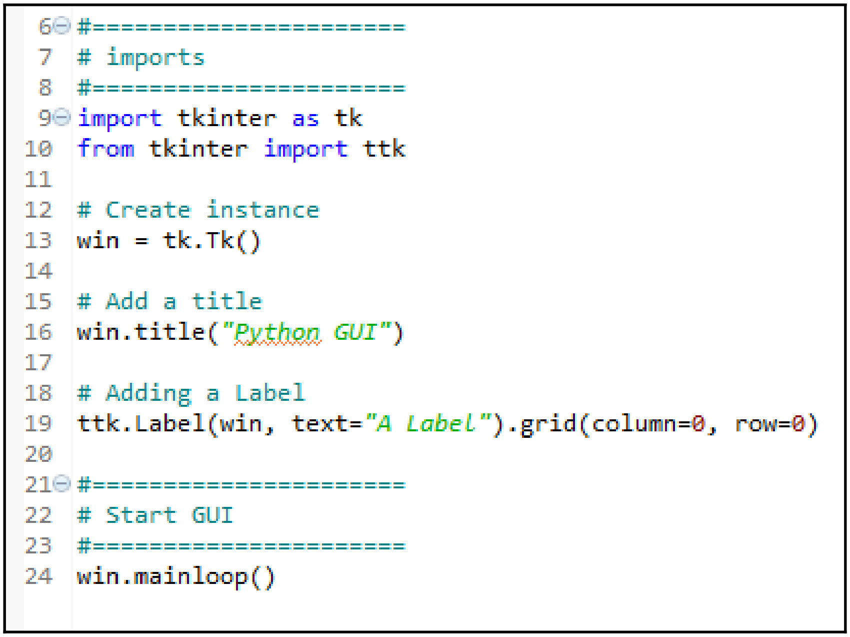
```
#======================
# imports
#======================
import tkinter as tk

# Create instance
win = tk.Tk()

# Add a title
win.title("Python GUI")

#======================
# Start GUI
#======================
win.mainloop()
```



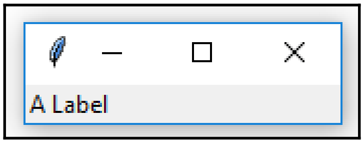


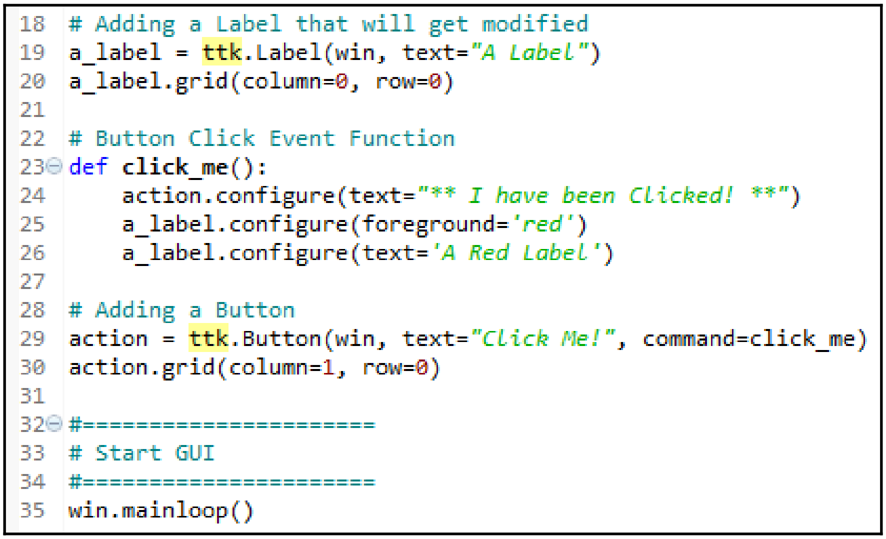
```
#======================
# imports
#======================
import tkinter as tk

# Create instance
win = tk.Tk()

# Add a title
win.title("Python GUI")

# Disable resizing the GUI by passing in False/False
win.resizable(False, False)

# Enable resizing x-dimension, disable y-dimension
# win.resizable(True, False)

#======================
# Start GUI
#======================
win.mainloop()
```



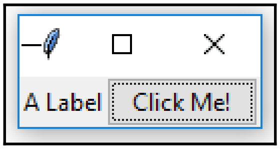


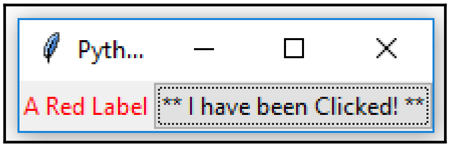
```python
#======================
# imports
#======================
import tkinter as tk
from tkinter import ttk

# Create instance
win = tk.Tk()

# Add a title
win.title("Python GUI")

# Adding a Label
ttk.Label(win, text="A Label").grid(column=0, row=0)

#======================
# Start GUI
#======================
win.mainloop()
```

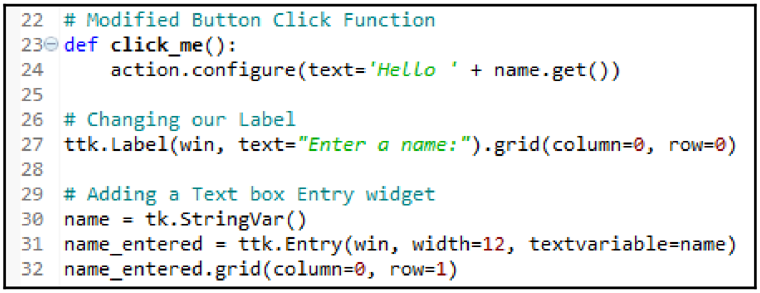




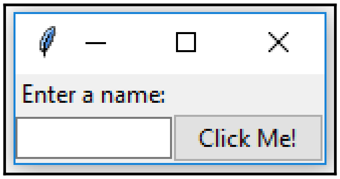
```python
# Adding a Label that will get modified
a_label = ttk.Label(win, text="A Label")
a_label.grid(column=0, row=0)

# Button Click Event Function
def click_me():
    action.configure(text="** I have been Clicked! **")
    a_label.configure(foreground='red')
    a_label.configure(text='A Red Label')

# Adding a Button
action = ttk.Button(win, text="Click Me!", command=click_me)
action.grid(column=1, row=0)

#======================
# Start GUI
#======================
win.mainloop()
```

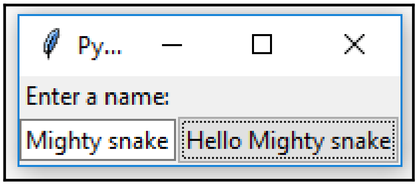


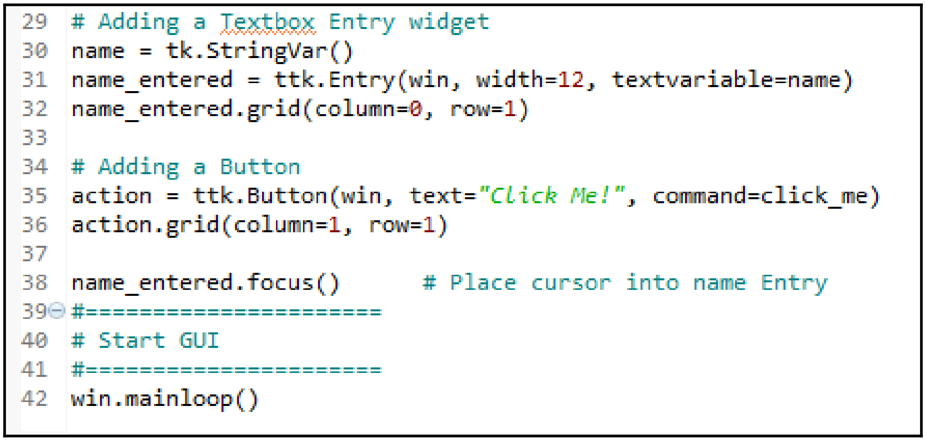




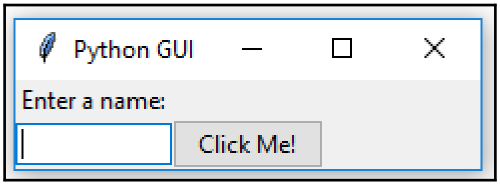
```
# Modified Button Click Function
def click_me():
    action.configure(text='Hello ' + name.get())

# Changing our Label
ttk.Label(win, text="Enter a name:").grid(column=0, row=0)

# Adding a Text box Entry widget
name = tk.StringVar()
name_entered = ttk.Entry(win, width=12, textvariable=name)
name_entered.grid(column=0, row=1)
```

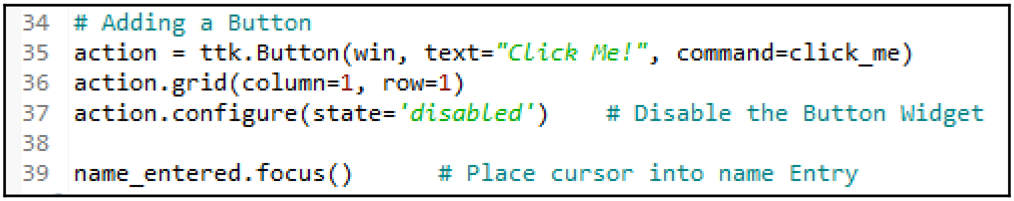




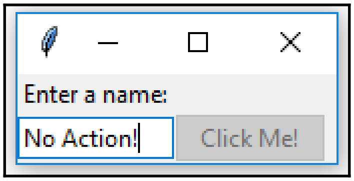


```
# Adding a Textbox Entry widget
name = tk.StringVar()
name_entered = ttk.Entry(win, width=12, textvariable=name)
name_entered.grid(column=0, row=1)

# Adding a Button
action = ttk.Button(win, text="Click Me!", command=click_me)
action.grid(column=1, row=1)

name_entered.focus()      # Place cursor into name Entry
#======================
# Start GUI
#======================
win.mainloop()
```

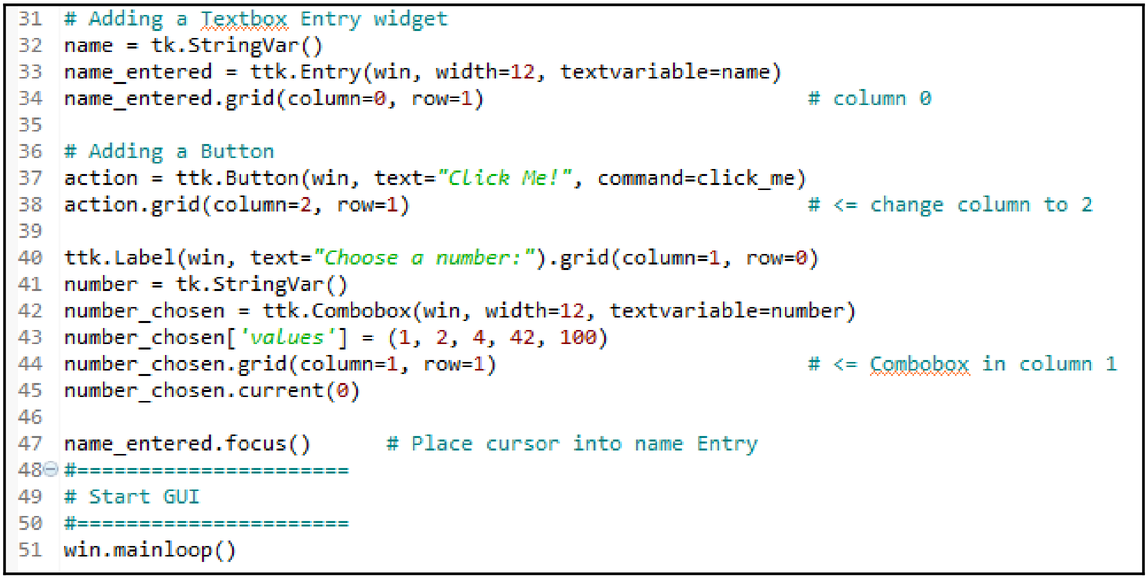




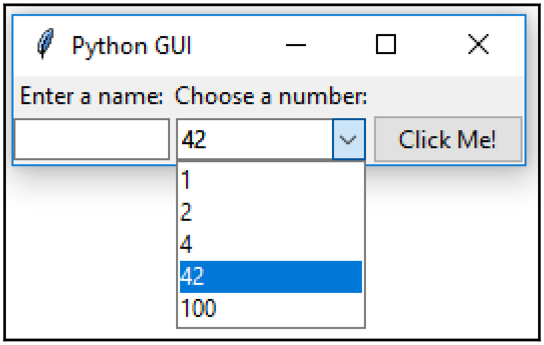
```
# Adding a Button
action = ttk.Button(win, text="Click Me!", command=click_me)
action.grid(column=1, row=1)
action.configure(state='disabled')     # Disable the Button Widget

name_entered.focus()      # Place cursor into name Entry
```

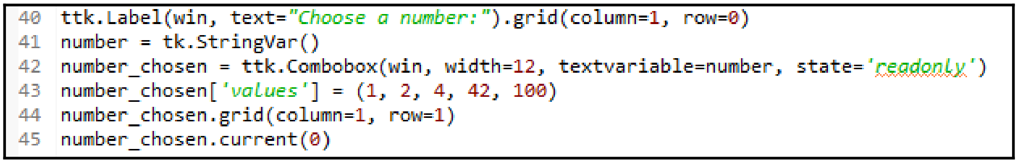


No Action!

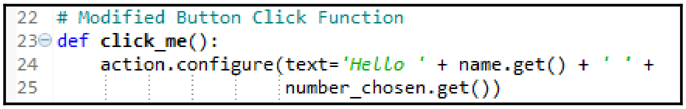


```python
# Adding a Textbox Entry widget
name = tk.StringVar()
name_entered = ttk.Entry(win, width=12, textvariable=name)
name_entered.grid(column=0, row=1)                          # column 0

# Adding a Button
action = ttk.Button(win, text="Click Me!", command=click_me)
action.grid(column=2, row=1)                                # <= change column to 2

ttk.Label(win, text="Choose a number:").grid(column=1, row=0)
number = tk.StringVar()
number_chosen = ttk.Combobox(win, width=12, textvariable=number)
number_chosen['values'] = (1, 2, 4, 42, 100)
number_chosen.grid(column=1, row=1)                         # <= Combobox in column 1
number_chosen.current(0)

name_entered.focus()      # Place cursor into name Entry
#======================
# Start GUI
#======================
win.mainloop()
```

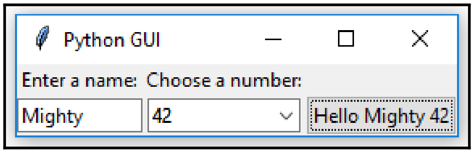




```python
ttk.Label(win, text="Choose a number:").grid(column=1, row=0)
number = tk.StringVar()
number_chosen = ttk.Combobox(win, width=12, textvariable=number, state='readonly')
number_chosen['values'] = (1, 2, 4, 42, 100)
number_chosen.grid(column=1, row=1)
number_chosen.current(0)
```

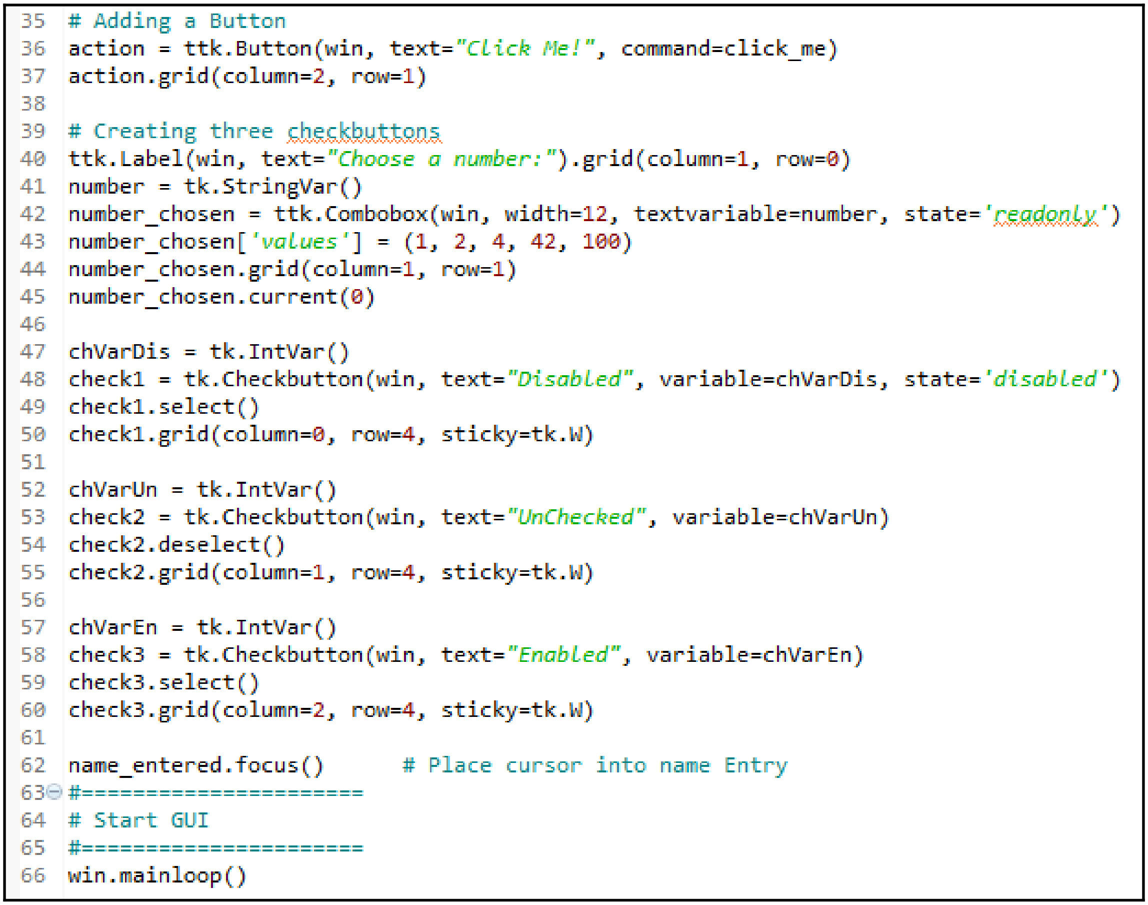


```python
# Modified Button Click Function
def click_me():
    action.configure(text='Hello ' + name.get() + ' ' +
                      number_chosen.get())
```

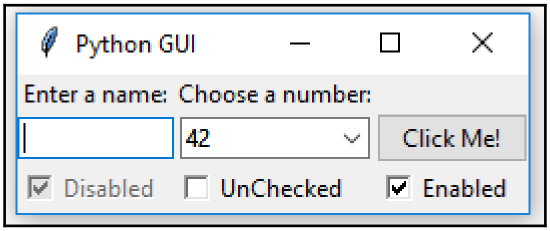


Mighty

42

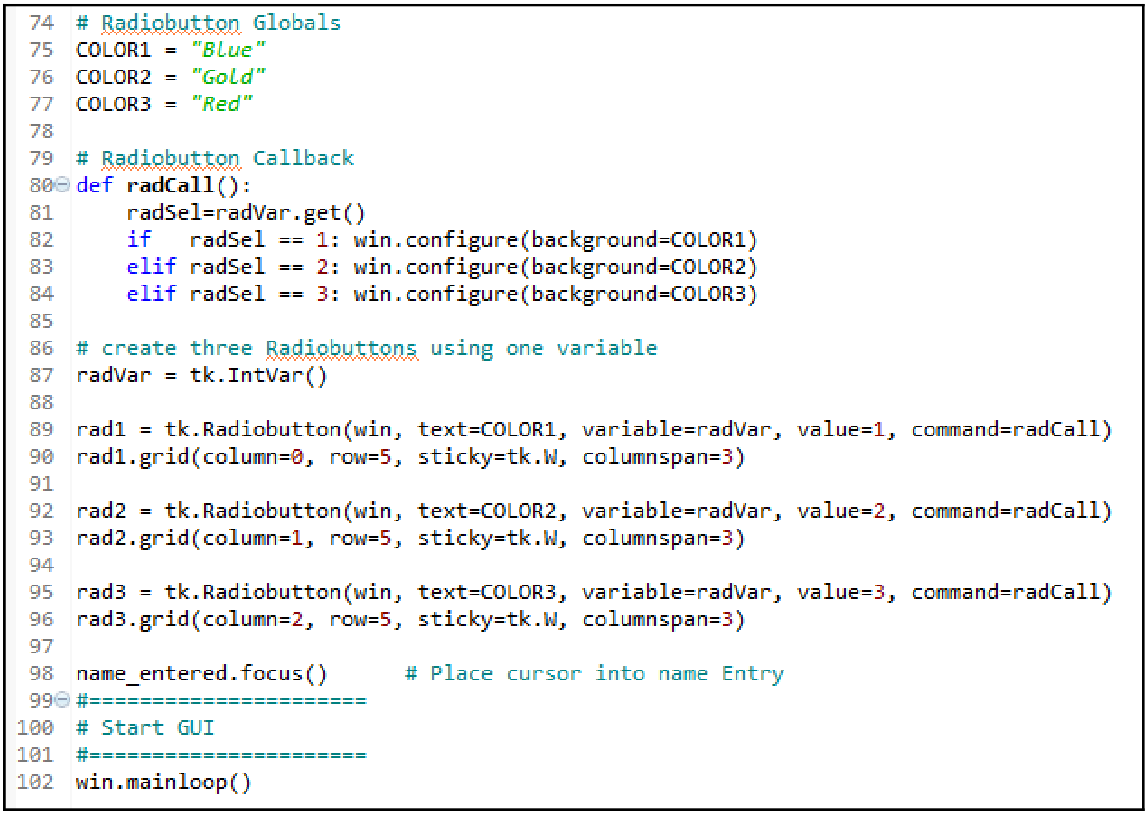


```
# Adding a Button
action = ttk.Button(win, text="Click Me!", command=click_me)
action.grid(column=2, row=1)

# Creating three checkbuttons
ttk.Label(win, text="Choose a number:").grid(column=1, row=0)
number = tk.StringVar()
number_chosen = ttk.Combobox(win, width=12, textvariable=number, state='readonly')
number_chosen['values'] = (1, 2, 4, 42, 100)
number_chosen.grid(column=1, row=1)
number_chosen.current(0)

chVarDis = tk.IntVar()
check1 = tk.Checkbutton(win, text="Disabled", variable=chVarDis, state='disabled')
check1.select()
check1.grid(column=0, row=4, sticky=tk.W)

chVarUn = tk.IntVar()
check2 = tk.Checkbutton(win, text="UnChecked", variable=chVarUn)
check2.deselect()
check2.grid(column=1, row=4, sticky=tk.W)

chVarEn = tk.IntVar()
check3 = tk.Checkbutton(win, text="Enabled", variable=chVarEn)
check3.select()
check3.grid(column=2, row=4, sticky=tk.W)

name_entered.focus()      # Place cursor into name Entry
#======================
# Start GUI
#======================
win.mainloop()
```

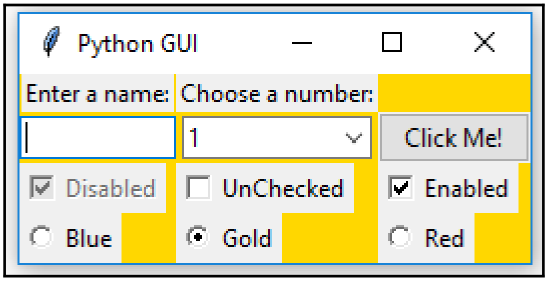


42

☑ Disabled ☐ UnChecked ☑ Enabled



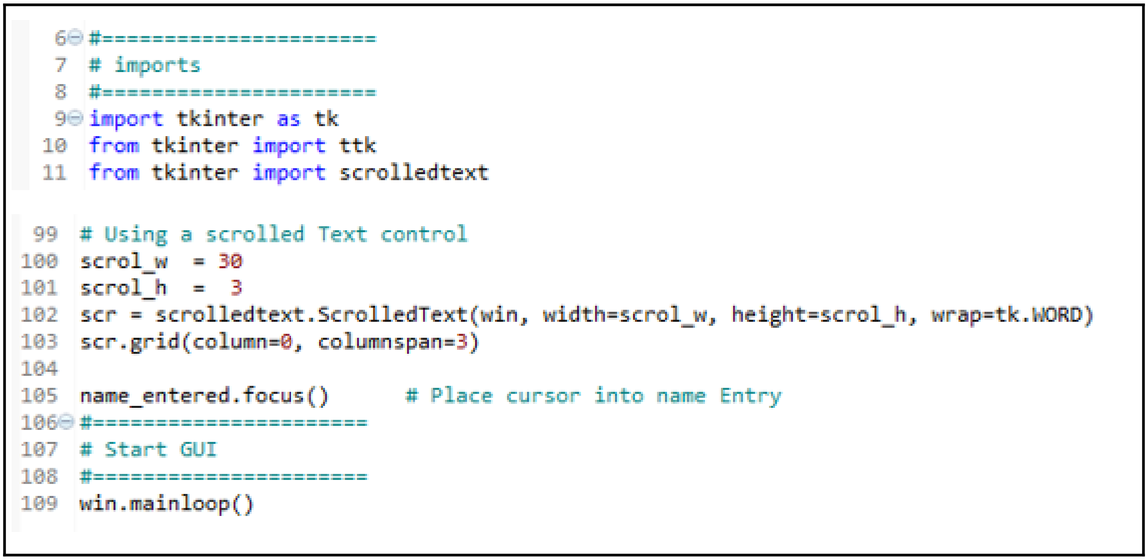
```
# Radiobutton Globals
COLOR1 = "Blue"
COLOR2 = "Gold"
COLOR3 = "Red"

# Radiobutton Callback
def radCall():
    radSel=radVar.get()
    if   radSel == 1: win.configure(background=COLOR1)
    elif radSel == 2: win.configure(background=COLOR2)
    elif radSel == 3: win.configure(background=COLOR3)

# create three Radiobuttons using one variable
radVar = tk.IntVar()

rad1 = tk.Radiobutton(win, text=COLOR1, variable=radVar, value=1, command=radCall)
rad1.grid(column=0, row=5, sticky=tk.W, columnspan=3)

rad2 = tk.Radiobutton(win, text=COLOR2, variable=radVar, value=2, command=radCall)
rad2.grid(column=1, row=5, sticky=tk.W, columnspan=3)

rad3 = tk.Radiobutton(win, text=COLOR3, variable=radVar, value=3, command=radCall)
rad3.grid(column=2, row=5, sticky=tk.W, columnspan=3)

name_entered.focus()      # Place cursor into name Entry
#======================
# Start GUI
#======================
win.mainloop()
```



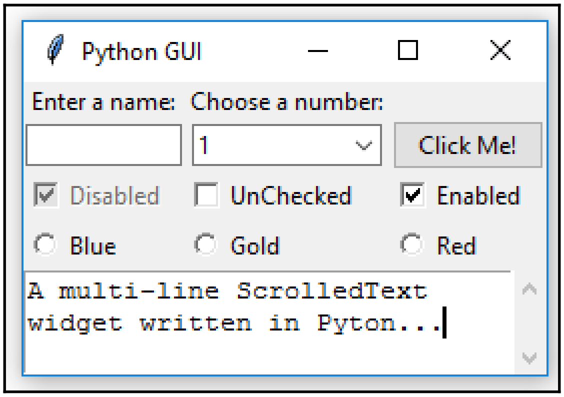
Enter a name:
Choose a number: 1

☑ Disabled □ UnChecked ☑ Enabled

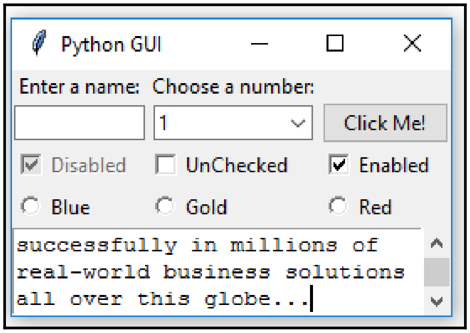
○ Blue ● Gold ○ Red



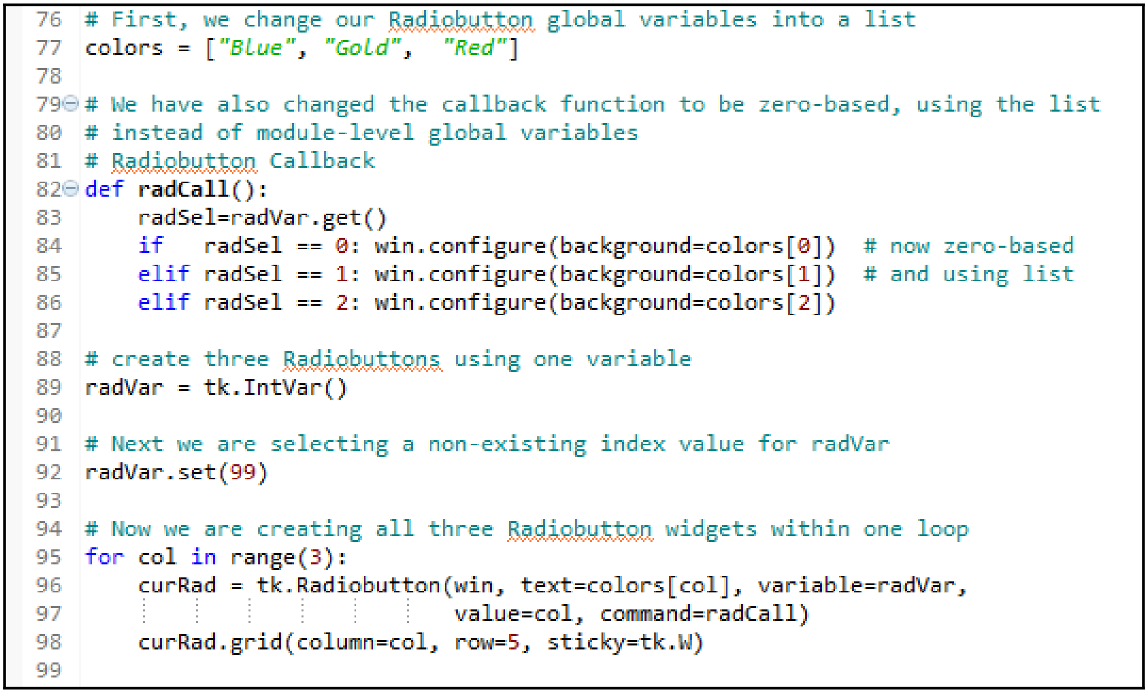
```
#======================
# imports
#======================
import tkinter as tk
from tkinter import ttk
from tkinter import scrolledtext

# Using a scrolled Text control
scrol_w  = 30
scrol_h  =  3
scr = scrolledtext.ScrolledText(win, width=scrol_w, height=scrol_h, wrap=tk.WORD)
scr.grid(column=0, columnspan=3)

name_entered.focus()      # Place cursor into name Entry
#======================
# Start GUI
#======================
win.mainloop()
```



```
A multi-line ScrolledText
widget written in Pyton...
```

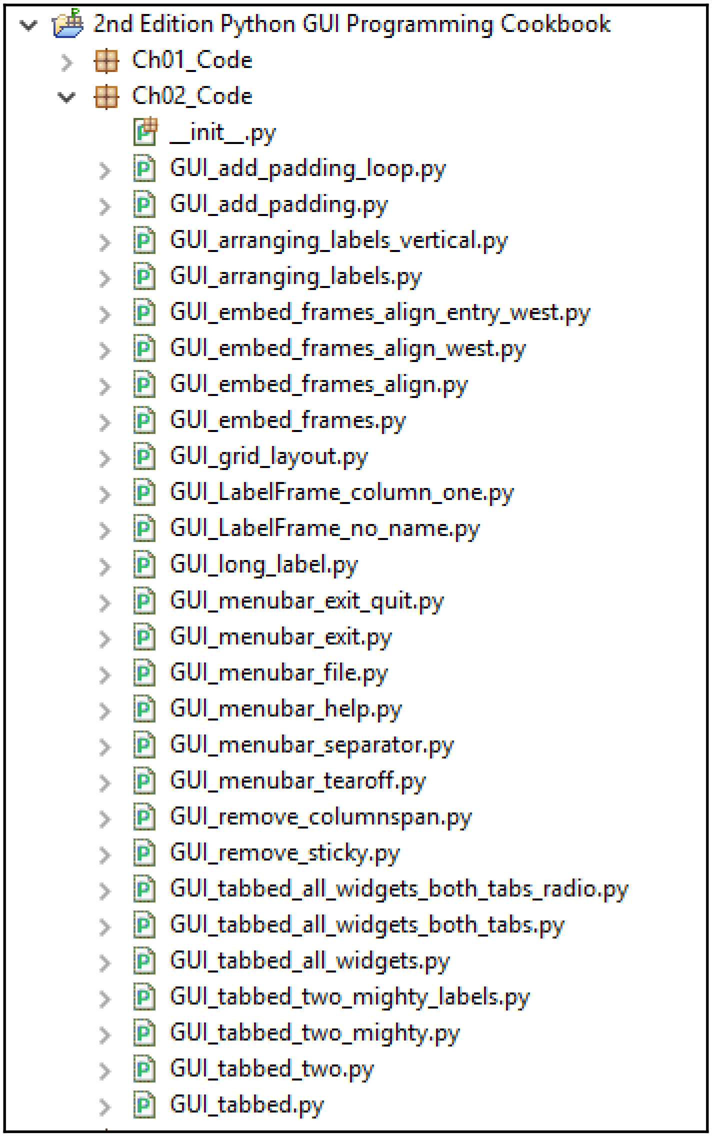


```
successfully in millions of
real-world business solutions
all over this globe...
```

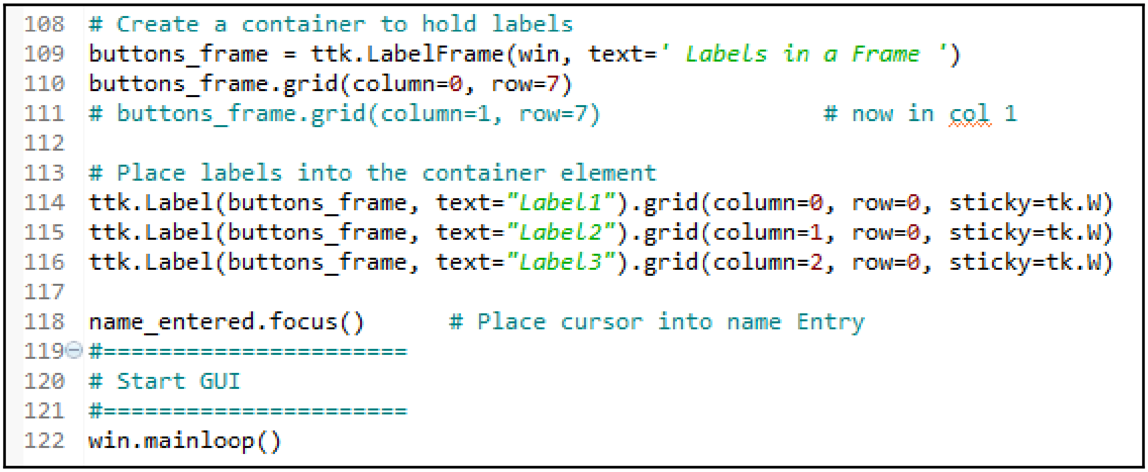


```python
# First, we change our Radiobutton global variables into a list
colors = ["Blue", "Gold",  "Red"]

# We have also changed the callback function to be zero-based, using the list
# instead of module-level global variables
# Radiobutton Callback
def radCall():
    radSel=radVar.get()
    if   radSel == 0: win.configure(background=colors[0])  # now zero-based
    elif radSel == 1: win.configure(background=colors[1])  # and using list
    elif radSel == 2: win.configure(background=colors[2])

# create three Radiobuttons using one variable
radVar = tk.IntVar()

# Next we are selecting a non-existing index value for radVar
radVar.set(99)

# Now we are creating all three Radiobutton widgets within one loop
for col in range(3):
    curRad = tk.Radiobutton(win, text=colors[col], variable=radVar,
                            value=col, command=radCall)
    curRad.grid(column=col, row=5, sticky=tk.W)

```

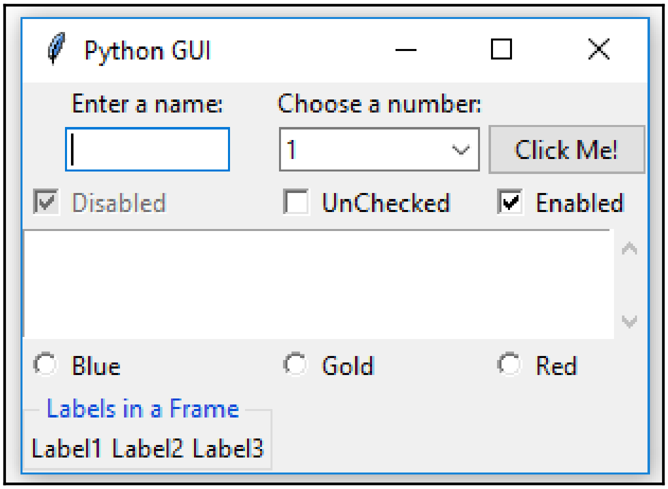
Chapter 2: Layout Management



```
2nd Edition Python GUI Programming Cookbook
  Ch01_Code
  Ch02_Code
    __init__.py
    GUI_add_padding_loop.py
    GUI_add_padding.py
    GUI_arranging_labels_vertical.py
    GUI_arranging_labels.py
    GUI_embed_frames_align_entry_west.py
    GUI_embed_frames_align_west.py
    GUI_embed_frames_align.py
    GUI_embed_frames.py
    GUI_grid_layout.py
    GUI_LabelFrame_column_one.py
    GUI_LabelFrame_no_name.py
    GUI_long_label.py
    GUI_menubar_exit_quit.py
    GUI_menubar_exit.py
    GUI_menubar_file.py
    GUI_menubar_help.py
    GUI_menubar_separator.py
    GUI_menubar_tearoff.py
    GUI_remove_columnspan.py
    GUI_remove_sticky.py
    GUI_tabbed_all_widgets_both_tabs_radio.py
    GUI_tabbed_all_widgets_both_tabs.py
    GUI_tabbed_all_widgets.py
    GUI_tabbed_two_mighty_labels.py
    GUI_tabbed_two_mighty.py
    GUI_tabbed_two.py
    GUI_tabbed.py
```



```python
# Create a container to hold labels
buttons_frame = ttk.LabelFrame(win, text=' Labels in a Frame ')
buttons_frame.grid(column=0, row=7)
# buttons_frame.grid(column=1, row=7)           # now in col 1

# Place labels into the container element
ttk.Label(buttons_frame, text="Label1").grid(column=0, row=0, sticky=tk.W)
ttk.Label(buttons_frame, text="Label2").grid(column=1, row=0, sticky=tk.W)
ttk.Label(buttons_frame, text="Label3").grid(column=2, row=0, sticky=tk.W)

name_entered.focus()      # Place cursor into name Entry
#======================
# Start GUI
#======================
win.mainloop()
```



Enter a name: Choose a number:

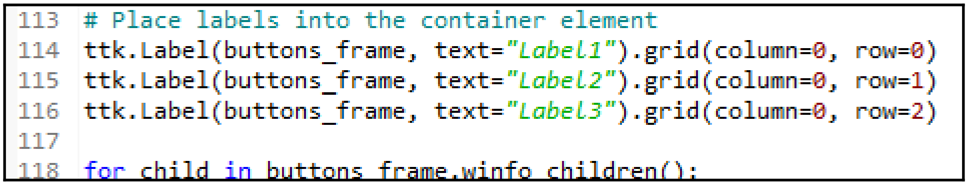
1

☑ Disabled ☐ UnChecked ☑ Enabled

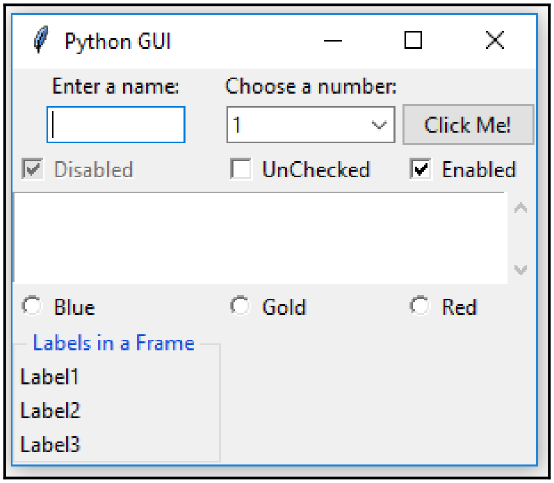
○ Blue ○ Gold ○ Red

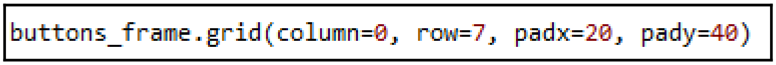
Labels in a Frame

Label1 Label2 Label3

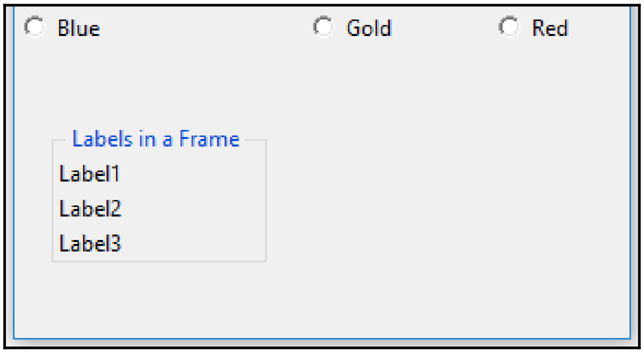


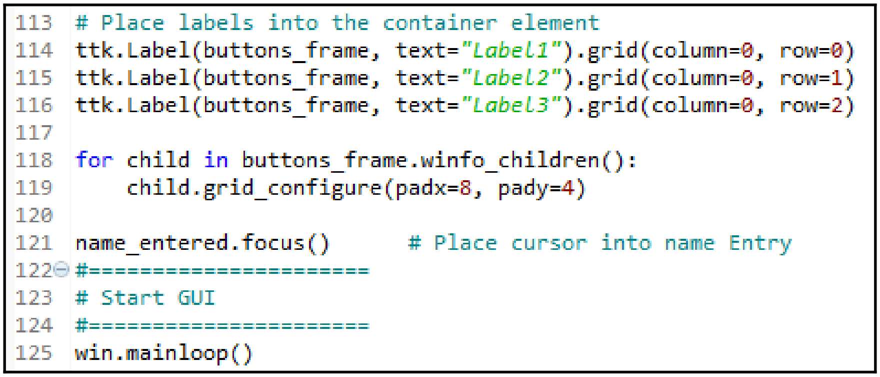
```python
# Place labels into the container element
ttk.Label(buttons_frame, text="Label1").grid(column=0, row=0)
ttk.Label(buttons_frame, text="Label2").grid(column=0, row=1)
ttk.Label(buttons_frame, text="Label3").grid(column=0, row=2)

for child in buttons_frame.winfo_children():
```



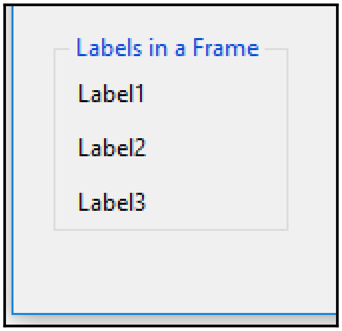


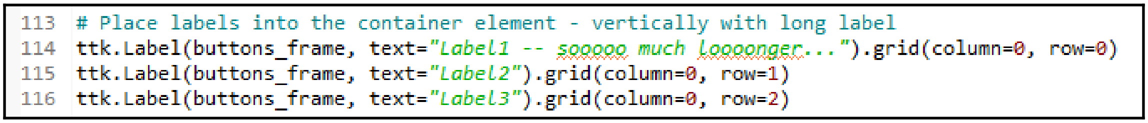
```
buttons_frame.grid(column=0, row=7, padx=20, pady=40)
```



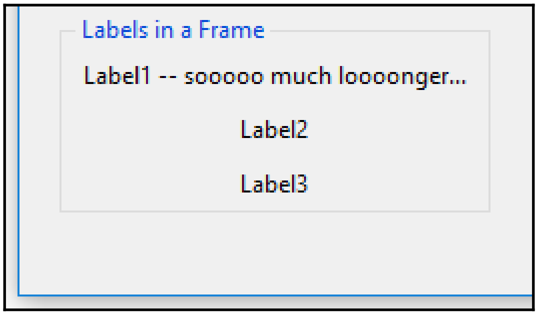


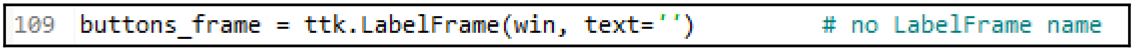
```
# Place labels into the container element
ttk.Label(buttons_frame, text="Label1").grid(column=0, row=0)
ttk.Label(buttons_frame, text="Label2").grid(column=0, row=1)
ttk.Label(buttons_frame, text="Label3").grid(column=0, row=2)

for child in buttons_frame.winfo_children():
    child.grid_configure(padx=8, pady=4)

name_entered.focus()      # Place cursor into name Entry
#======================
# Start GUI
#======================
win.mainloop()
```



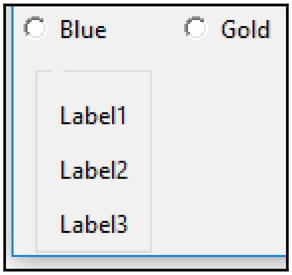


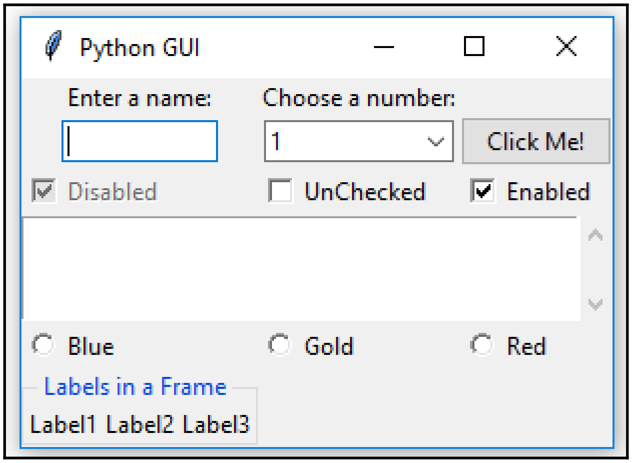
```
# Place labels into the container element - vertically with long label
ttk.Label(buttons_frame, text="Label1 -- sooooo much loooonger...").grid(column=0, row=0)
ttk.Label(buttons_frame, text="Label2").grid(column=0, row=1)
ttk.Label(buttons_frame, text="Label3").grid(column=0, row=2)
```

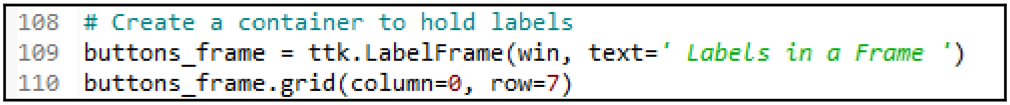




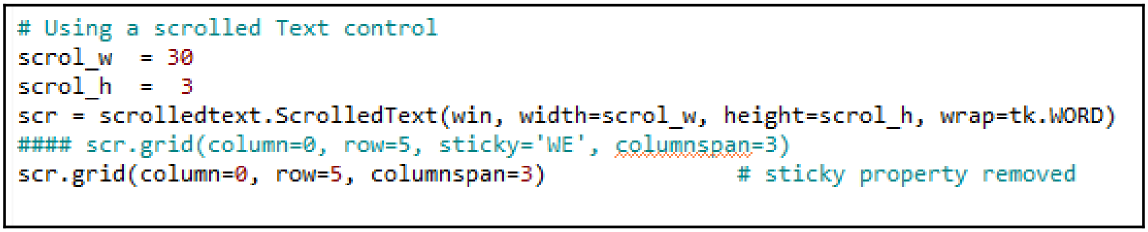
```
buttons_frame = ttk.LabelFrame(win, text='')        # no LabelFrame name
```



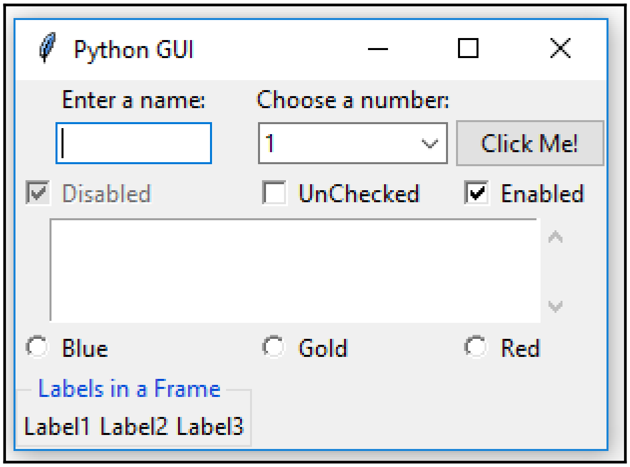


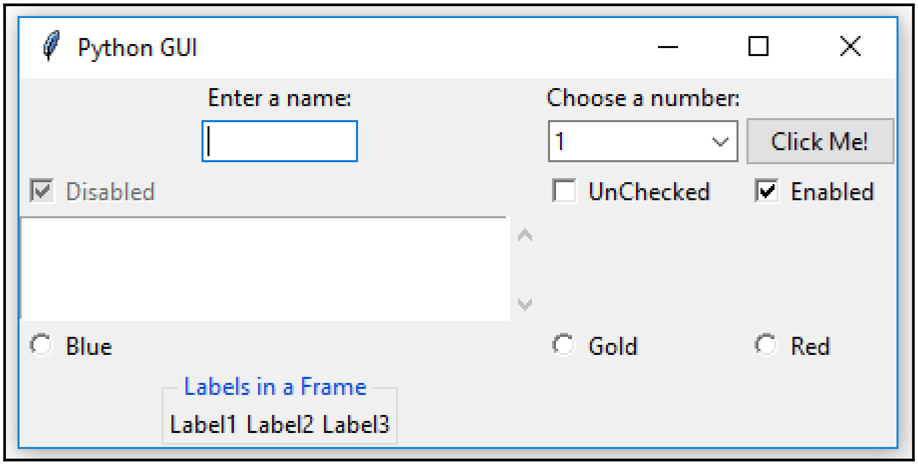


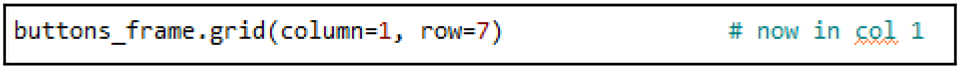
```
# Create a container to hold labels
buttons_frame = ttk.LabelFrame(win, text=' Labels in a Frame ')
buttons_frame.grid(column=0, row=7)
```



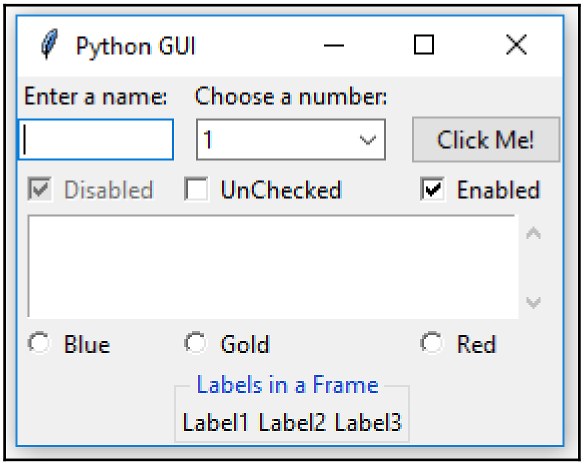
```
# Using a scrolled Text control
scrol_w  = 30
scrol_h  =  3
scr = scrolledtext.ScrolledText(win, width=scrol_w, height=scrol_h, wrap=tk.WORD)
#### scr.grid(column=0, row=5, sticky='WE', columnspan=3)
scr.grid(column=0, row=5, columnspan=3)              # sticky property removed
```

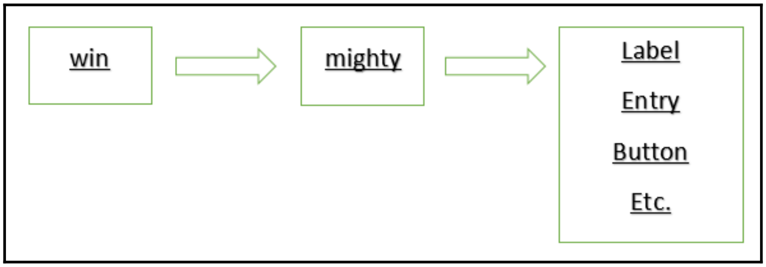


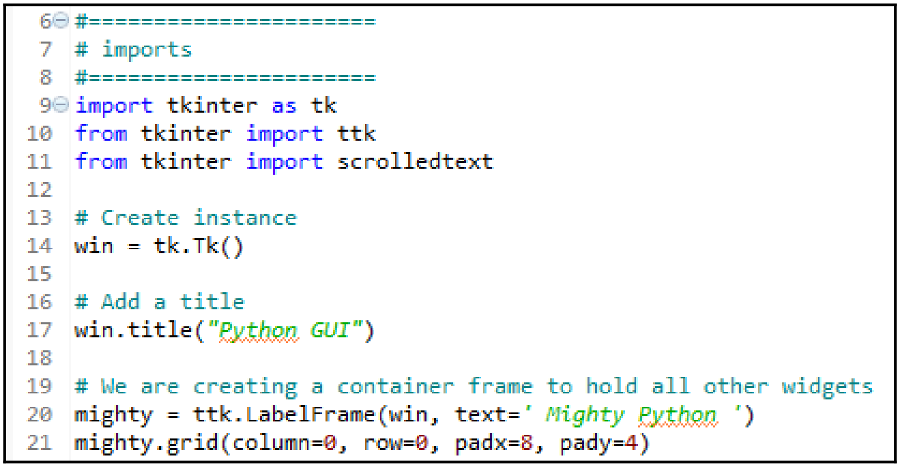




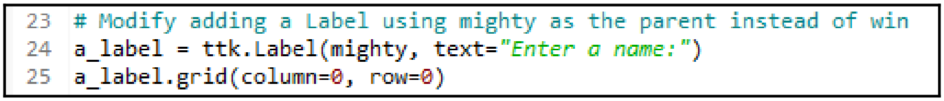
```
buttons_frame.grid(column=1, row=7)              # now in col 1
```



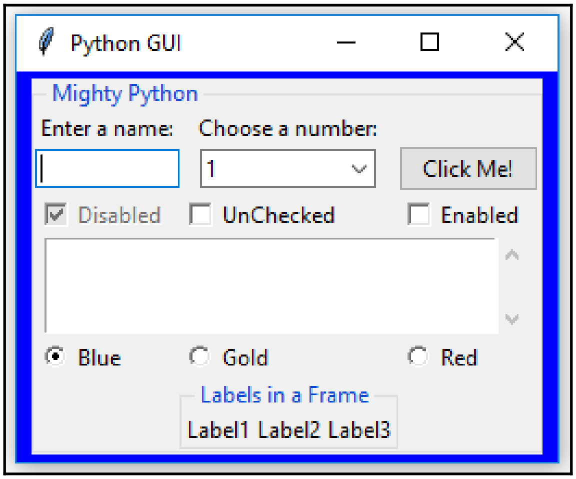


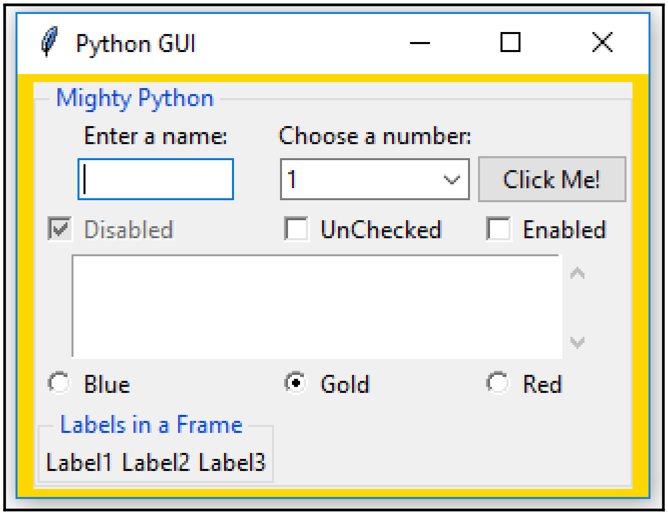


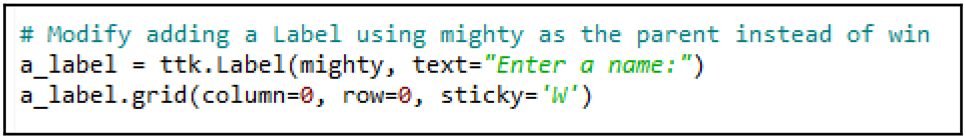
```
#======================
# imports
#======================
import tkinter as tk
from tkinter import ttk
from tkinter import scrolledtext

# Create instance
win = tk.Tk()

# Add a title
win.title("Python GUI")

# We are creating a container frame to hold all other widgets
mighty = ttk.LabelFrame(win, text=' Mighty Python ')
mighty.grid(column=0, row=0, padx=8, pady=4)
```



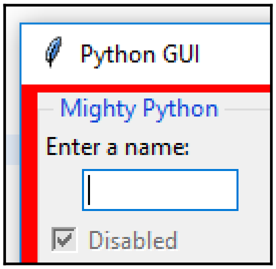
```
# Modify adding a Label using mighty as the parent instead of win
a_label = ttk.Label(mighty, text="Enter a name:")
a_label.grid(column=0, row=0)
```

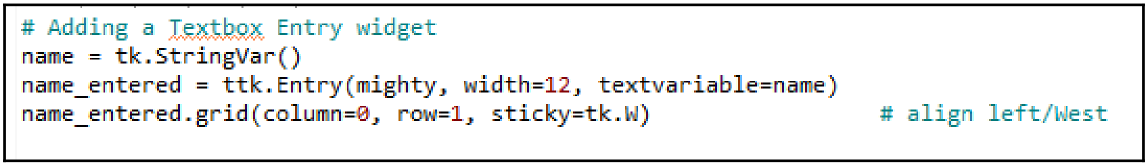




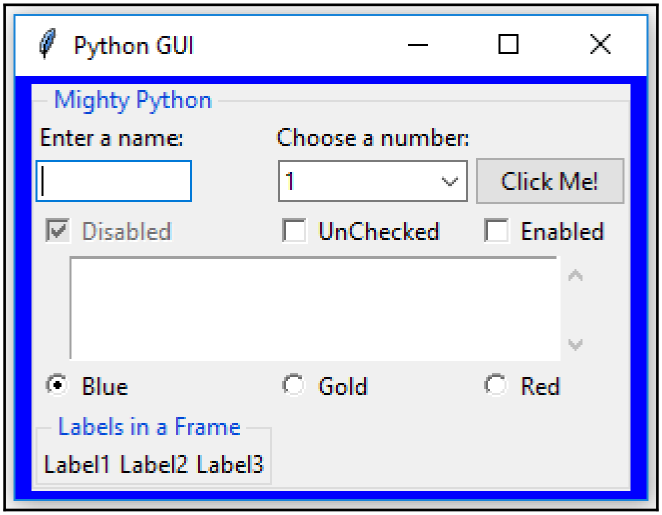


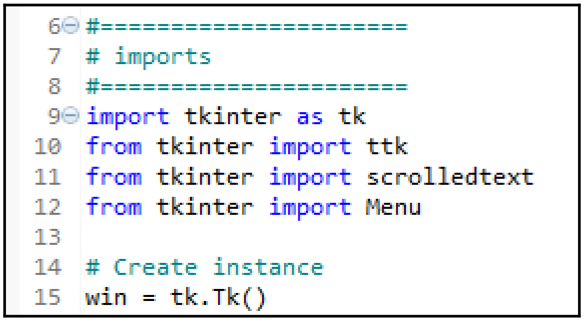
```
# Modify adding a Label using mighty as the parent instead of win
a_label = ttk.Label(mighty, text="Enter a name:")
a_label.grid(column=0, row=0, sticky='W')
```



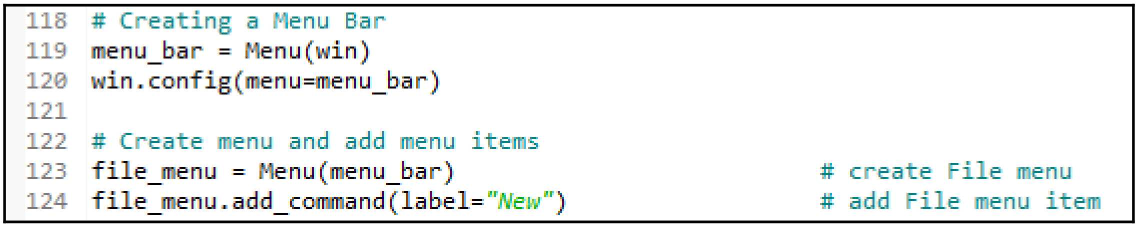


```
# Adding a Textbox Entry widget
name = tk.StringVar()
name_entered = ttk.Entry(mighty, width=12, textvariable=name)
name_entered.grid(column=0, row=1, sticky=tk.W)              # align left/West
```

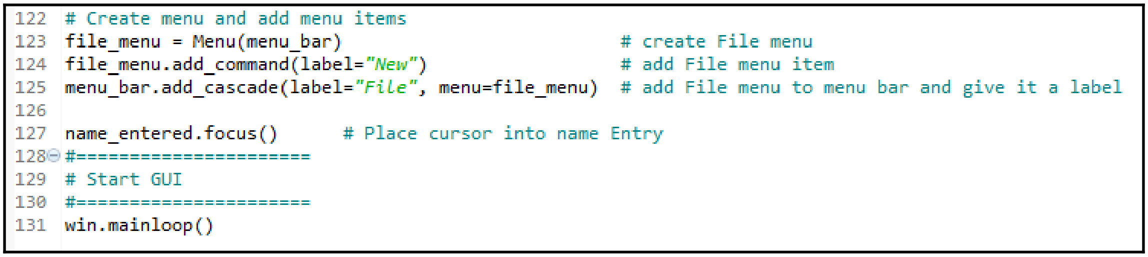




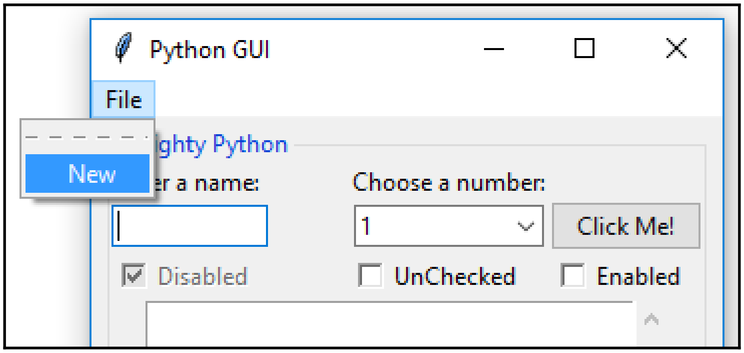
```
#======================
# imports
#======================
import tkinter as tk
from tkinter import ttk
from tkinter import scrolledtext
from tkinter import Menu

# Create instance
win = tk.Tk()
```

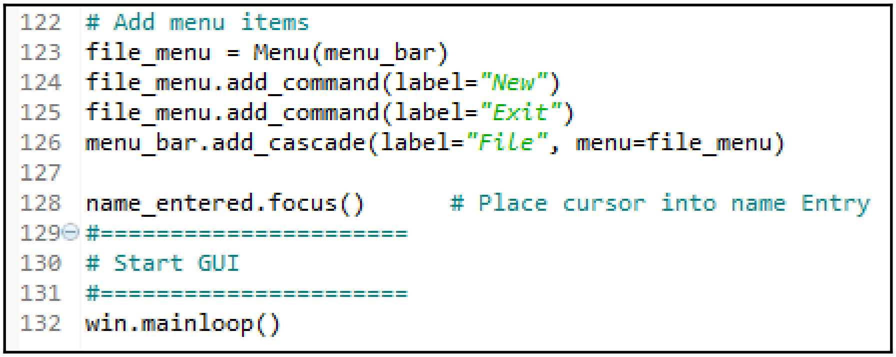


```
# Creating a Menu Bar
menu_bar = Menu(win)
win.config(menu=menu_bar)

# Create menu and add menu items
file_menu = Menu(menu_bar)                         # create File menu
file_menu.add_command(label="New")                 # add File menu item
```

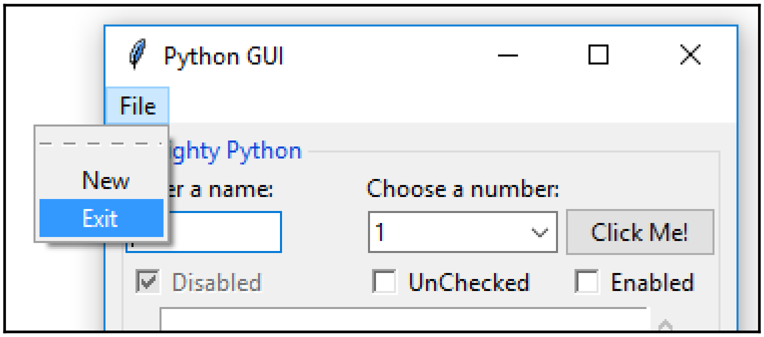


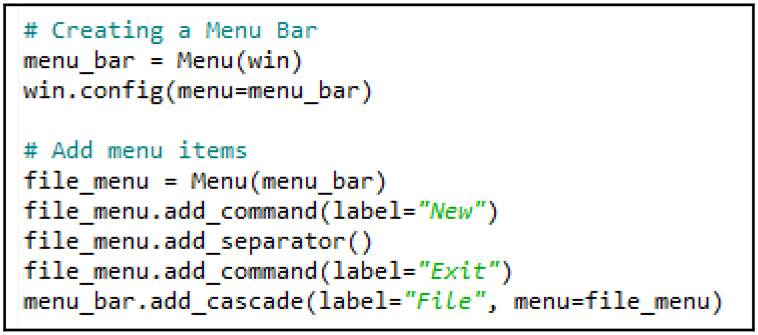
```
# Create menu and add menu items
file_menu = Menu(menu_bar)                          # create File menu
file_menu.add_command(label="New")                  # add File menu item
menu_bar.add_cascade(label="File", menu=file_menu)  # add File menu to menu bar and give it a label

name_entered.focus()      # Place cursor into name Entry
#======================
# Start GUI
#======================
win.mainloop()
```



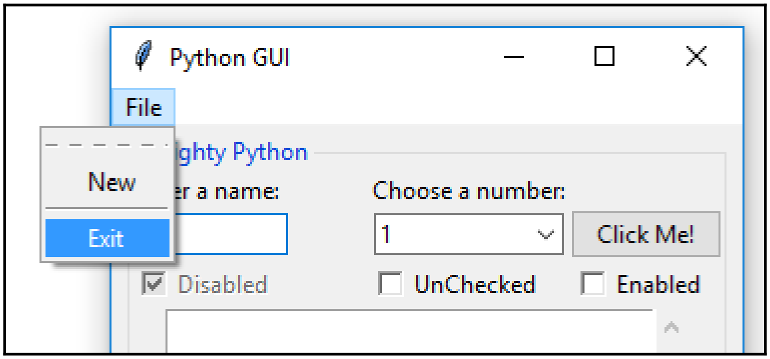


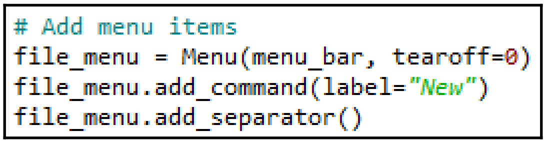
```
# Add menu items
file_menu = Menu(menu_bar)
file_menu.add_command(label="New")
file_menu.add_command(label="Exit")
menu_bar.add_cascade(label="File", menu=file_menu)

name_entered.focus()      # Place cursor into name Entry
#======================
# Start GUI
#======================
win.mainloop()
```



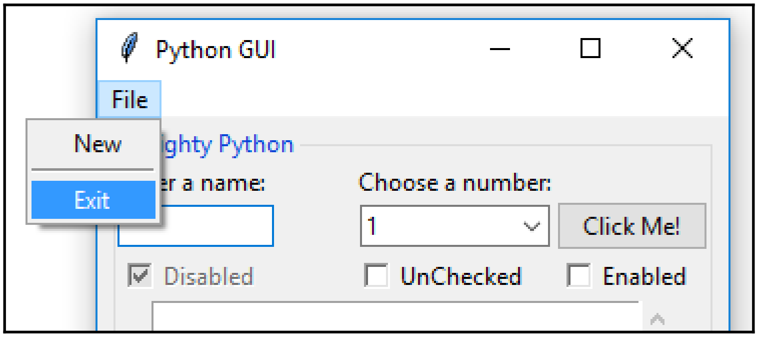


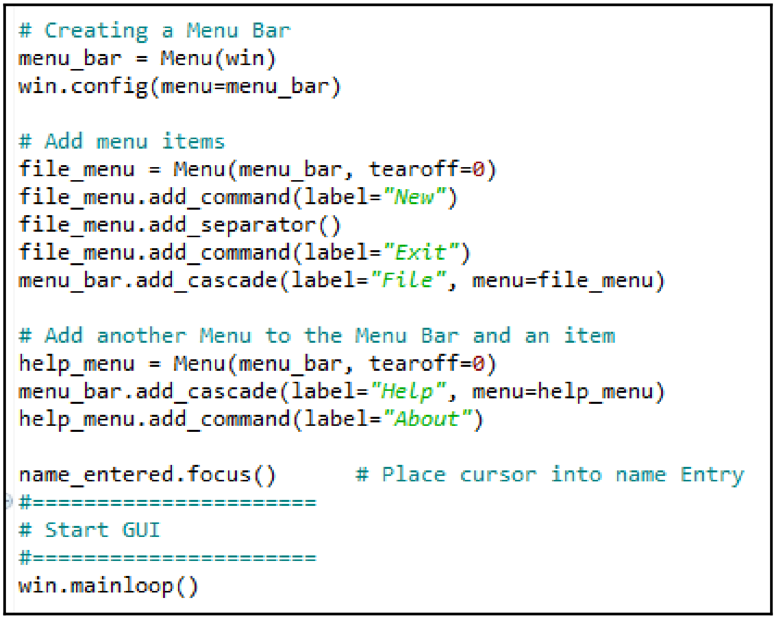
```
# Creating a Menu Bar
menu_bar = Menu(win)
win.config(menu=menu_bar)

# Add menu items
file_menu = Menu(menu_bar)
file_menu.add_command(label="New")
file_menu.add_separator()
file_menu.add_command(label="Exit")
menu_bar.add_cascade(label="File", menu=file_menu)
```



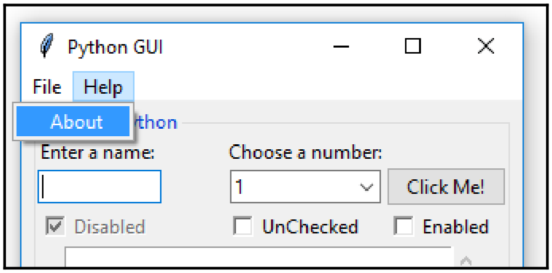


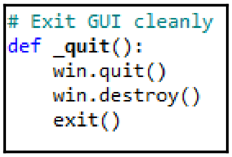
```
# Add menu items
file_menu = Menu(menu_bar, tearoff=0)
file_menu.add_command(label="New")
file_menu.add_separator()
```



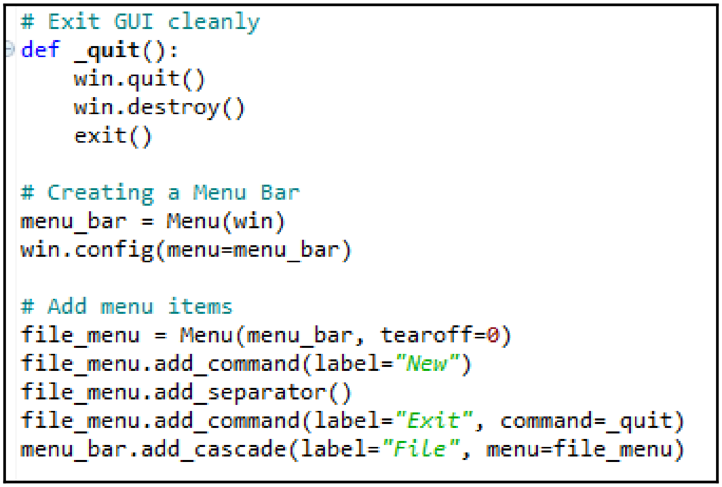


```
# Creating a Menu Bar
menu_bar = Menu(win)
win.config(menu=menu_bar)

# Add menu items
file_menu = Menu(menu_bar, tearoff=0)
file_menu.add_command(label="New")
file_menu.add_separator()
file_menu.add_command(label="Exit")
menu_bar.add_cascade(label="File", menu=file_menu)

# Add another Menu to the Menu Bar and an item
help_menu = Menu(menu_bar, tearoff=0)
menu_bar.add_cascade(label="Help", menu=help_menu)
help_menu.add_command(label="About")

name_entered.focus()      # Place cursor into name Entry
#======================
# Start GUI
#======================
win.mainloop()
```

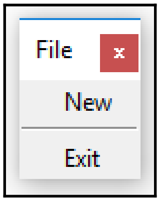


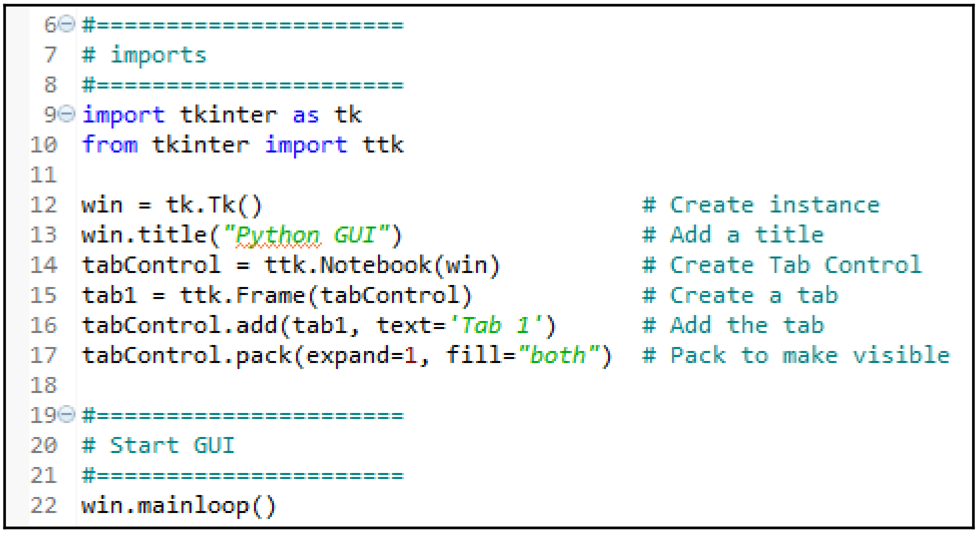


```
# Exit GUI cleanly
def _quit():
    win.quit()
    win.destroy()
    exit()
```

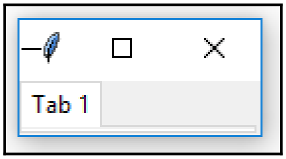


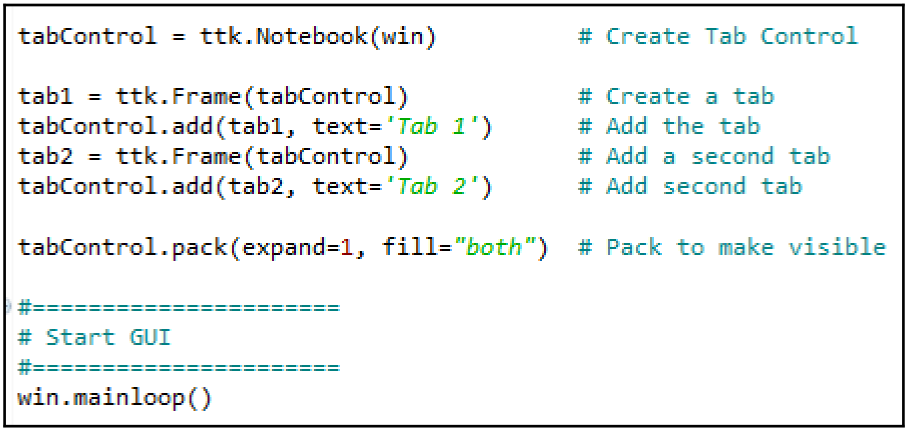
```python
# Exit GUI cleanly
def _quit():
    win.quit()
    win.destroy()
    exit()

# Creating a Menu Bar
menu_bar = Menu(win)
win.config(menu=menu_bar)

# Add menu items
file_menu = Menu(menu_bar, tearoff=0)
file_menu.add_command(label="New")
file_menu.add_separator()
file_menu.add_command(label="Exit", command=_quit)
menu_bar.add_cascade(label="File", menu=file_menu)
```



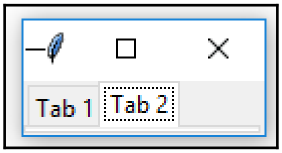


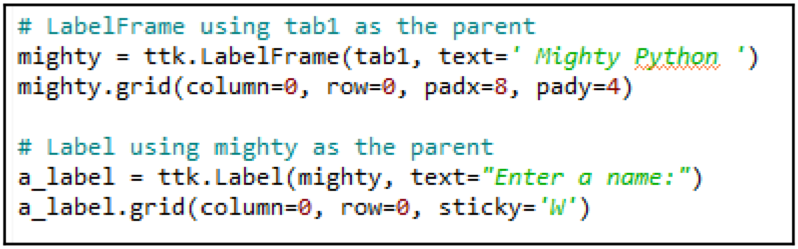
```python
#======================
# imports
#======================
import tkinter as tk
from tkinter import ttk

win = tk.Tk()                               # Create instance
win.title("Python GUI")                     # Add a title
tabControl = ttk.Notebook(win)              # Create Tab Control
tab1 = ttk.Frame(tabControl)                # Create a tab
tabControl.add(tab1, text='Tab 1')          # Add the tab
tabControl.pack(expand=1, fill="both")      # Pack to make visible

#======================
# Start GUI
#======================
win.mainloop()
```



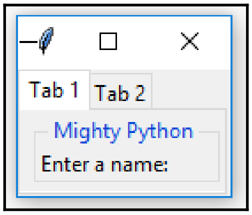


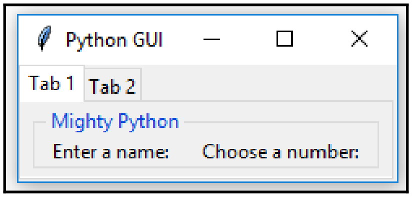
```
tabControl = ttk.Notebook(win)          # Create Tab Control

tab1 = ttk.Frame(tabControl)            # Create a tab
tabControl.add(tab1, text='Tab 1')      # Add the tab
tab2 = ttk.Frame(tabControl)            # Add a second tab
tabControl.add(tab2, text='Tab 2')      # Add second tab

tabControl.pack(expand=1, fill="both")  # Pack to make visible

#======================
# Start GUI
#======================
win.mainloop()
```

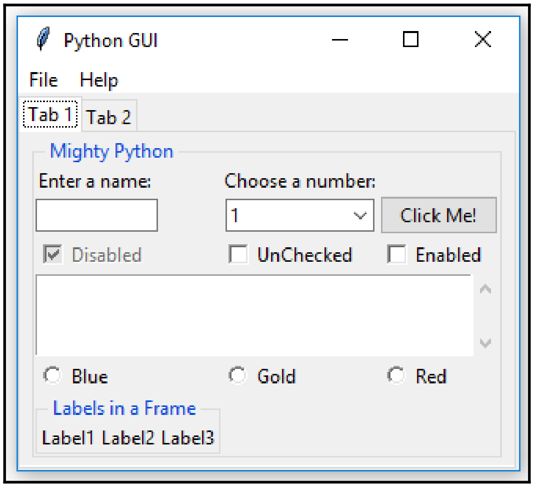


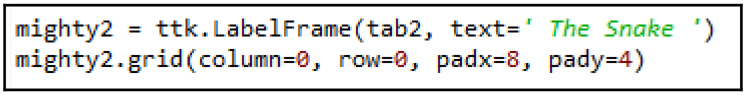


```
# LabelFrame using tab1 as the parent
mighty = ttk.LabelFrame(tab1, text=' Mighty Python ')
mighty.grid(column=0, row=0, padx=8, pady=4)

# Label using mighty as the parent
a_label = ttk.Label(mighty, text="Enter a name:")
a_label.grid(column=0, row=0, sticky='W')
```

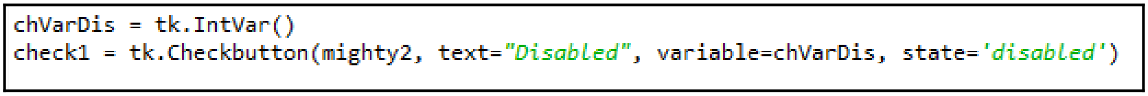




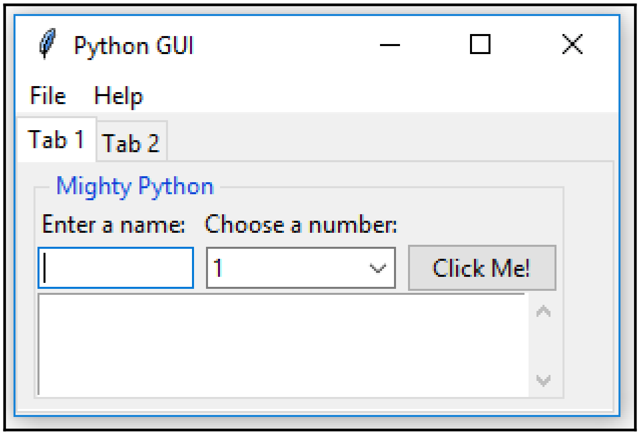


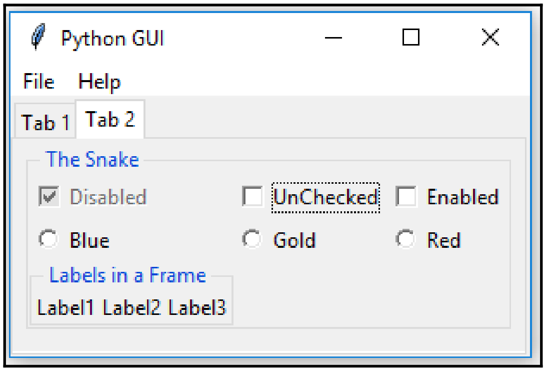


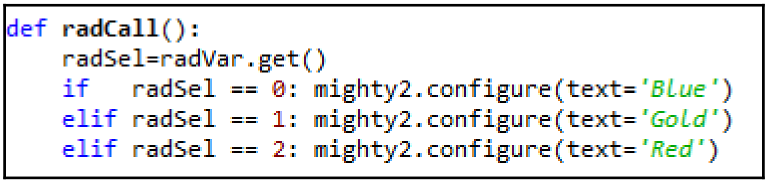
```
mighty2 = ttk.LabelFrame(tab2, text=' The Snake ')
mighty2.grid(column=0, row=0, padx=8, pady=4)
```



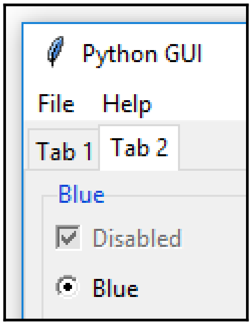
```
chVarDis = tk.IntVar()
check1 = tk.Checkbutton(mighty2, text="Disabled", variable=chVarDis, state='disabled')
```

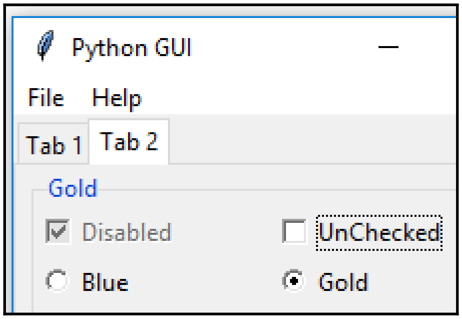


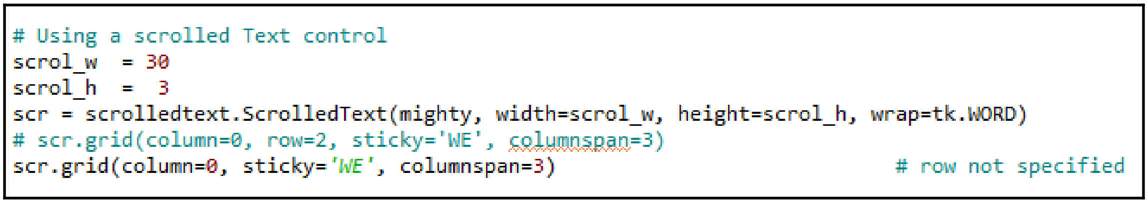




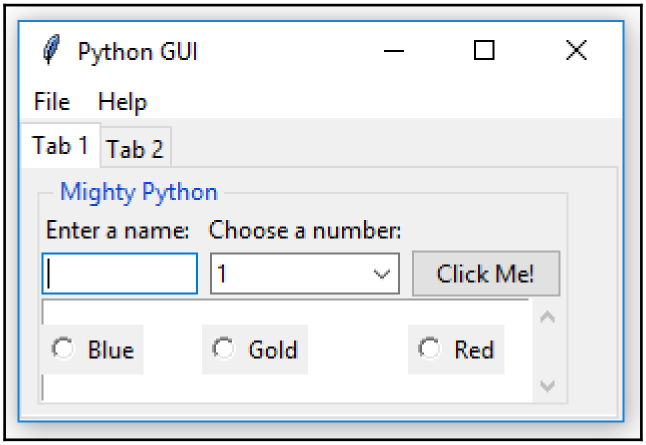
```
def radCall():
    radSel=radVar.get()
    if   radSel == 0: mighty2.configure(text='Blue')
    elif radSel == 1: mighty2.configure(text='Gold')
    elif radSel == 2: mighty2.configure(text='Red')
```

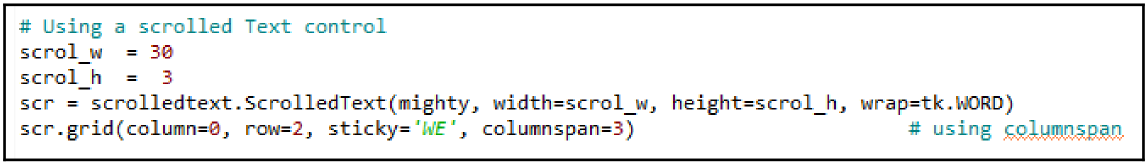






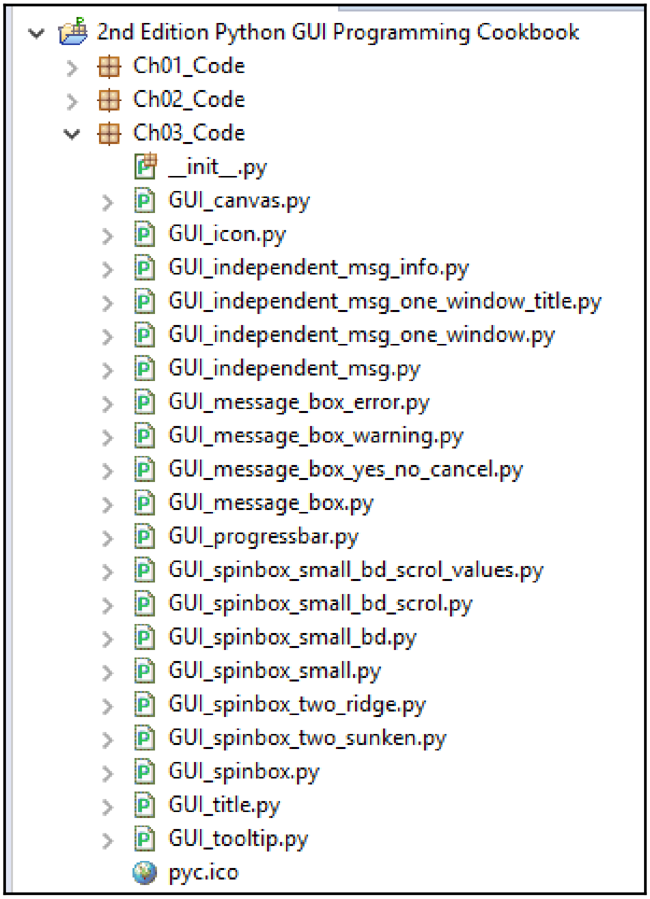
```
# Using a scrolled Text control
scrol_w  = 30
scrol_h  =  3
scr = scrolledtext.ScrolledText(mighty, width=scrol_w, height=scrol_h, wrap=tk.WORD)
# scr.grid(column=0, row=2, sticky='WE', columnspan=3)
scr.grid(column=0, sticky='WE', columnspan=3)                         # row not specified
```



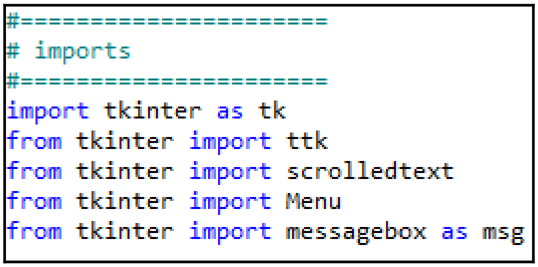


```
# Using a scrolled Text control
scrol_w  = 30
scrol_h  =  3
scr = scrolledtext.ScrolledText(mighty, width=scrol_w, height=scrol_h, wrap=tk.WORD)
scr.grid(column=0, row=2, sticky='WE', columnspan=3)                 # using columnspan
```

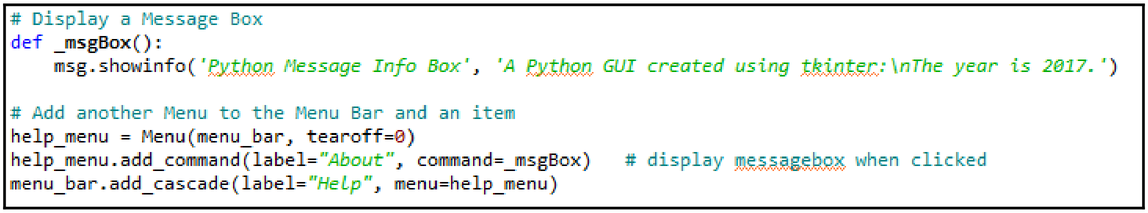
Chapter 3: Look and Feel Customization



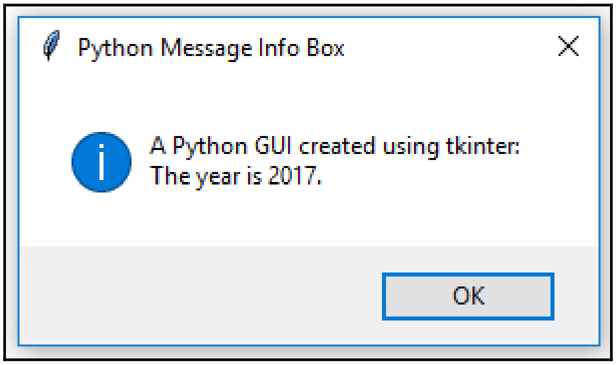
```
2nd Edition Python GUI Programming Cookbook
    Ch01_Code
    Ch02_Code
    Ch03_Code
        __init__.py
        GUI_canvas.py
        GUI_icon.py
        GUI_independent_msg_info.py
        GUI_independent_msg_one_window_title.py
        GUI_independent_msg_one_window.py
        GUI_independent_msg.py
        GUI_message_box_error.py
        GUI_message_box_warning.py
        GUI_message_box_yes_no_cancel.py
        GUI_message_box.py
        GUI_progressbar.py
        GUI_spinbox_small_bd_scrol_values.py
        GUI_spinbox_small_bd_scrol.py
        GUI_spinbox_small_bd.py
        GUI_spinbox_small.py
        GUI_spinbox_two_ridge.py
        GUI_spinbox_two_sunken.py
        GUI_spinbox.py
        GUI_title.py
        GUI_tooltip.py
        pyc.ico
```



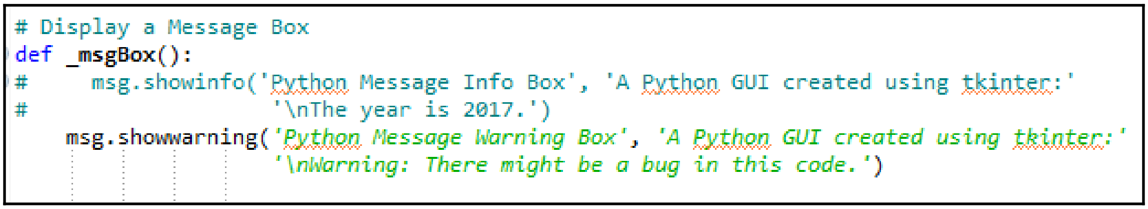
```python
#======================
# imports
#======================
import tkinter as tk
from tkinter import ttk
from tkinter import scrolledtext
from tkinter import Menu
from tkinter import messagebox as msg
```



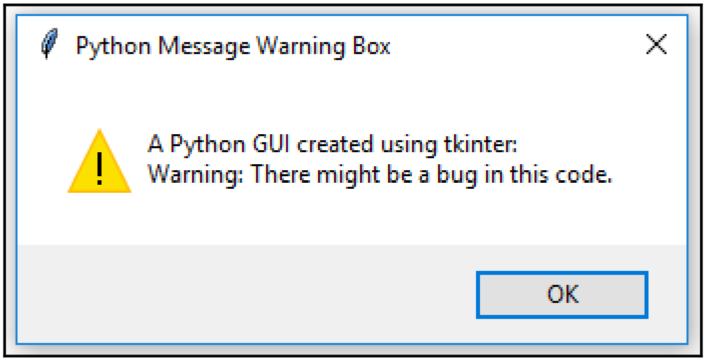
```
# Display a Message Box
def _msgBox():
    msg.showinfo('Python Message Info Box', 'A Python GUI created using tkinter:\nThe year is 2017.')

# Add another Menu to the Menu Bar and an item
help_menu = Menu(menu_bar, tearoff=0)
help_menu.add_command(label="About", command=_msgBox)   # display messagebox when clicked
menu_bar.add_cascade(label="Help", menu=help_menu)
```



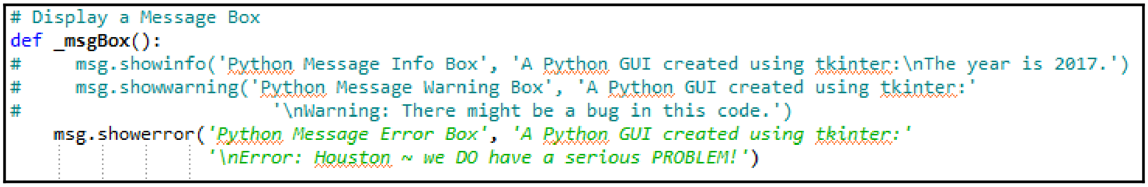
A Python GUI created using tkinter:
The year is 2017.



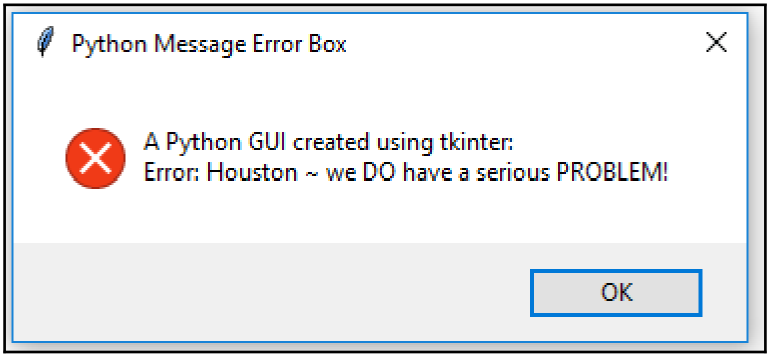
```
# Display a Message Box
def _msgBox():
#     msg.showinfo('Python Message Info Box', 'A Python GUI created using tkinter:'
#                  '\nThe year is 2017.')
    msg.showwarning('Python Message Warning Box', 'A Python GUI created using tkinter:'
                    '\nWarning: There might be a bug in this code.')
```



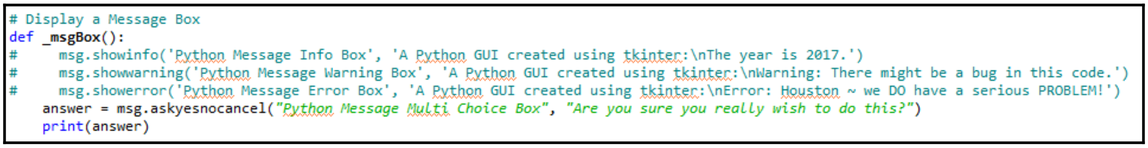
A Python GUI created using tkinter:
Warning: There might be a bug in this code.



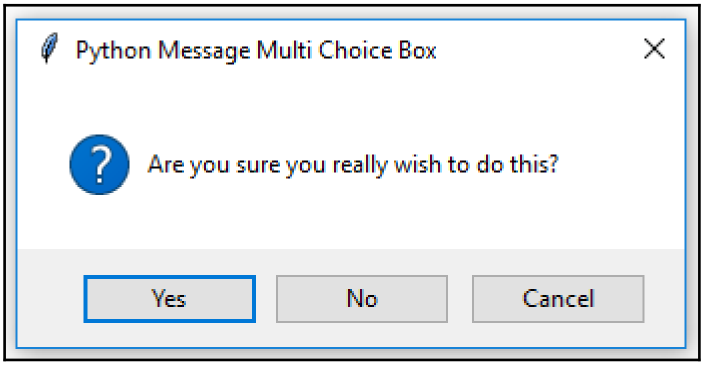
```
# Display a Message Box
def _msgBox():
#     msg.showinfo('Python Message Info Box', 'A Python GUI created using tkinter:\nThe year is 2017.')
#     msg.showwarning('Python Message Warning Box', 'A Python GUI created using tkinter:'
#                     '\nWarning: There might be a bug in this code.')
    msg.showerror('Python Message Error Box', 'A Python GUI created using tkinter:'
                  '\nError: Houston ~ we DO have a serious PROBLEM!')
```



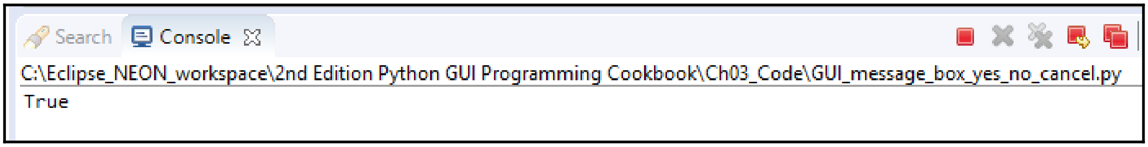
A Python GUI created using tkinter:
Error: Houston ~ we DO have a serious PROBLEM!



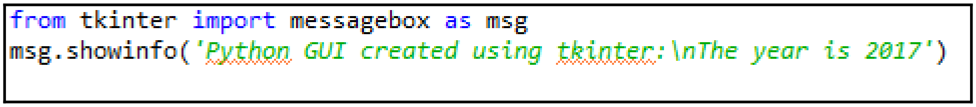
```
# Display a Message Box
def _msgBox():
#     msg.showinfo('Python Message Info Box', 'A Python GUI created using tkinter:\nThe year is 2017.')
#     msg.showwarning('Python Message Warning Box', 'A Python GUI created using tkinter:\nWarning: There might be a bug in this code.')
#     msg.showerror('Python Message Error Box', 'A Python GUI created using tkinter:\nError: Houston ~ we DO have a serious PROBLEM!')
    answer = msg.askyesnocancel("Python Message Multi Choice Box", "Are you sure you really wish to do this?")
    print(answer)
```



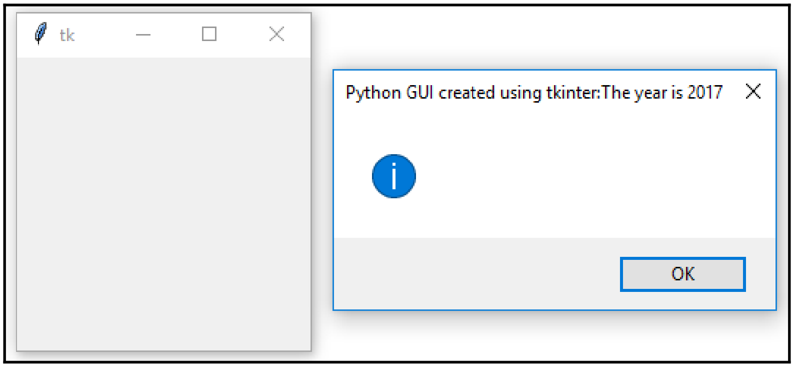
Are you sure you really wish to do this?



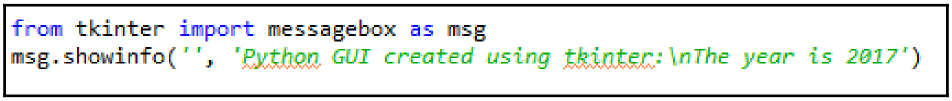
```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch03_Code\GUI_message_box_yes_no_cancel.py
True
```



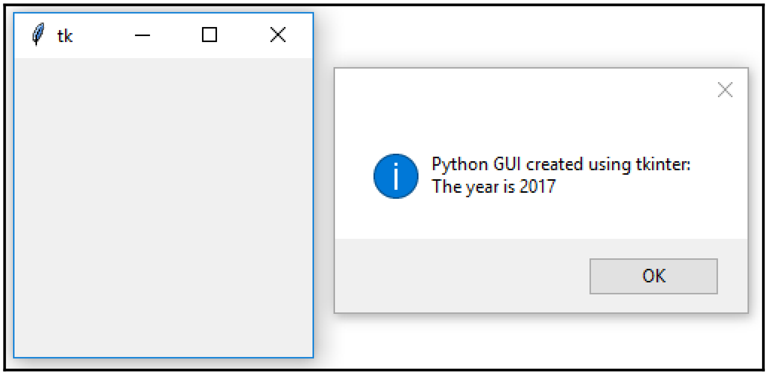
```
from tkinter import messagebox as msg
msg.showinfo('Python GUI created using tkinter:\nThe year is 2017')
```



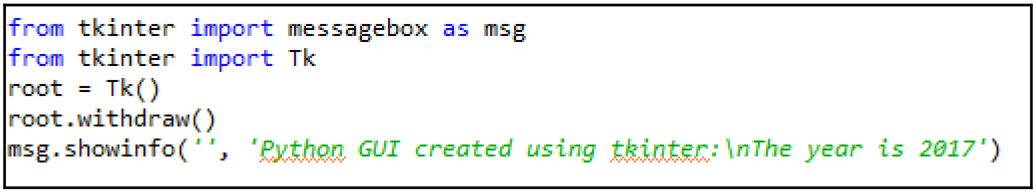
Python GUI created using tkinter:The year is 2017



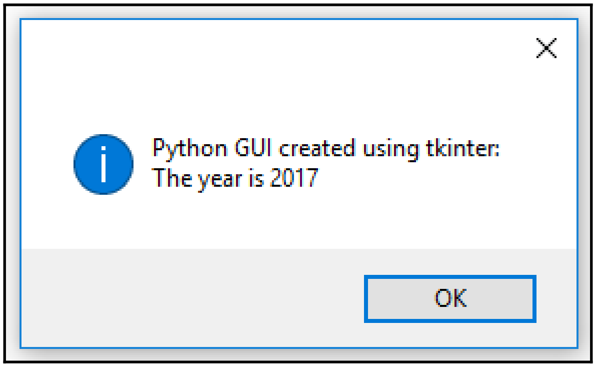
```
from tkinter import messagebox as msg
msg.showinfo('', 'Python GUI created using tkinter:\nThe year is 2017')
```



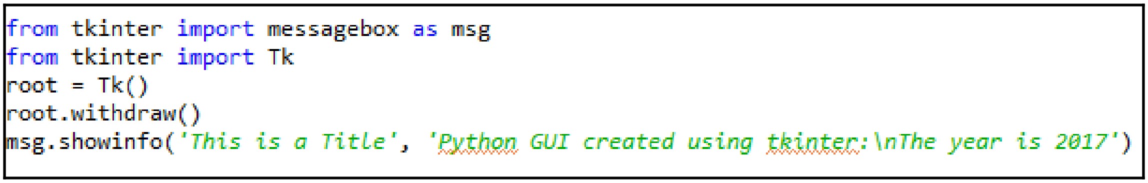
Python GUI created using tkinter:
The year is 2017



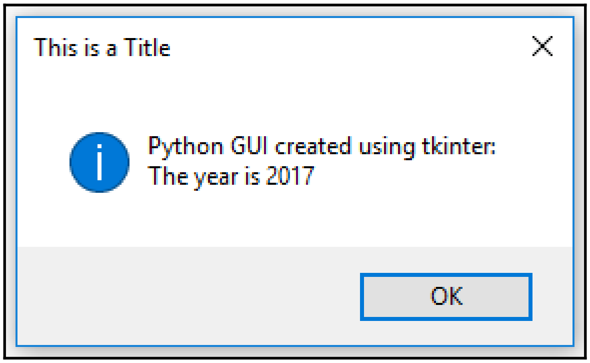
```
from tkinter import messagebox as msg
from tkinter import Tk
root = Tk()
root.withdraw()
msg.showinfo('', 'Python GUI created using tkinter:\nThe year is 2017')
```



Python GUI created using tkinter:
The year is 2017

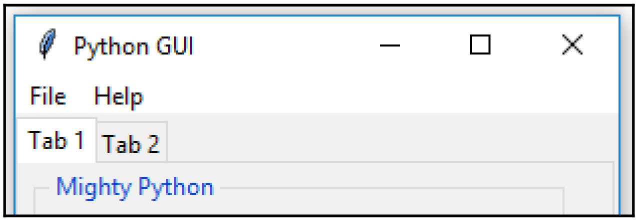


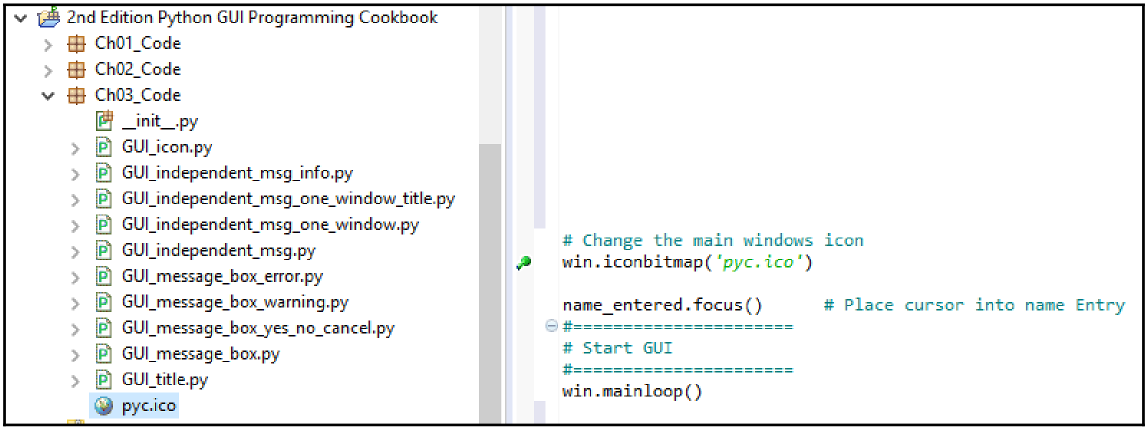
```
from tkinter import messagebox as msg
from tkinter import Tk
root = Tk()
root.withdraw()
msg.showinfo('This is a Title', 'Python GUI created using tkinter:\nThe year is 2017')
```



This is a Title

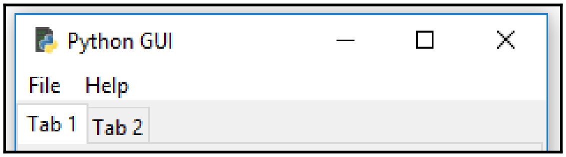
Python GUI created using tkinter:
The year is 2017

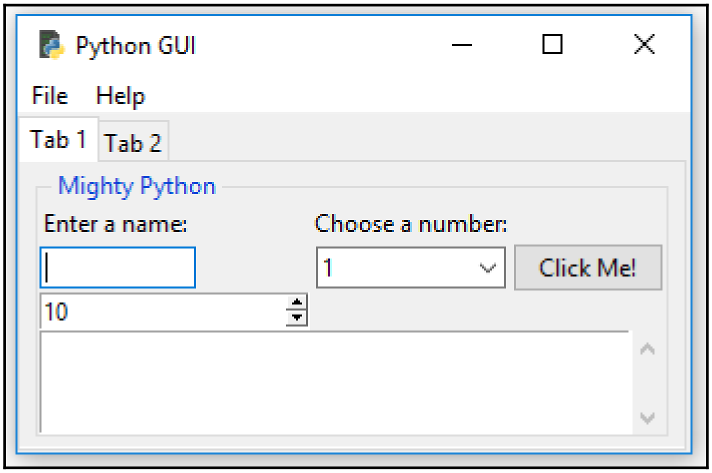


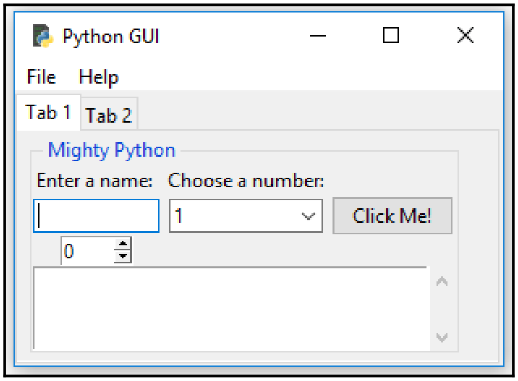


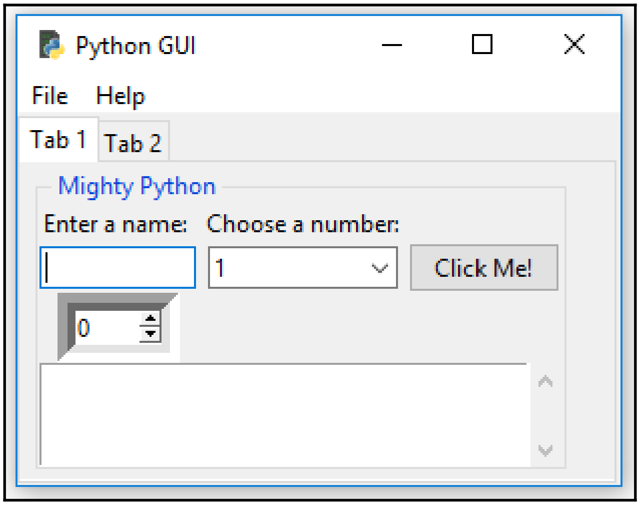
```
2nd Edition Python GUI Programming Cookbook
  Ch01_Code
  Ch02_Code
  Ch03_Code
    __init__.py
    GUI_icon.py
    GUI_independent_msg_info.py
    GUI_independent_msg_one_window_title.py
    GUI_independent_msg_one_window.py
    GUI_independent_msg.py
    GUI_message_box_error.py
    GUI_message_box_warning.py
    GUI_message_box_yes_no_cancel.py
    GUI_message_box.py
    GUI_title.py
    pyc.ico
```

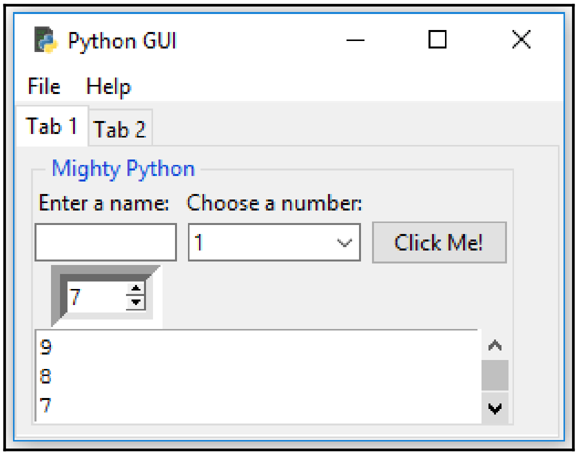
```
# Change the main windows icon
win.iconbitmap('pyc.ico')

name_entered.focus()      # Place cursor into name Entry
#======================
# Start GUI
#======================
win.mainloop()
```



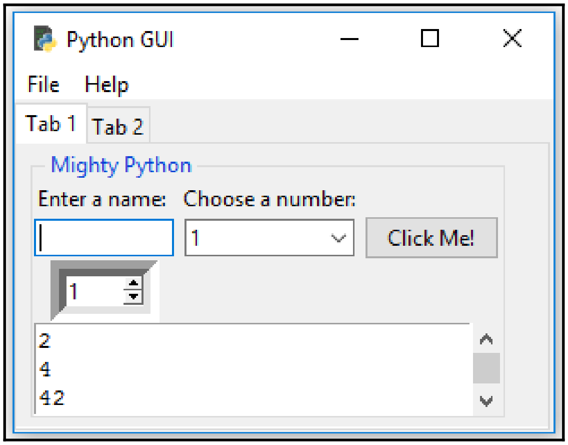



Python GUI
File
Help
Tab 1
Tab 2
Mighty Python
Enter a name:
Choose a number:
1
Click Me!
0


Python GUI
File
Help
Tab 1
Tab 2
Mighty Python
Enter a name:
Choose a number:
1
Click Me!
0


Python GUI
File
Help
Tab 1
Tab 2
Mighty Python
Enter a name:
Choose a number:
1
Click Me!
7

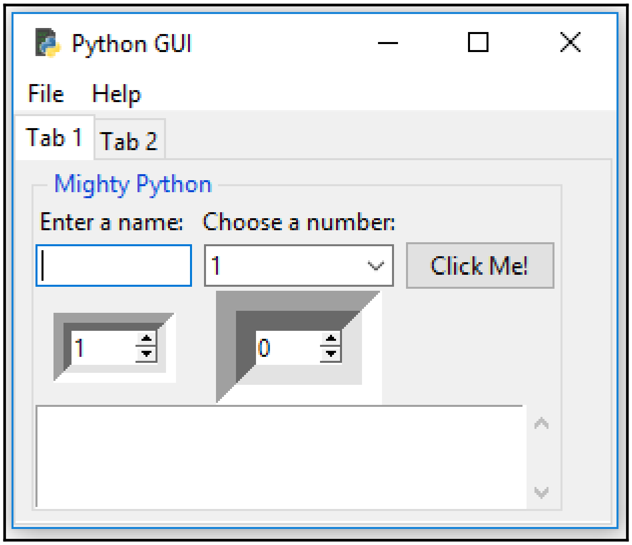
```
9
8
7
```


Python GUI
File
Help
Tab 1
Tab 2
Mighty Python
Enter a name:
Choose a number:
Click Me!

1

1

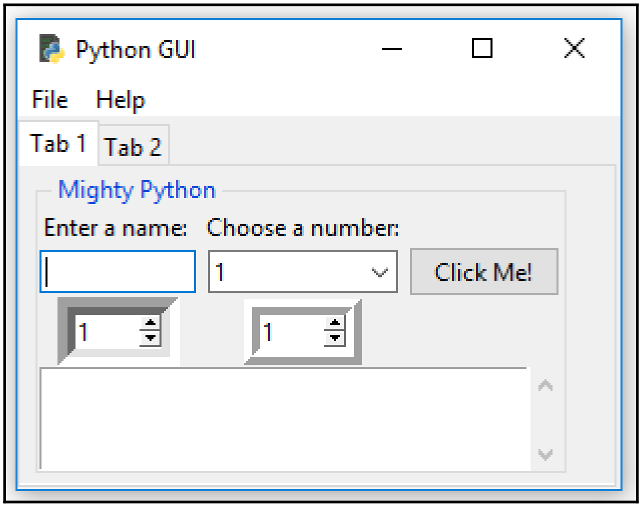
2
4
42

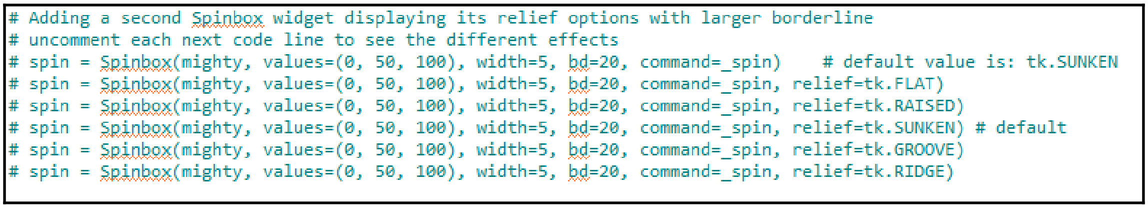

Python GUI
File
Help
Tab 1
Tab 2
Mighty Python
Enter a name:
Choose a number:
Click Me!

1

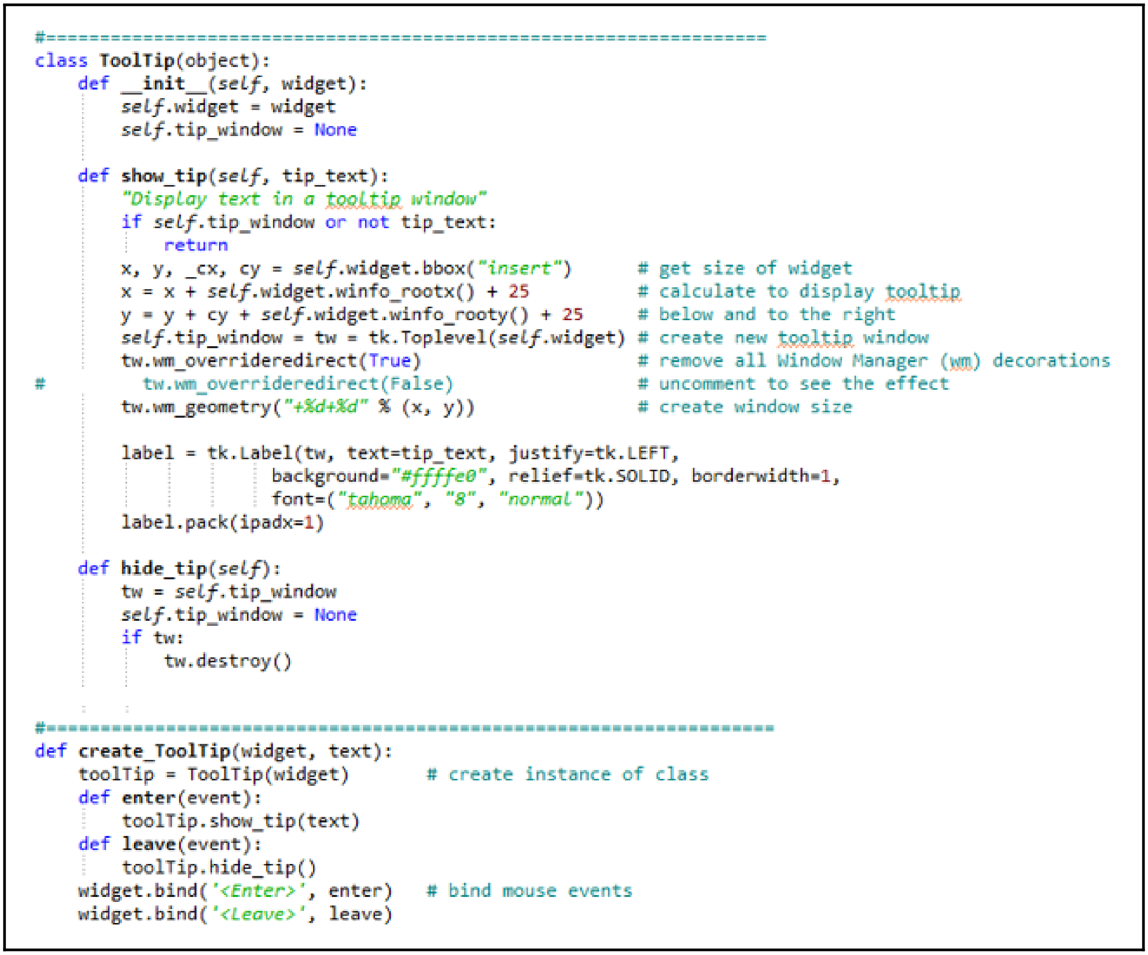
1

0

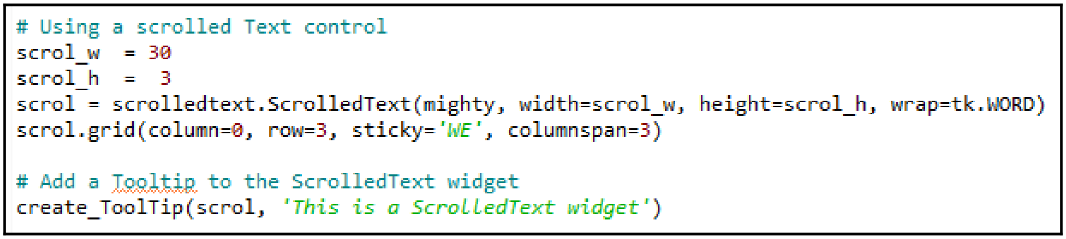




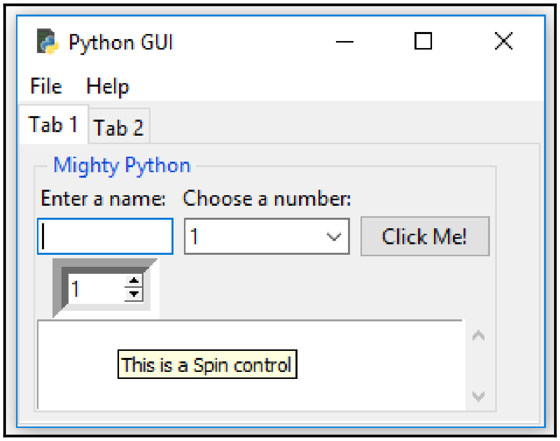
```
# Adding a second Spinbox widget displaying its relief options with larger borderline
# uncomment each next code line to see the different effects
# spin = Spinbox(mighty, values=(0, 50, 100), width=5, bd=20, command=_spin)    # default value is: tk.SUNKEN
# spin = Spinbox(mighty, values=(0, 50, 100), width=5, bd=20, command=_spin, relief=tk.FLAT)
# spin = Spinbox(mighty, values=(0, 50, 100), width=5, bd=20, command=_spin, relief=tk.RAISED)
# spin = Spinbox(mighty, values=(0, 50, 100), width=5, bd=20, command=_spin, relief=tk.SUNKEN) # default
# spin = Spinbox(mighty, values=(0, 50, 100), width=5, bd=20, command=_spin, relief=tk.GROOVE)
# spin = Spinbox(mighty, values=(0, 50, 100), width=5, bd=20, command=_spin, relief=tk.RIDGE)
```

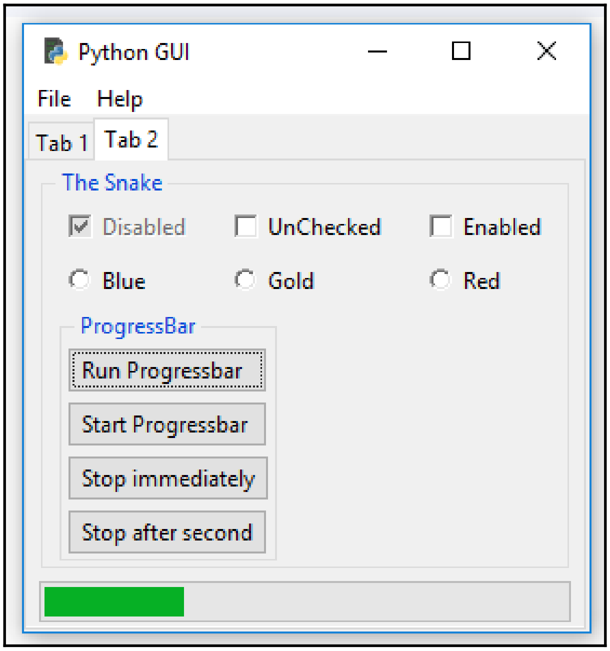


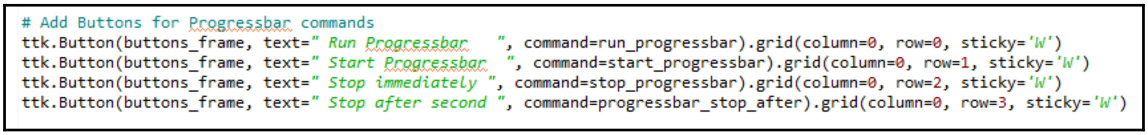
```
#===================================================================
class ToolTip(object):
    def __init__(self, widget):
        self.widget = widget
        self.tip_window = None

    def show_tip(self, tip_text):
        "Display text in a tooltip window"
        if self.tip_window or not tip_text:
            return
        x, y, _cx, cy = self.widget.bbox("insert")      # get size of widget
        x = x + self.widget.winfo_rootx() + 25          # calculate to display tooltip
        y = y + cy + self.widget.winfo_rooty() + 25     # below and to the right
        self.tip_window = tw = tk.Toplevel(self.widget) # create new tooltip window
        tw.wm_overrideredirect(True)                    # remove all Window Manager (wm) decorations
#         tw.wm_overrideredirect(False)                 # uncomment to see the effect
        tw.wm_geometry("+%d+%d" % (x, y))               # create window size

        label = tk.Label(tw, text=tip_text, justify=tk.LEFT,
                      background="#ffffe0", relief=tk.SOLID, borderwidth=1,
                      font=("tahoma", "8", "normal"))
        label.pack(ipadx=1)

    def hide_tip(self):
        tw = self.tip_window
        self.tip_window = None
        if tw:
            tw.destroy()


#===================================================================
def create_ToolTip(widget, text):
    toolTip = ToolTip(widget)       # create instance of class
    def enter(event):
        toolTip.show_tip(text)
    def leave(event):
        toolTip.hide_tip()
    widget.bind('<Enter>', enter)   # bind mouse events
    widget.bind('<Leave>', leave)
```



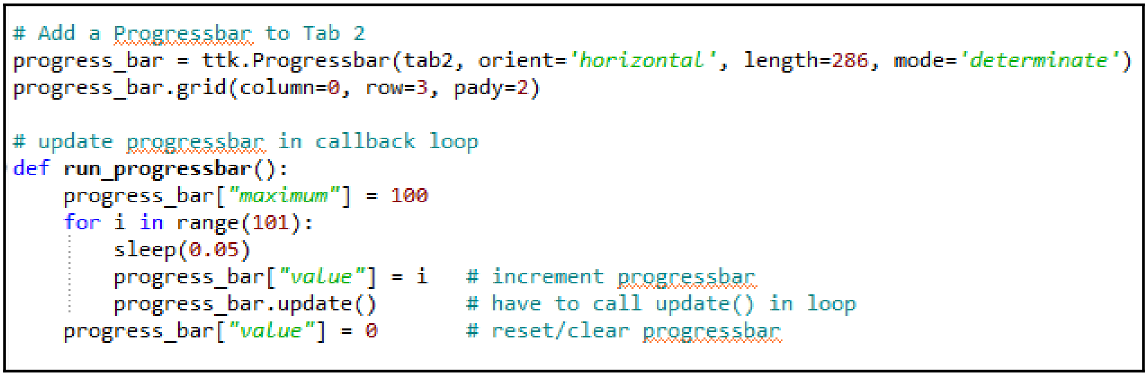
```
# Using a scrolled Text control
scrol_w  = 30
scrol_h  =  3
scrol = scrolledtext.ScrolledText(mighty, width=scrol_w, height=scrol_h, wrap=tk.WORD)
scrol.grid(column=0, row=3, sticky='WE', columnspan=3)

# Add a Tooltip to the ScrolledText widget
create_ToolTip(scrol, 'This is a ScrolledText widget')
```



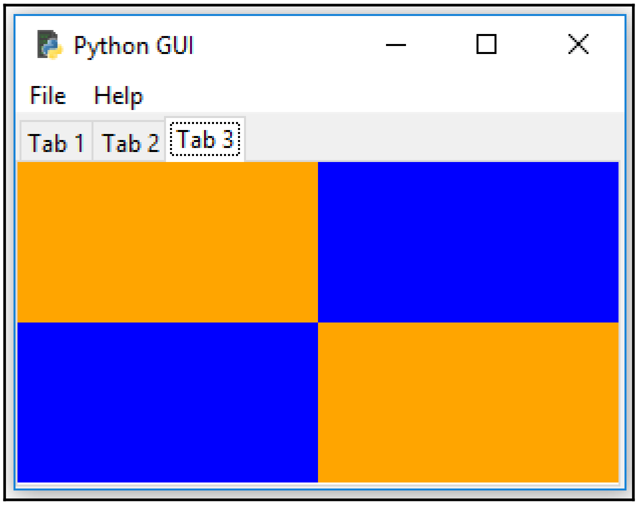




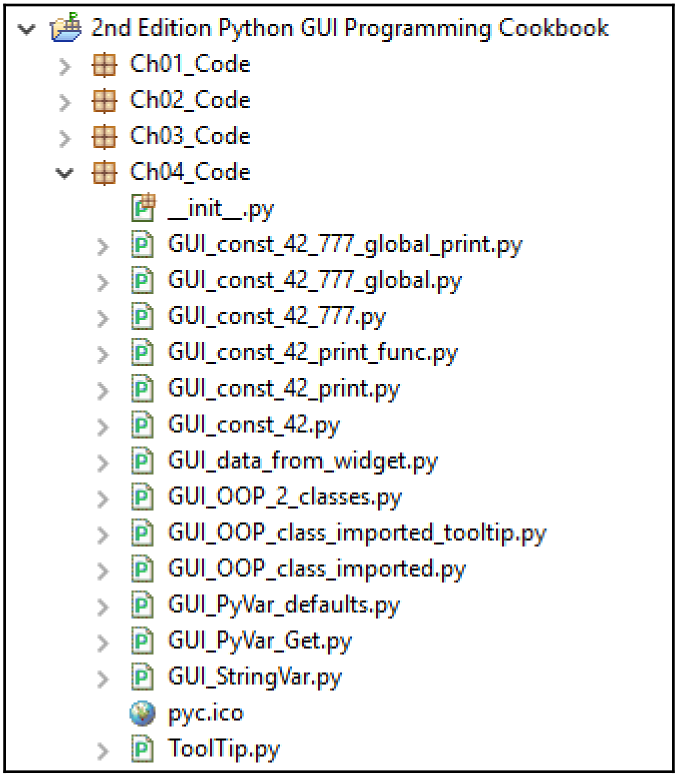
```
# Add Buttons for Progressbar commands
ttk.Button(buttons_frame, text=" Run Progressbar   ", command=run_progressbar).grid(column=0, row=0, sticky='W')
ttk.Button(buttons_frame, text=" Start Progressbar  ", command=start_progressbar).grid(column=0, row=1, sticky='W')
ttk.Button(buttons_frame, text=" Stop immediately ", command=stop_progressbar).grid(column=0, row=2, sticky='W')
ttk.Button(buttons_frame, text=" Stop after second ", command=progressbar_stop_after).grid(column=0, row=3, sticky='W')
```



```python
# Add a Progressbar to Tab 2
progress_bar = ttk.Progressbar(tab2, orient='horizontal', length=286, mode='determinate')
progress_bar.grid(column=0, row=3, pady=2)

# update progressbar in callback loop
def run_progressbar():
    progress_bar["maximum"] = 100
    for i in range(101):
        sleep(0.05)
        progress_bar["value"] = i   # increment progressbar
        progress_bar.update()       # have to call update() in loop
    progress_bar["value"] = 0       # reset/clear progressbar
```

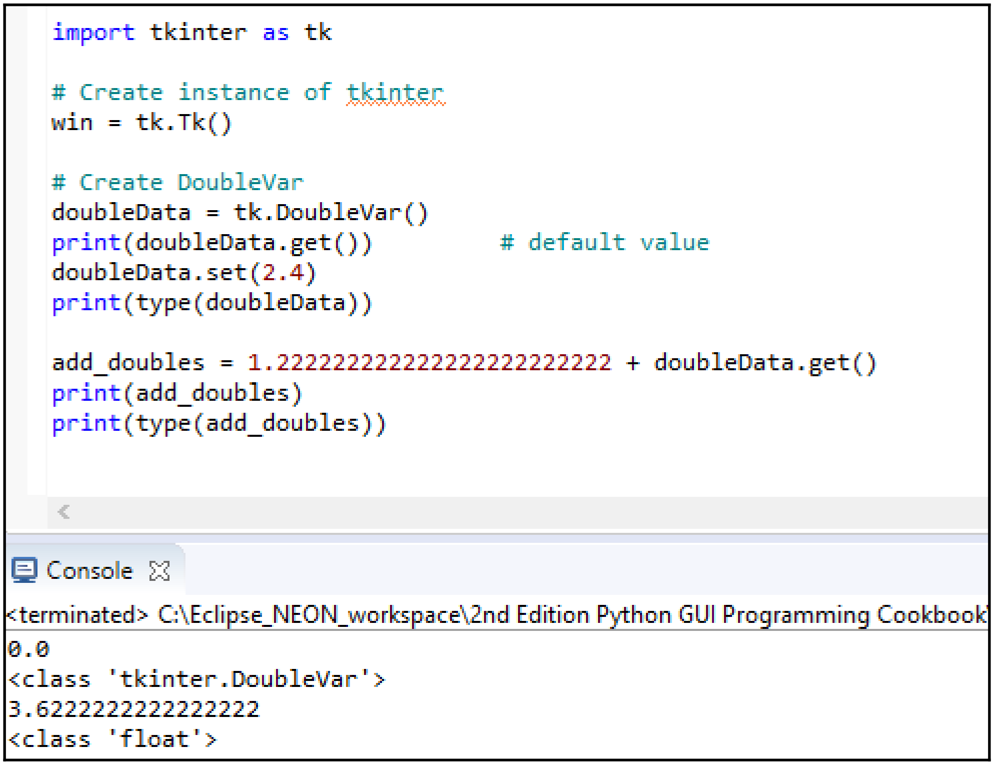


Chapter 4: Data and Classes



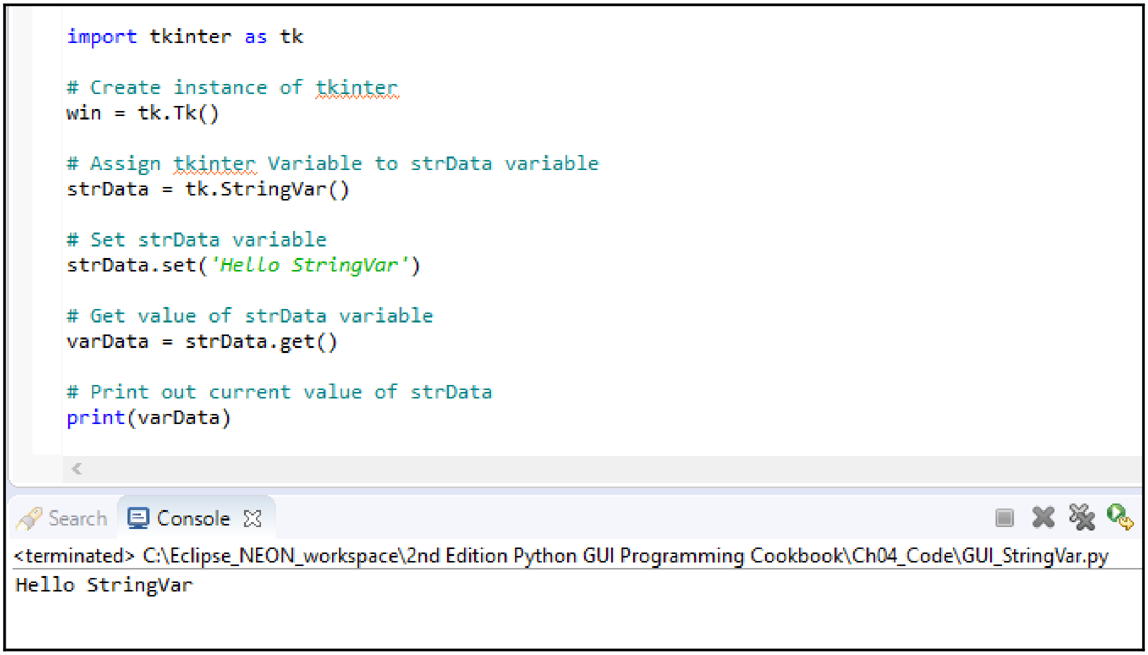
2nd Edition Python GUI Programming Cookbook

- Ch01_Code
- Ch02_Code
- Ch03_Code
- Ch04_Code
  - __init__.py
  - GUI_const_42_777_global_print.py
  - GUI_const_42_777_global.py
  - GUI_const_42_777.py
  - GUI_const_42_print_func.py
  - GUI_const_42_print.py
  - GUI_const_42.py
  - GUI_data_from_widget.py
  - GUI_OOP_2_classes.py
  - GUI_OOP_class_imported_tooltip.py
  - GUI_OOP_class_imported.py
  - GUI_PyVar_defaults.py
  - GUI_PyVar_Get.py
  - GUI_StringVar.py
  - pyc.ico
  - ToolTip.py



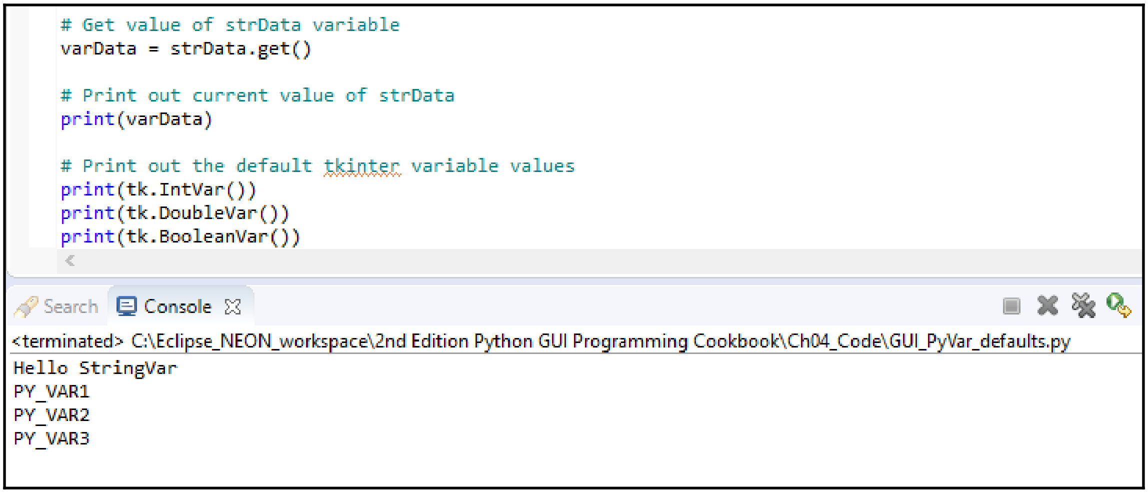
```
import tkinter as tk

# Create instance of tkinter
win = tk.Tk()

# Create DoubleVar
doubleData = tk.DoubleVar()
print(doubleData.get())         # default value
doubleData.set(2.4)
print(type(doubleData))

add_doubles = 1.22222222222222222222222 + doubleData.get()
print(add_doubles)
print(type(add_doubles))
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook
0.0
<class 'tkinter.DoubleVar'>
3.6222222222222222
<class 'float'>
```



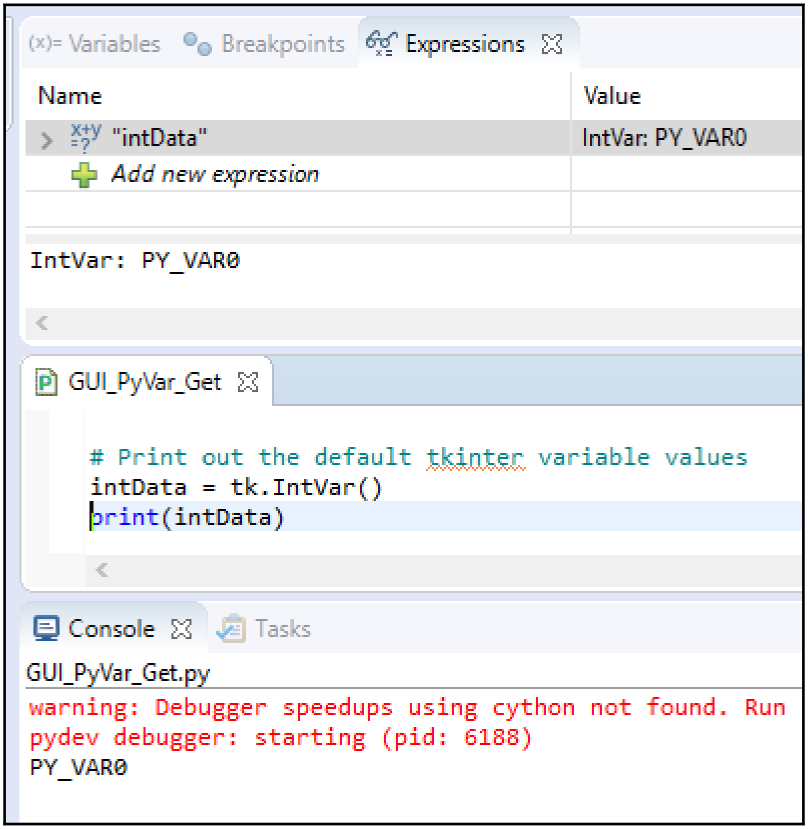
```python
import tkinter as tk

# Create instance of tkinter
win = tk.Tk()

# Assign tkinter Variable to strData variable
strData = tk.StringVar()

# Set strData variable
strData.set('Hello StringVar')

# Get value of strData variable
varData = strData.get()

# Print out current value of strData
print(varData)
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch04_Code\GUI_StringVar.py
Hello StringVar
```



```python
# Get value of strData variable
varData = strData.get()

# Print out current value of strData
print(varData)

# Print out the default tkinter variable values
print(tk.IntVar())
print(tk.DoubleVar())
print(tk.BooleanVar())
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch04_Code\GUI_PyVar_defaults.py
Hello StringVar
PY_VAR1
PY_VAR2
PY_VAR3
```

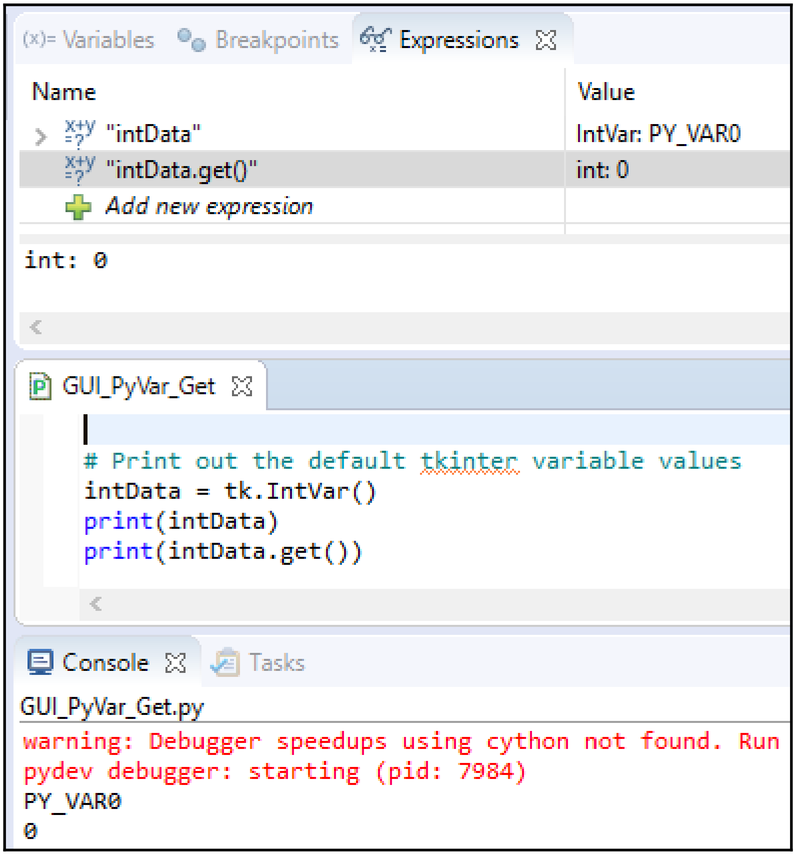


```
IntVar: PY_VAR0
```

```
# Print out the default tkinter variable values
intData = tk.IntVar()
print(intData)
```

GUI_PyVar_Get.py

```
warning: Debugger speedups using cython not found. Run
pydev debugger: starting (pid: 6188)
PY_VAR0
```



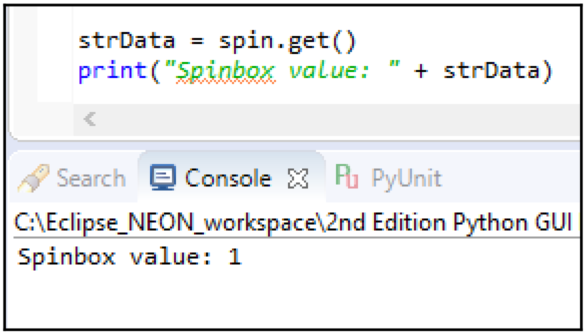
Name	Value
"intData"	IntVar: PY_VAR0
"intData.get()"	int: 0

```
int: 0
```

```
# Print out the default tkinter variable values
intData = tk.IntVar()
print(intData)
print(intData.get())
```

GUI_PyVar_Get.py

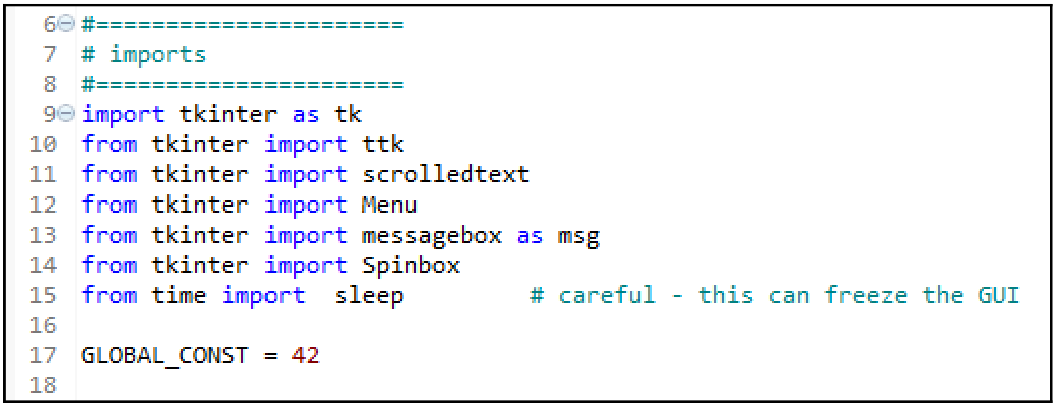
```
warning: Debugger speedups using cython not found. Run
pydev debugger: starting (pid: 7984)
PY_VAR0
0
```



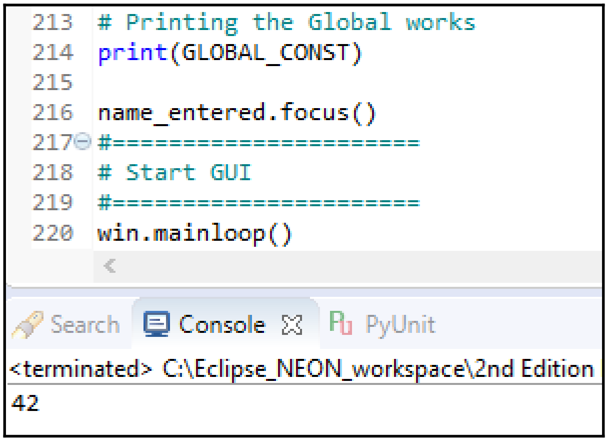
```
strData = spin.get()
print("Spinbox value: " + strData)
```

C:\Eclipse_NEON_workspace\2nd Edition Python GUI

```
Spinbox value: 1
```

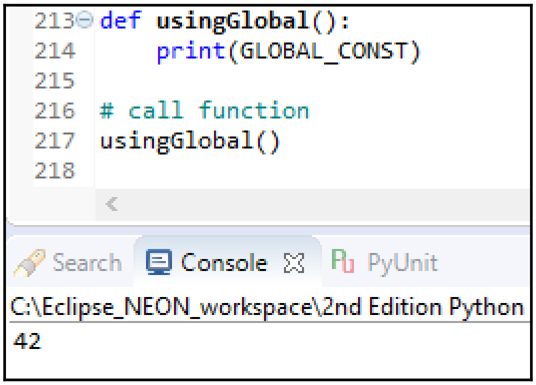


```python
#======================
# imports
#======================
import tkinter as tk
from tkinter import ttk
from tkinter import scrolledtext
from tkinter import Menu
from tkinter import messagebox as msg
from tkinter import Spinbox
from time import  sleep         # careful - this can freeze the GUI

GLOBAL_CONST = 42

```



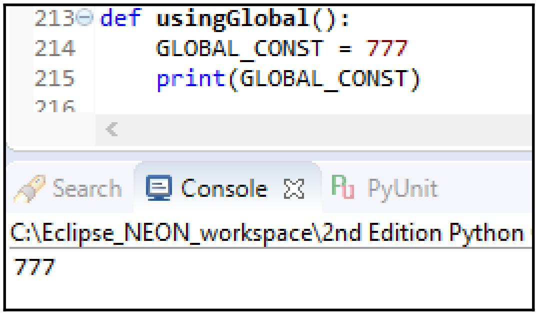
```python
# Printing the Global works
print(GLOBAL_CONST)

name_entered.focus()
#======================
# Start GUI
#======================
win.mainloop()
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition
42
```



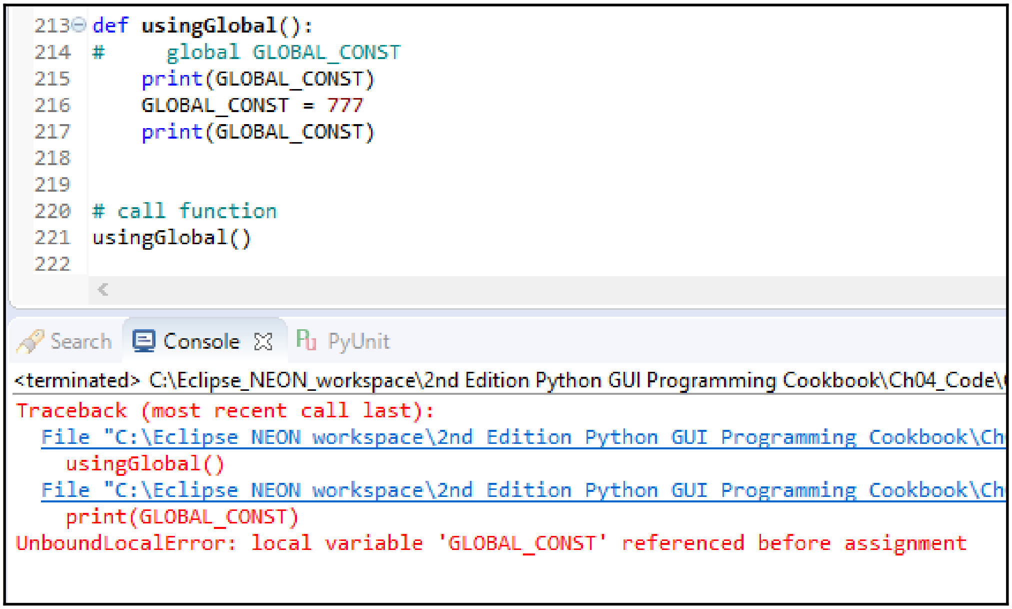
```python
def usingGlobal():
    print(GLOBAL_CONST)

# call function
usingGlobal()

```

```
C:\Eclipse_NEON_workspace\2nd Edition Python
42
```



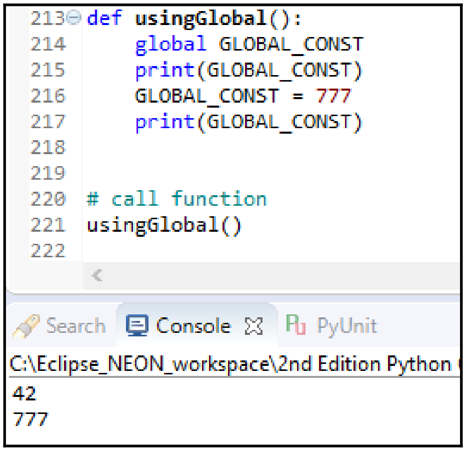
```
def usingGlobal():
    GLOBAL_CONST = 777
    print(GLOBAL_CONST)
```

```
C:\Eclipse_NEON_workspace\2nd Edition Python
777
```



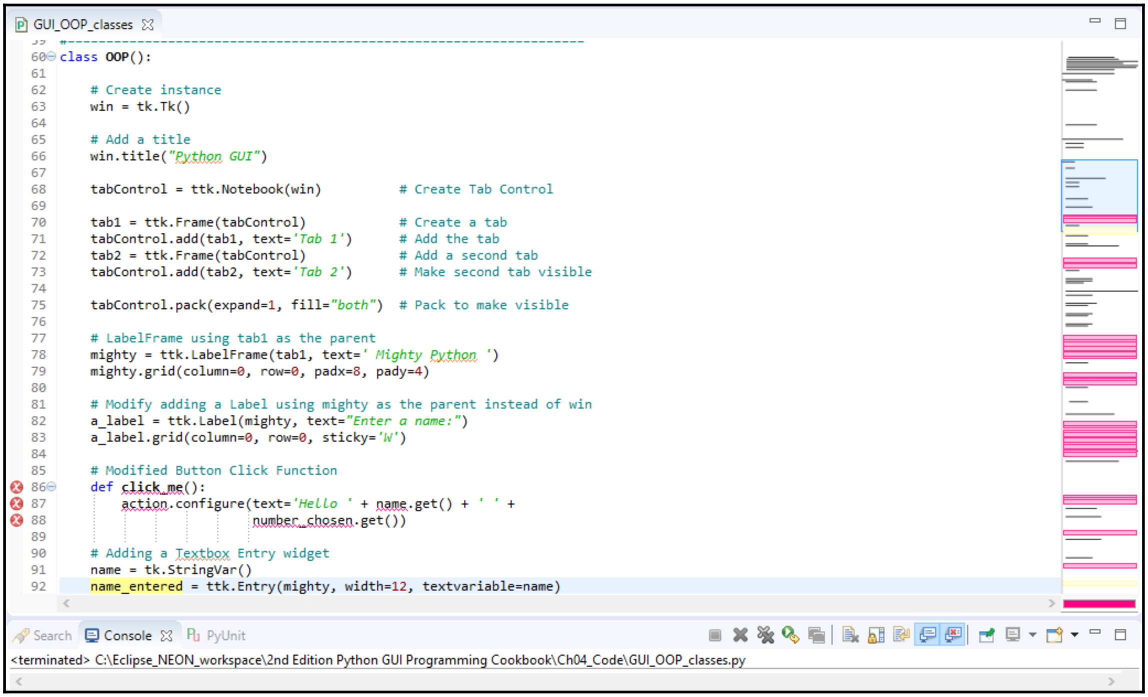
```
def usingGlobal():
#     global GLOBAL_CONST
    print(GLOBAL_CONST)
    GLOBAL_CONST = 777
    print(GLOBAL_CONST)


# call function
usingGlobal()
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch04_Code\
Traceback (most recent call last):
  File "C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch
    usingGlobal()
  File "C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch
    print(GLOBAL_CONST)
UnboundLocalError: local variable 'GLOBAL_CONST' referenced before assignment
```



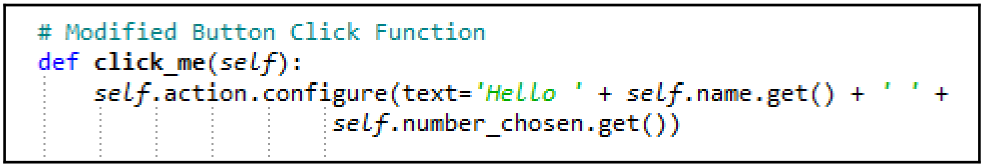
```
def usingGlobal():
    global GLOBAL_CONST
    print(GLOBAL_CONST)
    GLOBAL_CONST = 777
    print(GLOBAL_CONST)


# call function
usingGlobal()
```

```
C:\Eclipse_NEON_workspace\2nd Edition Python
42
777
```

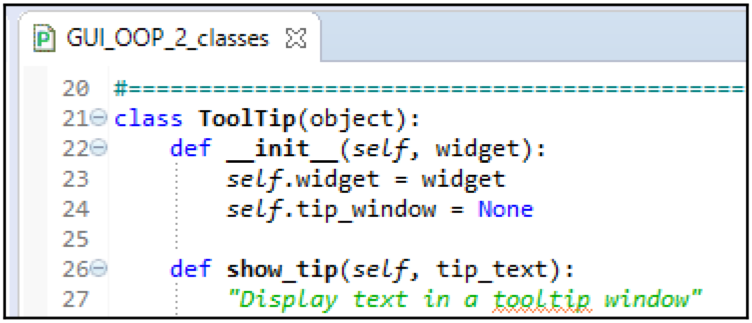


```
#----------------------------------------------------------------------
class OOP():

    # Create instance
    win = tk.Tk()

    # Add a title
    win.title("Python GUI")

    tabControl = ttk.Notebook(win)          # Create Tab Control

    tab1 = ttk.Frame(tabControl)            # Create a tab
    tabControl.add(tab1, text='Tab 1')      # Add the tab
    tab2 = ttk.Frame(tabControl)            # Add a second tab
    tabControl.add(tab2, text='Tab 2')      # Make second tab visible

    tabControl.pack(expand=1, fill="both")  # Pack to make visible

    # LabelFrame using tab1 as the parent
    mighty = ttk.LabelFrame(tab1, text=' Mighty Python ')
    mighty.grid(column=0, row=0, padx=8, pady=4)

    # Modify adding a Label using mighty as the parent instead of win
    a_label = ttk.Label(mighty, text="Enter a name:")
    a_label.grid(column=0, row=0, sticky='W')

    # Modified Button Click Function
    def click_me():
        action.configure(text='Hello ' + name.get() + ' ' +
                                 number_chosen.get())

    # Adding a Textbox Entry widget
    name = tk.StringVar()
    name_entered = ttk.Entry(mighty, width=12, textvariable=name)
```

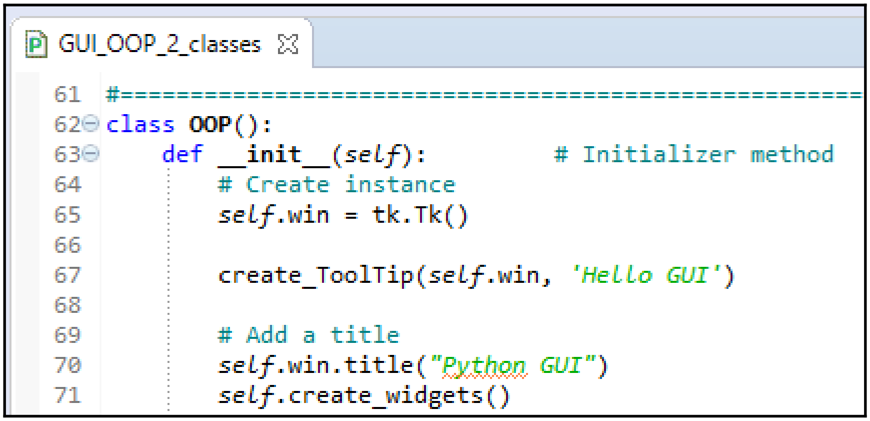
```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch04_Code\GUI_OOP_classes.py
```



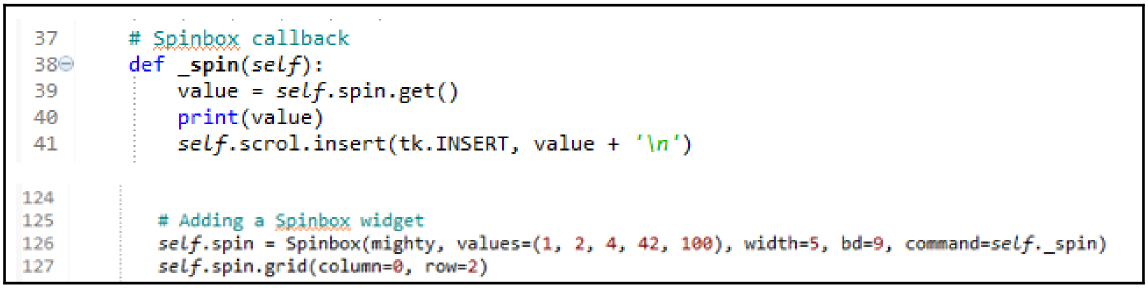
```
# Modified Button Click Function
def click_me(self):
    self.action.configure(text='Hello ' + self.name.get() + ' ' +
                               self.number_chosen.get())
```



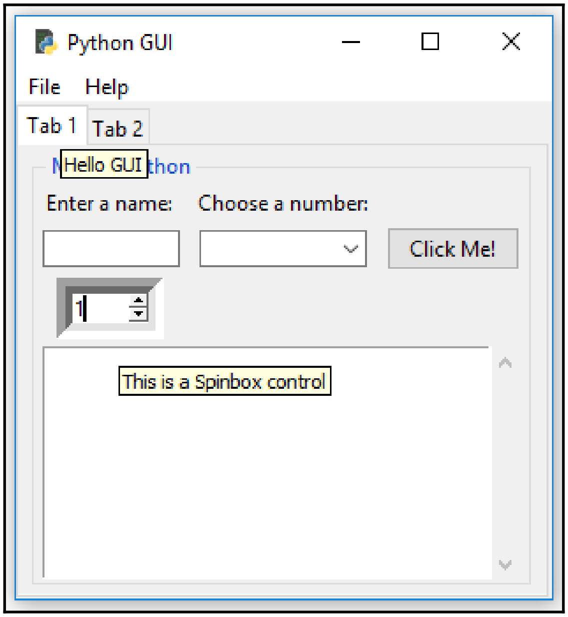
```
#===========================================
class ToolTip(object):
    def __init__(self, widget):
        self.widget = widget
        self.tip_window = None

    def show_tip(self, tip_text):
        "Display text in a tooltip window"
```

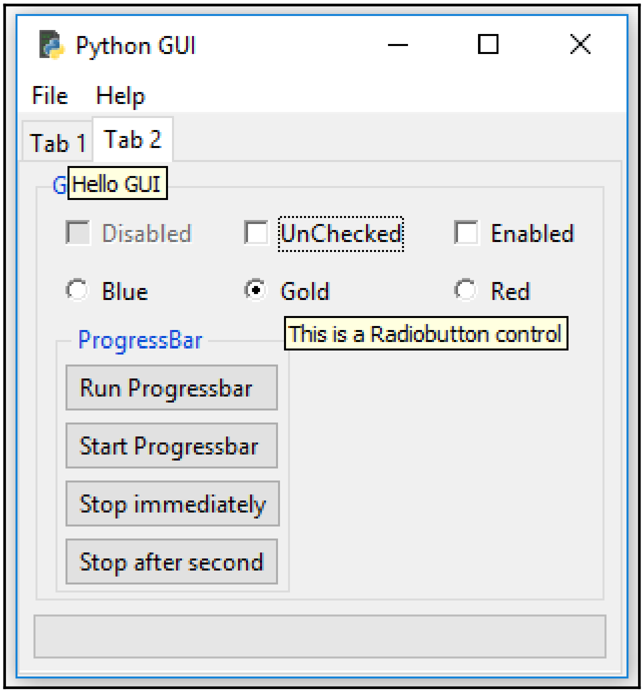


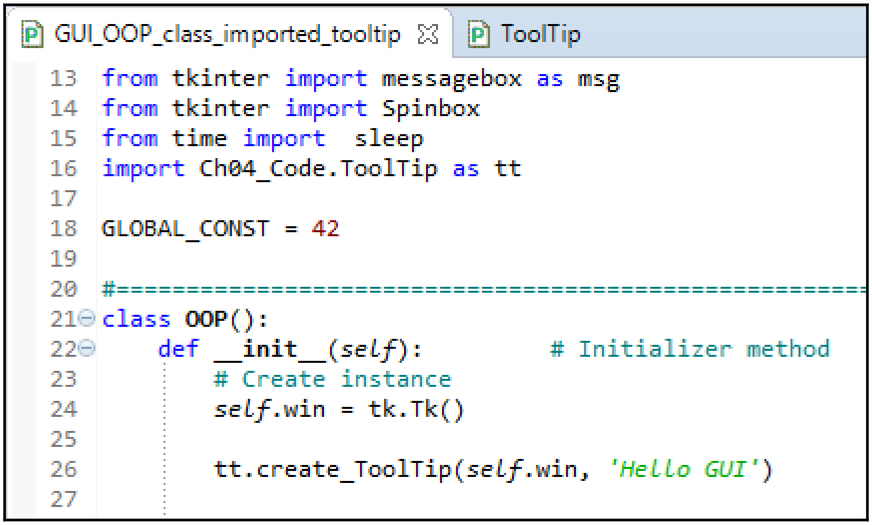
```
#======================================================
class OOP():
    def __init__(self):          # Initializer method
        # Create instance
        self.win = tk.Tk()

        create_ToolTip(self.win, 'Hello GUI')

        # Add a title
        self.win.title("Python GUI")
        self.create_widgets()
```



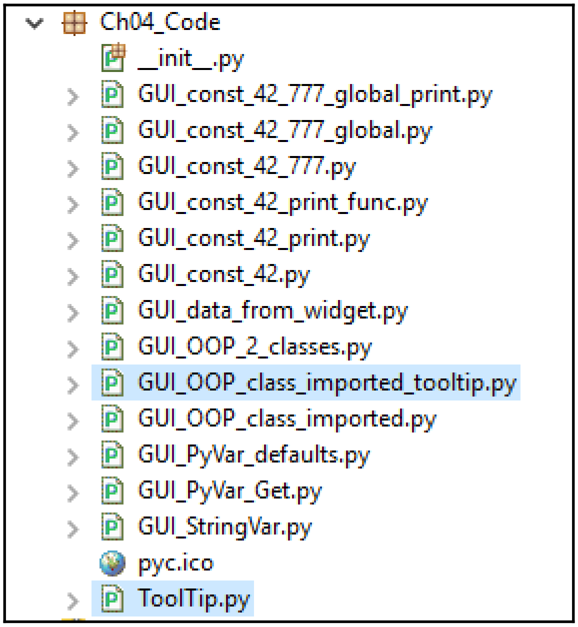
```
    # Spinbox callback
    def _spin(self):
        value = self.spin.get()
        print(value)
        self.scrol.insert(tk.INSERT, value + '\n')

        # Adding a Spinbox widget
        self.spin = Spinbox(mighty, values=(1, 2, 4, 42, 100), width=5, bd=9, command=self._spin)
        self.spin.grid(column=0, row=2)
```


Python GUI
File
Help
Tab 1
Tab 2
Hello GUI
thon
Enter a name:
Choose a number:
Click Me!
1
This is a Spinbox control

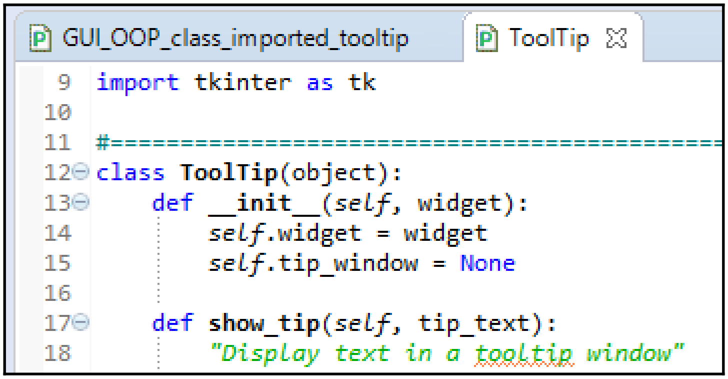

Python GUI
File
Help
Tab 1
Tab 2
G
Hello GUI
□ Disabled
□ UnChecked
□ Enabled
○ Blue
● Gold
○ Red
ProgressBar
This is a Radiobutton control
Run Progressbar
Start Progressbar
Stop immediately
Stop after second



```
from tkinter import messagebox as msg
from tkinter import Spinbox
from time import  sleep
import Ch04_Code.ToolTip as tt

GLOBAL_CONST = 42

#=====================================================
class OOP():
    def __init__(self):         # Initializer method
        # Create instance
        self.win = tk.Tk()

        tt.create_ToolTip(self.win, 'Hello GUI')

```

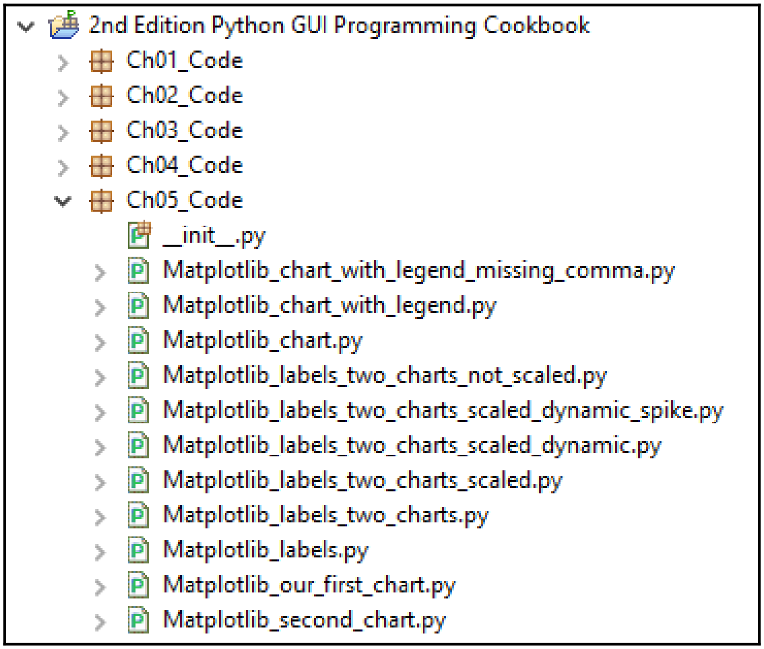


- Ch04_Code
  - __init__.py
  - GUI_const_42_777_global_print.py
  - GUI_const_42_777_global.py
  - GUI_const_42_777.py
  - GUI_const_42_print_func.py
  - GUI_const_42_print.py
  - GUI_const_42.py
  - GUI_data_from_widget.py
  - GUI_OOP_2_classes.py
  - GUI_OOP_class_imported_tooltip.py
  - GUI_OOP_class_imported.py
  - GUI_PyVar_defaults.py
  - GUI_PyVar_Get.py
  - GUI_StringVar.py
  - pyc.ico
  - ToolTip.py

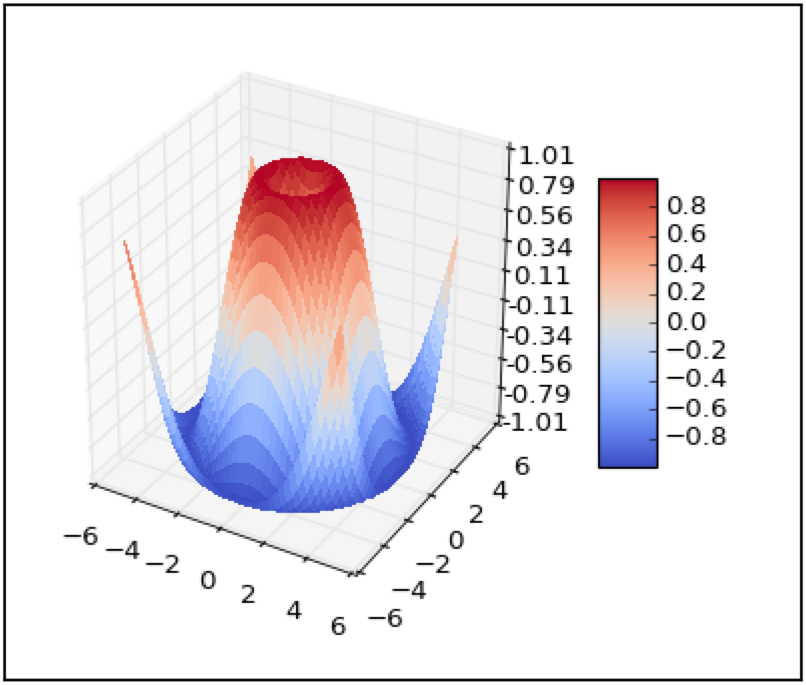


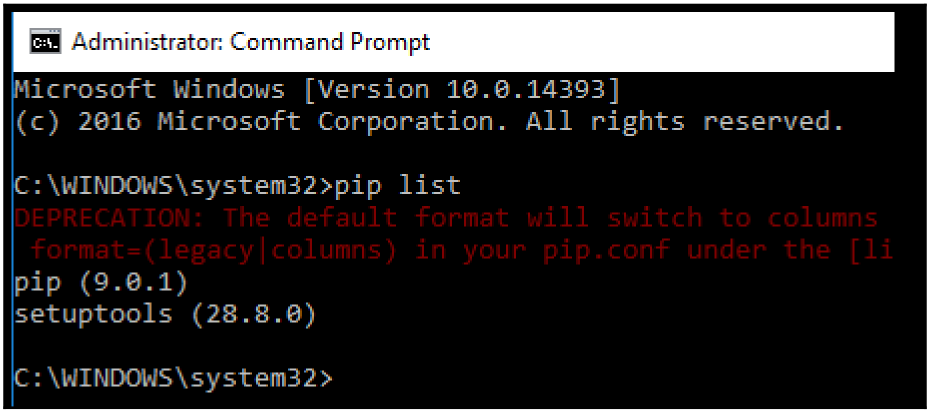
```
import tkinter as tk

#============================================
class ToolTip(object):
    def __init__(self, widget):
        self.widget = widget
        self.tip_window = None

    def show_tip(self, tip_text):
        "Display text in a tooltip window"
```

Chapter 5: Matplotlib Charts

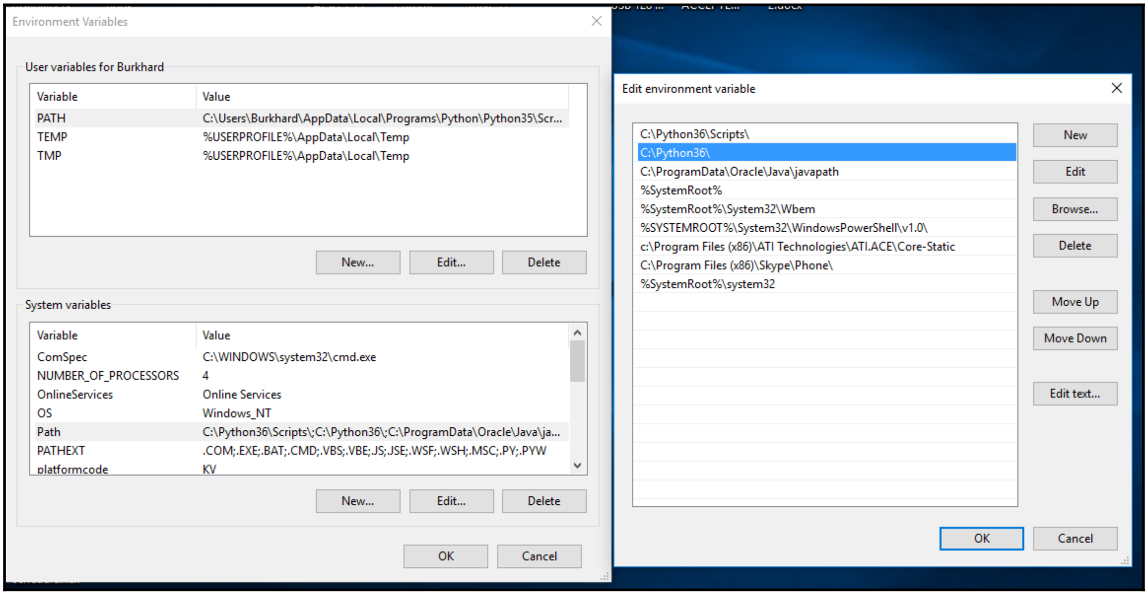


```
2nd Edition Python GUI Programming Cookbook
    Ch01_Code
    Ch02_Code
    Ch03_Code
    Ch04_Code
    Ch05_Code
        __init__.py
        Matplotlib_chart_with_legend_missing_comma.py
        Matplotlib_chart_with_legend.py
        Matplotlib_chart.py
        Matplotlib_labels_two_charts_not_scaled.py
        Matplotlib_labels_two_charts_scaled_dynamic_spike.py
        Matplotlib_labels_two_charts_scaled_dynamic.py
        Matplotlib_labels_two_charts_scaled.py
        Matplotlib_labels_two_charts.py
        Matplotlib_labels.py
        Matplotlib_our_first_chart.py
        Matplotlib_second_chart.py
```





```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>pip list
DEPRECATION: The default format will switch to columns
 format=(legacy|columns) in your pip.conf under the [li
pip (9.0.1)
setuptools (28.8.0)

C:\WINDOWS\system32>
```



Environment Variables

User variables for Burkhard

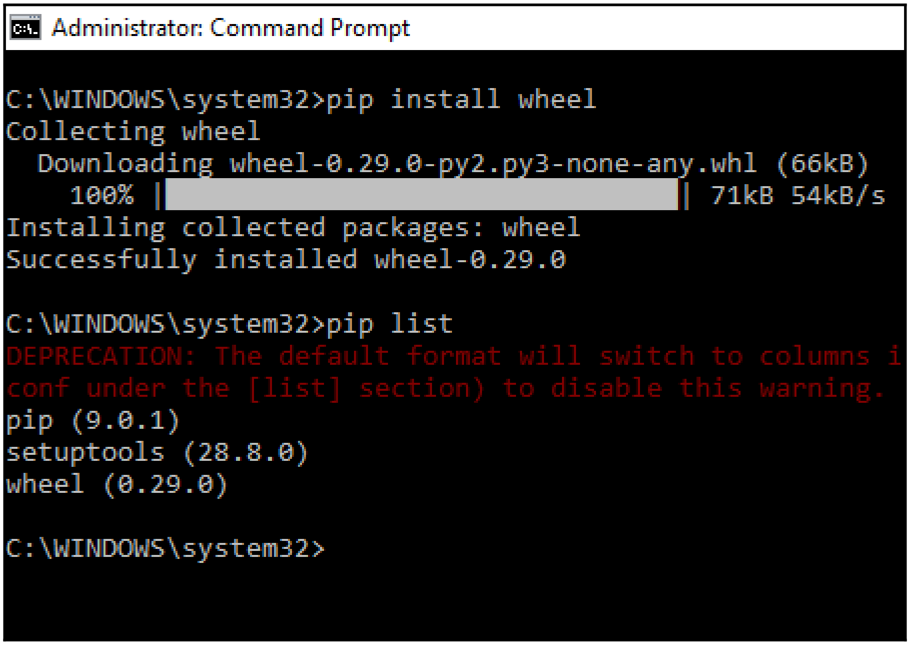
Variable	Value
PATH	C:\Users\Burkhard\AppData\Local\Programs\Python\Python35\Scr...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
NUMBER_OF_PROCESSORS	4
OnlineServices	Online Services
OS	Windows_NT
Path	C:\Python36\Scripts\;C:\Python36\;C:\ProgramData\Oracle\Java\ja...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW
platformcode	KV

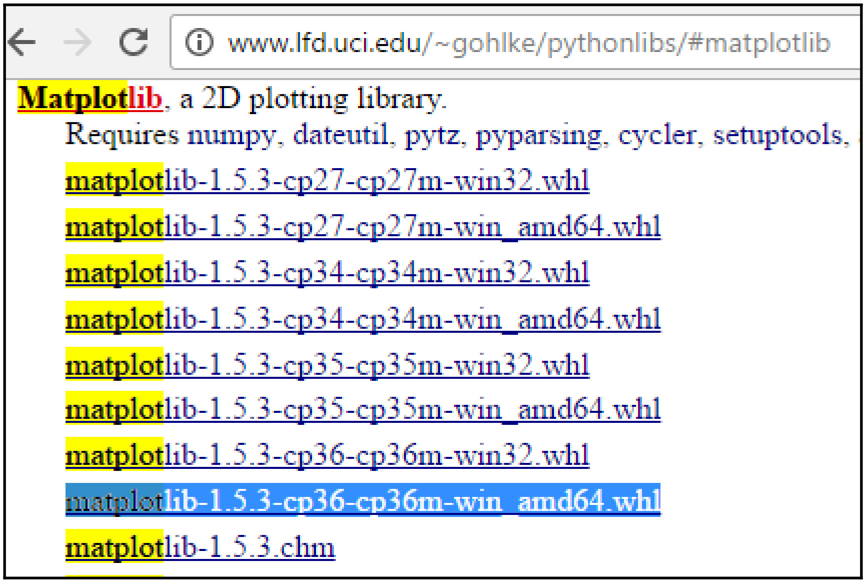
Edit environment variable

```
C:\Python36\Scripts\
C:\Python36\
C:\ProgramData\Oracle\Java\javapath
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
c:\Program Files (x86)\ATI Technologies\ATI.ACE\Core-Static
C:\Program Files (x86)\Skype\Phone\
%SystemRoot%\system32
```



Administrator: Command Prompt

```
C:\WINDOWS\system32>pip install wheel
Collecting wheel
  Downloading wheel-0.29.0-py2.py3-none-any.whl (66kB)
    100% |████████████████████████████████| 71kB 54kB/s
Installing collected packages: wheel
Successfully installed wheel-0.29.0

C:\WINDOWS\system32>pip list
DEPRECATION: The default format will switch to columns i
conf under the [list] section) to disable this warning.
pip (9.0.1)
setuptools (28.8.0)
wheel (0.29.0)

C:\WINDOWS\system32>
```



www.lfd.uci.edu/~gohlke/pythonlibs/#matplotlib

Matplotlib, a 2D plotting library.

Requires numpy, dateutil, pytz, pyparsing, cycler, setuptools,

matplotlib-1.5.3-cp27-cp27m-win32.whl

matplotlib-1.5.3-cp27-cp27m-win_amd64.whl

matplotlib-1.5.3-cp34-cp34m-win32.whl

matplotlib-1.5.3-cp34-cp34m-win_amd64.whl

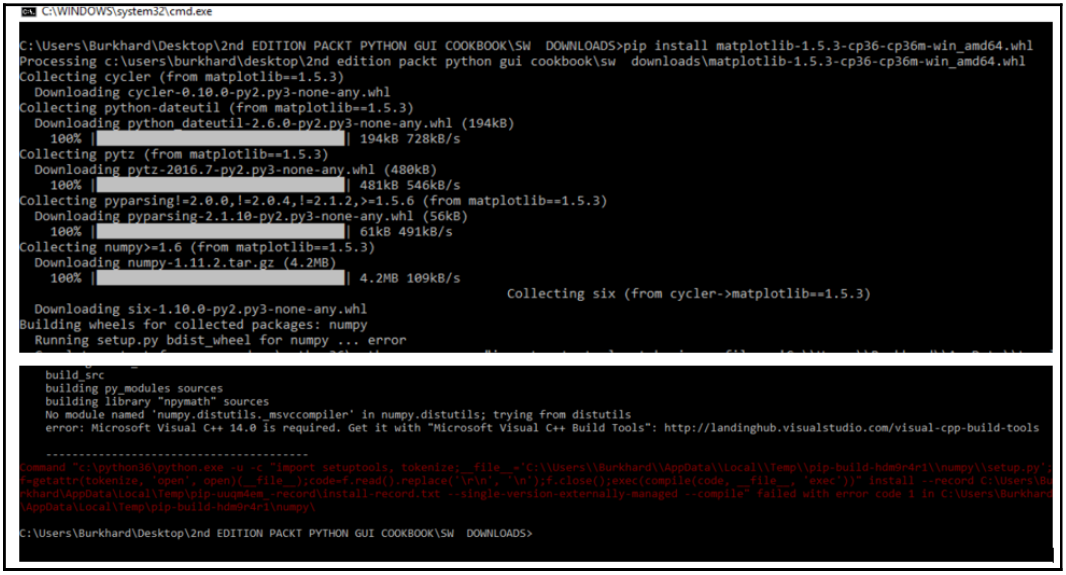
matplotlib-1.5.3-cp35-cp35m-win32.whl

matplotlib-1.5.3-cp35-cp35m-win_amd64.whl

matplotlib-1.5.3-cp36-cp36m-win32.whl

matplotlib-1.5.3-cp36-cp36m-win_amd64.whl

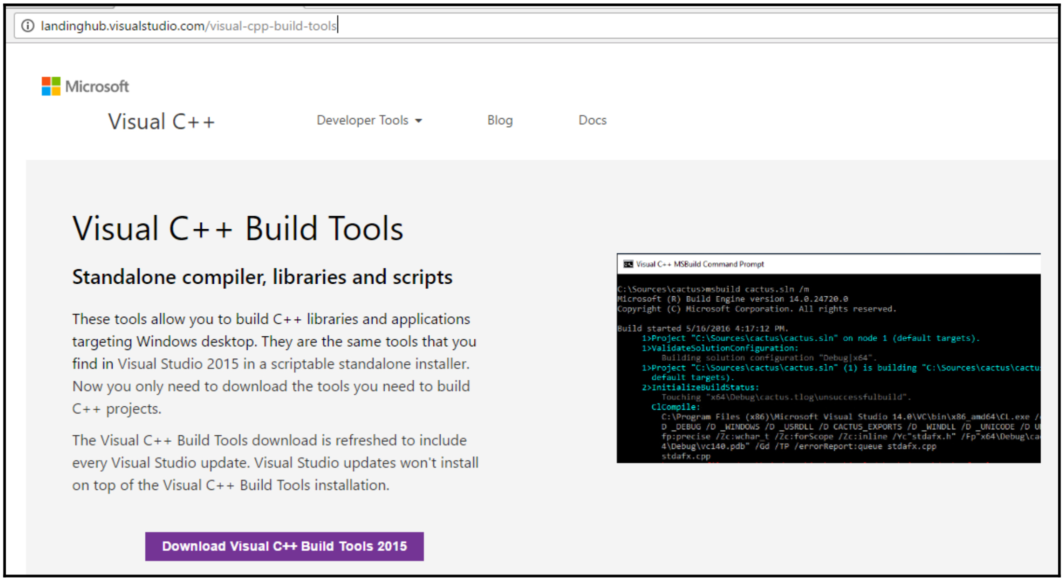
matplotlib-1.5.3.chm



C:\WINDOWS\system32\cmd.exe

```
C:\Users\Burkhard\Desktop\2nd EDITION PACKT PYTHON GUI COOKBOOK\SW  DOWNLOADS>pip install matplotlib-1.5.3-cp36-cp36m-win_amd64.whl
Processing c:\users\burkhard\desktop\2nd edition packt python gui cookbook\sw  downloads\matplotlib-1.5.3-cp36-cp36m-win_amd64.whl
Collecting cycler (from matplotlib==1.5.3)
  Downloading cycler-0.10.0-py2.py3-none-any.whl
Collecting python-dateutil (from matplotlib==1.5.3)
  Downloading python_dateutil-2.6.0-py2.py3-none-any.whl (194kB)
    100% |████████████████████████████████| 194kB 728kB/s
Collecting pytz (from matplotlib==1.5.3)
  Downloading pytz-2016.7-py2.py3-none-any.whl (480kB)
    100% |████████████████████████████████| 481kB 546kB/s
Collecting pyparsing!=2.0.0,!=2.0.4,!=2.1.2,>=1.5.6 (from matplotlib==1.5.3)
  Downloading pyparsing-2.1.10-py2.py3-none-any.whl (56kB)
    100% |████████████████████████████████| 61kB 491kB/s
Collecting numpy>=1.6 (from matplotlib==1.5.3)
  Downloading numpy-1.11.2.tar.gz (4.2MB)
    100% |████████████████████████████████| 4.2MB 109kB/s
                                                               Collecting six (from cycler->matplotlib==1.5.3)
  Downloading six-1.10.0-py2.py3-none-any.whl
Building wheels for collected packages: numpy
  Running setup.py bdist_wheel for numpy ... error
```

```
    build_src
    building py_modules sources
    building library "npymath" sources
    No module named 'numpy.distutils._msvccompiler' in numpy.distutils; trying from distutils
    error: Microsoft Visual C++ 14.0 is required. Get it with "Microsoft Visual C++ Build Tools": http://landinghub.visualstudio.com/visual-cpp-build-tools

    ----------------------------------------
Command "c:\python36\python.exe -u -c "import setuptools, tokenize;__file__='C:\\Users\\Burkhard\\AppData\\Local\\Temp\\pip-build-hdm9r4r1\\numpy\\setup.py';
f=getattr(tokenize, 'open', open)(__file__);code=f.read().replace('\r\n', '\n');f.close();exec(compile(code, __file__, 'exec'))" install --record C:\Users\Bu
rkhard\AppData\Local\Temp\pip-uuqm4em_-record\install-record.txt --single-version-externally-managed --compile" failed with error code 1 in C:\Users\Burkhard
\AppData\Local\Temp\pip-build-hdm9r4r1\numpy\

C:\Users\Burkhard\Desktop\2nd EDITION PACKT PYTHON GUI COOKBOOK\SW  DOWNLOADS>
```



Visual C++ Build Tools

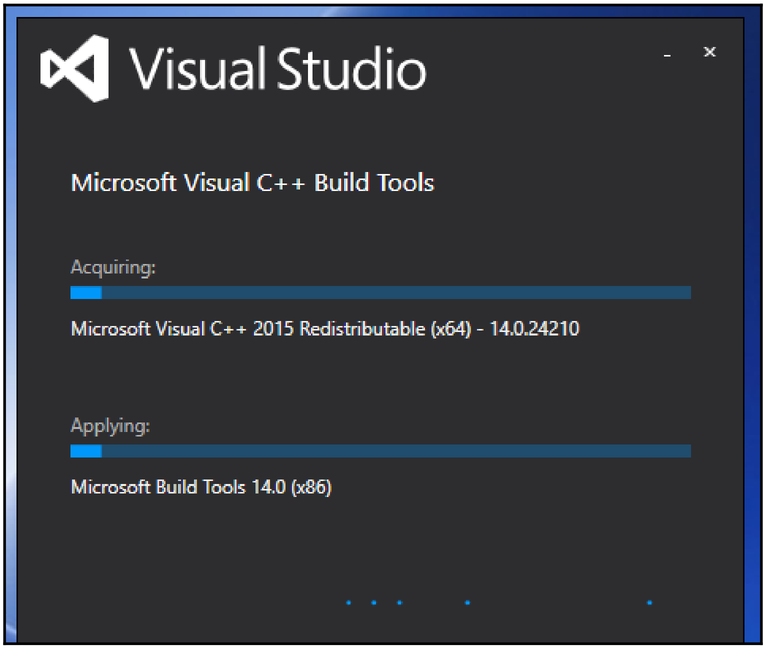
Standalone compiler, libraries and scripts

These tools allow you to build C++ libraries and applications targeting Windows desktop. They are the same tools that you find in Visual Studio 2015 in a scriptable standalone installer. Now you only need to download the tools you need to build C++ projects.

The Visual C++ Build Tools download is refreshed to include every Visual Studio update. Visual Studio updates won't install on top of the Visual C++ Build Tools installation.

Download Visual C++ Build Tools 2015
Visual C++ MSBuild Command Prompt

```
C:\Sources\cactus>msbuild cactus.sln /m
Microsoft (R) Build Engine version 14.0.24720.0
Copyright (C) Microsoft Corporation. All rights reserved.

Build started 5/16/2016 4:17:12 PM.
     1>Project "C:\Sources\cactus\cactus.sln" on node 1 (default targets).
     1>ValidateSolutionConfiguration:
         Building solution configuration "Debug|x64".
     1>Project "C:\Sources\cactus\cactus.sln" (1) is building "C:\Sources\cactus\cactu
       default targets).
     2>InitializeBuildStatus:
         Touching "x64\Debug\cactus.tlog\unsuccessfulbuild".
       ClCompile:
         C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\bin\x86_amd64\CL.exe /
         D _DEBUG /D _WINDOWS /D _USRDLL /D CACTUS_EXPORTS /D _WINDLL /D _UNICODE /D U
         fp:precise /Zc:wchar_t /Zc:forScope /Zc:inline /Yc"stdafx.h" /Fp"x64\Debug\ca
         4\Debug\vc140.pdb" /Gd /TP /errorReport:queue stdafx.cpp
         stdafx.cpp
```



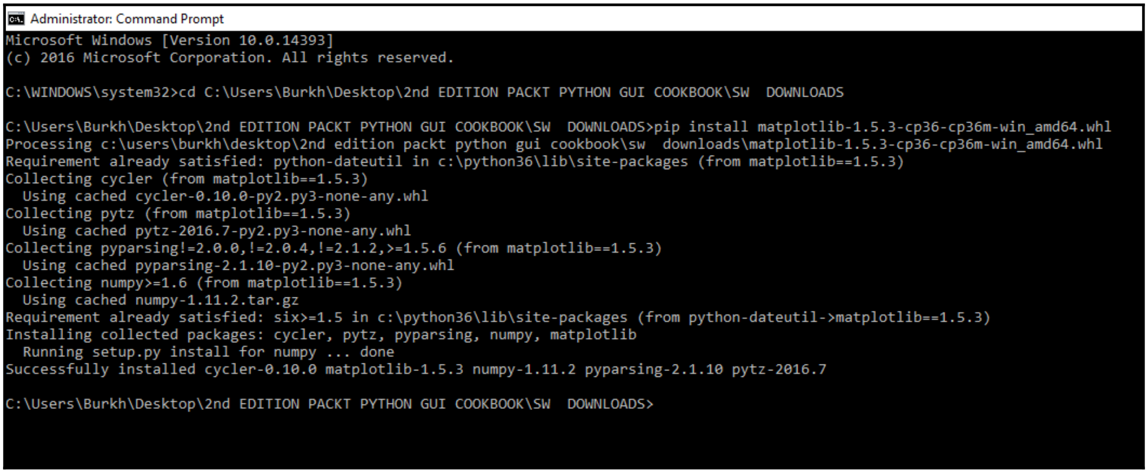
Microsoft Visual C++ Build Tools

Acquiring:

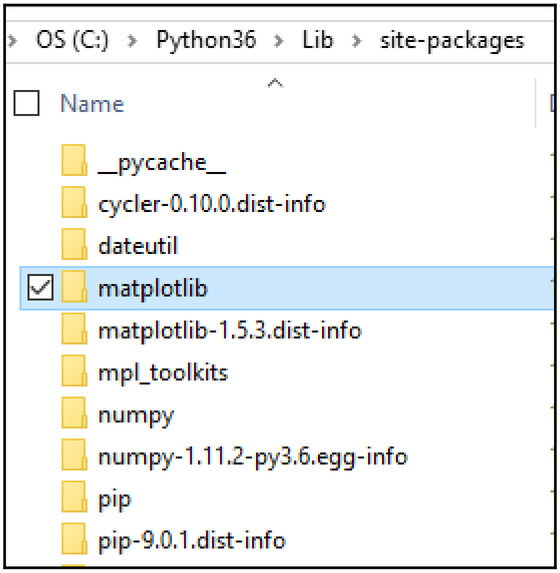
Microsoft Visual C++ 2015 Redistributable (x64) - 14.0.24210

Applying:

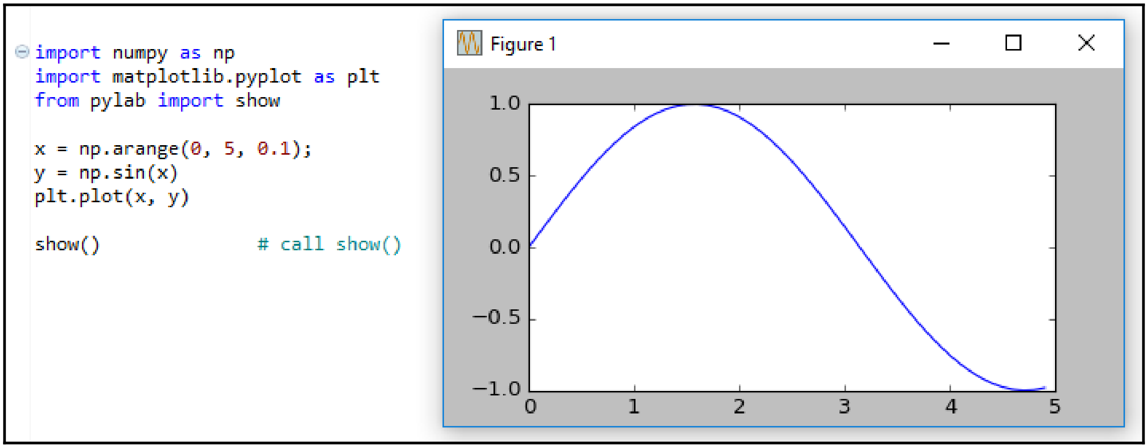
Microsoft Build Tools 14.0 (x86)



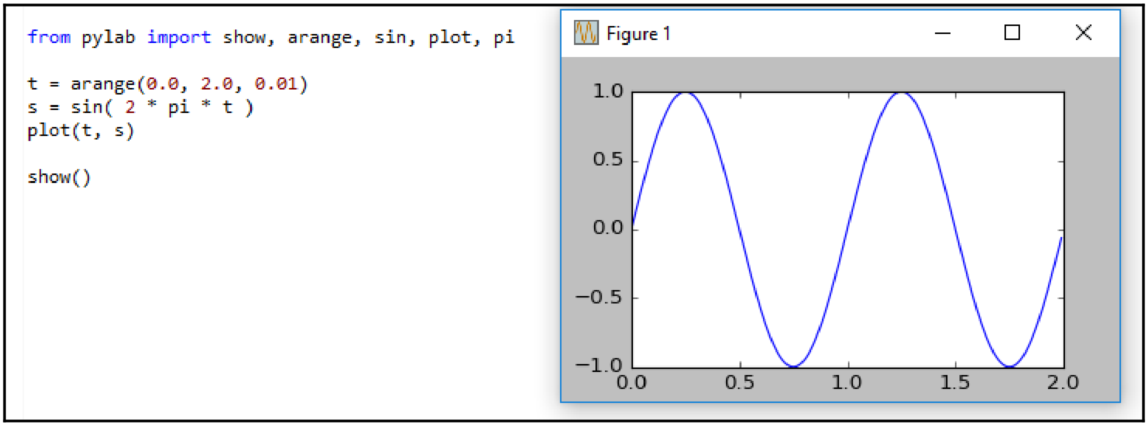
```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\Burkh\Desktop\2nd EDITION PACKT PYTHON GUI COOKBOOK\SW  DOWNLOADS

C:\Users\Burkh\Desktop\2nd EDITION PACKT PYTHON GUI COOKBOOK\SW  DOWNLOADS>pip install matplotlib-1.5.3-cp36-cp36m-win_amd64.whl
Processing c:\users\burkh\desktop\2nd edition packt python gui cookbook\sw  downloads\matplotlib-1.5.3-cp36-cp36m-win_amd64.whl
Requirement already satisfied: python-dateutil in c:\python36\lib\site-packages (from matplotlib==1.5.3)
Collecting cycler (from matplotlib==1.5.3)
  Using cached cycler-0.10.0-py2.py3-none-any.whl
Collecting pytz (from matplotlib==1.5.3)
  Using cached pytz-2016.7-py2.py3-none-any.whl
Collecting pyparsing!=2.0.0,!=2.0.4,!=2.1.2,>=1.5.6 (from matplotlib==1.5.3)
  Using cached pyparsing-2.1.10-py2.py3-none-any.whl
Collecting numpy>=1.6 (from matplotlib==1.5.3)
  Using cached numpy-1.11.2.tar.gz
Requirement already satisfied: six>=1.5 in c:\python36\lib\site-packages (from python-dateutil->matplotlib==1.5.3)
Installing collected packages: cycler, pytz, pyparsing, numpy, matplotlib
  Running setup.py install for numpy ... done
Successfully installed cycler-0.10.0 matplotlib-1.5.3 numpy-1.11.2 pyparsing-2.1.10 pytz-2016.7

C:\Users\Burkh\Desktop\2nd EDITION PACKT PYTHON GUI COOKBOOK\SW  DOWNLOADS>
```



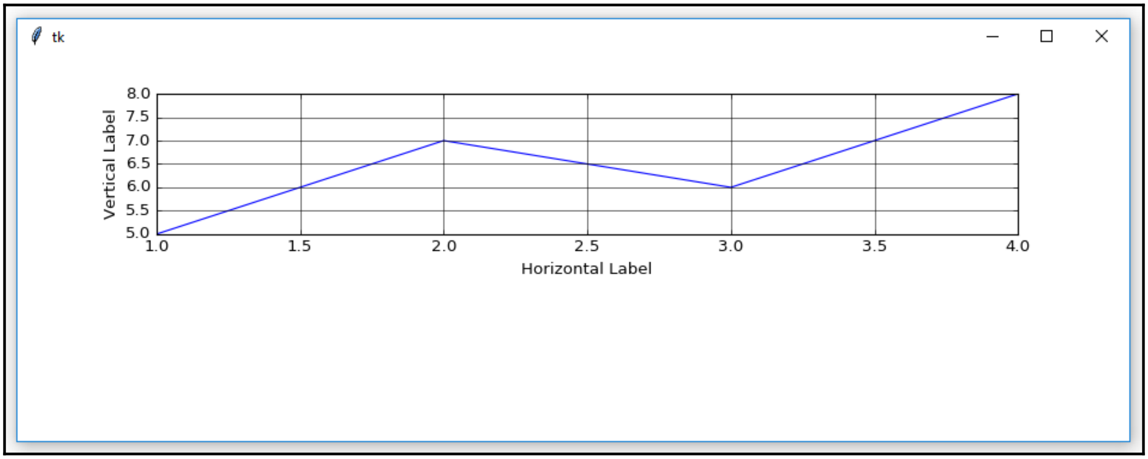
__pycache__
cycler-0.10.0.dist-info
dateutil
☑ matplotlib
matplotlib-1.5.3.dist-info
mpl_toolkits
numpy
numpy-1.11.2-py3.6.egg-info
pip
pip-9.0.1.dist-info

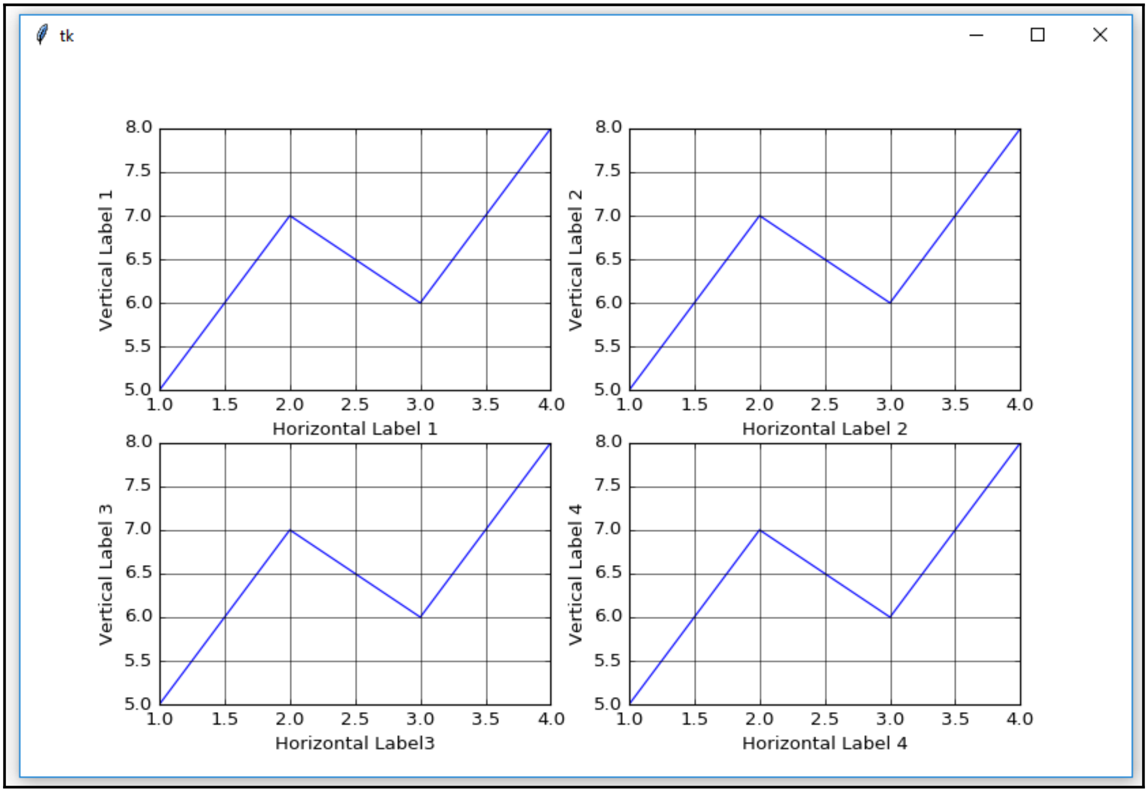


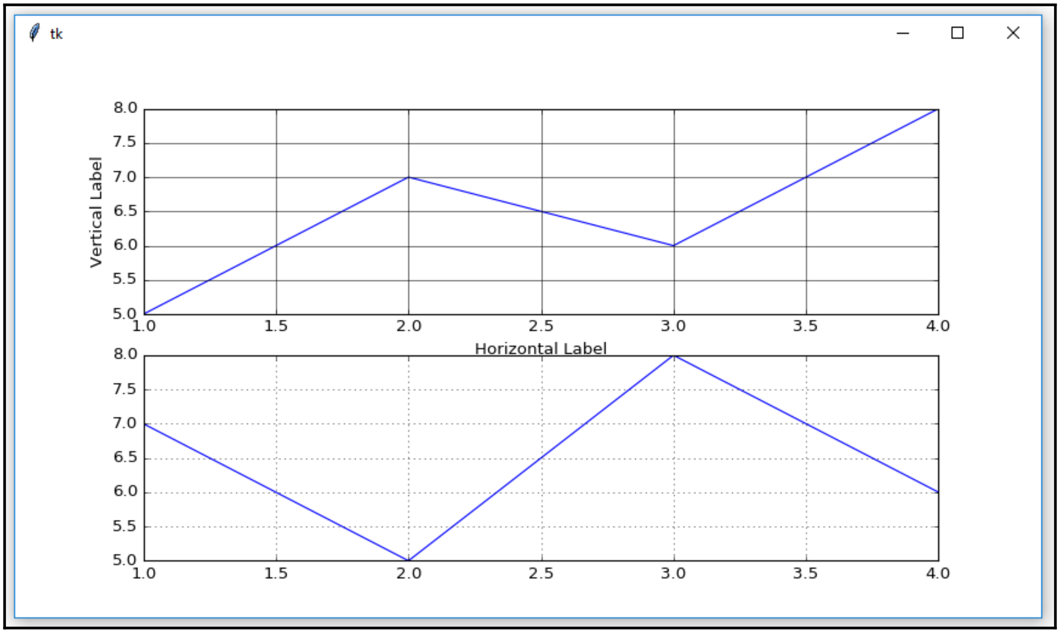
```
import numpy as np
import matplotlib.pyplot as plt
from pylab import show

x = np.arange(0, 5, 0.1);
y = np.sin(x)
plt.plot(x, y)

show()              # call show()
```

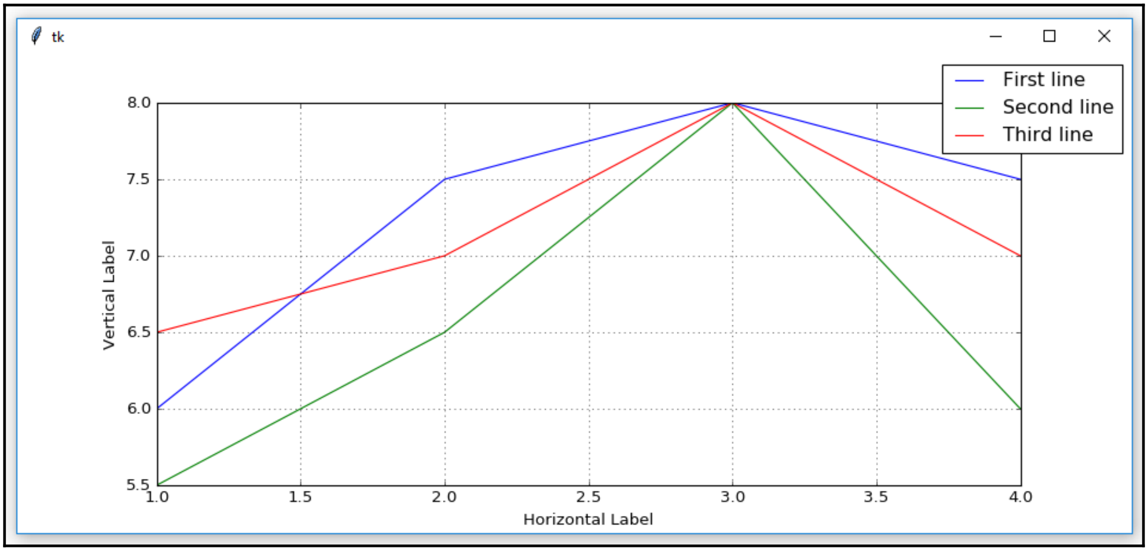


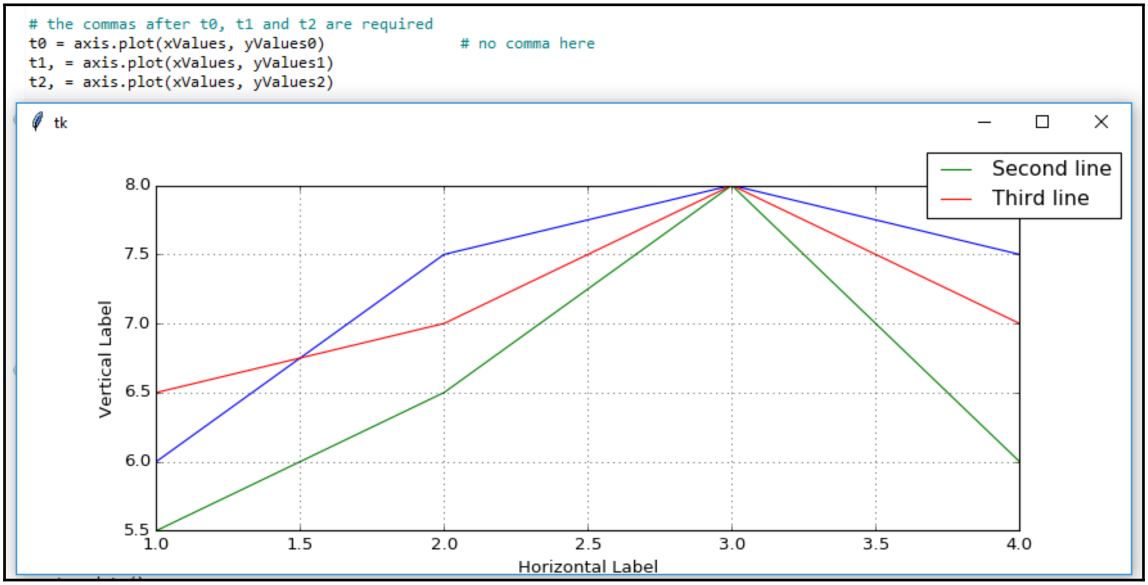
```
from pylab import show, arange, sin, plot, pi

t = arange(0.0, 2.0, 0.01)
s = sin( 2 * pi * t )
plot(t, s)

show()
```



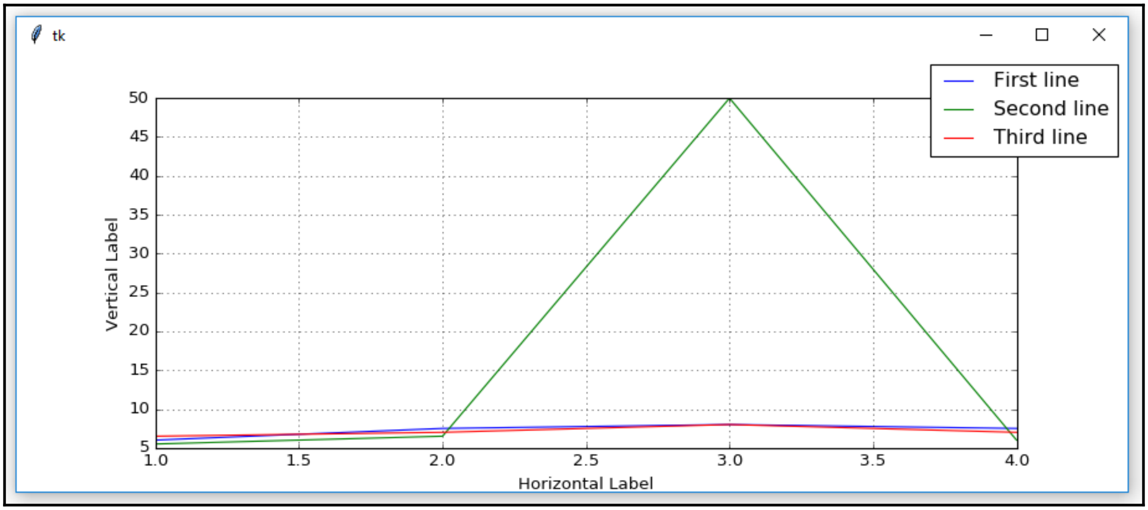

tk
8.0
7.5
7.0
6.5
6.0
5.5
5.0
Vertical Label 1
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Horizontal Label 1
Vertical Label 2
Horizontal Label 2
Vertical Label 3
Horizontal Label3
Vertical Label 4
Horizontal Label 4

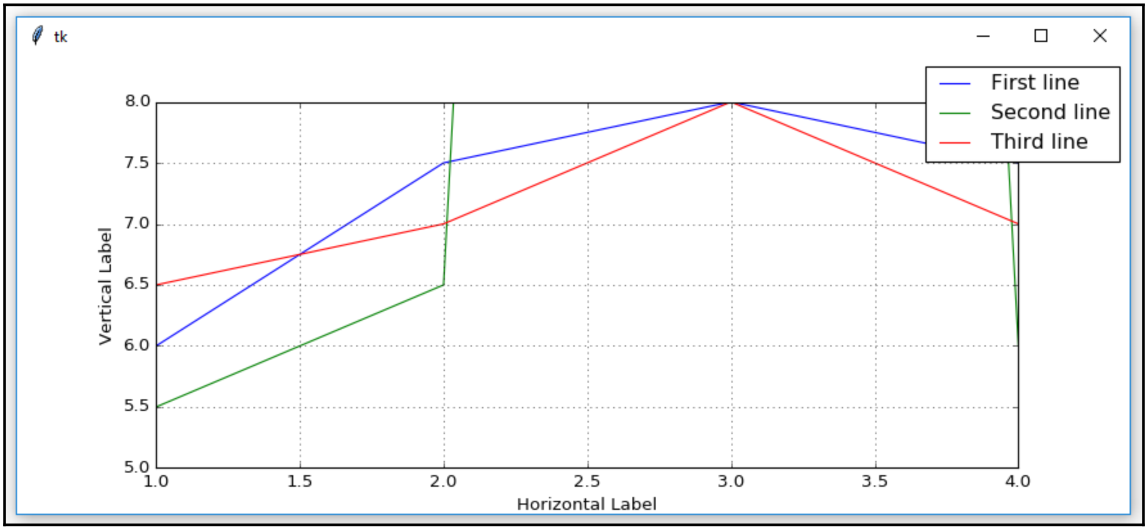

tk
Vertical Label
8.0
7.5
7.0
6.5
6.0
5.5
5.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Horizontal Label
8.0
7.5
7.0
6.5
6.0
5.5
5.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0

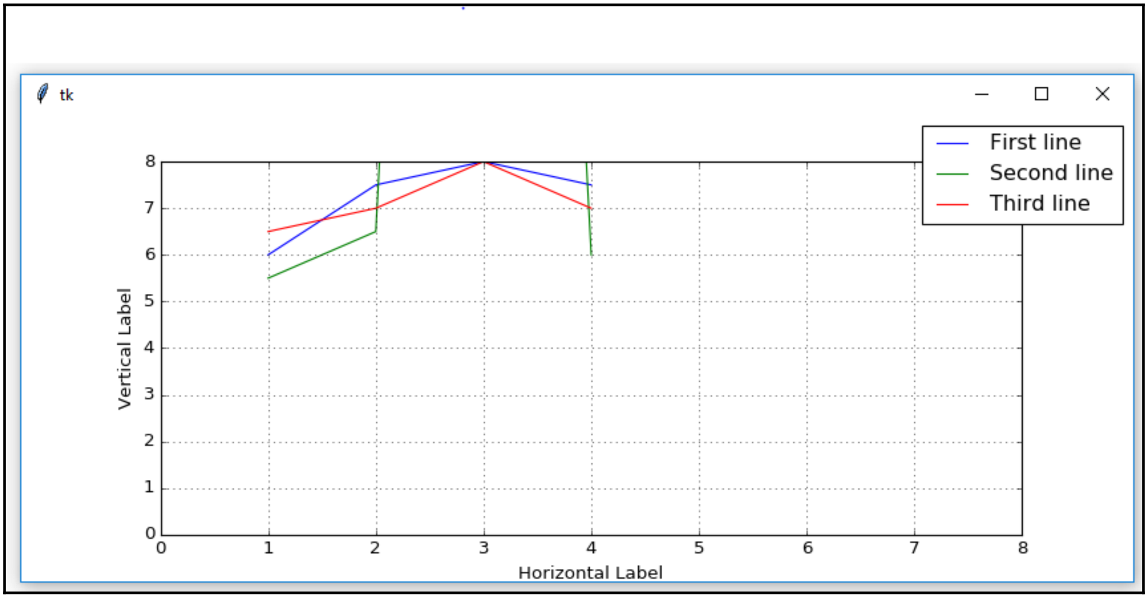

tk
First line
Second line
Third line
Vertical Label
8.0
7.5
7.0
6.5
6.0
5.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Horizontal Label

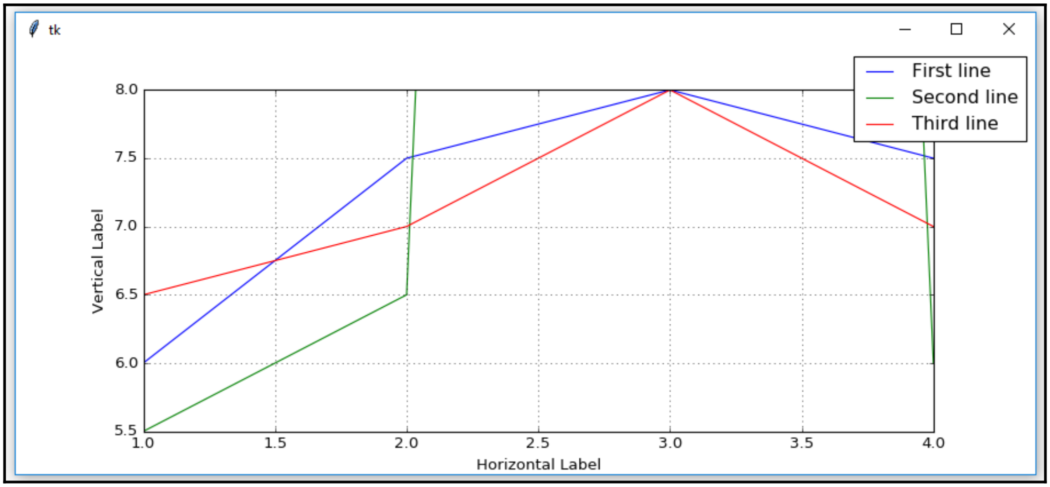


```
# the commas after t0, t1 and t2 are required
t0 = axis.plot(xValues, yValues0)                # no comma here
t1, = axis.plot(xValues, yValues1)
t2, = axis.plot(xValues, yValues2)
```

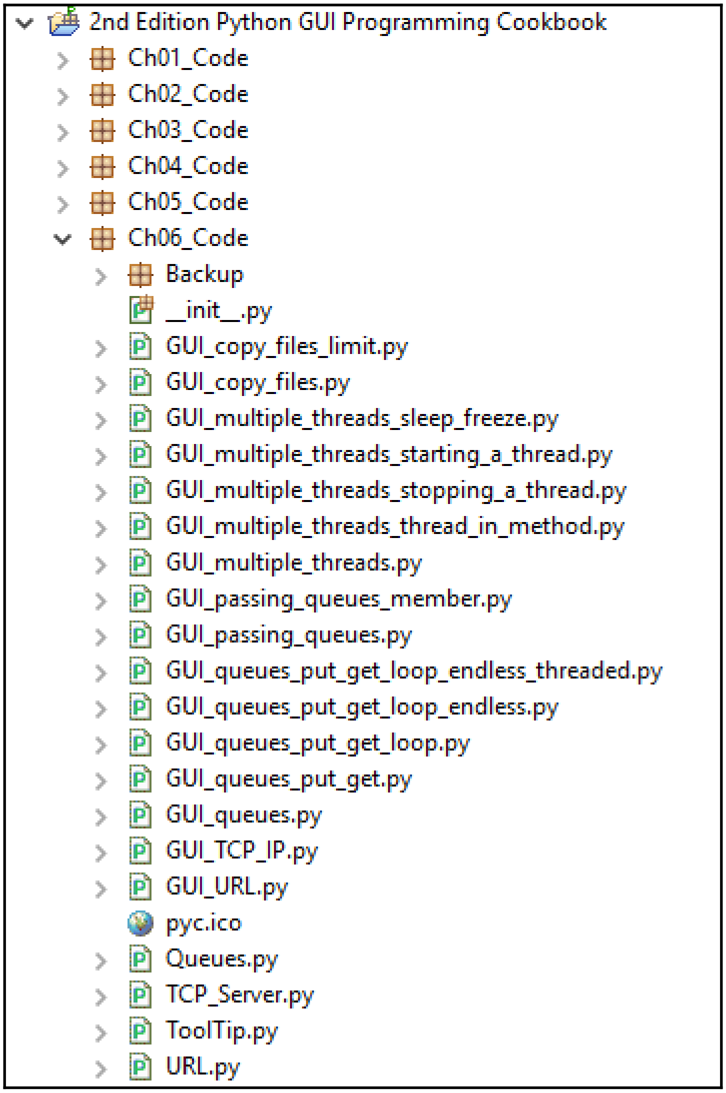



tk
First line
Second line
Third line
8.0
7.5
7.0
6.5
6.0
5.5
5.0
Vertical Label
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Horizontal Label

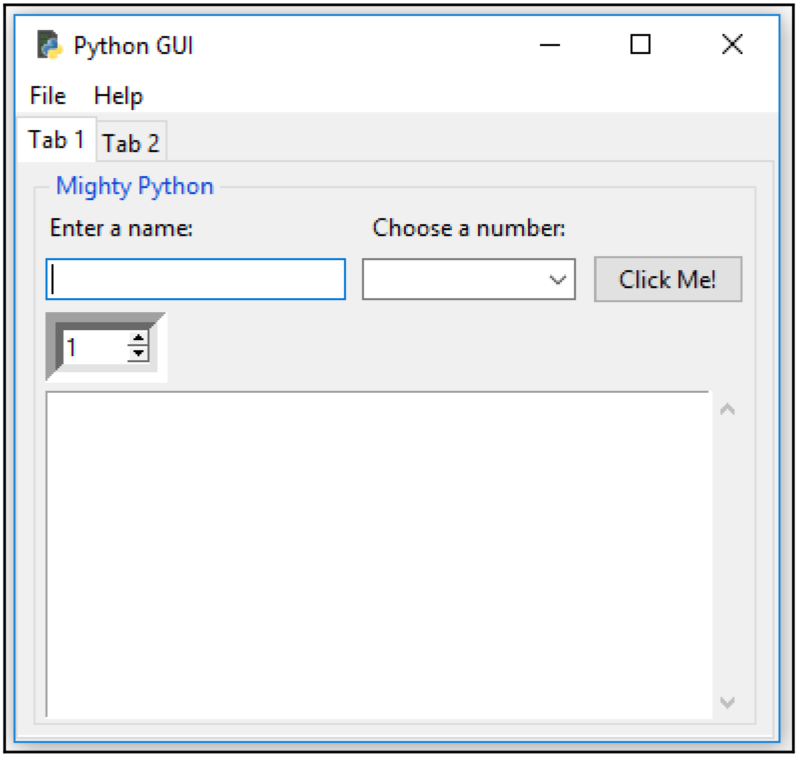

tk
First line
Second line
Third line
8
7
6
5
4
3
2
1
0
Vertical Label
0
1
2
3
4
5
6
7
8
Horizontal Label

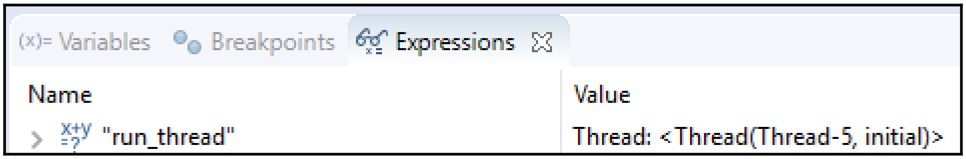

tk
First line
Second line
Third line
8.0
7.5
7.0
6.5
6.0
5.5
Vertical Label
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Horizontal Label

Chapter 6: Threads and Networking

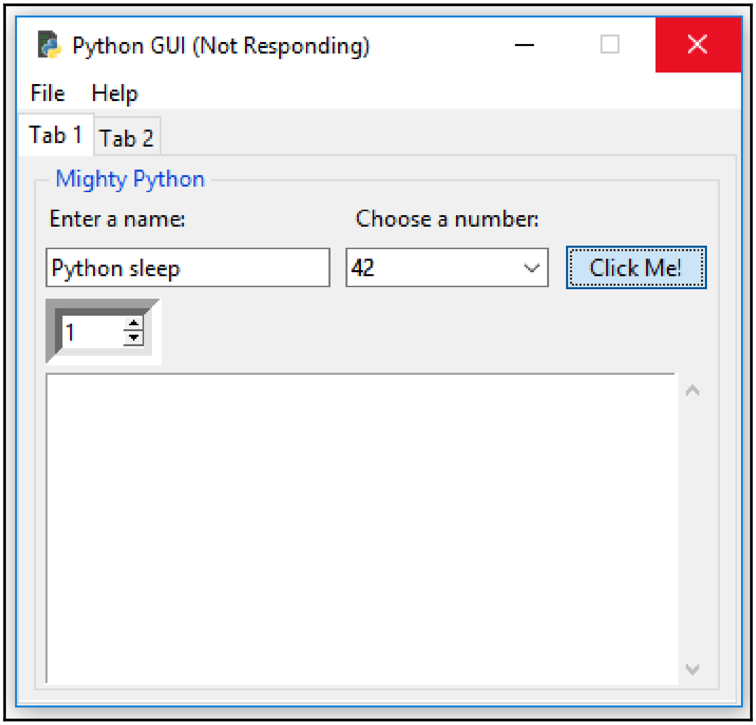


2nd Edition Python GUI Programming Cookbook
- Ch01_Code
- Ch02_Code
- Ch03_Code
- Ch04_Code
- Ch05_Code
- Ch06_Code
  - Backup
  - __init__.py
  - GUI_copy_files_limit.py
  - GUI_copy_files.py
  - GUI_multiple_threads_sleep_freeze.py
  - GUI_multiple_threads_starting_a_thread.py
  - GUI_multiple_threads_stopping_a_thread.py
  - GUI_multiple_threads_thread_in_method.py
  - GUI_multiple_threads.py
  - GUI_passing_queues_member.py
  - GUI_passing_queues.py
  - GUI_queues_put_get_loop_endless_threaded.py
  - GUI_queues_put_get_loop_endless.py
  - GUI_queues_put_get_loop.py
  - GUI_queues_put_get.py
  - GUI_queues.py
  - GUI_TCP_IP.py
  - GUI_URL.py
  - pyc.ico
  - Queues.py
  - TCP_Server.py
  - ToolTip.py
  - URL.py





Name	Value
"run_thread"	Thread: <Thread(Thread-5, initial)>

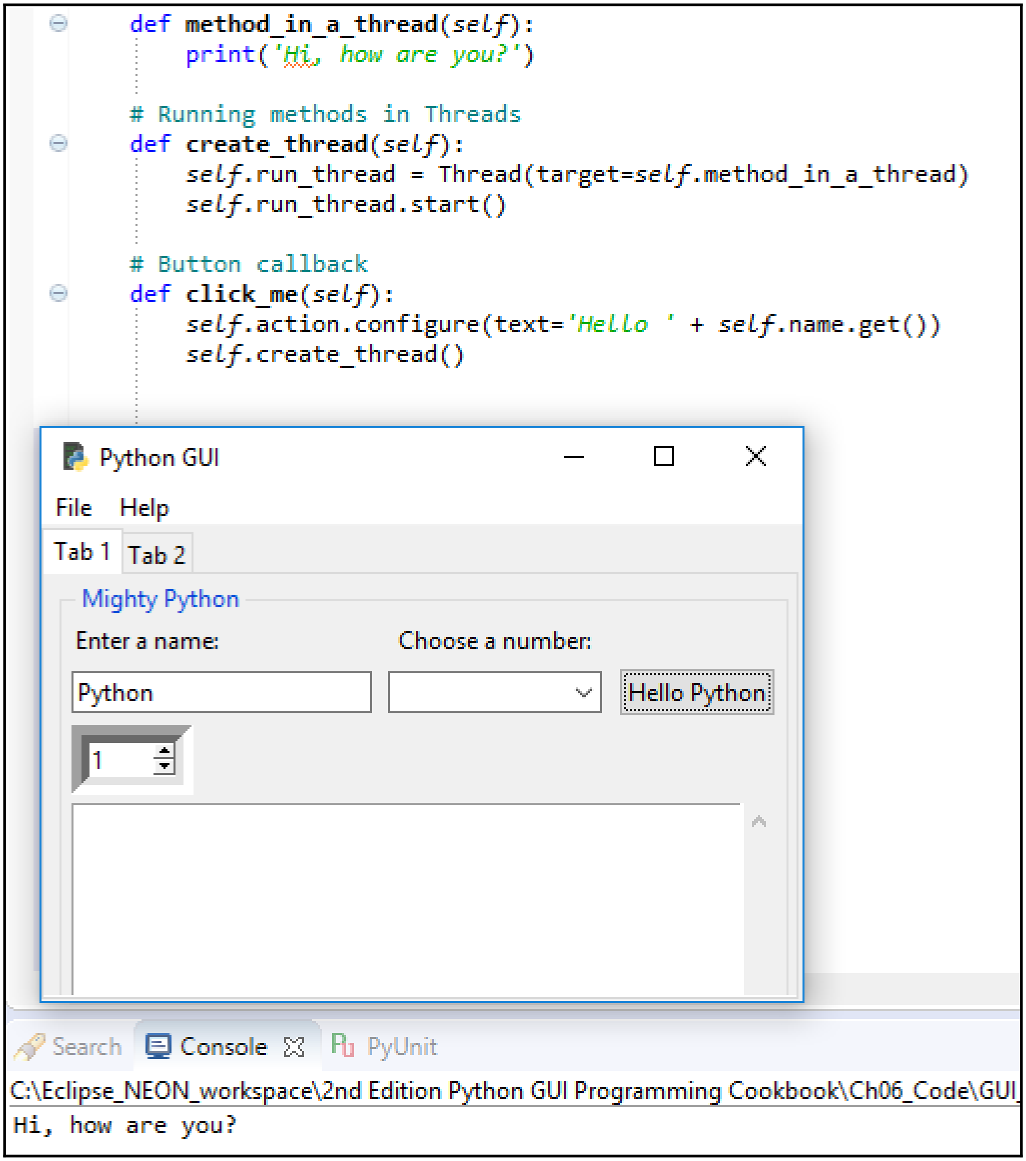


Python GUI (Not Responding)

Enter a name: Python sleep

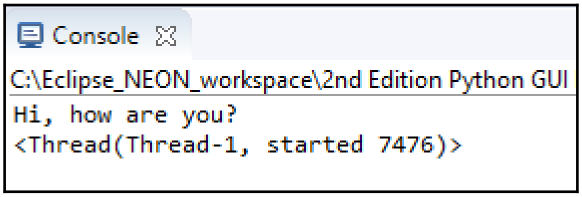
Choose a number: 42

1

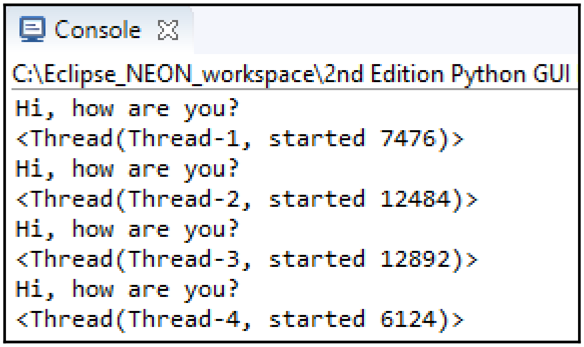


```
    def method_in_a_thread(self):
        print('Hi, how are you?')

    # Running methods in Threads
    def create_thread(self):
        self.run_thread = Thread(target=self.method_in_a_thread)
        self.run_thread.start()

    # Button callback
    def click_me(self):
        self.action.configure(text='Hello ' + self.name.get())
        self.create_thread()
```

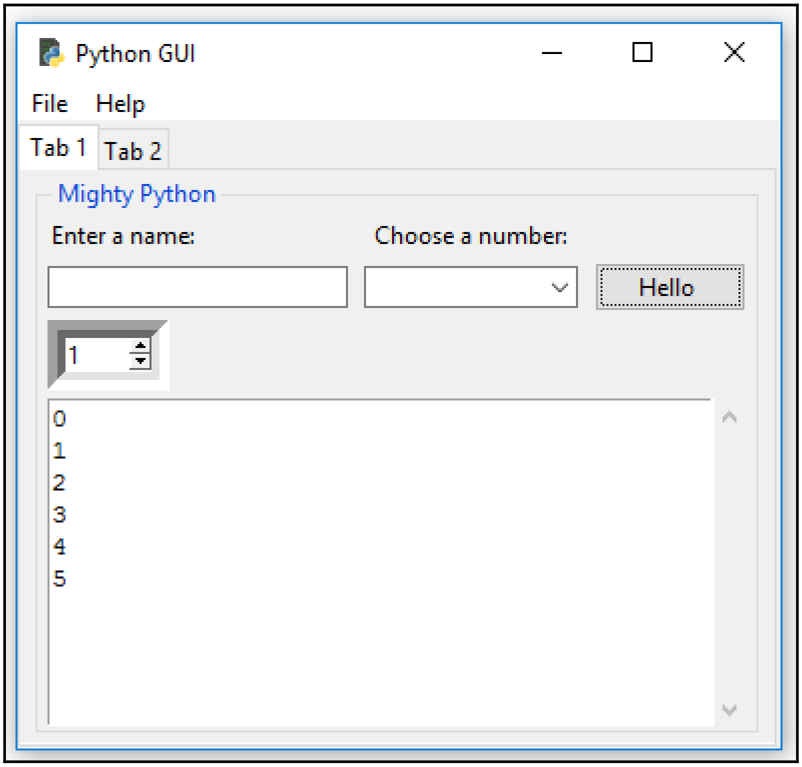
```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06_Code\GUI_
Hi, how are you?
```



```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI
Hi, how are you?
<Thread(Thread-1, started 7476)>
```

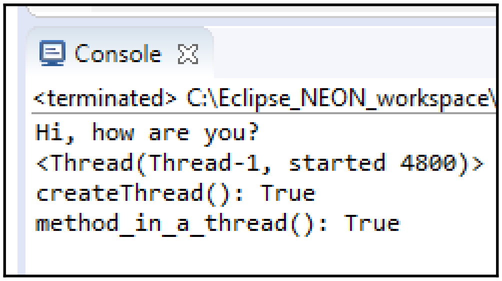


```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI
Hi, how are you?
<Thread(Thread-1, started 7476)>
Hi, how are you?
<Thread(Thread-2, started 12484)>
Hi, how are you?
<Thread(Thread-3, started 12892)>
Hi, how are you?
<Thread(Thread-4, started 6124)>
```

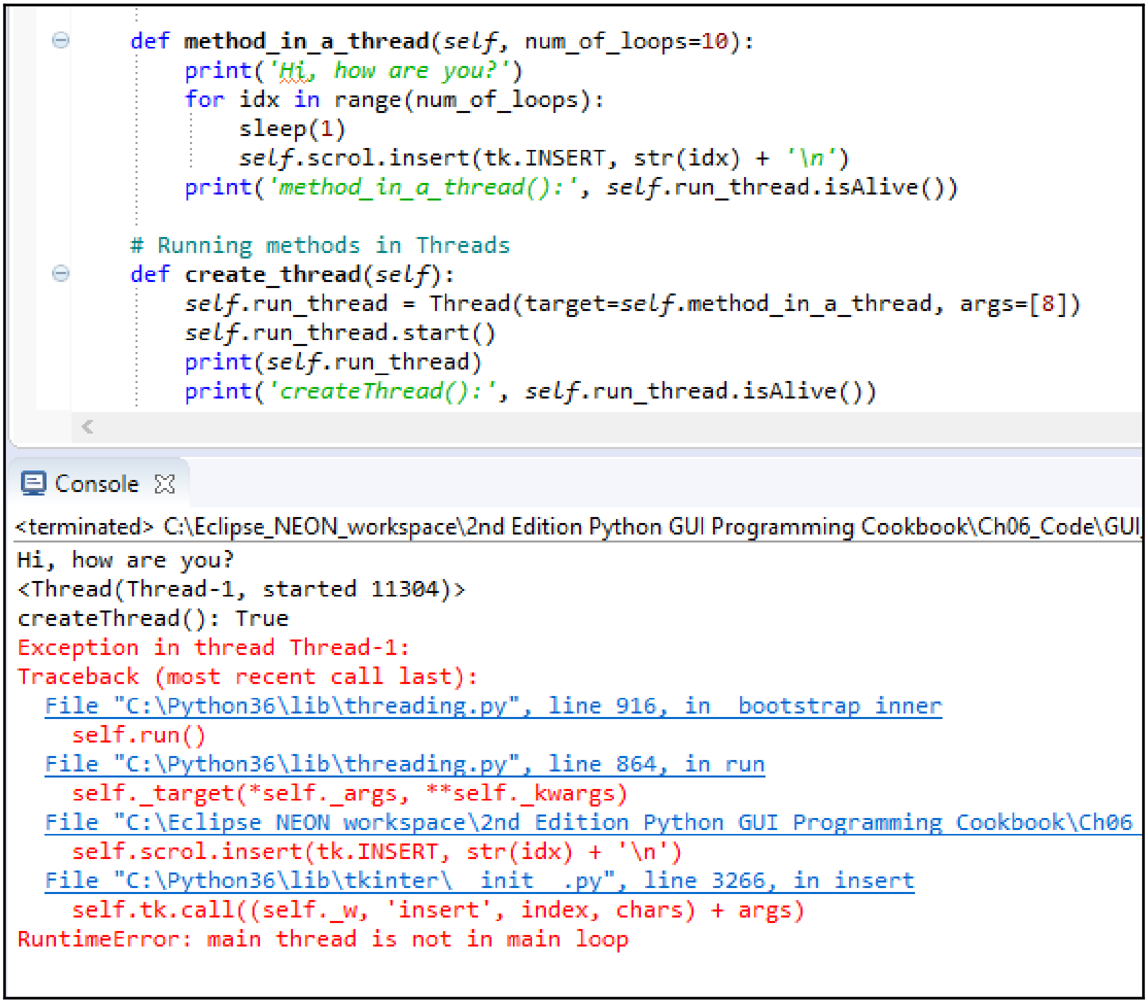


Spinbox: 1

```
0
1
2
3
4
5
```

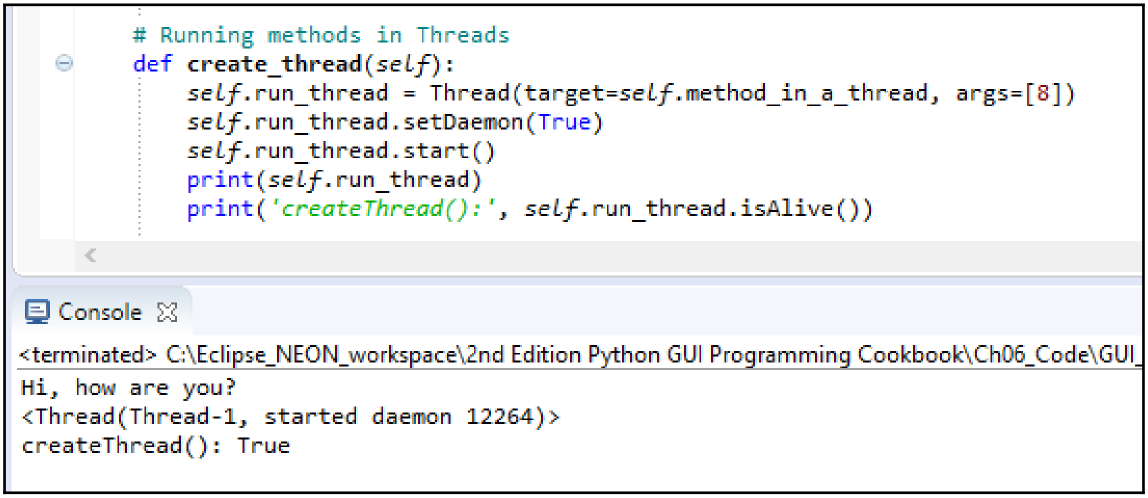


```
<terminated> C:\Eclipse_NEON_workspace\
Hi, how are you?
<Thread(Thread-1, started 4800)>
createThread(): True
method_in_a_thread(): True
```



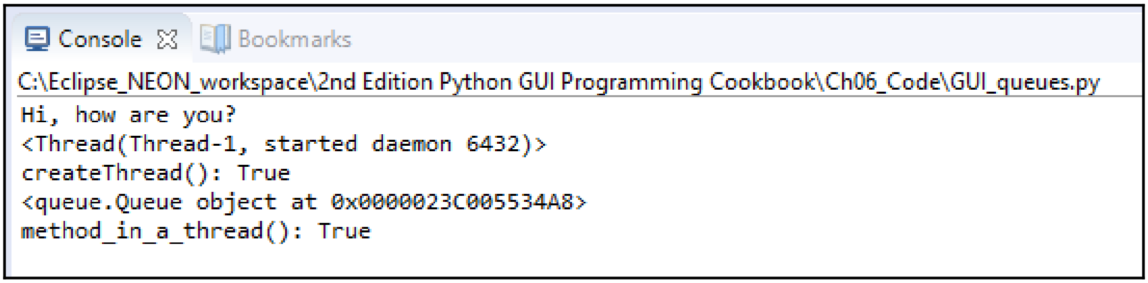
```
    def method_in_a_thread(self, num_of_loops=10):
        print('Hi, how are you?')
        for idx in range(num_of_loops):
            sleep(1)
            self.scrol.insert(tk.INSERT, str(idx) + '\n')
        print('method_in_a_thread():', self.run_thread.isAlive())

    # Running methods in Threads
    def create_thread(self):
        self.run_thread = Thread(target=self.method_in_a_thread, args=[8])
        self.run_thread.start()
        print(self.run_thread)
        print('createThread():', self.run_thread.isAlive())
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06_Code\GUI
Hi, how are you?
<Thread(Thread-1, started 11304)>
createThread(): True
Exception in thread Thread-1:
Traceback (most recent call last):
  File "C:\Python36\lib\threading.py", line 916, in _bootstrap_inner
    self.run()
  File "C:\Python36\lib\threading.py", line 864, in run
    self._target(*self._args, **self._kwargs)
  File "C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06
    self.scrol.insert(tk.INSERT, str(idx) + '\n')
  File "C:\Python36\lib\tkinter\__init__.py", line 3266, in insert
    self.tk.call((self._w, 'insert', index, chars) + args)
RuntimeError: main thread is not in main loop
```

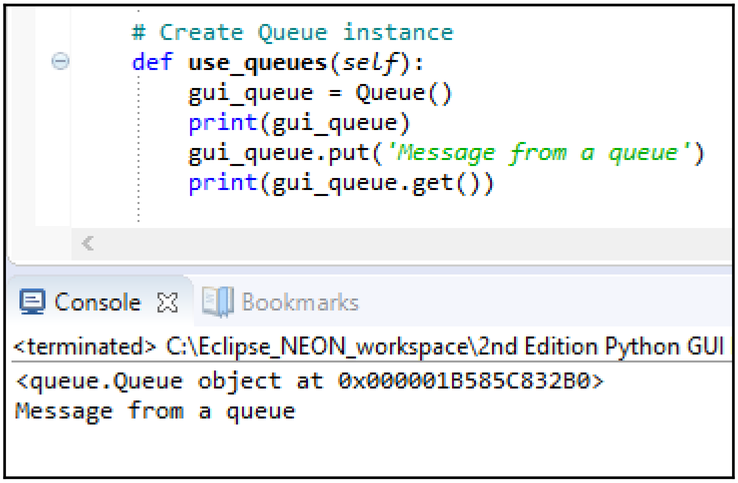


```
    # Running methods in Threads
    def create_thread(self):
        self.run_thread = Thread(target=self.method_in_a_thread, args=[8])
        self.run_thread.setDaemon(True)
        self.run_thread.start()
        print(self.run_thread)
        print('createThread():', self.run_thread.isAlive())
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06_Code\GUI_
Hi, how are you?
<Thread(Thread-1, started daemon 12264)>
createThread(): True
```

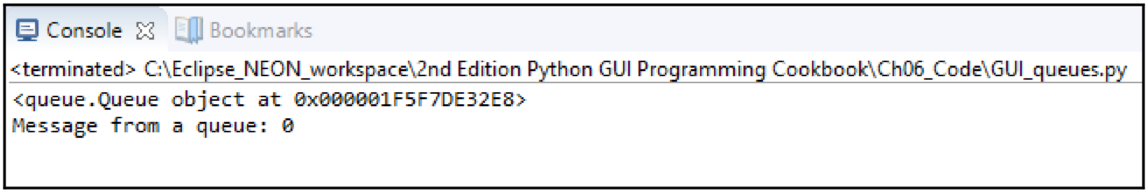


```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06_Code\GUI_queues.py
Hi, how are you?
<Thread(Thread-1, started daemon 6432)>
createThread(): True
<queue.Queue object at 0x0000023C005534A8>
method_in_a_thread(): True
```

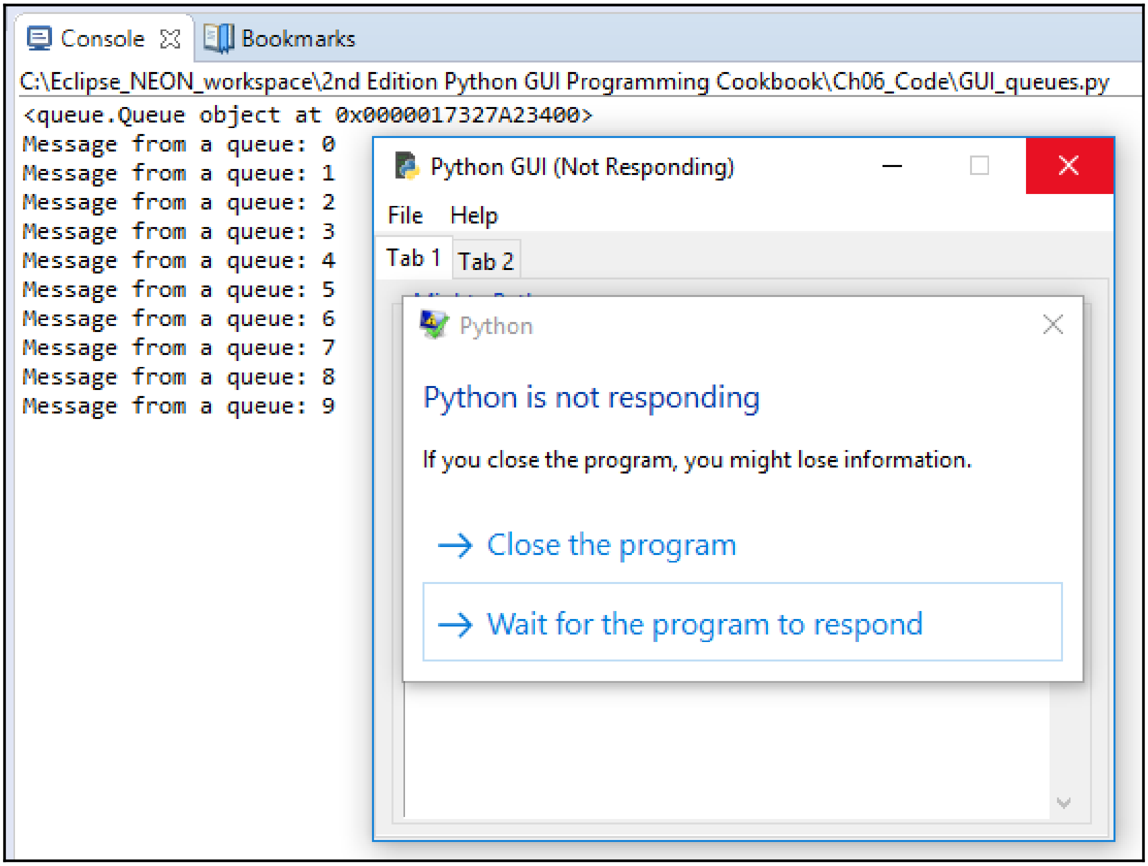


```
    # Create Queue instance
    def use_queues(self):
        gui_queue = Queue()
        print(gui_queue)
        gui_queue.put('Message from a queue')
        print(gui_queue.get())
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI
<queue.Queue object at 0x000001B585C832B0>
Message from a queue
```



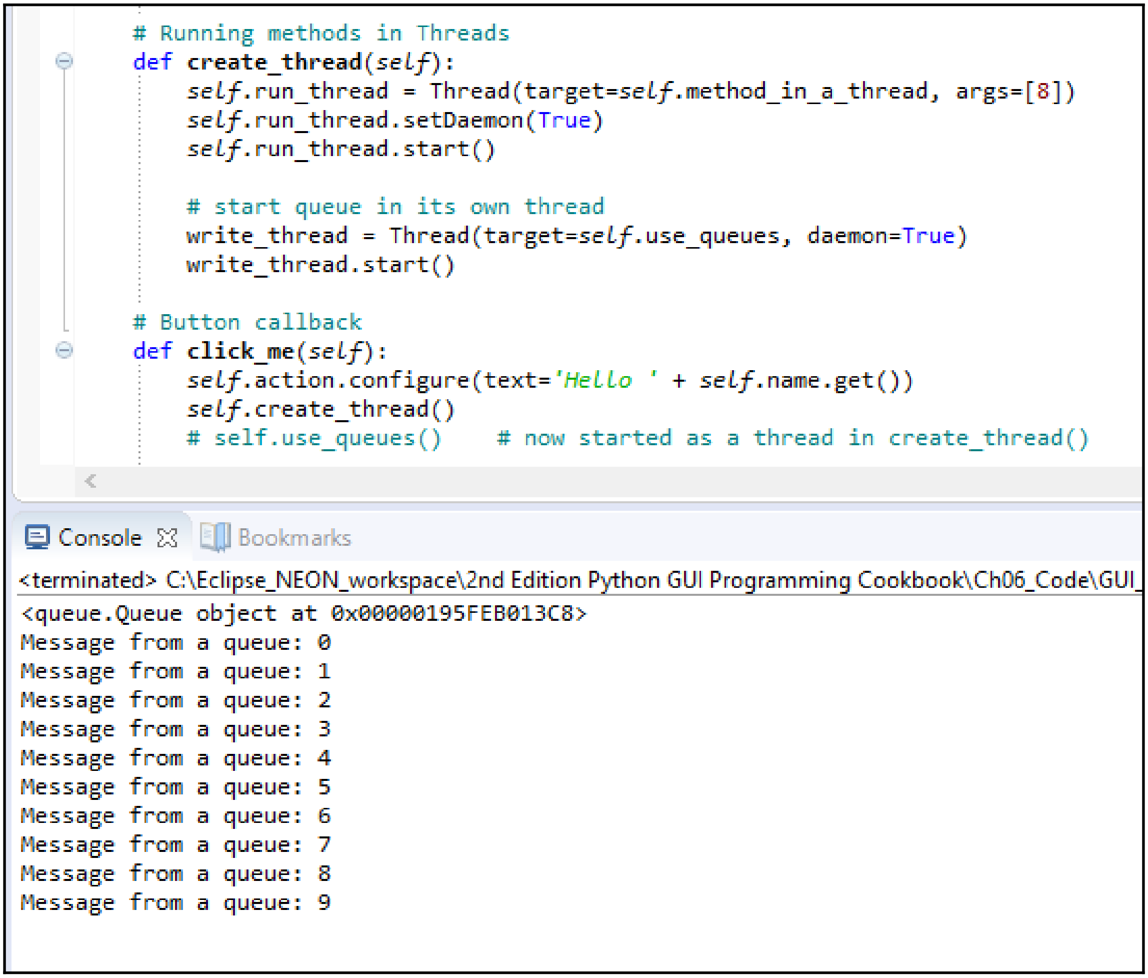
```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06_Code\GUI_queues.py
<queue.Queue object at 0x000001F5F7DE32E8>
Message from a queue: 0
```



```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06_Code\GUI_queues.py
<queue.Queue object at 0x0000017327A23400>
Message from a queue: 0
Message from a queue: 1
Message from a queue: 2
Message from a queue: 3
Message from a queue: 4
Message from a queue: 5
Message from a queue: 6
Message from a queue: 7
Message from a queue: 8
Message from a queue: 9
```

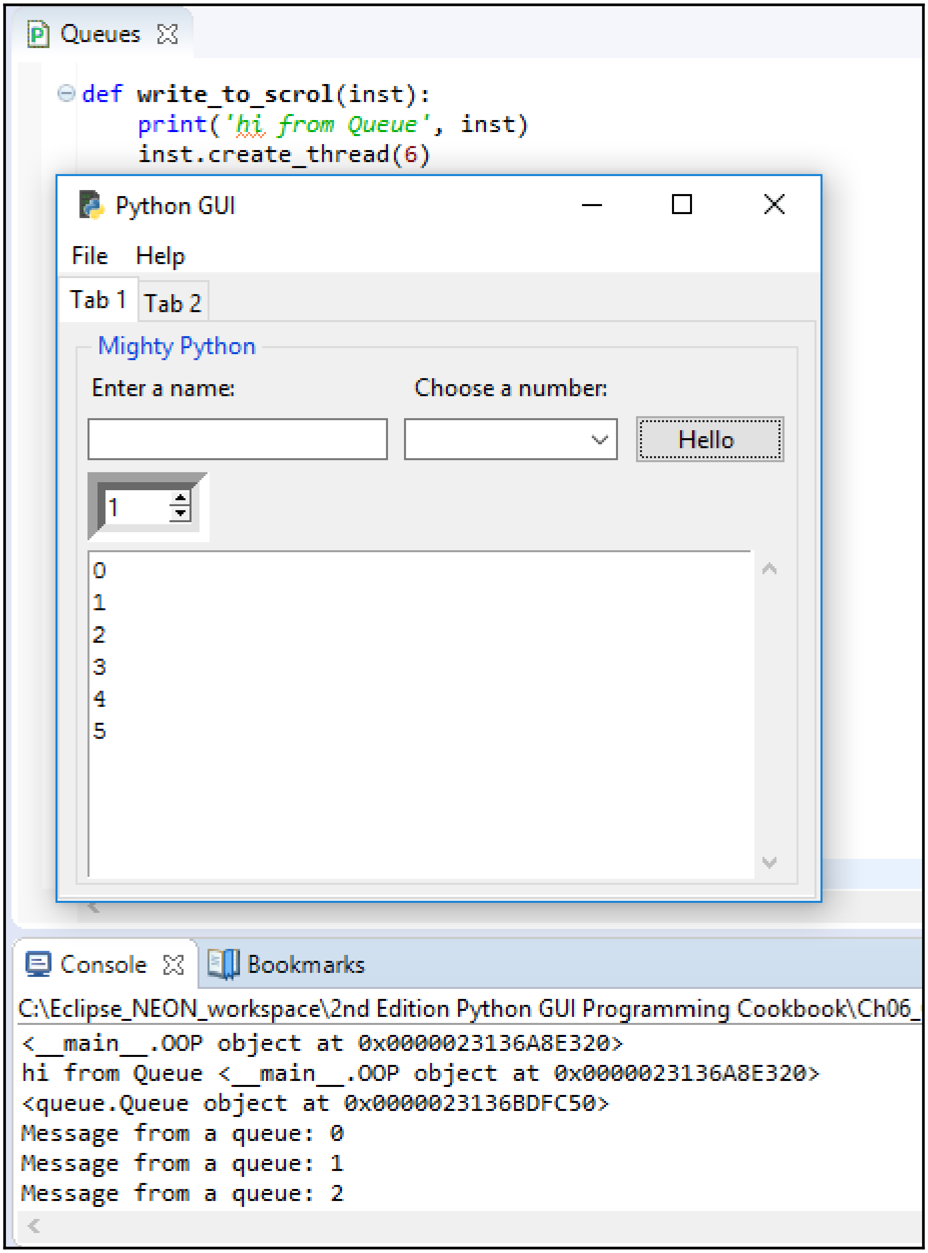
Python is not responding

If you close the program, you might lose information.



```
    # Running methods in Threads
    def create_thread(self):
        self.run_thread = Thread(target=self.method_in_a_thread, args=[8])
        self.run_thread.setDaemon(True)
        self.run_thread.start()

        # start queue in its own thread
        write_thread = Thread(target=self.use_queues, daemon=True)
        write_thread.start()

    # Button callback
    def click_me(self):
        self.action.configure(text='Hello ' + self.name.get())
        self.create_thread()
        # self.use_queues()    # now started as a thread in create_thread()
```

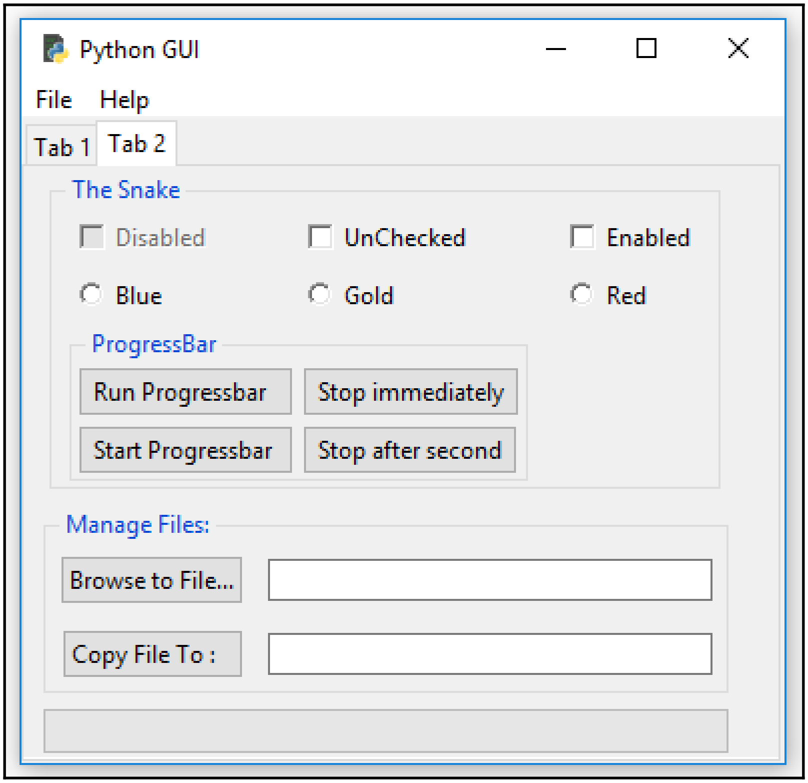
```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06_Code\GUI_
<queue.Queue object at 0x00000195FEB013C8>
Message from a queue: 0
Message from a queue: 1
Message from a queue: 2
Message from a queue: 3
Message from a queue: 4
Message from a queue: 5
Message from a queue: 6
Message from a queue: 7
Message from a queue: 8
Message from a queue: 9
```

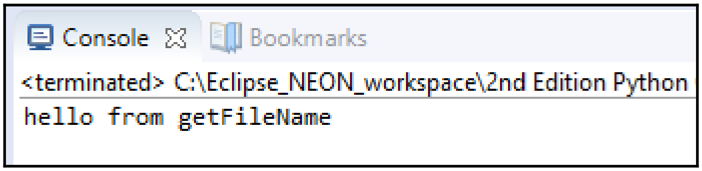


```
def write_to_scrol(inst):
    print('hi from Queue', inst)
    inst.create_thread(6)
```

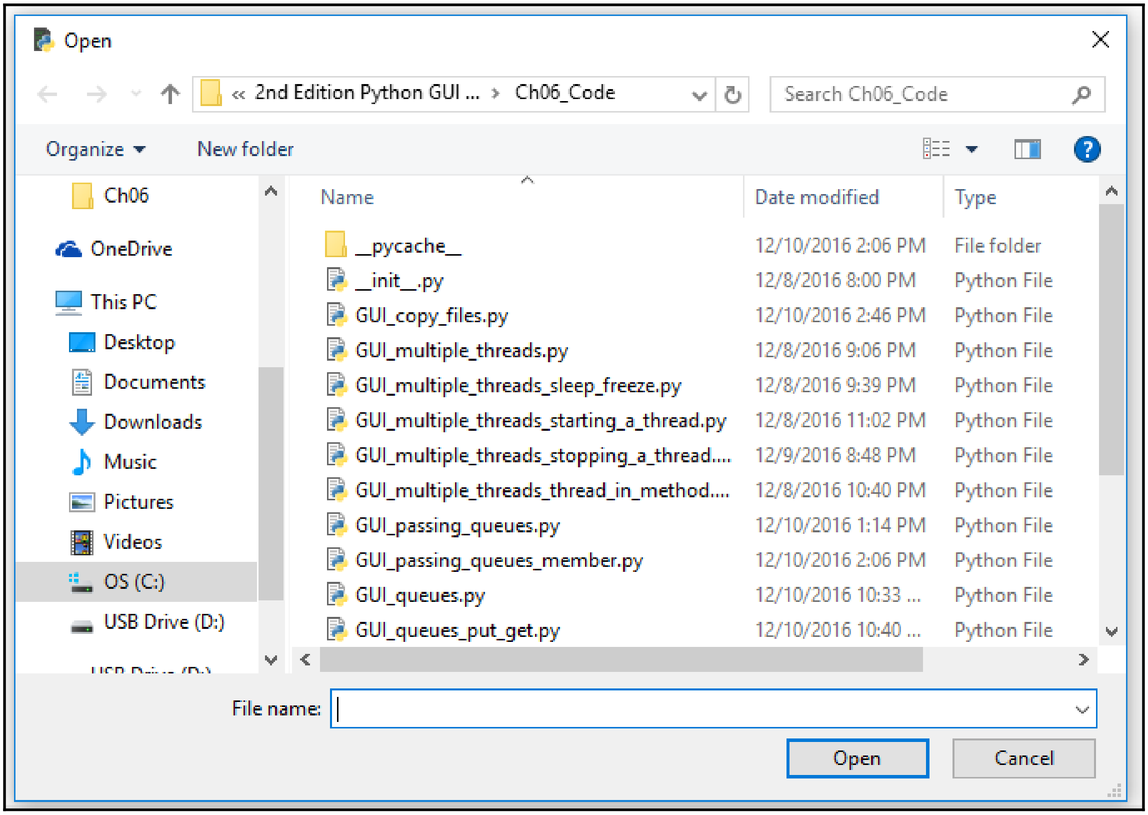
```
0
1
2
3
4
5
```

```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06_
<__main__.OOP object at 0x0000023136A8E320>
hi from Queue <__main__.OOP object at 0x0000023136A8E320>
<queue.Queue object at 0x0000023136BDFC50>
Message from a queue: 0
Message from a queue: 1
Message from a queue: 2
```

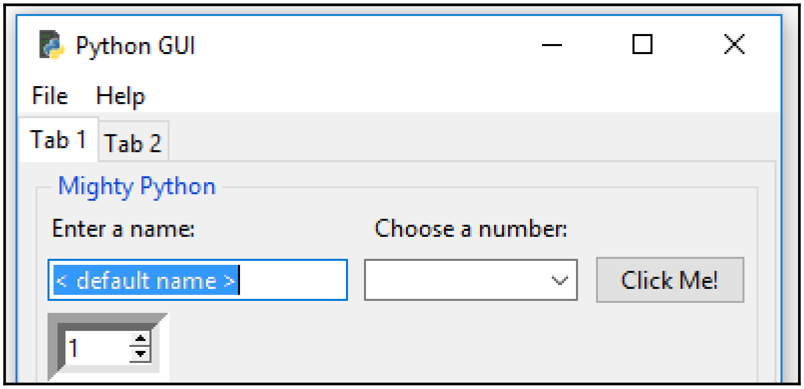




```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python
hello from getFileName
```

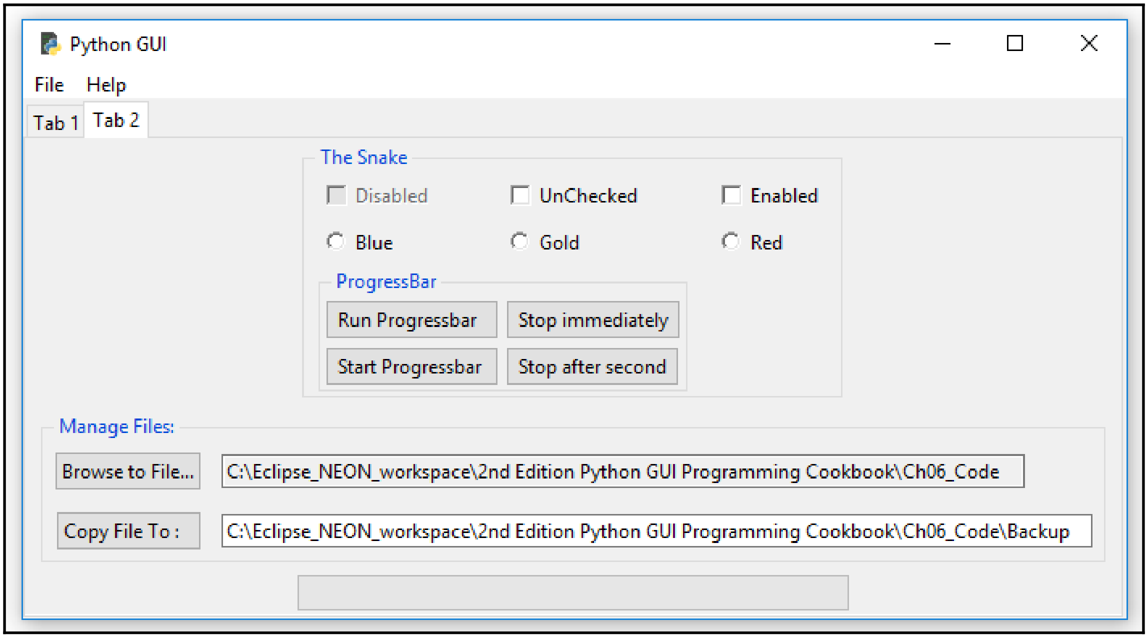


Name	Date modified	Type
__pycache__	12/10/2016 2:06 PM	File folder
__init__.py	12/8/2016 8:00 PM	Python File
GUI_copy_files.py	12/10/2016 2:46 PM	Python File
GUI_multiple_threads.py	12/8/2016 9:06 PM	Python File
GUI_multiple_threads_sleep_freeze.py	12/8/2016 9:39 PM	Python File
GUI_multiple_threads_starting_a_thread.py	12/8/2016 11:02 PM	Python File
GUI_multiple_threads_stopping_a_thread....	12/9/2016 8:48 PM	Python File
GUI_multiple_threads_thread_in_method....	12/8/2016 10:40 PM	Python File
GUI_passing_queues.py	12/10/2016 1:14 PM	Python File
GUI_passing_queues_member.py	12/10/2016 2:06 PM	Python File
GUI_queues.py	12/10/2016 10:33 ...	Python File
GUI_queues_put_get.py	12/10/2016 10:40 ...	Python File



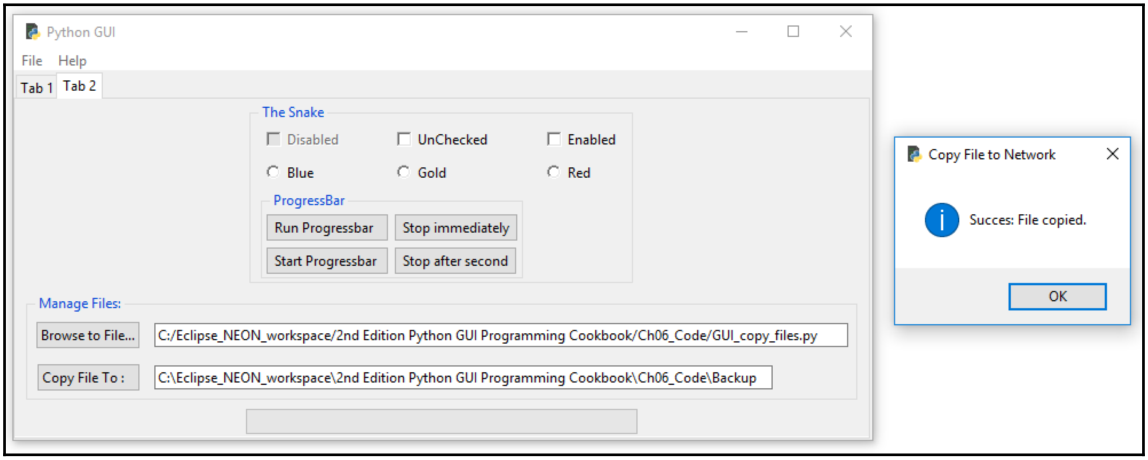
Enter a name: < default name >

1



C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06_Code

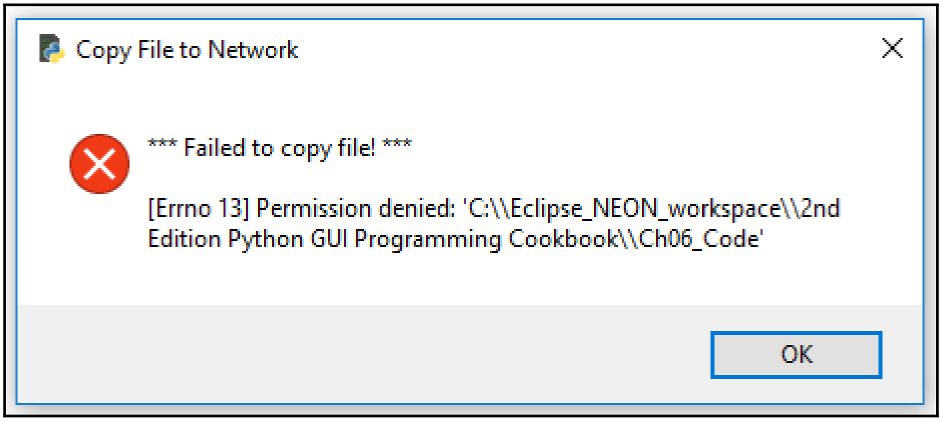
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06_Code\Backup



C:/Eclipse_NEON_workspace/2nd Edition Python GUI Programming Cookbook/Ch06_Code/GUI_copy_files.py

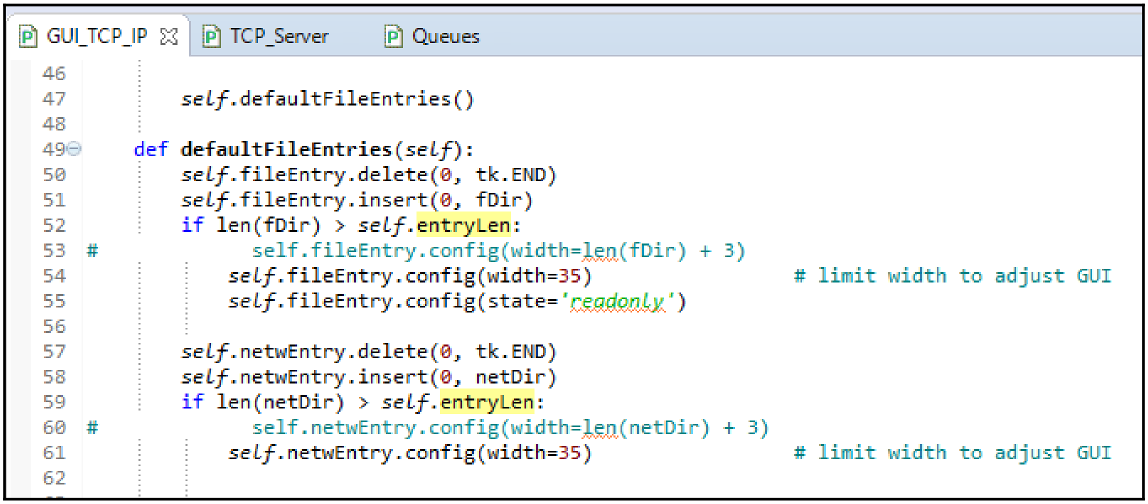
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06_Code\Backup

Succes: File copied.

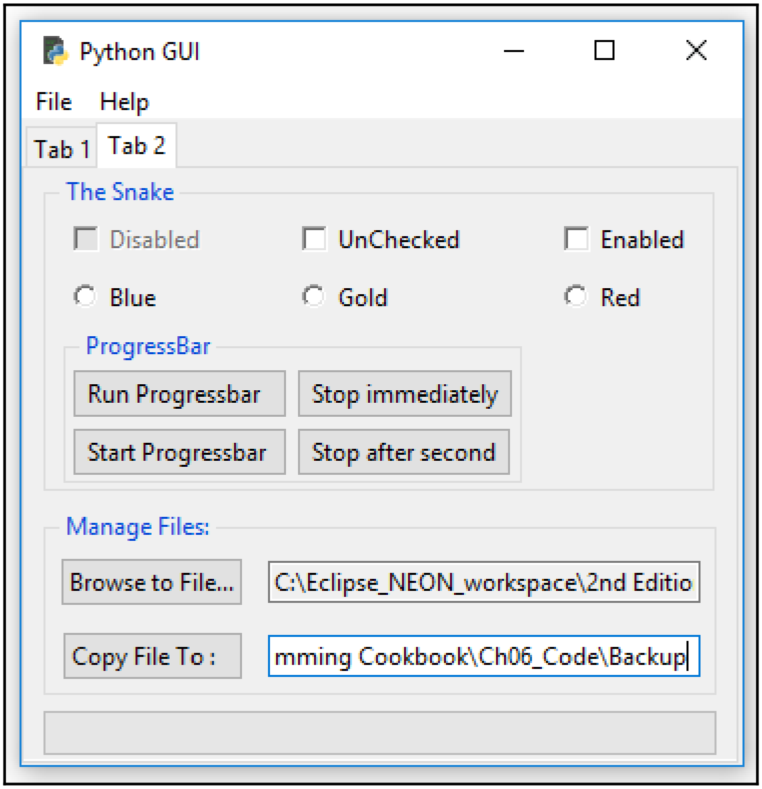


*** Failed to copy file! ***

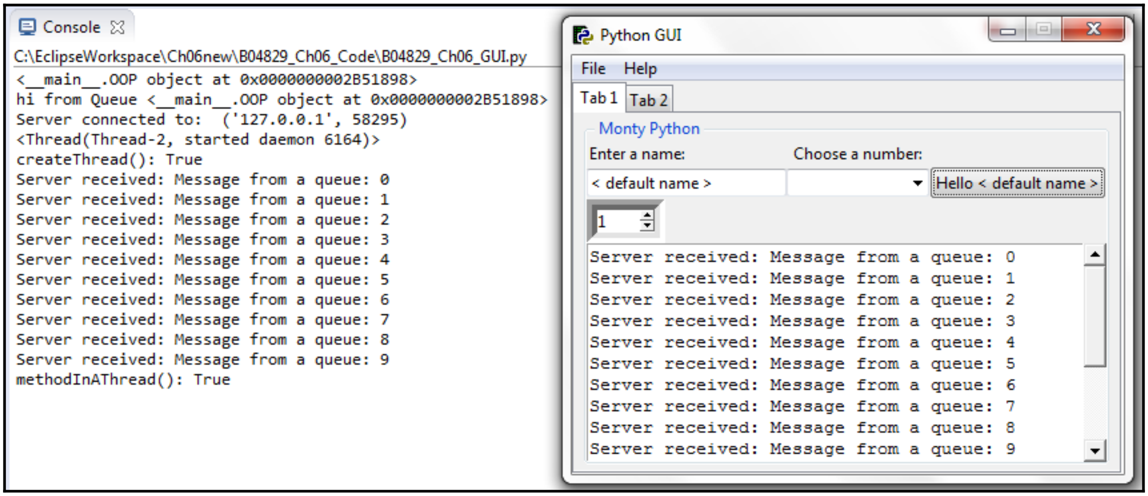
[Errno 13] Permission denied: 'C:\\\\Eclipse_NEON_workspace\\\\2nd Edition Python GUI Programming Cookbook\\\\Ch06_Code'



```
        self.defaultFileEntries()

    def defaultFileEntries(self):
        self.fileEntry.delete(0, tk.END)
        self.fileEntry.insert(0, fDir)
        if len(fDir) > self.entryLen:
#             self.fileEntry.config(width=len(fDir) + 3)
            self.fileEntry.config(width=35)                 # limit width to adjust GUI
            self.fileEntry.config(state='readonly')

        self.netwEntry.delete(0, tk.END)
        self.netwEntry.insert(0, netDir)
        if len(netDir) > self.entryLen:
#             self.netwEntry.config(width=len(netDir) + 3)
            self.netwEntry.config(width=35)                 # limit width to adjust GUI

```

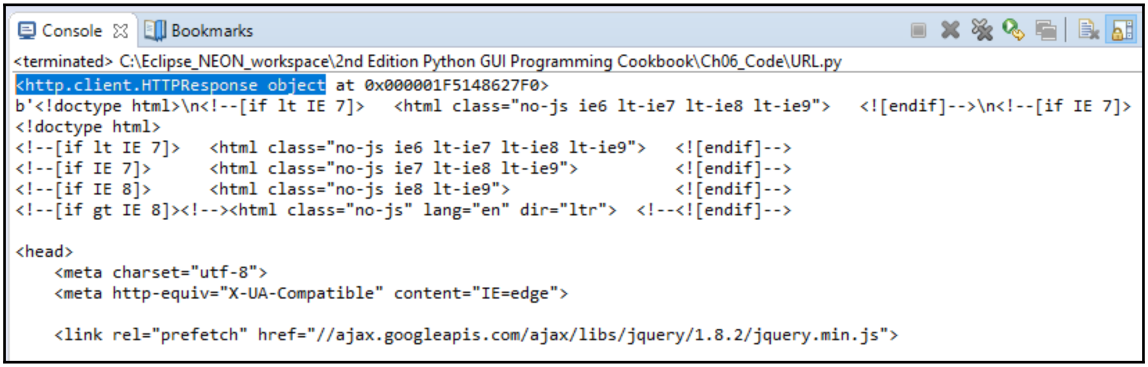


```
C:\Eclipse_NEON_workspace\2nd Editio
mming Cookbook\Ch06_Code\Backup
```

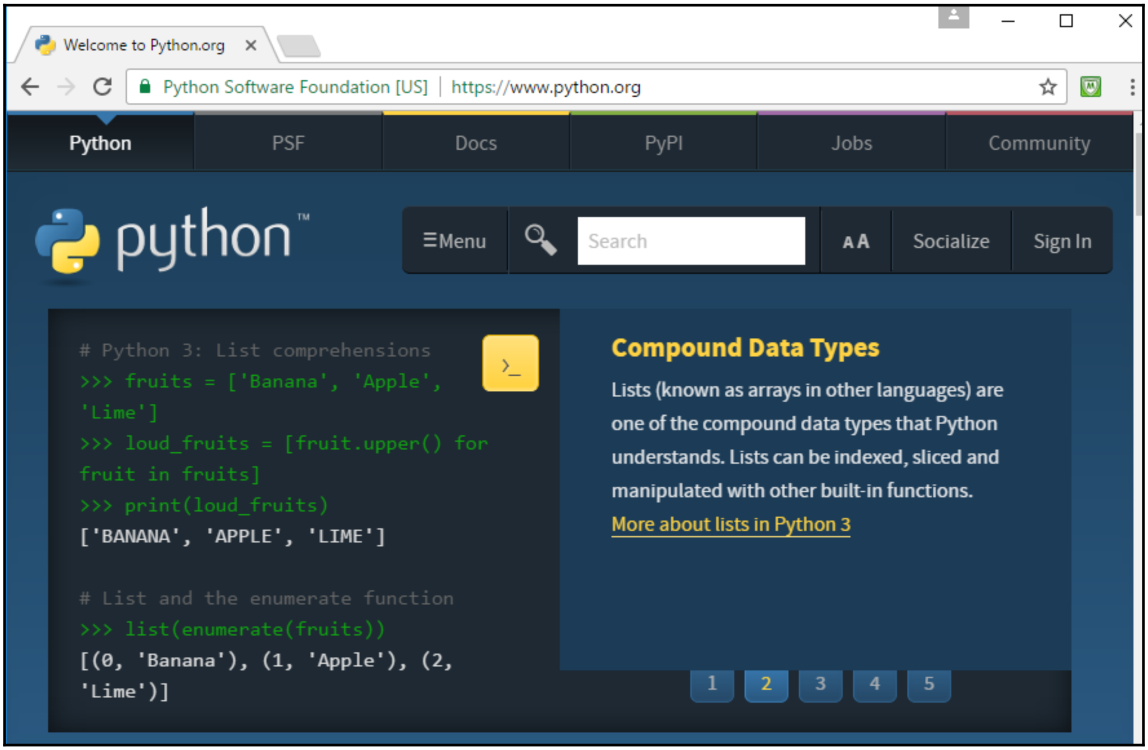


```
C:\EclipseWorkspace\Ch06new\B04829_Ch06_Code\B04829_Ch06_GUI.py
<__main__.OOP object at 0x0000000002B51898>
hi from Queue <__main__.OOP object at 0x0000000002B51898>
Server connected to:  ('127.0.0.1', 58295)
<Thread(Thread-2, started daemon 6164)>
createThread(): True
Server received: Message from a queue: 0
Server received: Message from a queue: 1
Server received: Message from a queue: 2
Server received: Message from a queue: 3
Server received: Message from a queue: 4
Server received: Message from a queue: 5
Server received: Message from a queue: 6
Server received: Message from a queue: 7
Server received: Message from a queue: 8
Server received: Message from a queue: 9
methodInAThread(): True
```

```
< default name >
1
Server received: Message from a queue: 0
Server received: Message from a queue: 1
Server received: Message from a queue: 2
Server received: Message from a queue: 3
Server received: Message from a queue: 4
Server received: Message from a queue: 5
Server received: Message from a queue: 6
Server received: Message from a queue: 7
Server received: Message from a queue: 8
Server received: Message from a queue: 9
```



```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch06_Code\URL.py
<http.client.HTTPResponse object at 0x000001F5148627F0>
b'<!doctype html>\n<!--[if lt IE 7]>   <html class="no-js ie6 lt-ie7 lt-ie8 lt-ie9">   <![endif]-->\n<!--[if IE 7]>
<!doctype html>
<!--[if lt IE 7]>   <html class="no-js ie6 lt-ie7 lt-ie8 lt-ie9">   <![endif]-->
<!--[if IE 7]>      <html class="no-js ie7 lt-ie8 lt-ie9">          <![endif]-->
<!--[if IE 8]>      <html class="no-js ie8 lt-ie9">                 <![endif]-->
<!--[if gt IE 8]><!--><html class="no-js" lang="en" dir="ltr">  <!--<![endif]-->

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <link rel="prefetch" href="//ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js">
```

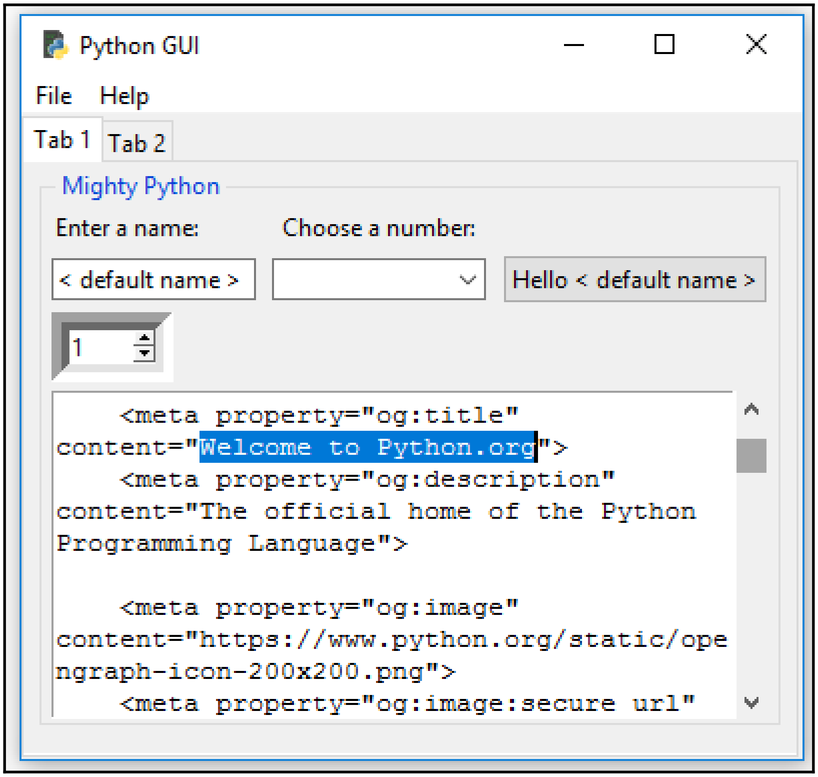


```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple',
'Lime']
>>> loud_fruits = [fruit.upper() for
fruit in fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2,
'Lime')]
```

Compound Data Types

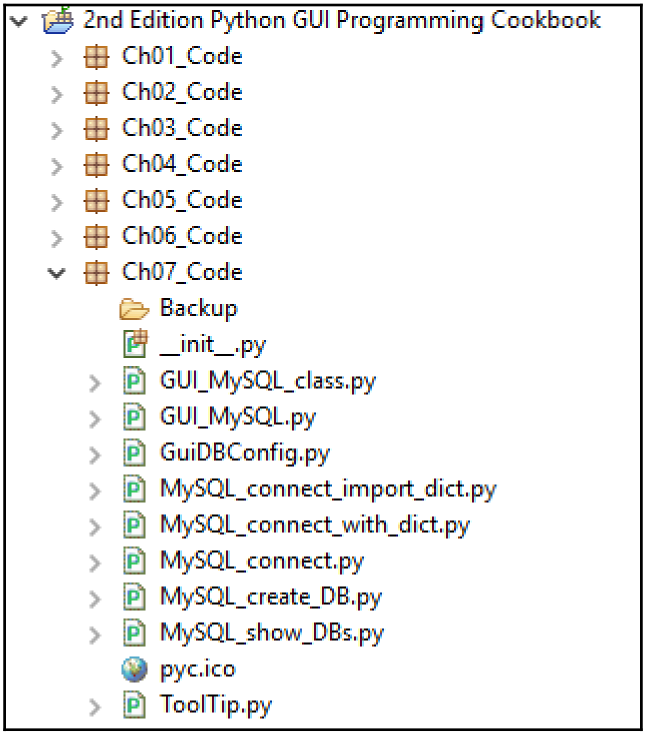
Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions.

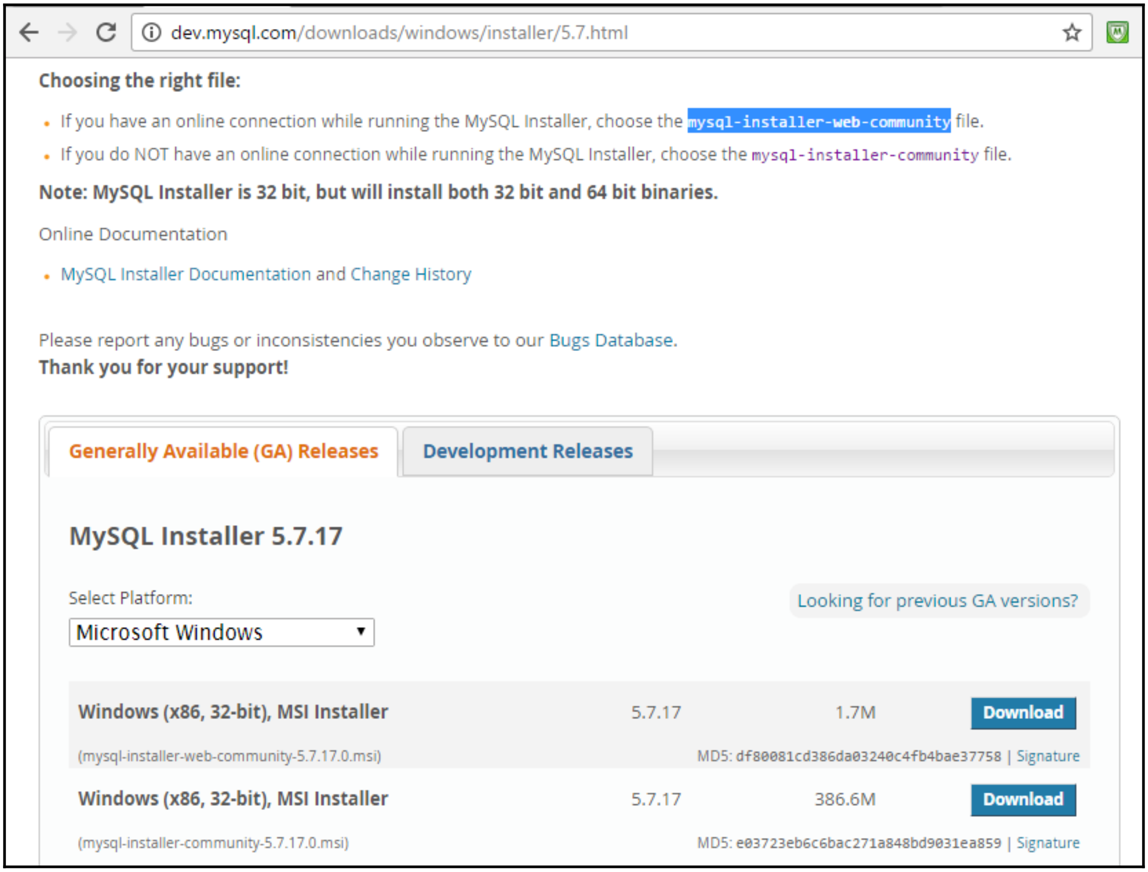
More about lists in Python 3



```
    <meta property="og:title"
content="Welcome to Python.org">
    <meta property="og:description"
content="The official home of the Python
Programming Language">

    <meta property="og:image"
content="https://www.python.org/static/ope
ngraph-icon-200x200.png">
    <meta property="og:image:secure url"
```

Chapter 7:
Storing Data in our MySQL Database via our GUI





Choosing the right file:

- If you have an online connection while running the MySQL Installer, choose the `mysql-installer-web-community` file.
- If you do NOT have an online connection while running the MySQL Installer, choose the `mysql-installer-community` file.

Note: MySQL Installer is 32 bit, but will install both 32 bit and 64 bit binaries.

Online Documentation

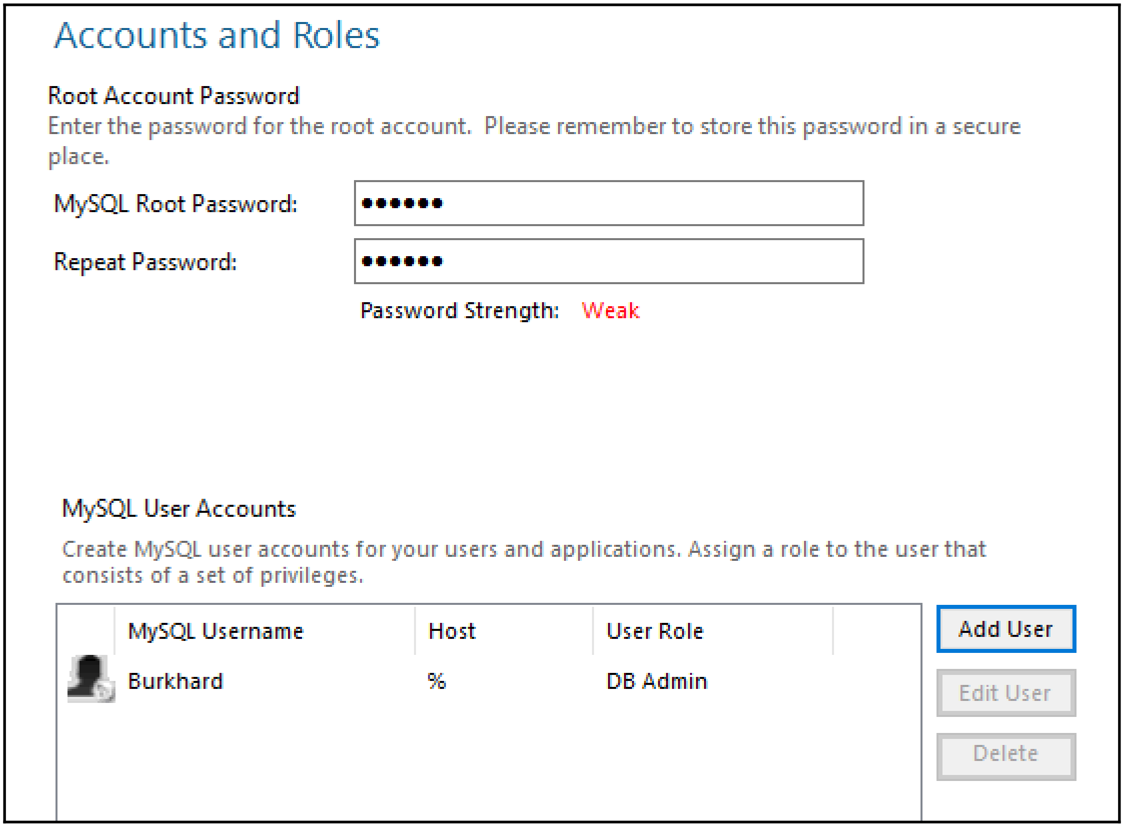
- MySQL Installer Documentation and Change History

Please report any bugs or inconsistencies you observe to our Bugs Database.
Thank you for your support!

MySQL Installer 5.7.17

Select Platform: Microsoft Windows ▾

Windows (x86, 32-bit), MSI Installer	5.7.17	1.7M	Download
(mysql-installer-web-community-5.7.17.0.msi)		MD5: `df80081cd386da03240c4fb4bae37758` | Signature	
Windows (x86, 32-bit), MSI Installer	5.7.17	386.6M	Download
(mysql-installer-community-5.7.17.0.msi)		MD5: `e03723eb6c6bac271a848bd9031ea859` | Signature	



Accounts and Roles

Root Account Password

Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password: ••••••

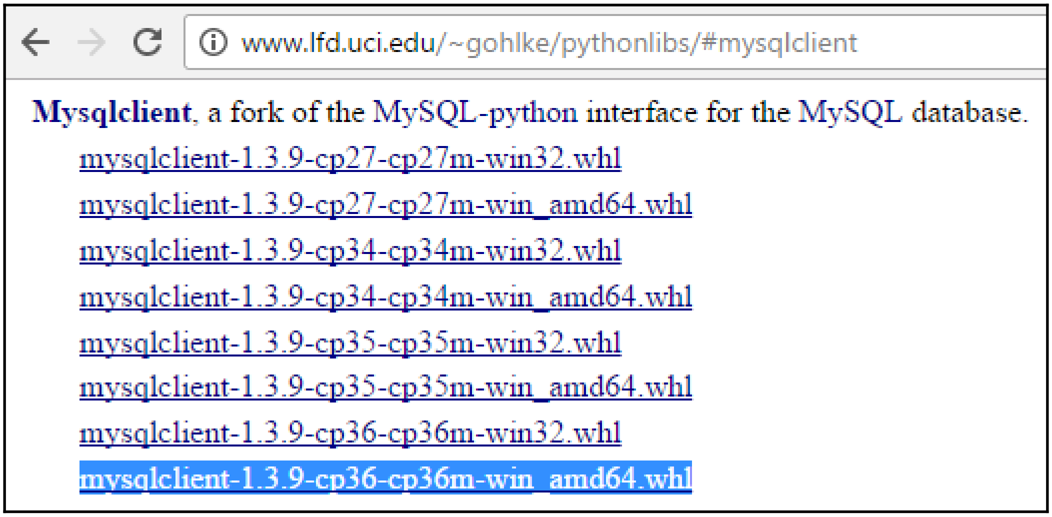
Repeat Password: ••••••

Password Strength: Weak

MySQL User Accounts

Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

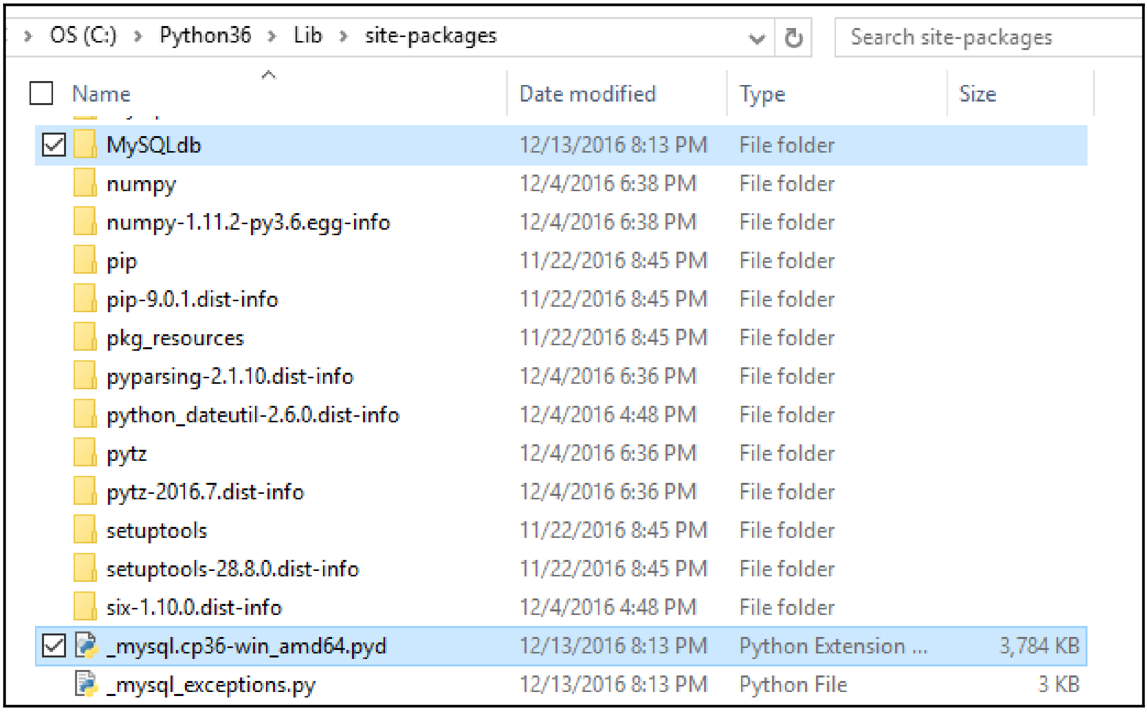
MySQL Username	Host	User Role
Burkhard	%	DB Admin



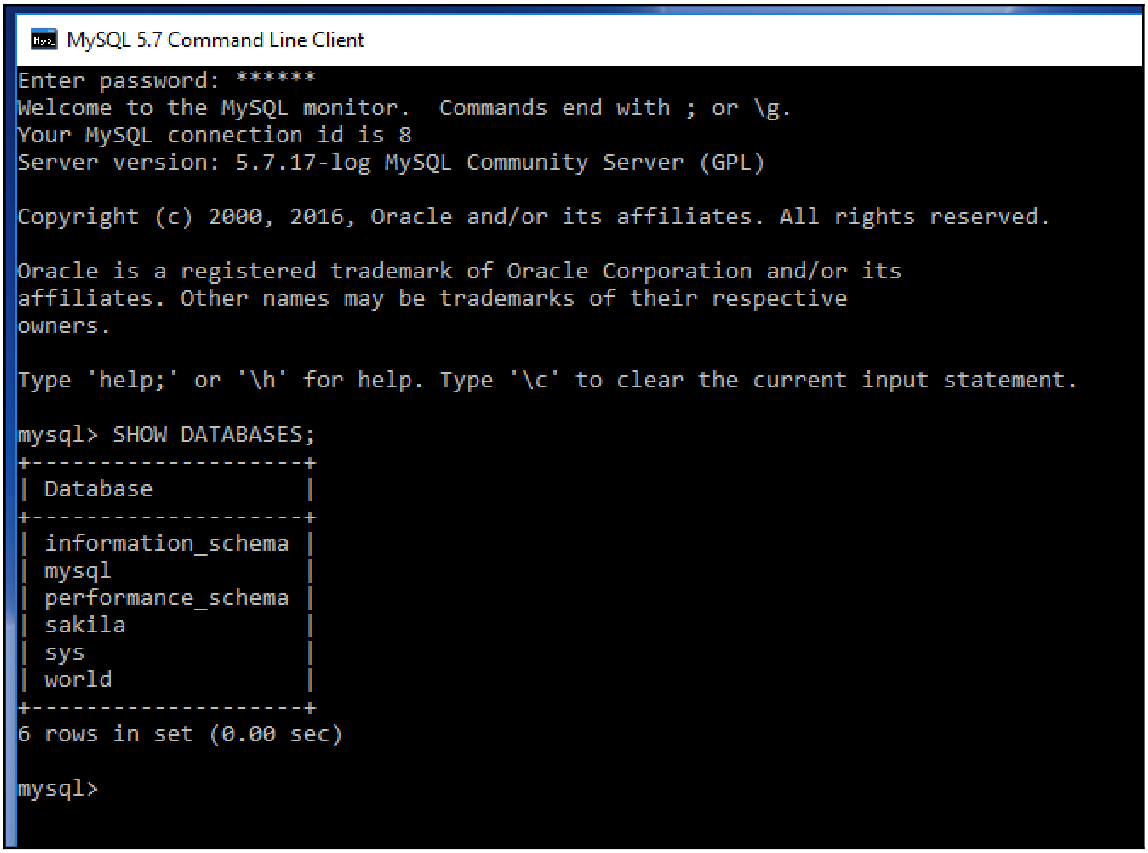
www.lfd.uci.edu/~gohlke/pythonlibs/#mysqlclient

Mysqlclient, a fork of the MySQL-python interface for the MySQL database.

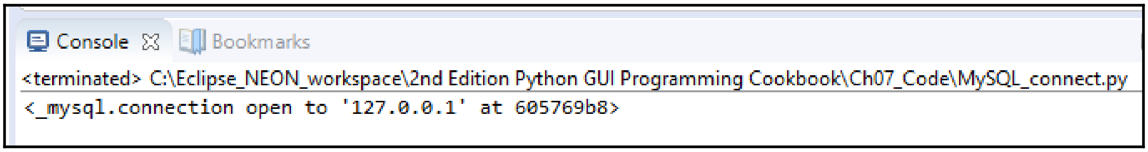
- mysqlclient-1.3.9-cp27-cp27m-win32.whl
- mysqlclient-1.3.9-cp27-cp27m-win_amd64.whl
- mysqlclient-1.3.9-cp34-cp34m-win32.whl
- mysqlclient-1.3.9-cp34-cp34m-win_amd64.whl
- mysqlclient-1.3.9-cp35-cp35m-win32.whl
- mysqlclient-1.3.9-cp35-cp35m-win_amd64.whl
- mysqlclient-1.3.9-cp36-cp36m-win32.whl
- mysqlclient-1.3.9-cp36-cp36m-win_amd64.whl



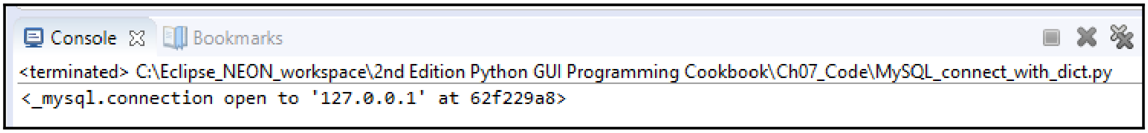
□	Name	Date modified	Type	Size
☑	MySQLdb	12/13/2016 8:13 PM	File folder	
	numpy	12/4/2016 6:38 PM	File folder	
	numpy-1.11.2-py3.6.egg-info	12/4/2016 6:38 PM	File folder	
	pip	11/22/2016 8:45 PM	File folder	
	pip-9.0.1.dist-info	11/22/2016 8:45 PM	File folder	
	pkg_resources	11/22/2016 8:45 PM	File folder	
	pyparsing-2.1.10.dist-info	12/4/2016 6:36 PM	File folder	
	python_dateutil-2.6.0.dist-info	12/4/2016 4:48 PM	File folder	
	pytz	12/4/2016 6:36 PM	File folder	
	pytz-2016.7.dist-info	12/4/2016 6:36 PM	File folder	
	setuptools	11/22/2016 8:45 PM	File folder	
	setuptools-28.8.0.dist-info	11/22/2016 8:45 PM	File folder	
	six-1.10.0.dist-info	12/4/2016 4:48 PM	File folder	
☑	_mysql.cp36-win_amd64.pyd	12/13/2016 8:13 PM	Python Extension ...	3,784 KB
	_mysql_exceptions.py	12/13/2016 8:13 PM	Python File	3 KB



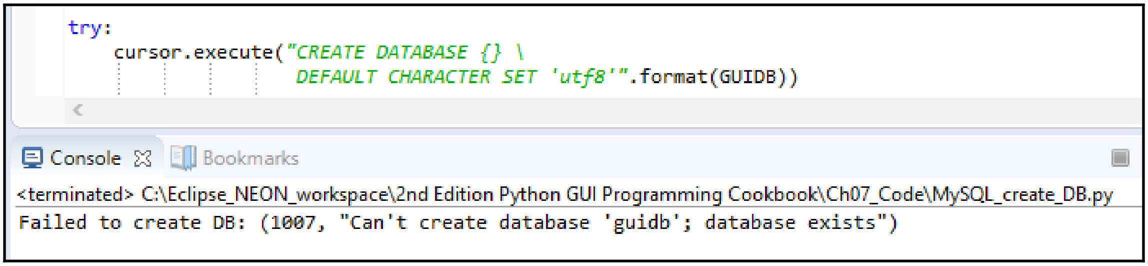
```
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.17-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| world              |
+--------------------+
6 rows in set (0.00 sec)

mysql>
```



```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch07_Code\MySQL_connect.py
<_mysql.connection open to '127.0.0.1' at 605769b8>
```

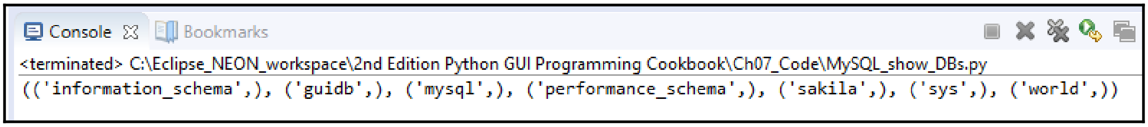


```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch07_Code\MySQL_connect_with_dict.py
<_mysql.connection open to '127.0.0.1' at 62f229a8>
```

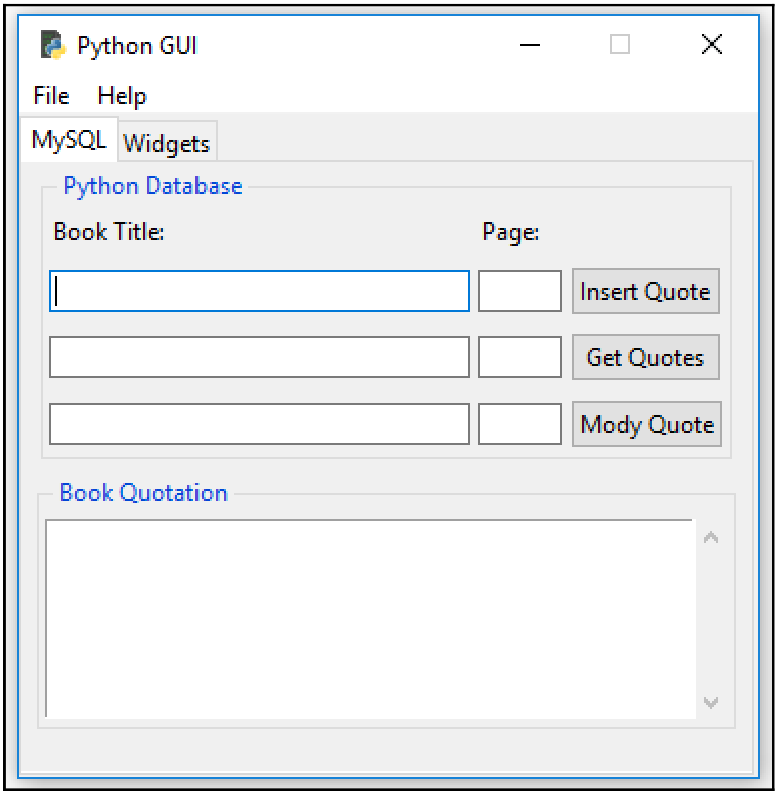


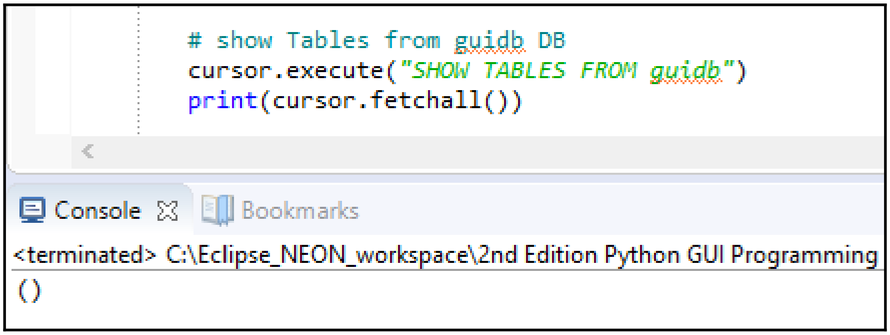
```
try:
    cursor.execute("CREATE DATABASE {} \
                DEFAULT CHARACTER SET 'utf8'".format(GUIDB))
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch07_Code\MySQL_create_DB.py
Failed to create DB: (1007, "Can't create database 'guidb'; database exists")
```



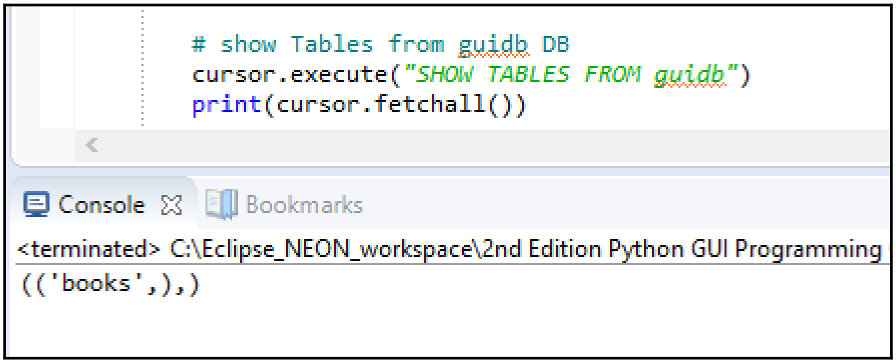
```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch07_Code\MySQL_show_DBs.py
(('information_schema',), ('guidb',), ('mysql',), ('performance_schema',), ('sakila',), ('sys',), ('world',))
```





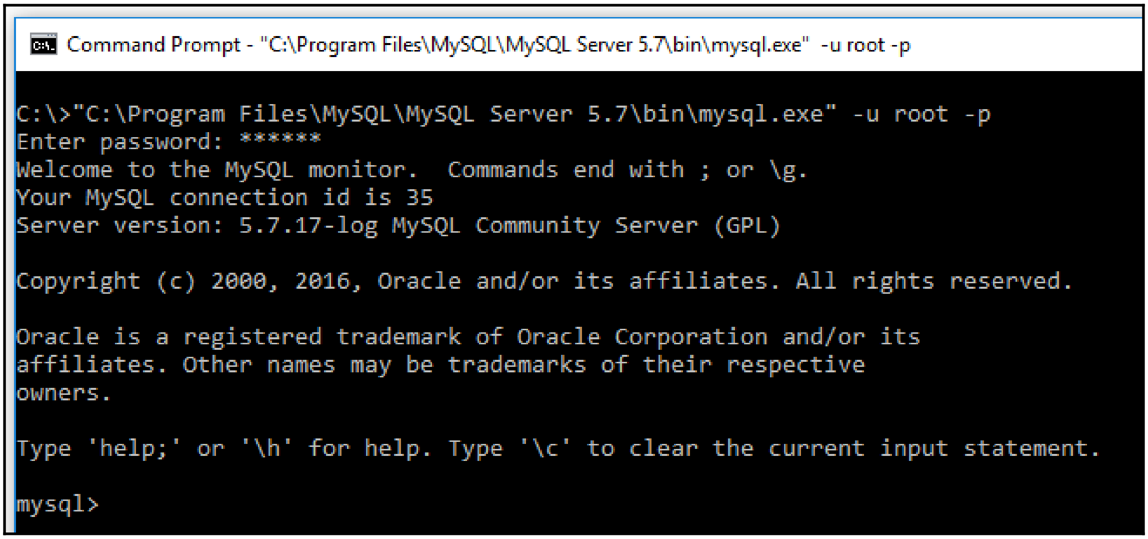
```
# show Tables from guidb DB
cursor.execute("SHOW TABLES FROM guidb")
print(cursor.fetchall())
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming
()
```



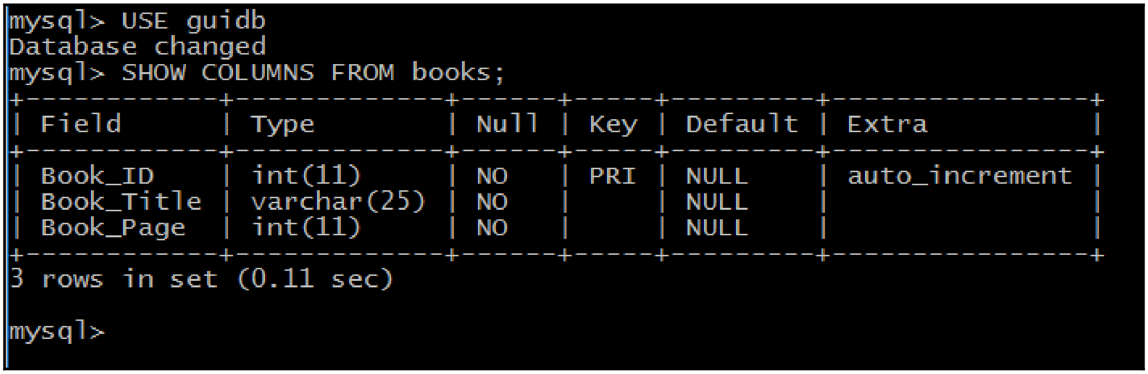
```
# show Tables from guidb DB
cursor.execute("SHOW TABLES FROM guidb")
print(cursor.fetchall())
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming
(('books',),)
```

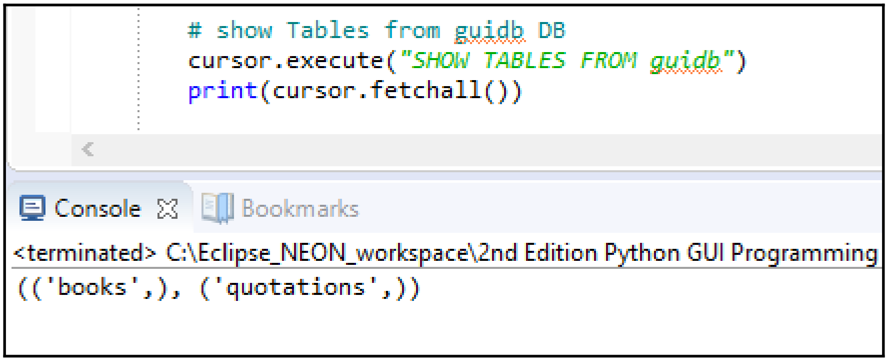


Command Prompt - "C:\Program Files\MySQL\MySQL Server 5.7\bin\mysql.exe" -u root -p

```
C:\>"C:\Program Files\MySQL\MySQL Server 5.7\bin\mysql.exe" -u root -p
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 35
Server version: 5.7.17-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

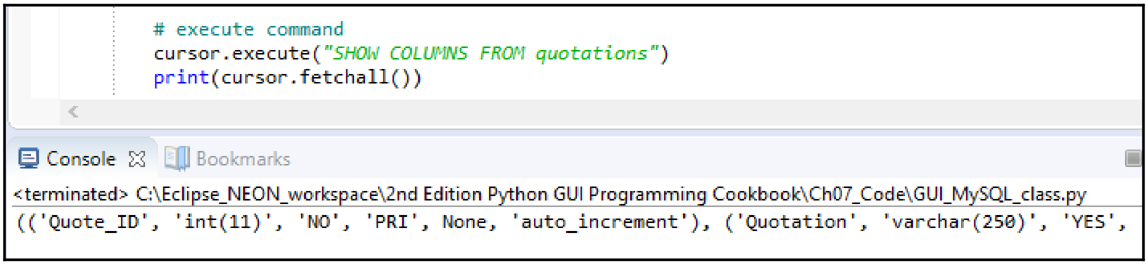


```
mysql> USE guidb
Database changed
mysql> SHOW COLUMNS FROM books;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| Book_ID    | int(11)      | NO   | PRI | NULL    | auto_increment |
| Book_Title | varchar(25)  | NO   |     | NULL    |                |
| Book_Page  | int(11)      | NO   |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
3 rows in set (0.11 sec)

mysql>
```



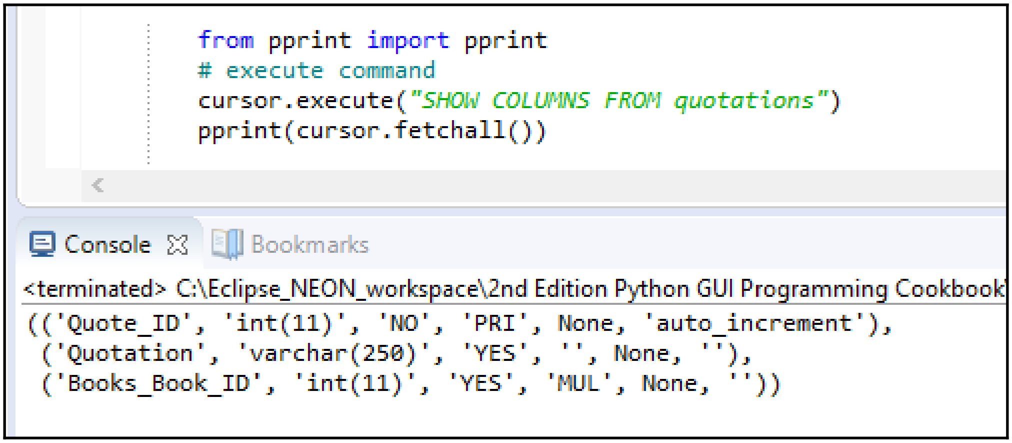
```
    # show Tables from guidb DB
    cursor.execute("SHOW TABLES FROM guidb")
    print(cursor.fetchall())
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming
(('books',), ('quotations',))
```



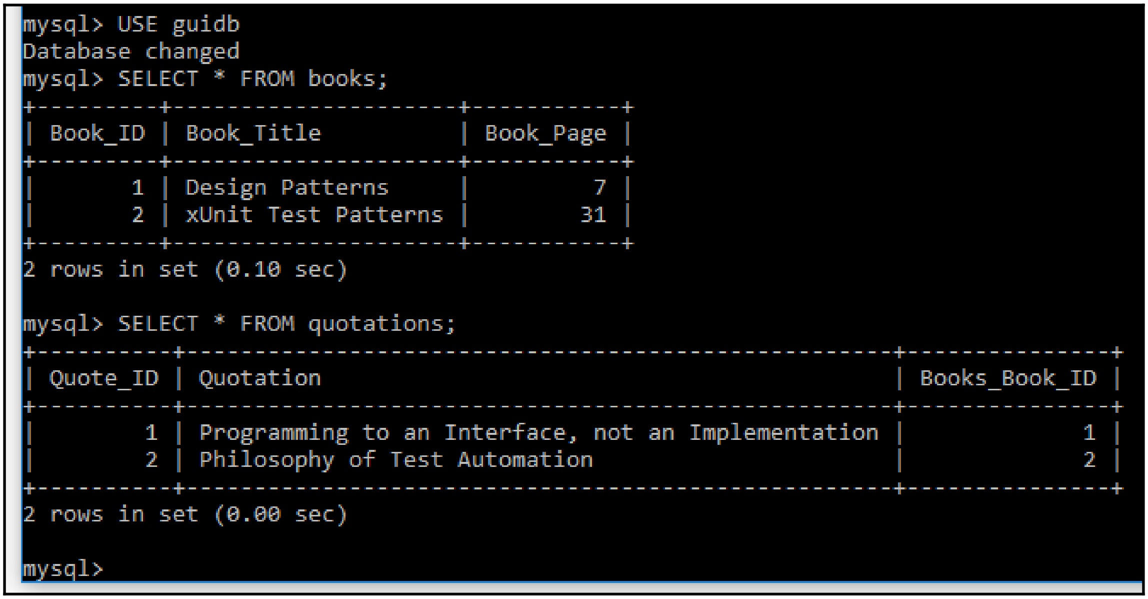
```
    # execute command
    cursor.execute("SHOW COLUMNS FROM quotations")
    print(cursor.fetchall())
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch07_Code\GUI_MySQL_class.py
(('Quote_ID', 'int(11)', 'NO', 'PRI', None, 'auto_increment'), ('Quotation', 'varchar(250)', 'YES',
```

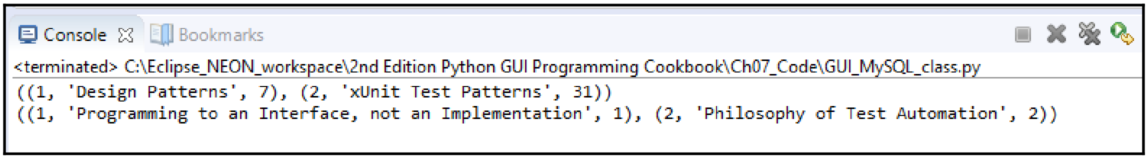


```
from pprint import pprint
# execute command
cursor.execute("SHOW COLUMNS FROM quotations")
pprint(cursor.fetchall())
```

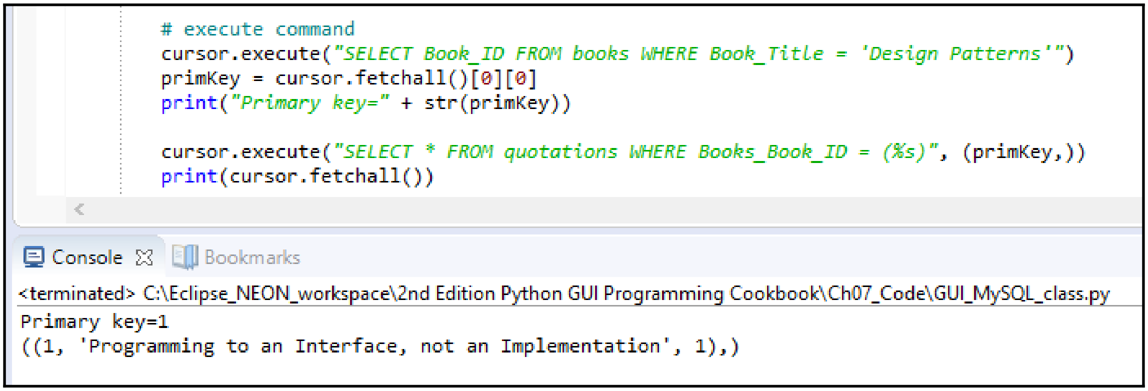
```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook
(('Quote_ID', 'int(11)', 'NO', 'PRI', None, 'auto_increment'),
 ('Quotation', 'varchar(250)', 'YES', '', None, ''),
 ('Books_Book_ID', 'int(11)', 'YES', 'MUL', None, ''))
```



```
mysql> USE guidb
Database changed
mysql> SELECT * FROM books;
+---------+---------------------+----------+
| Book_ID | Book_Title          | Book_Page |
+---------+---------------------+----------+
|       1 | Design Patterns     |        7 |
|       2 | xUnit Test Patterns |       31 |
+---------+---------------------+----------+
2 rows in set (0.10 sec)

mysql> SELECT * FROM quotations;
+----------+-------------------------------------------------+---------------+
| Quote_ID | Quotation                                       | Books_Book_ID |
+----------+-------------------------------------------------+---------------+
|        1 | Programming to an Interface, not an Implementation |             1 |
|        2 | Philosophy of Test Automation                   |             2 |
+----------+-------------------------------------------------+---------------+
2 rows in set (0.00 sec)

mysql>
```

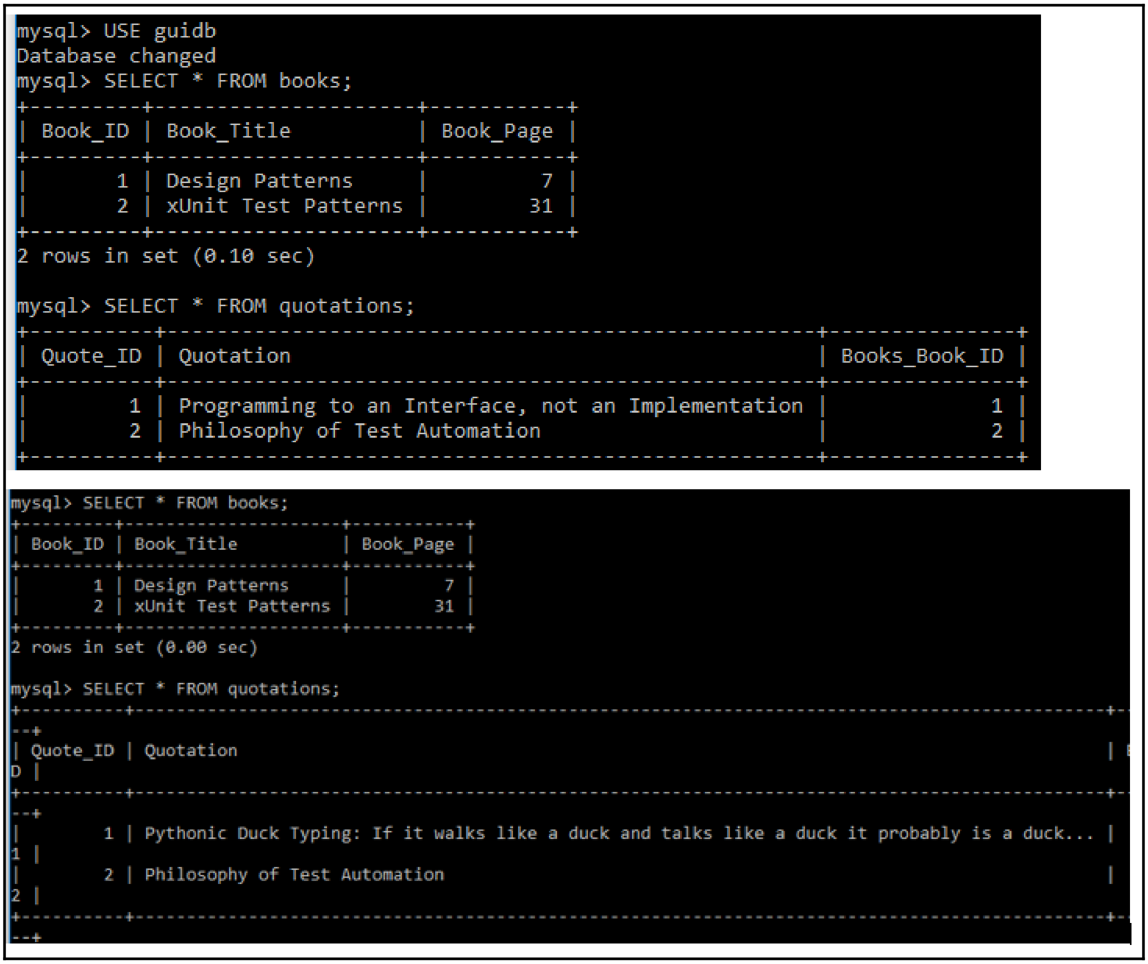


```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch07_Code\GUI_MySQL_class.py
((1, 'Design Patterns', 7), (2, 'xUnit Test Patterns', 31))
((1, 'Programming to an Interface, not an Implementation', 1), (2, 'Philosophy of Test Automation', 2))
```



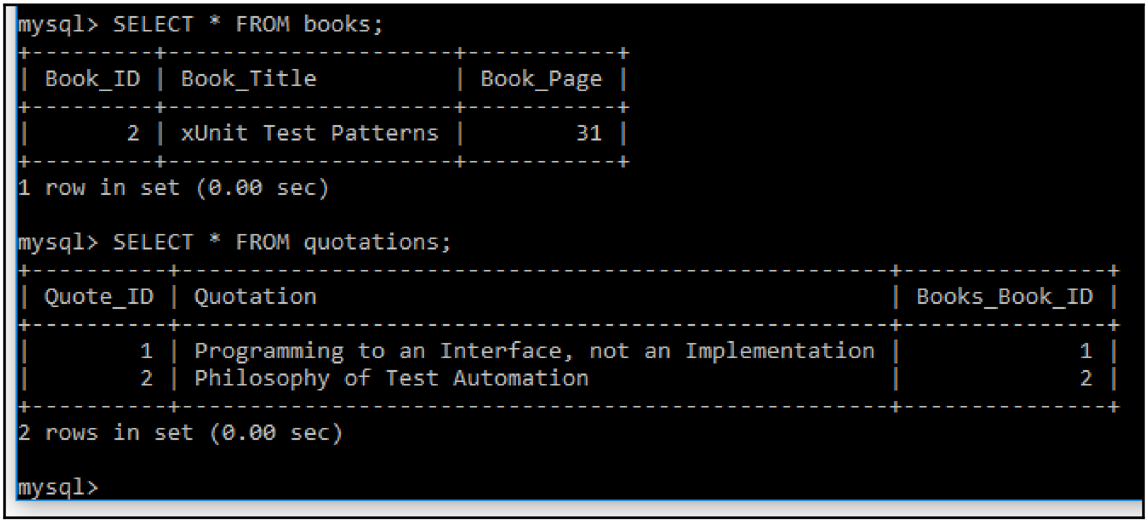
```
        # execute command
        cursor.execute("SELECT Book_ID FROM books WHERE Book_Title = 'Design Patterns'")
        primKey = cursor.fetchall()[0][0]
        print("Primary key=" + str(primKey))

        cursor.execute("SELECT * FROM quotations WHERE Books_Book_ID = (%s)", (primKey,))
        print(cursor.fetchall())
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch07_Code\GUI_MySQL_class.py
Primary key=1
((1, 'Programming to an Interface, not an Implementation', 1),)
```

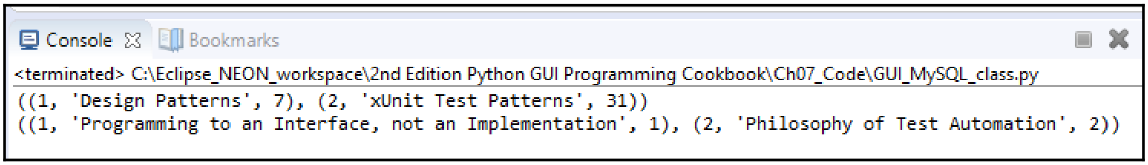


```
mysql> USE guidb
Database changed
mysql> SELECT * FROM books;
+---------+---------------------+-----------+
| Book_ID | Book_Title          | Book_Page |
+---------+---------------------+-----------+
|       1 | Design Patterns     |         7 |
|       2 | xUnit Test Patterns |        31 |
+---------+---------------------+-----------+
2 rows in set (0.10 sec)

mysql> SELECT * FROM quotations;
+----------+----------------------------------------------------+---------------+
| Quote_ID | Quotation                                          | Books_Book_ID |
+----------+----------------------------------------------------+---------------+
|        1 | Programming to an Interface, not an Implementation |             1 |
|        2 | Philosophy of Test Automation                      |             2 |
+----------+----------------------------------------------------+---------------+
```

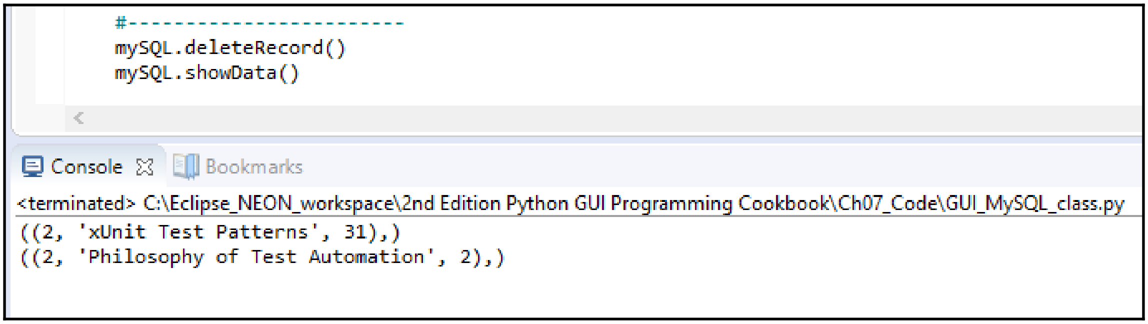
```
mysql> SELECT * FROM books;
+---------+---------------------+-----------+
| Book_ID | Book_Title          | Book_Page |
+---------+---------------------+-----------+
|       1 | Design Patterns     |         7 |
|       2 | xUnit Test Patterns |        31 |
+---------+---------------------+-----------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM quotations;
+----------+----------------------------------------------------------------------------------------------+-
--+
| Quote_ID | Quotation                                                                                    | B
D |
+----------+----------------------------------------------------------------------------------------------+-
--+
|        1 | Pythonic Duck Typing: If it walks like a duck and talks like a duck it probably is a duck... |
1 |
|        2 | Philosophy of Test Automation                                                                |
2 |
+----------+----------------------------------------------------------------------------------------------+-
--+
```



```
mysql> SELECT * FROM books;
+---------+---------------------+-----------+
| Book_ID | Book_Title          | Book_Page |
+---------+---------------------+-----------+
|       2 | xUnit Test Patterns |        31 |
+---------+---------------------+-----------+
1 row in set (0.00 sec)

mysql> SELECT * FROM quotations;
+----------+------------------------------------------------------+---------------+
| Quote_ID | Quotation                                            | Books_Book_ID |
+----------+------------------------------------------------------+---------------+
|        1 | Programming to an Interface, not an Implementation |             1 |
|        2 | Philosophy of Test Automation                        |             2 |
+----------+------------------------------------------------------+---------------+
2 rows in set (0.00 sec)

mysql>
```

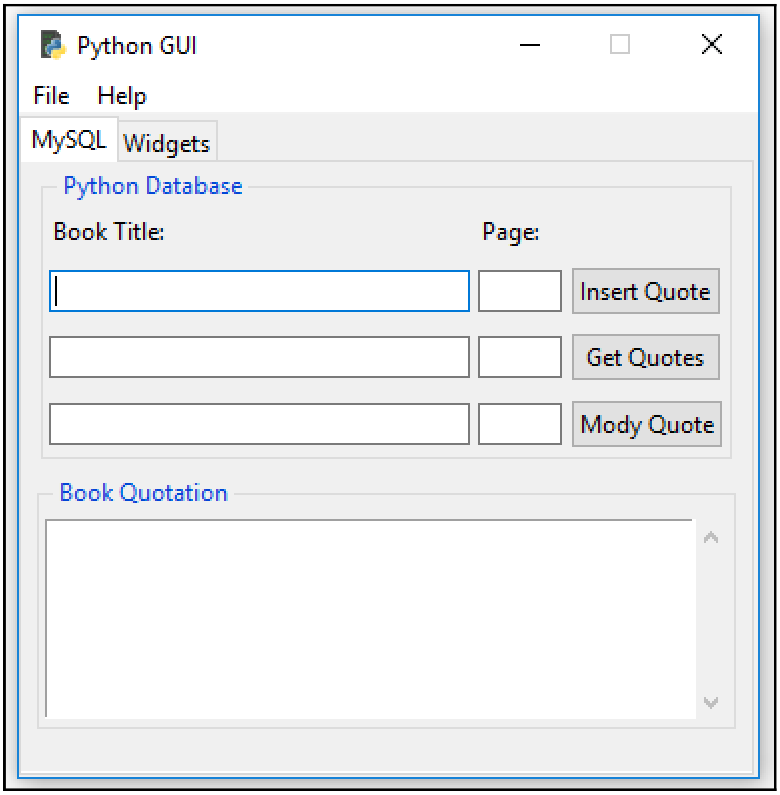


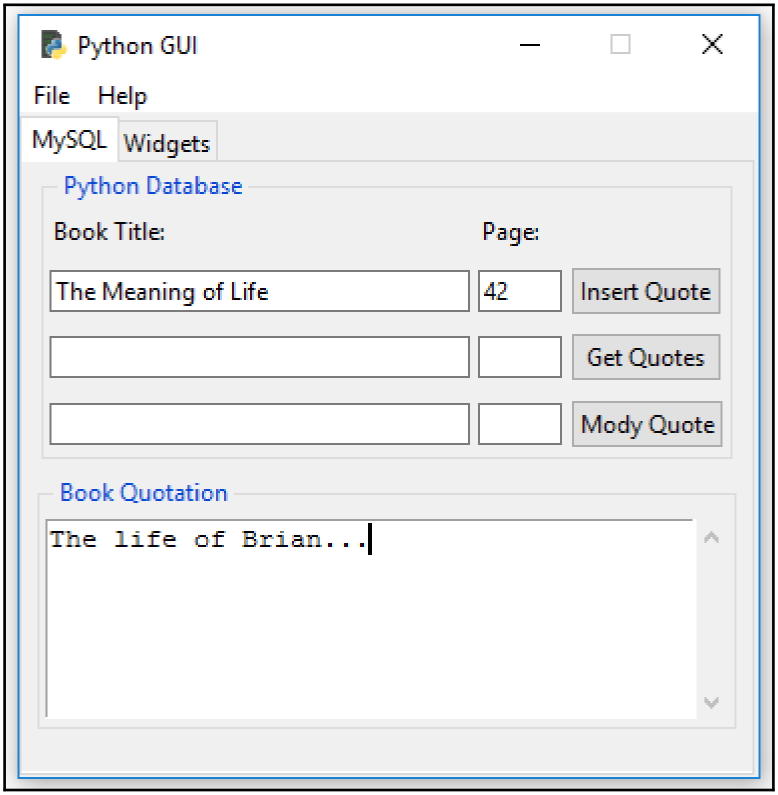
```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch07_Code\GUI_MySQL_class.py
((1, 'Design Patterns', 7), (2, 'xUnit Test Patterns', 31))
((1, 'Programming to an Interface, not an Implementation', 1), (2, 'Philosophy of Test Automation', 2))
```



```
    #-----------------------
    mySQL.deleteRecord()
    mySQL.showData()
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch07_Code\GUI_MySQL_class.py
((2, 'xUnit Test Patterns', 31),)
((2, 'Philosophy of Test Automation', 2),)
```

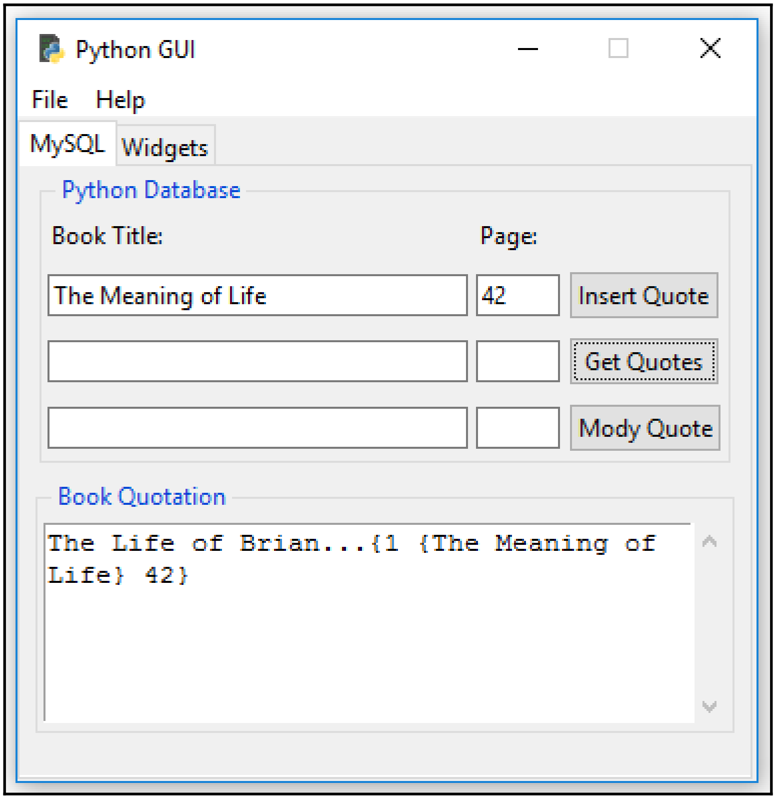

Python GUI
File
Help
MySQL
Widgets
Python Database
Book Title:
Page:
Insert Quote
Get Quotes
Mody Quote
Book Quotation



Book Title: The Meaning of Life

Page: 42

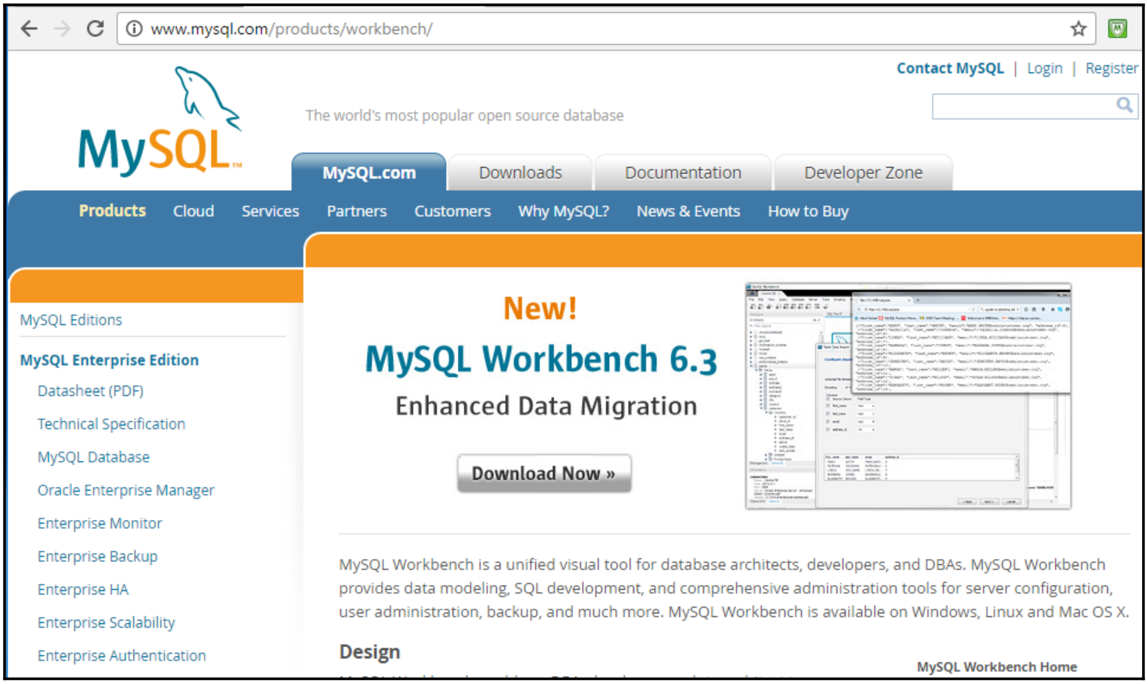
The life of Brian...



The Meaning of Life

42

```
The Life of Brian...{1 {The Meaning of
Life} 42}
```



The world's most popular open source database

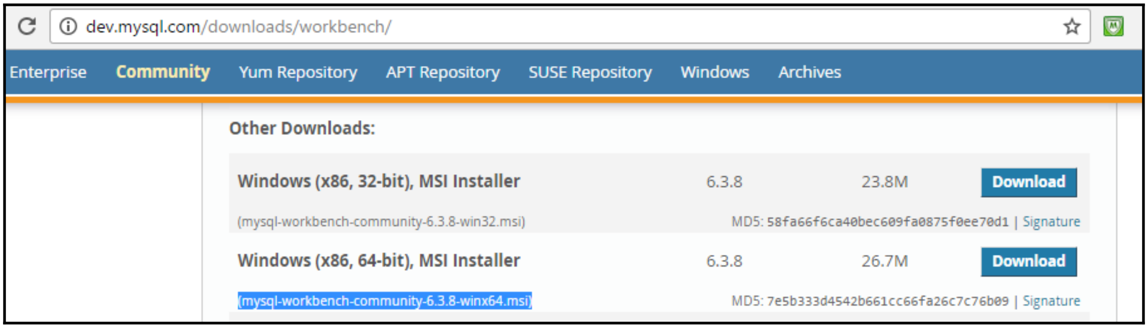
New!

MySQL Workbench 6.3

Enhanced Data Migration

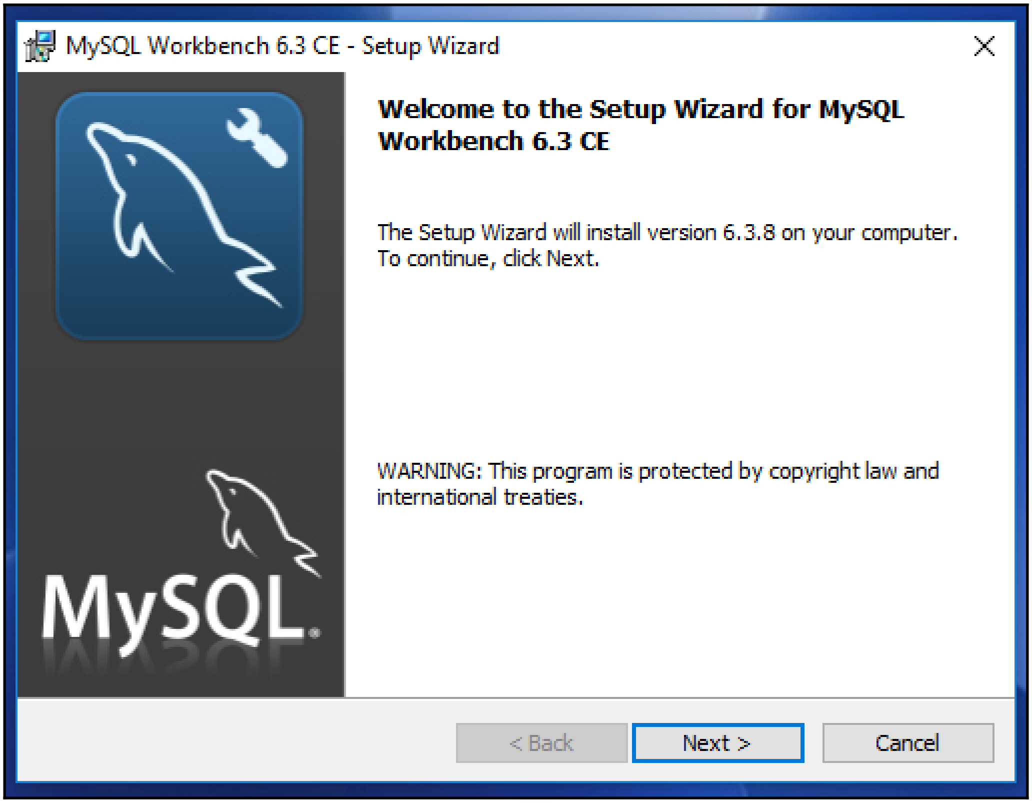
MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OS X.

Design



Other Downloads:

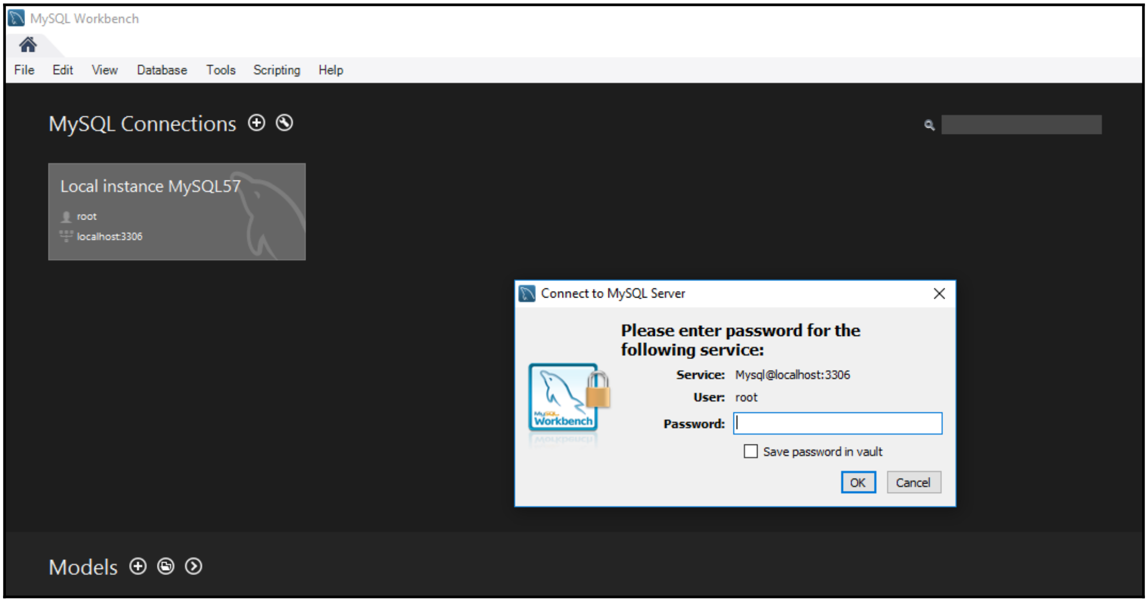
Windows (x86, 32-bit), MSI Installer	6.3.8	23.8M	Download
(mysql-workbench-community-6.3.8-win32.msi)	MD5: 58fa66f6ca40bec609fa0875f0ee70d1 | Signature		
Windows (x86, 64-bit), MSI Installer	6.3.8	26.7M	Download
(mysql-workbench-community-6.3.8-winx64.msi)	MD5: 7e5b333d4542b661cc66fa26c7c76b09 | Signature		



Welcome to the Setup Wizard for MySQL Workbench 6.3 CE

The Setup Wizard will install version 6.3.8 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.



Connect to MySQL Server

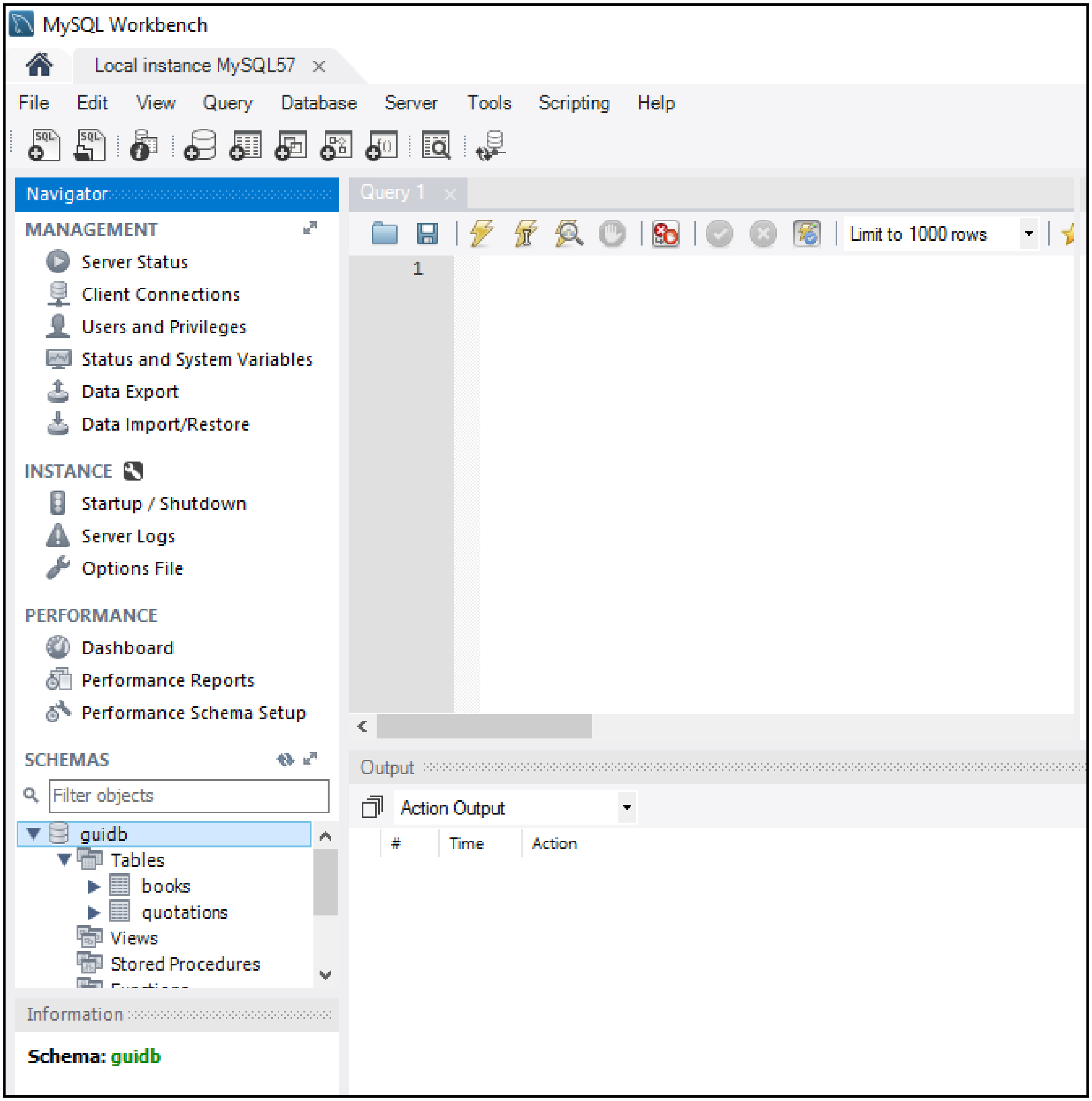
Please enter password for the following service:

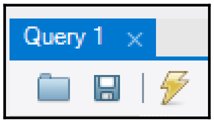
Service: Mysql@localhost:3306

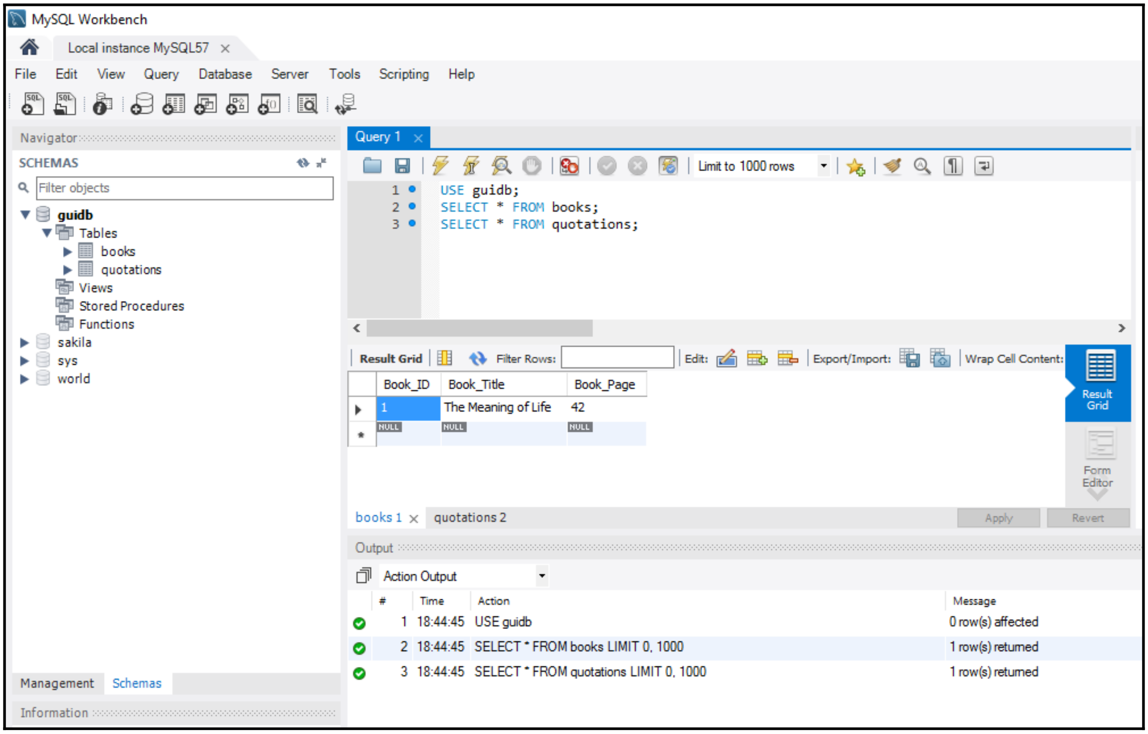
User: root

Password:

□ Save password in vault


MySQL Workbench
Local instance MySQL57
File
Edit
View
Query
Database
Server
Tools
Scripting
Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
SCHEMAS
Filter objects
guidb
Tables
books
quotations
Views
Stored Procedures
Information
Schema: guidb
Query 1
Limit to 1000 rows
1
Output
Action Output
#
Time
Action


Query 1

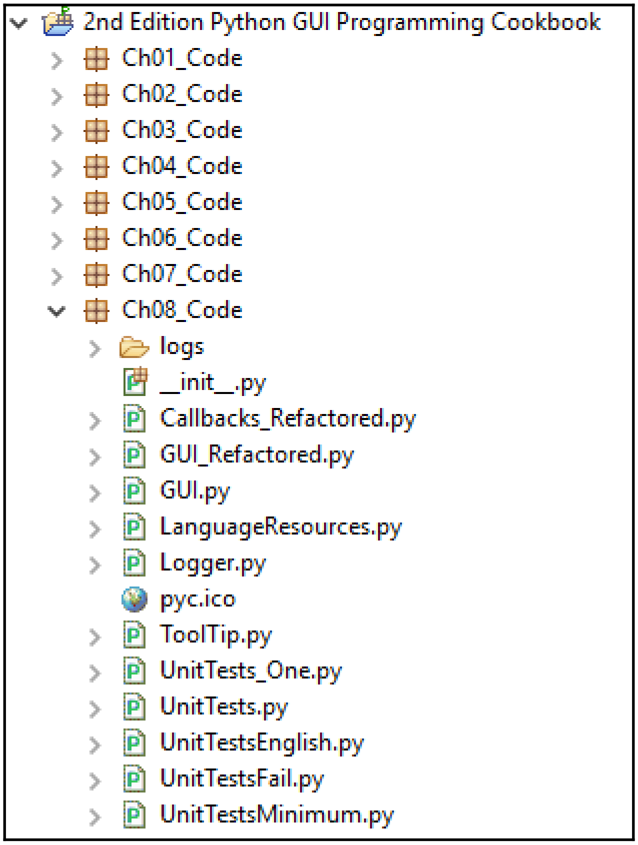


```
USE guidb;
SELECT * FROM books;
SELECT * FROM quotations;
```

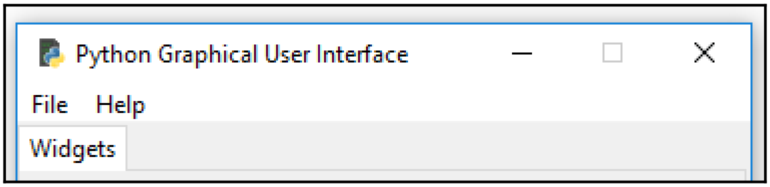
Book_ID	Book_Title	Book_Page
1	The Meaning of Life	42
NULL	NULL	NULL

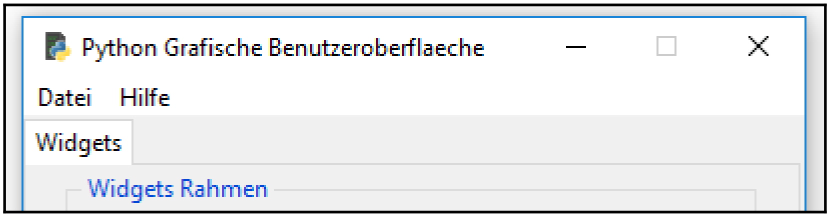
```
#   Time       Action                                    Message
1   18:44:45   USE guidb                                 0 row(s) affected
2   18:44:45   SELECT * FROM books LIMIT 0, 1000         1 row(s) returned
3   18:44:45   SELECT * FROM quotations LIMIT 0, 1000    1 row(s) returned
```

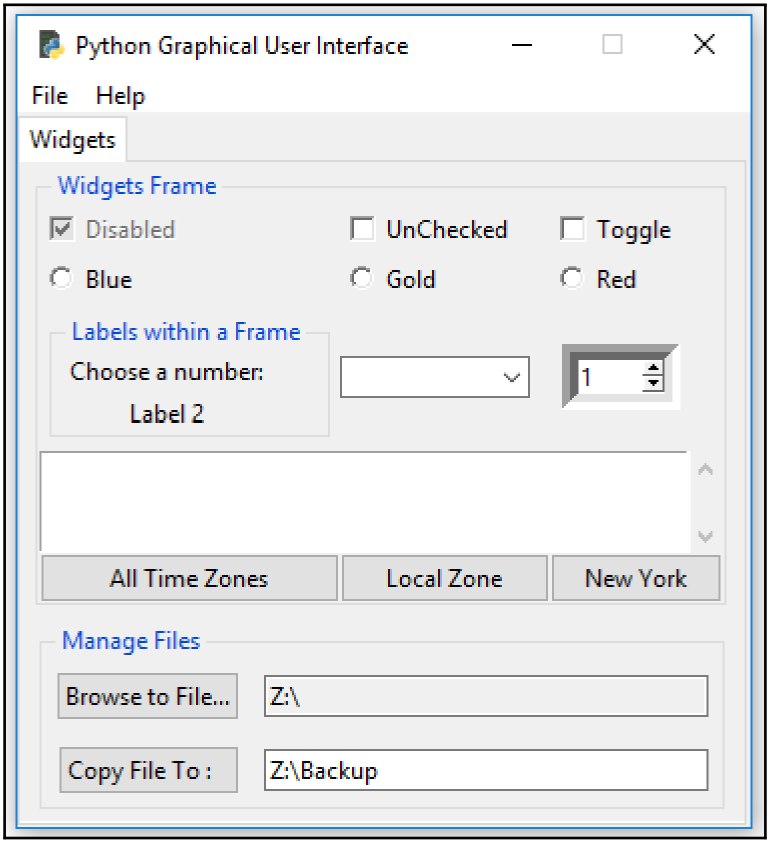
Chapter 8: Internationalization and Testing



```
2nd Edition Python GUI Programming Cookbook
    Ch01_Code
    Ch02_Code
    Ch03_Code
    Ch04_Code
    Ch05_Code
    Ch06_Code
    Ch07_Code
    Ch08_Code
        logs
        __init__.py
        Callbacks_Refactored.py
        GUI_Refactored.py
        GUI.py
        LanguageResources.py
        Logger.py
        pyc.ico
        ToolTip.py
        UnitTests_One.py
        UnitTests.py
        UnitTestsEnglish.py
        UnitTestsFail.py
        UnitTestsMinimum.py
```

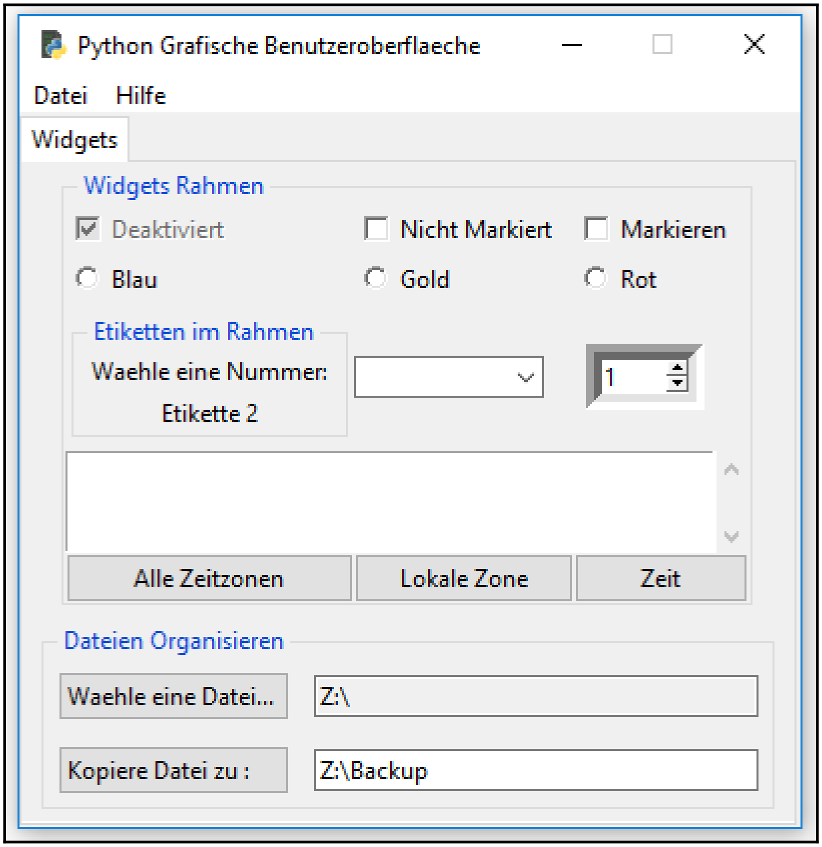






Browse to File... Z:\

Copy File To : Z:\Backup



☑ Deaktiviert

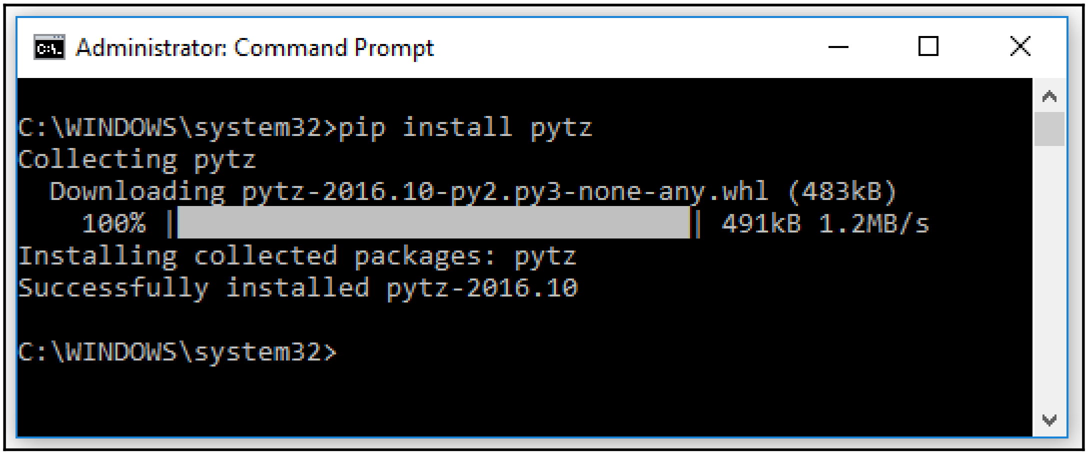
□ Nicht Markiert

□ Markieren

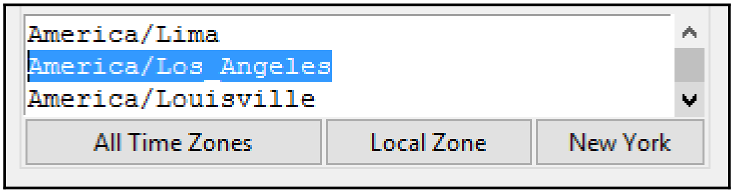
1

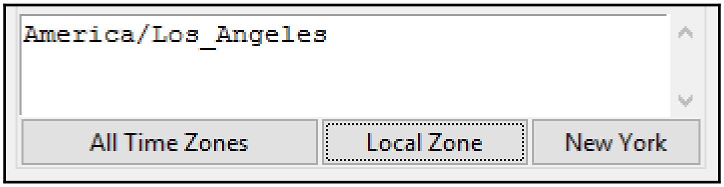
Waehle eine Datei... Z:\

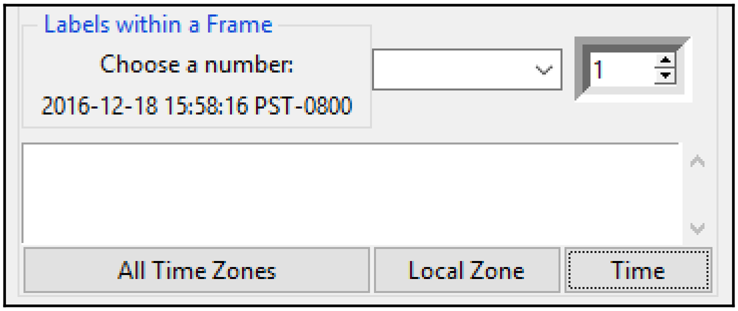
Kopiere Datei zu : Z:\Backup

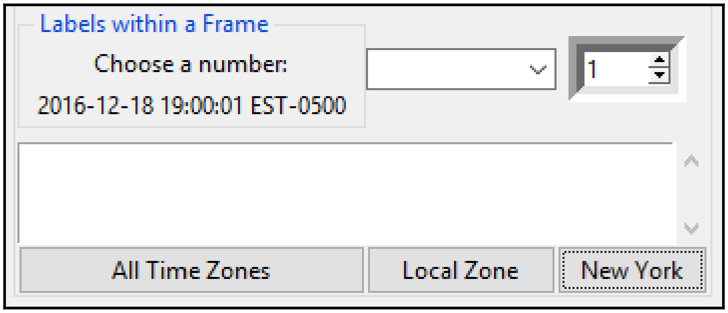


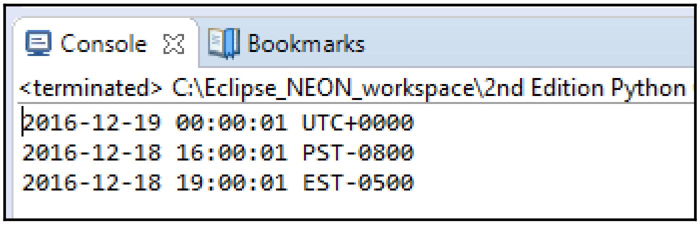
```
C:\WINDOWS\system32>pip install pytz
Collecting pytz
  Downloading pytz-2016.10-py2.py3-none-any.whl (483kB)
    100% |████████████████████████████████| 491kB 1.2MB/s
Installing collected packages: pytz
Successfully installed pytz-2016.10

C:\WINDOWS\system32>
```



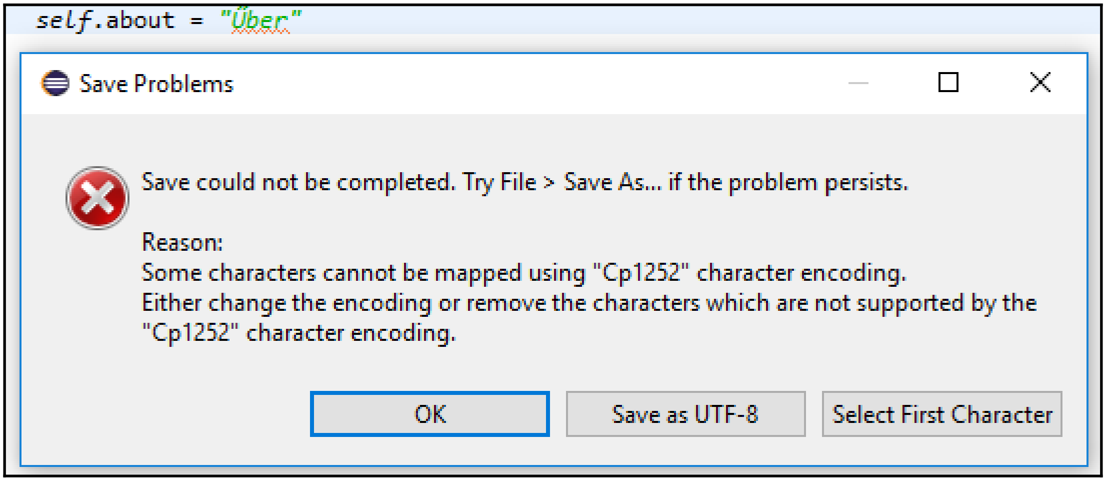








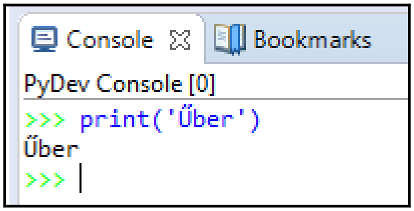
```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python
2016-12-19 00:00:01 UTC+0000
2016-12-18 16:00:01 PST-0800
2016-12-18 19:00:01 EST-0500
```



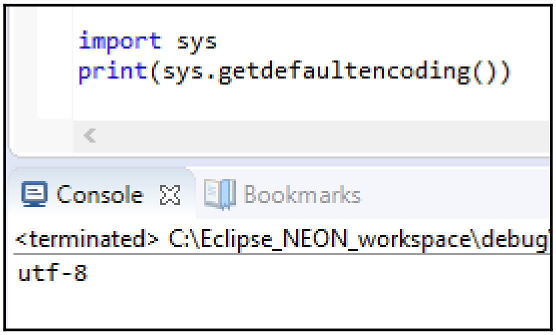
```
self.about = "Űber"
```

Save could not be completed. Try File > Save As... if the problem persists.

Reason:
Some characters cannot be mapped using "Cp1252" character encoding.
Either change the encoding or remove the characters which are not supported by the "Cp1252" character encoding.

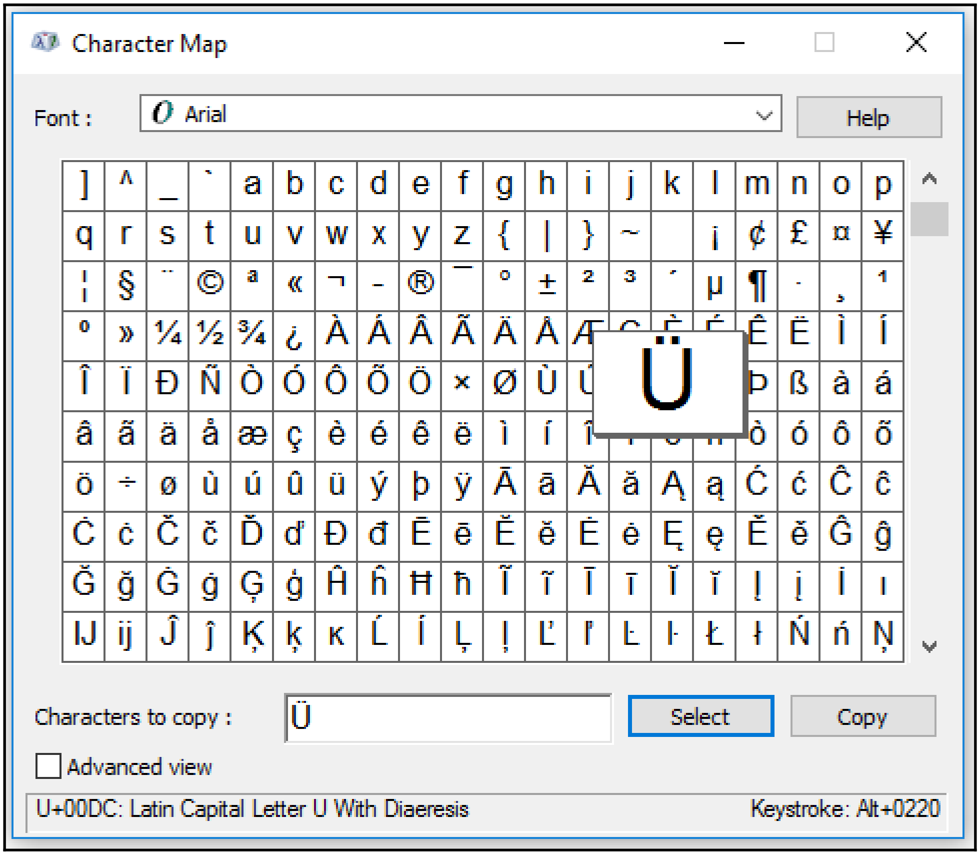


```
>>> print('Űber')
Űber
>>> 
```



```
import sys
print(sys.getdefaultencoding())
```

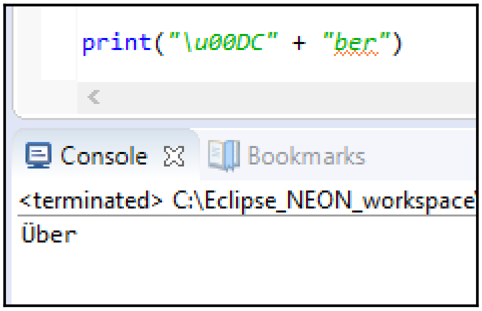
```
<terminated> C:\Eclipse_NEON_workspace\debug
utf-8
```


Character Map
Font : Arial
Help
] ^ _ ` a b c d e f g h i j k l m n o p
q r s t u v w x y z { | } ~ ¡ ¢ £ ¤ ¥
¦ § ¨ © ª « ¬ - ® ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹
º » ¼ ½ ¾ ¿ À Á Â Ã Ä Å Ê Ë Ì Í
Î Ï Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Þ ß à á
â ã ä å æ ç è é ê ë ì í ò ó ô õ
ö ÷ ø ù ú û ü ý þ ÿ Ā ā Ă ă Ą ą Ć ć Ĉ ĉ
Ċ ċ Č č Ď ď Đ đ Ē ē Ĕ ĕ Ė ė Ę ę Ě ě Ĝ ĝ
Ğ ğ Ġ ġ Ģ ģ Ĥ ĥ Ħ ħ Ĩ ĩ Ī ī Ĭ ĭ Į į İ ı
Ĳ ĳ Ĵ ĵ Ķ ķ ĸ Ĺ ĺ Ļ ļ Ľ ľ Ŀ ŀ Ł ł Ń ń Ņ
Ü
Select
Copy

Characters to copy : Ü

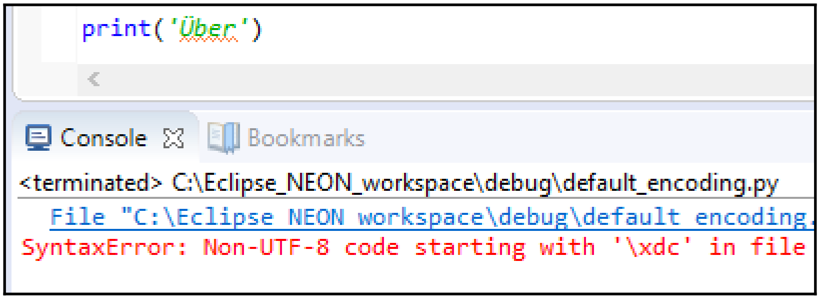
□ Advanced view

U+00DC: Latin Capital Letter U With Diaeresis Keystroke: Alt+0220



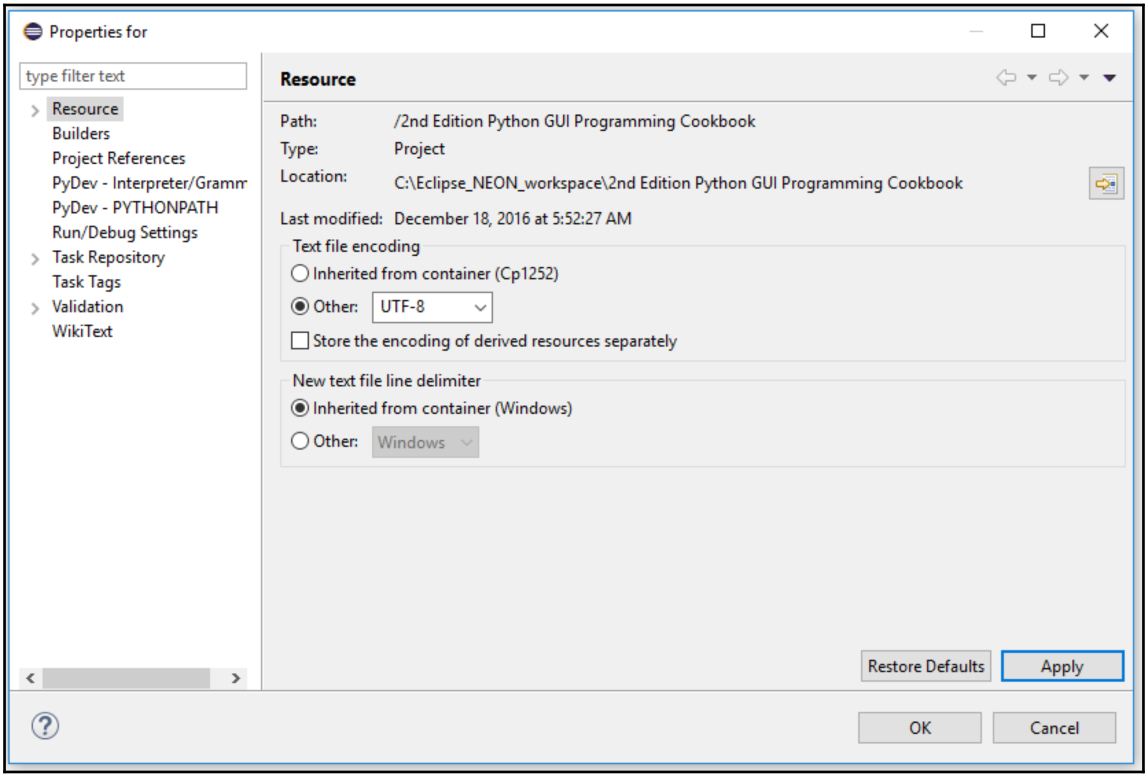
```
print("\u00DC" + "ber")
```

```
<terminated> C:\Eclipse_NEON_workspace
Über
```



```
print('Über')
```

```
<terminated> C:\Eclipse_NEON_workspace\debug\default_encoding.py
  File "C:\Eclipse_NEON_workspace\debug\default_encoding.
SyntaxError: Non-UTF-8 code starting with '\xdc' in file
```



Resource

Path: /2nd Edition Python GUI Programming Cookbook

Type: Project

Location: C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook

Last modified: December 18, 2016 at 5:52:27 AM

Text file encoding

○ Inherited from container (Cp1252)

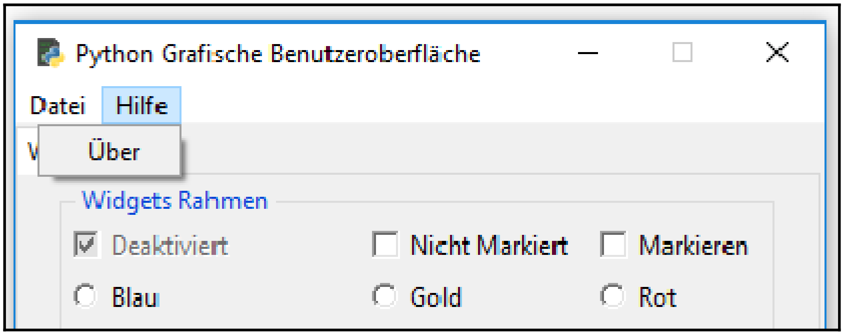
● Other: UTF-8

□ Store the encoding of derived resources separately

New text file line delimiter

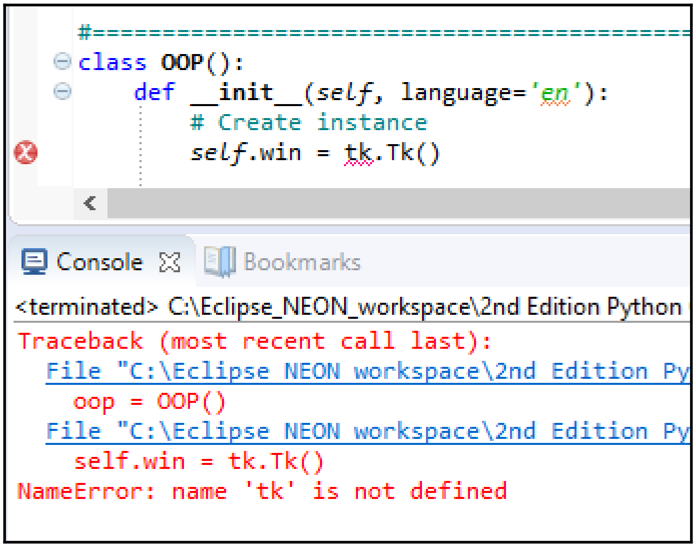
● Inherited from container (Windows)

○ Other: Windows



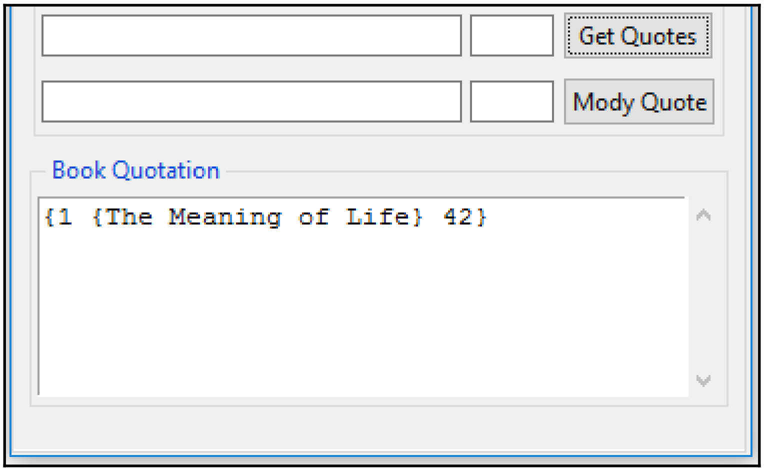
☑ Deaktiviert   □ Nicht Markiert   □ Markieren

○ Blau   ○ Gold   ○ Rot

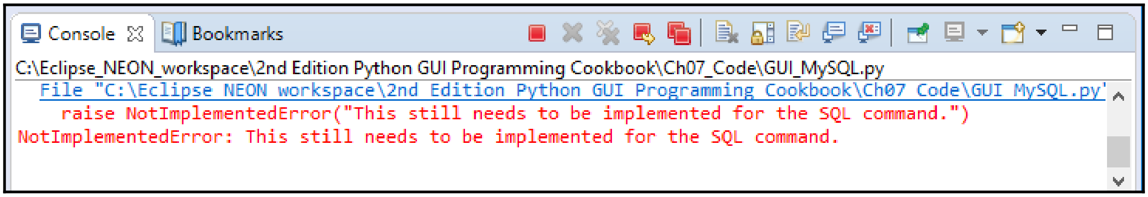


```
#=========================================
class OOP():
    def __init__(self, language='en'):
        # Create instance
        self.win = tk.Tk()
```

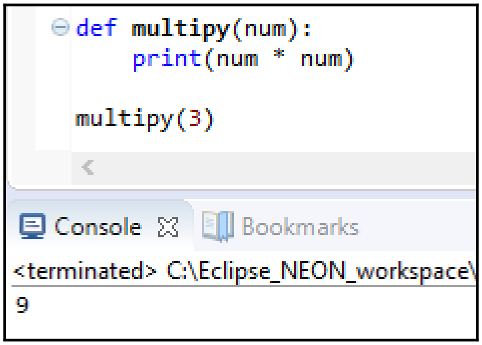
```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python
Traceback (most recent call last):
  File "C:\Eclipse NEON workspace\2nd Edition Py
    oop = OOP()
  File "C:\Eclipse NEON workspace\2nd Edition Py
    self.win = tk.Tk()
NameError: name 'tk' is not defined
```



```
{1 {The Meaning of Life} 42}
```

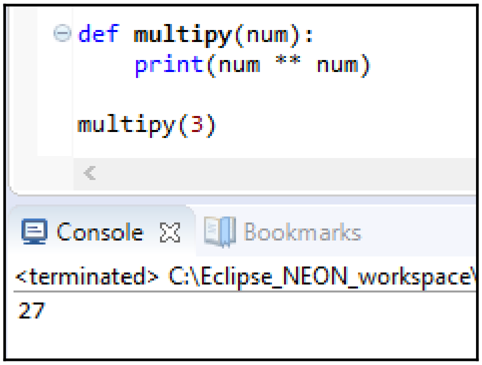


```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch07_Code\GUI_MySQL.py
  File "C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch07_Code\GUI_MySQL.py'
    raise NotImplementedError("This still needs to be implemented for the SQL command.")
NotImplementedError: This still needs to be implemented for the SQL command.
```



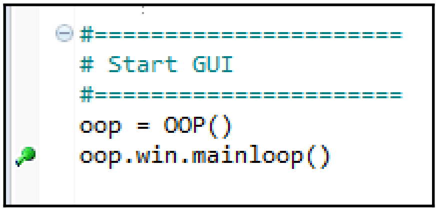
```
def multipy(num):
    print(num * num)

multipy(3)
```

```
<terminated> C:\Eclipse_NEON_workspace\
9
```

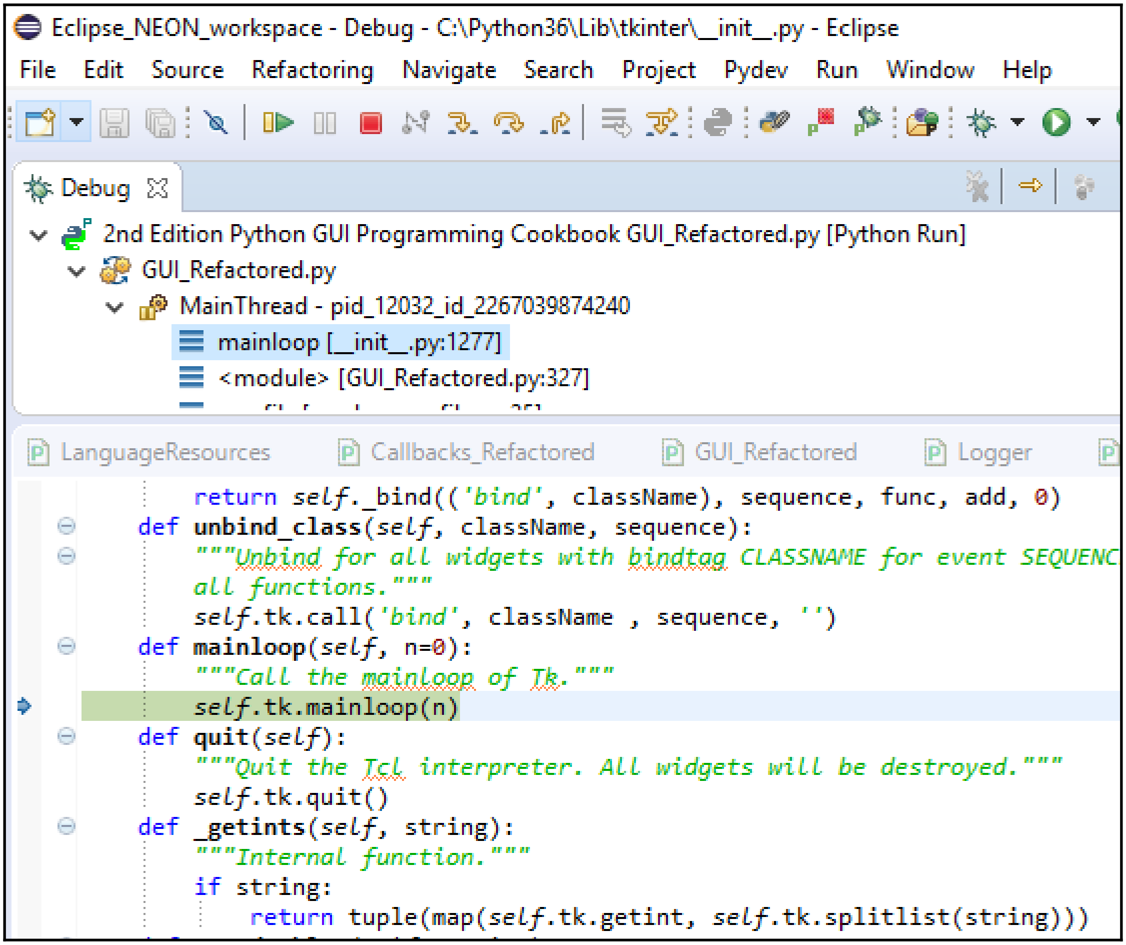


```
def multipy(num):
    print(num ** num)

multipy(3)
```

```
<terminated> C:\Eclipse_NEON_workspace\
27
```



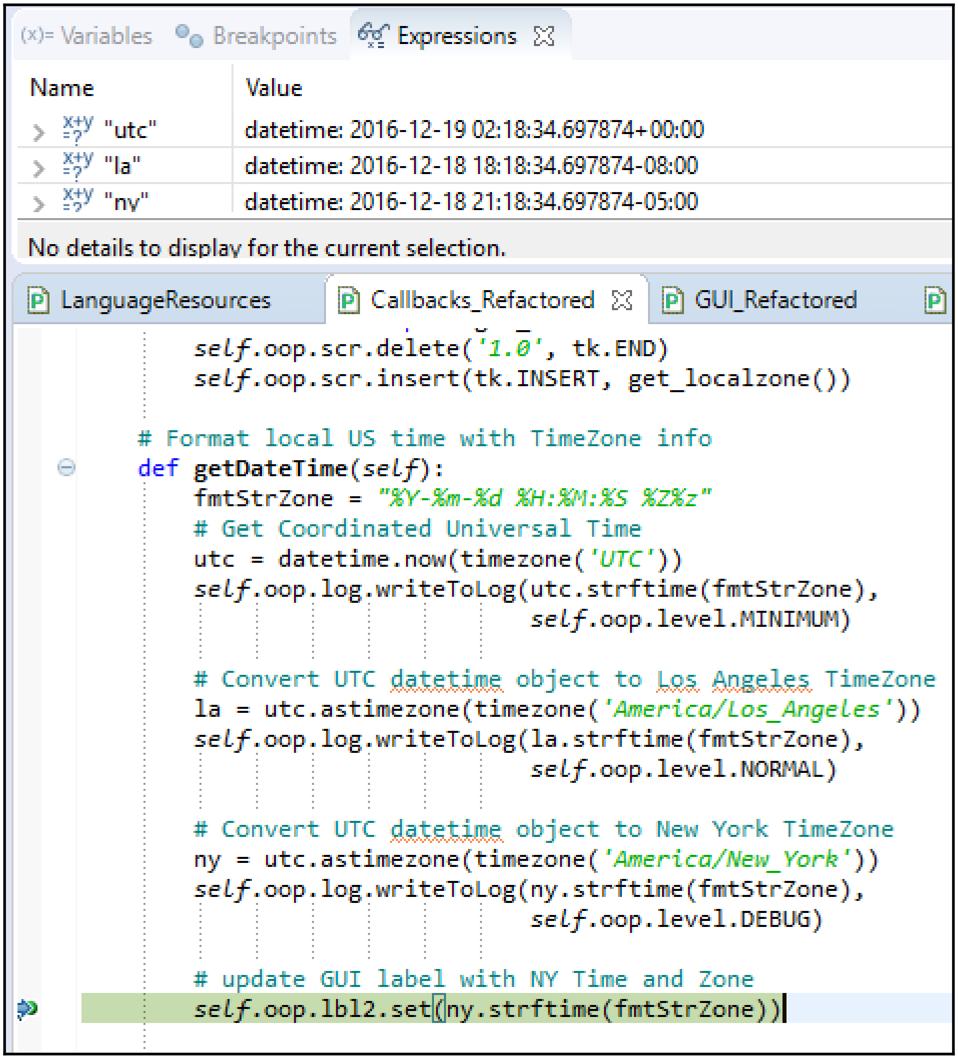
```
#======================
# Start GUI
#======================
oop = OOP()
oop.win.mainloop()
```



Eclipse_NEON_workspace - Debug - C:\Python36\Lib\tkinter\__init__.py - Eclipse

```
2nd Edition Python GUI Programming Cookbook GUI_Refactored.py [Python Run]
  GUI_Refactored.py
    MainThread - pid_12032_id_2267039874240
      mainloop [__init__.py:1277]
      <module> [GUI_Refactored.py:327]
```

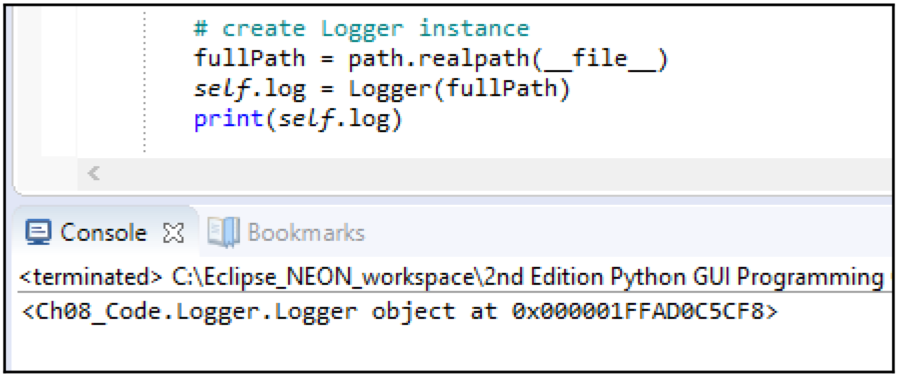
```
        return self._bind(('bind', className), sequence, func, add, 0)
    def unbind_class(self, className, sequence):
        """Unbind for all widgets with bindtag CLASSNAME for event SEQUENC
        all functions."""
        self.tk.call('bind', className , sequence, '')
    def mainloop(self, n=0):
        """Call the mainloop of Tk."""
        self.tk.mainloop(n)
    def quit(self):
        """Quit the Tcl interpreter. All widgets will be destroyed."""
        self.tk.quit()
    def _getints(self, string):
        """Internal function."""
        if string:
            return tuple(map(self.tk.getint, self.tk.splitlist(string)))
```



Name	Value
"utc"	datetime: 2016-12-19 02:18:34.697874+00:00
"la"	datetime: 2016-12-18 18:18:34.697874-08:00
"ny"	datetime: 2016-12-18 21:18:34.697874-05:00

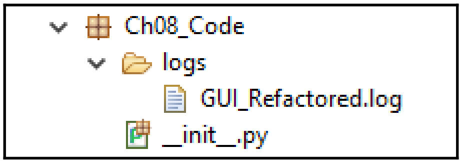
No details to display for the current selection.

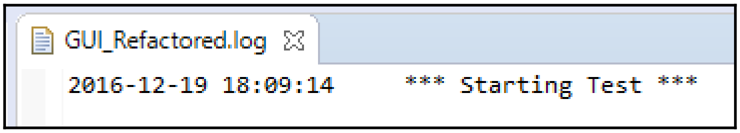
```
        self.oop.scr.delete('1.0', tk.END)
        self.oop.scr.insert(tk.INSERT, get_localzone())

    # Format local US time with TimeZone info
    def getDateTime(self):
        fmtStrZone = "%Y-%m-%d %H:%M:%S %Z%z"
        # Get Coordinated Universal Time
        utc = datetime.now(timezone('UTC'))
        self.oop.log.writeToLog(utc.strftime(fmtStrZone),
                                self.oop.level.MINIMUM)

        # Convert UTC datetime object to Los Angeles TimeZone
        la = utc.astimezone(timezone('America/Los_Angeles'))
        self.oop.log.writeToLog(la.strftime(fmtStrZone),
                                self.oop.level.NORMAL)

        # Convert UTC datetime object to New York TimeZone
        ny = utc.astimezone(timezone('America/New_York'))
        self.oop.log.writeToLog(ny.strftime(fmtStrZone),
                                self.oop.level.DEBUG)

        # update GUI label with NY Time and Zone
        self.oop.lbl2.set(ny.strftime(fmtStrZone))
```



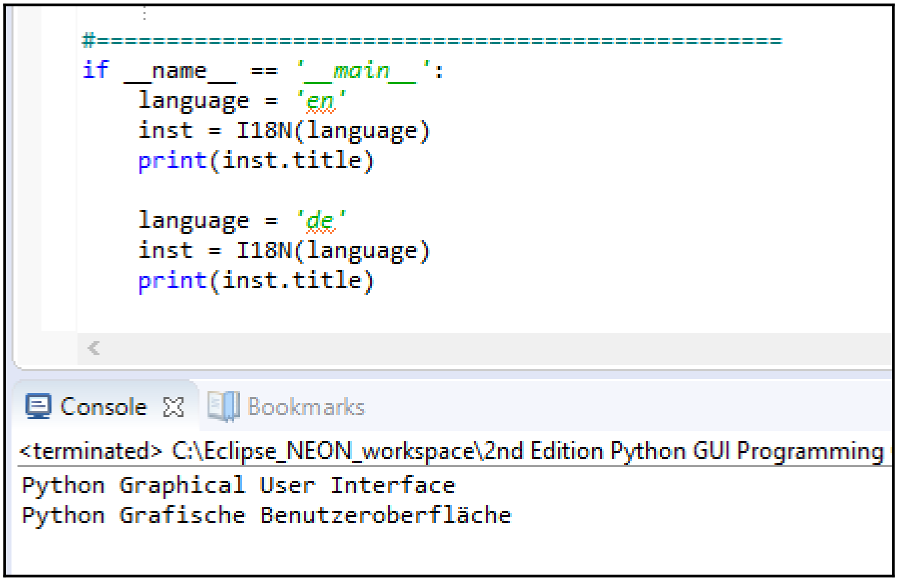
```
        # create Logger instance
        fullPath = path.realpath(__file__)
        self.log = Logger(fullPath)
        print(self.log)
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming
<Ch08_Code.Logger.Logger object at 0x000001FFAD0C5CF8>
```



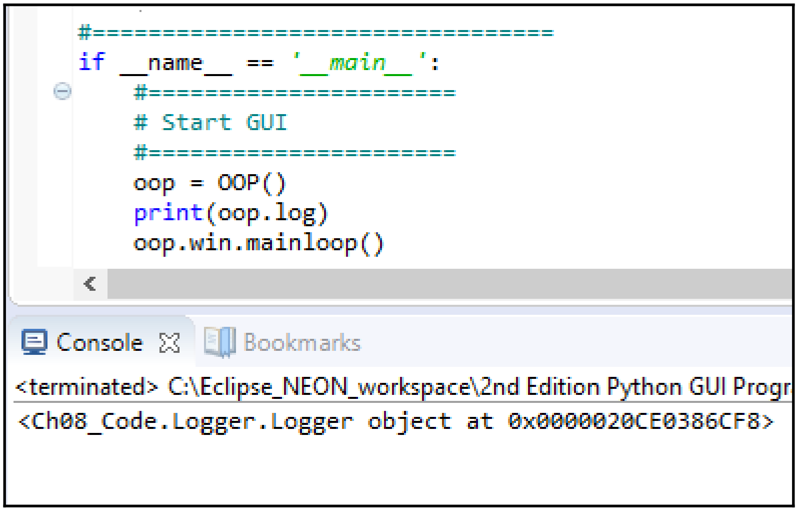


```
2016-12-19 18:09:14     *** Starting Test ***
```



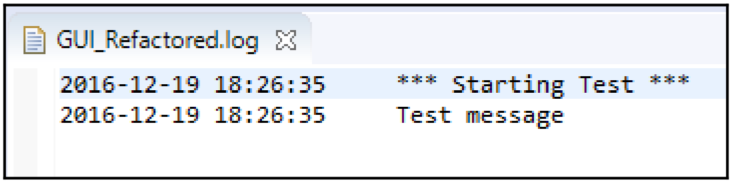
```
#======================================================
if __name__ == '__main__':
    language = 'en'
    inst = I18N(language)
    print(inst.title)

    language = 'de'
    inst = I18N(language)
    print(inst.title)
```

```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming
Python Graphical User Interface
Python Grafische Benutzeroberfläche
```

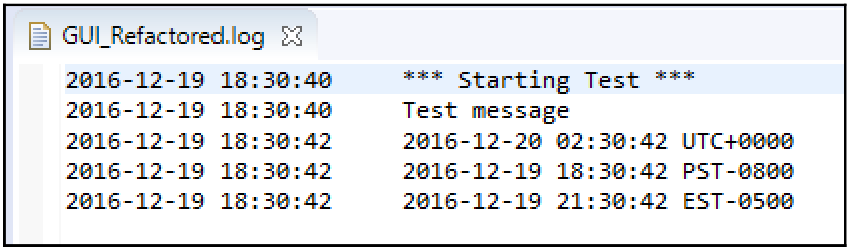


```
#==================================
if __name__ == '__main__':
    #======================
    # Start GUI
    #======================
    oop = OOP()
    print(oop.log)
    oop.win.mainloop()
```

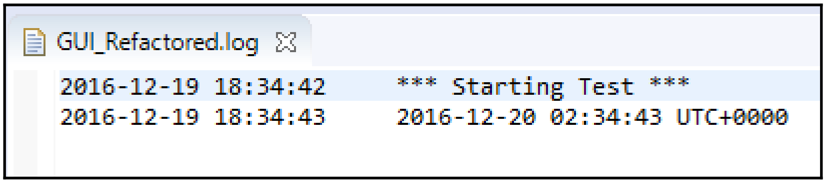
```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Progr
<Ch08_Code.Logger.Logger object at 0x0000020CE0386CF8>
```



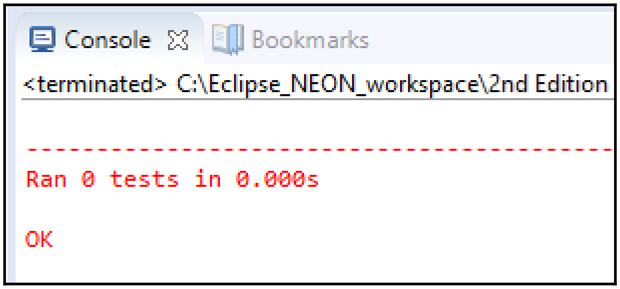
```
2016-12-19 18:26:35      *** Starting Test ***
2016-12-19 18:26:35      Test message
```



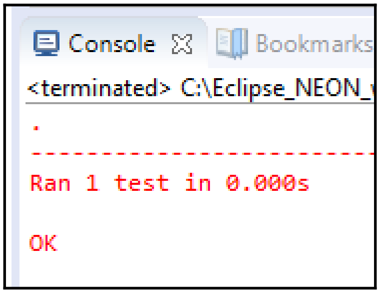
```
2016-12-19 18:30:40      *** Starting Test ***
2016-12-19 18:30:40      Test message
2016-12-19 18:30:42      2016-12-20 02:30:42 UTC+0000
2016-12-19 18:30:42      2016-12-19 18:30:42 PST-0800
2016-12-19 18:30:42      2016-12-19 21:30:42 EST-0500
```



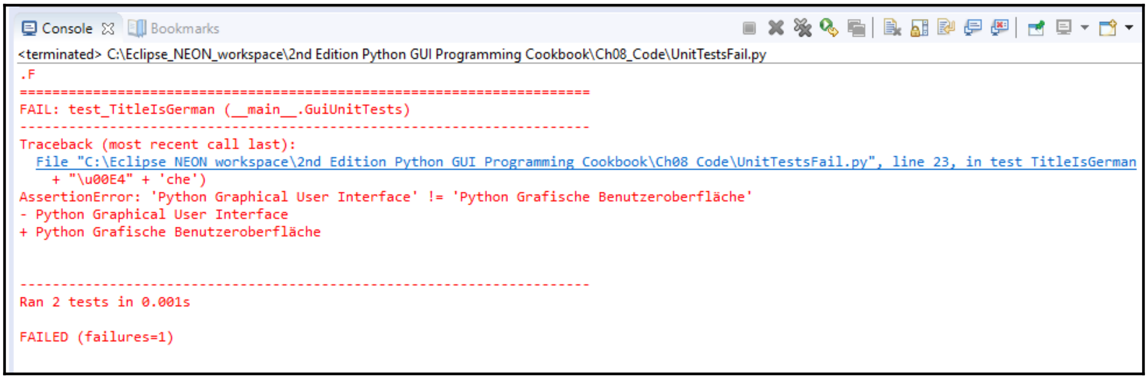
```
2016-12-19 18:34:42      *** Starting Test ***
2016-12-19 18:34:43      2016-12-20 02:34:43 UTC+0000
```



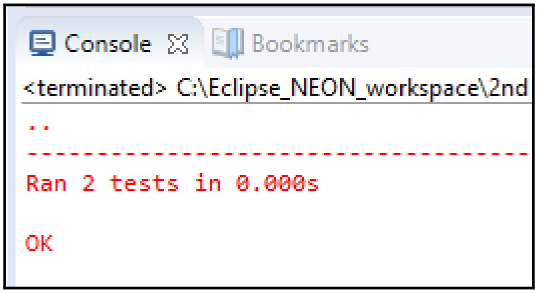
```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition

----------------------------------------
Ran 0 tests in 0.000s

OK
```



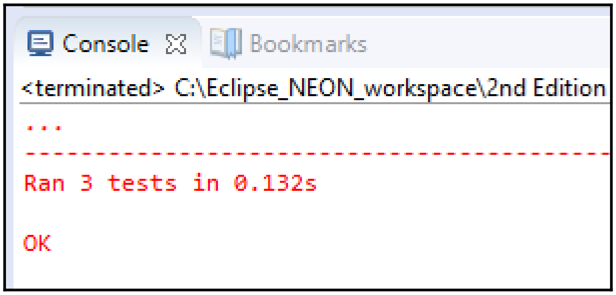
```
<terminated> C:\Eclipse_NEON_
.
--------------------------
Ran 1 test in 0.000s

OK
```



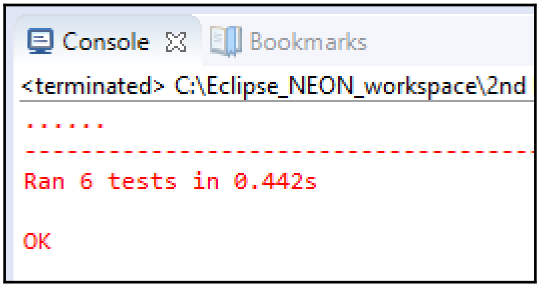
```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch08_Code\UnitTestsFail.py
.F
======================================================================
FAIL: test_TitleIsGerman (__main__.GuiUnitTests)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Eclipse_NEON_workspace\2nd_Edition_Python_GUI_Programming_Cookbook\Ch08_Code\UnitTestsFail.py", line 23, in test_TitleIsGerman
    + "\u00E4" + 'che')
AssertionError: 'Python Graphical User Interface' != 'Python Grafische Benutzeroberfläche'
- Python Graphical User Interface
+ Python Grafische Benutzeroberfläche


----------------------------------------------------------------------
Ran 2 tests in 0.001s

FAILED (failures=1)
```



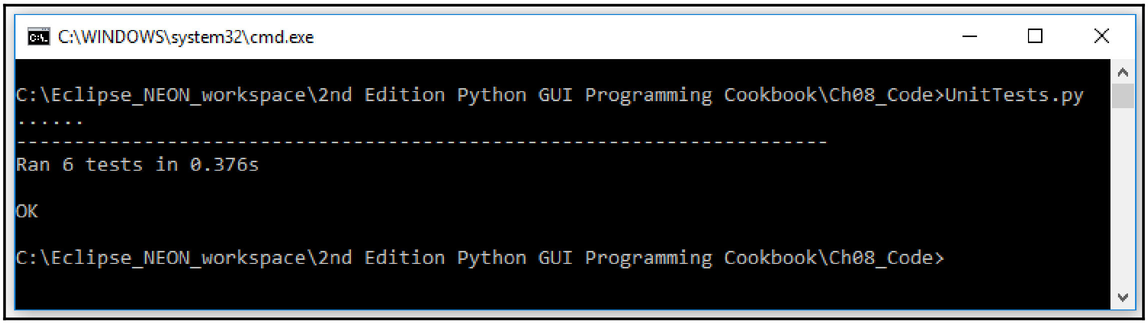
```
<terminated> C:\Eclipse_NEON_workspace\2nd
..
-----------------------------------
Ran 2 tests in 0.000s

OK
```



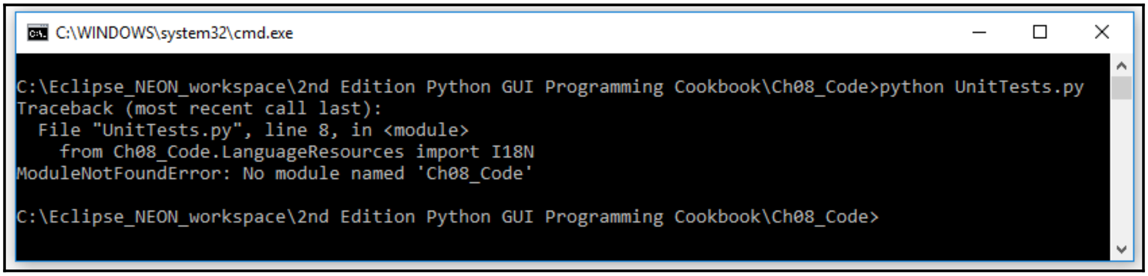
```
<terminated> C:\Eclipse_NEON_workspace\2nd Edition
...
----------------------------------------
Ran 3 tests in 0.132s

OK
```



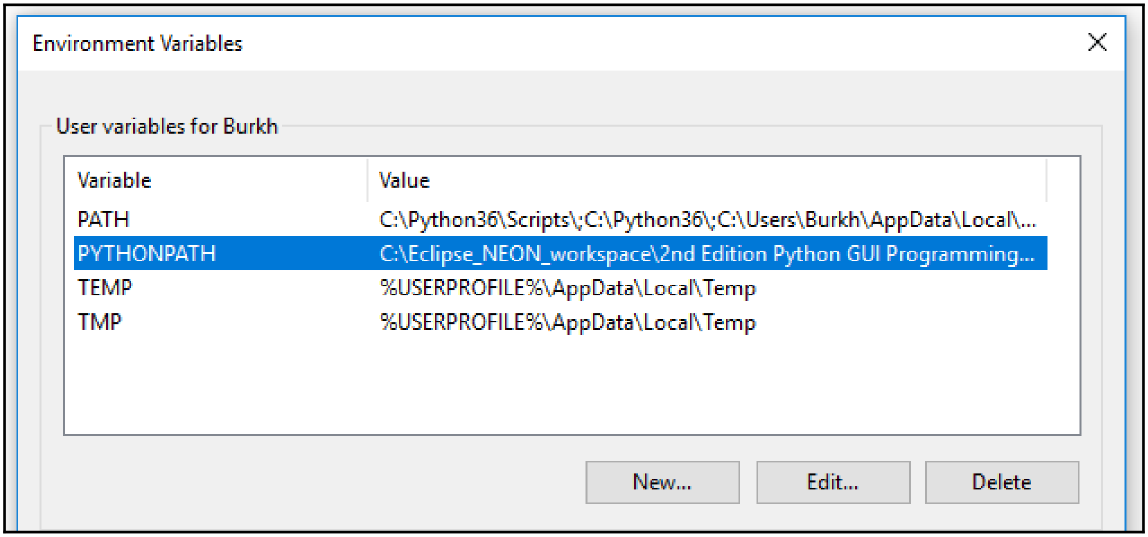
```
<terminated> C:\Eclipse_NEON_workspace\2nd
......
-----------------------------------
Ran 6 tests in 0.442s

OK
```



```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch08_Code>UnitTests.py
......
----------------------------------------------------------------------
Ran 6 tests in 0.376s

OK

C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch08_Code>
```



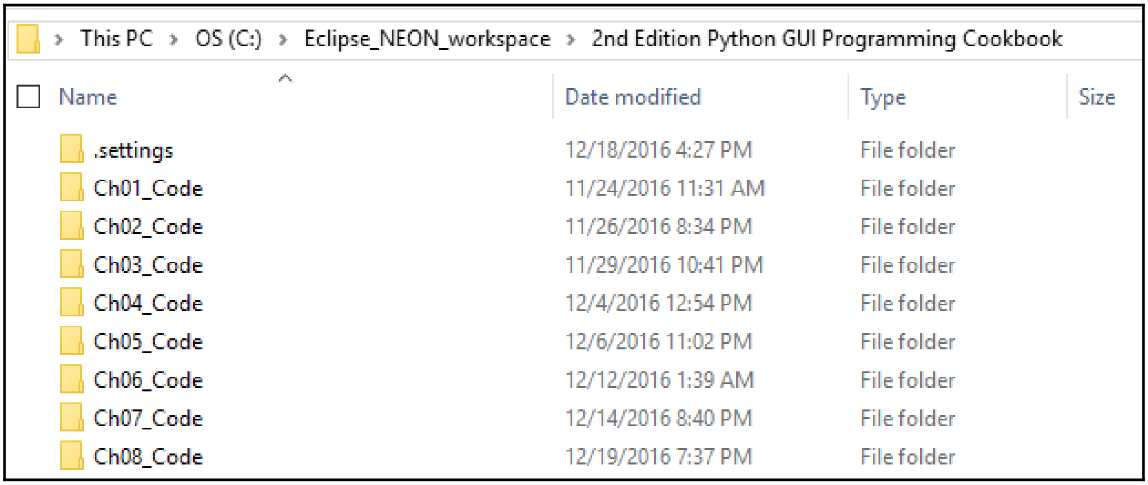
```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch08_Code>python UnitTests.py
Traceback (most recent call last):
  File "UnitTests.py", line 8, in <module>
    from Ch08_Code.LanguageResources import I18N
ModuleNotFoundError: No module named 'Ch08_Code'

C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch08_Code>
```



Environment Variables

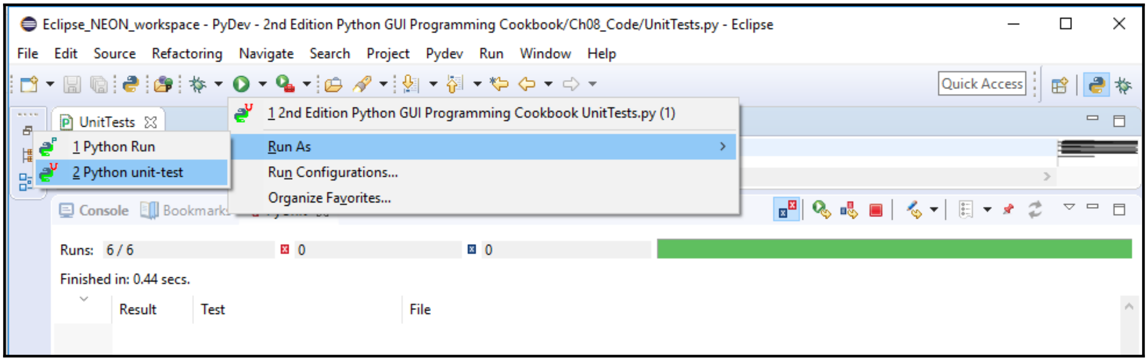
User variables for Burkh

Variable	Value
PATH	C:\Python36\Scripts\;C:\Python36\;C:\Users\Burkh\AppData\Local\...
PYTHONPATH	C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp



This PC › OS (C:) › Eclipse_NEON_workspace › 2nd Edition Python GUI Programming Cookbook

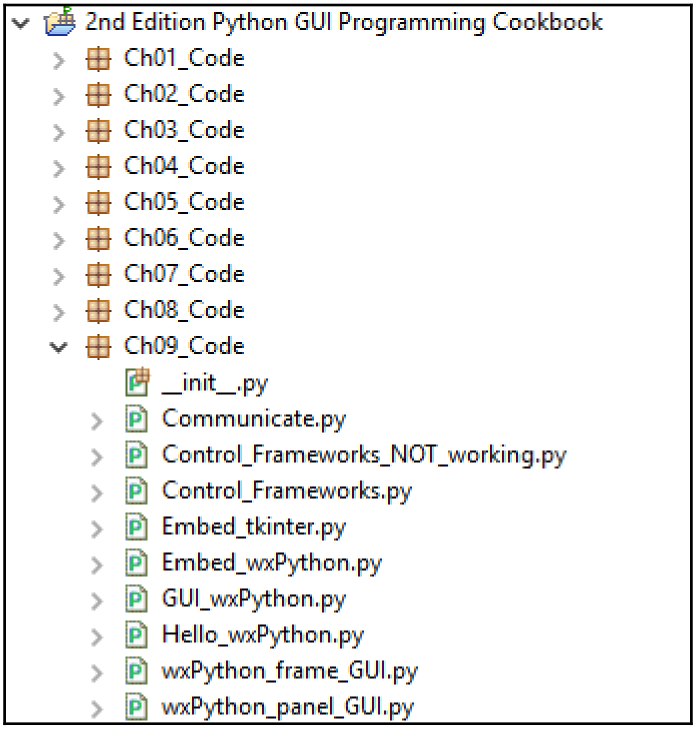
Name	Date modified	Type	Size
.settings	12/18/2016 4:27 PM	File folder	
Ch01_Code	11/24/2016 11:31 AM	File folder	
Ch02_Code	11/26/2016 8:34 PM	File folder	
Ch03_Code	11/29/2016 10:41 PM	File folder	
Ch04_Code	12/4/2016 12:54 PM	File folder	
Ch05_Code	12/6/2016 11:02 PM	File folder	
Ch06_Code	12/12/2016 1:39 AM	File folder	
Ch07_Code	12/14/2016 8:40 PM	File folder	
Ch08_Code	12/19/2016 7:37 PM	File folder	



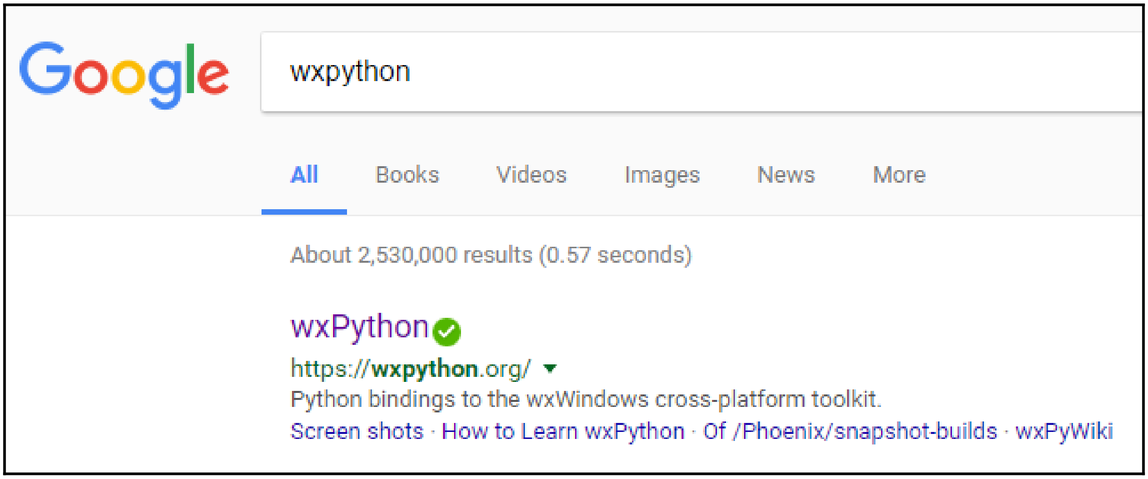
Runs: 6 / 6 0 0

Finished in: 0.44 secs.

Chapter 9:
Extending Our GUI with the wxPython Librar y



2nd Edition Python GUI Programming Cookbook
Ch01_Code
Ch02_Code
Ch03_Code
Ch04_Code
Ch05_Code
Ch06_Code
Ch07_Code
Ch08_Code
Ch09_Code
__init__.py
Communicate.py
Control_Frameworks_NOT_working.py
Control_Frameworks.py
Embed_tkinter.py
Embed_wxPython.py
GUI_wxPython.py
Hello_wxPython.py
wxPython_frame_GUI.py
wxPython_panel_GUI.py



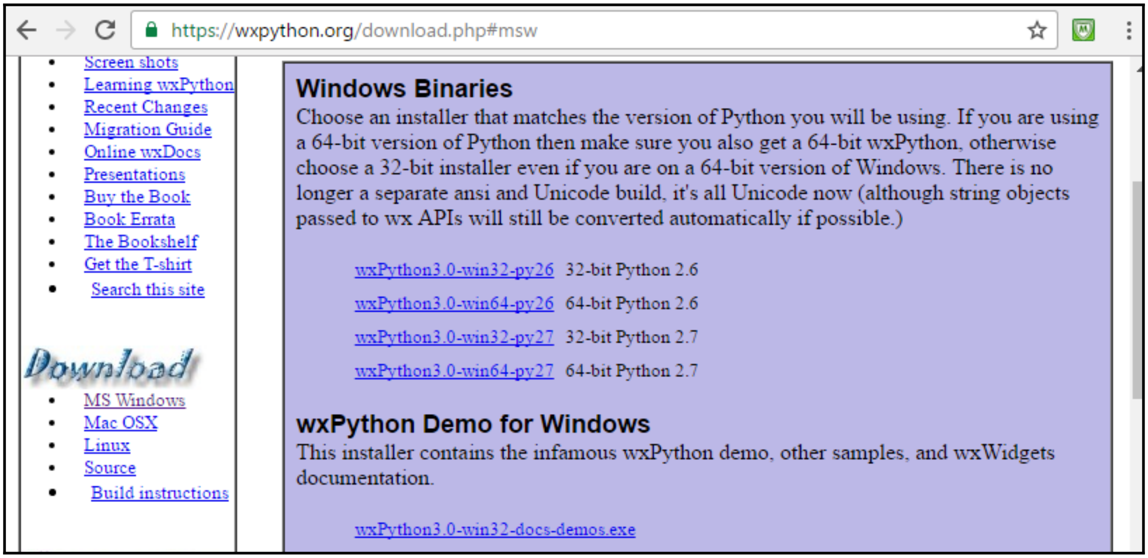
About 2,530,000 results (0.57 seconds)

wxPython

https://wxpython.org/

Python bindings to the wxWindows cross-platform toolkit.

Screen shots · How to Learn wxPython · Of /Phoenix/snapshot-builds · wxPyWiki



Windows Binaries

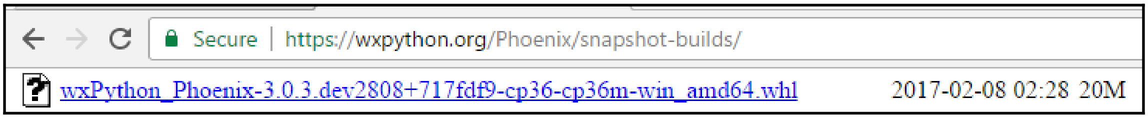
Choose an installer that matches the version of Python you will be using. If you are using a 64-bit version of Python then make sure you also get a 64-bit wxPython, otherwise choose a 32-bit installer even if you are on a 64-bit version of Windows. There is no longer a separate ansi and Unicode build, it's all Unicode now (although string objects passed to wx APIs will still be converted automatically if possible.)

wxPython3.0-win32-py26	32-bit Python 2.6
wxPython3.0-win64-py26	64-bit Python 2.6
wxPython3.0-win32-py27	32-bit Python 2.7
wxPython3.0-win64-py27	64-bit Python 2.7

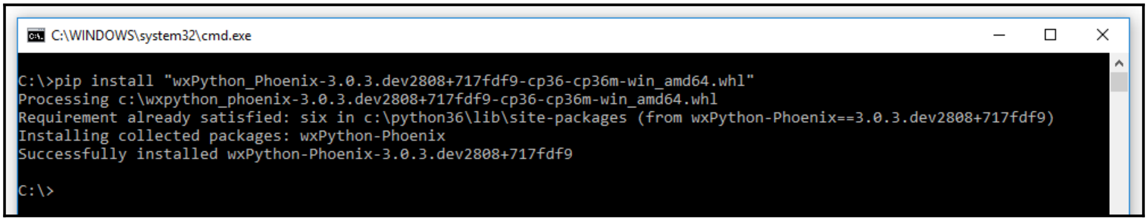
wxPython Demo for Windows

This installer contains the infamous wxPython demo, other samples, and wxWidgets documentation.

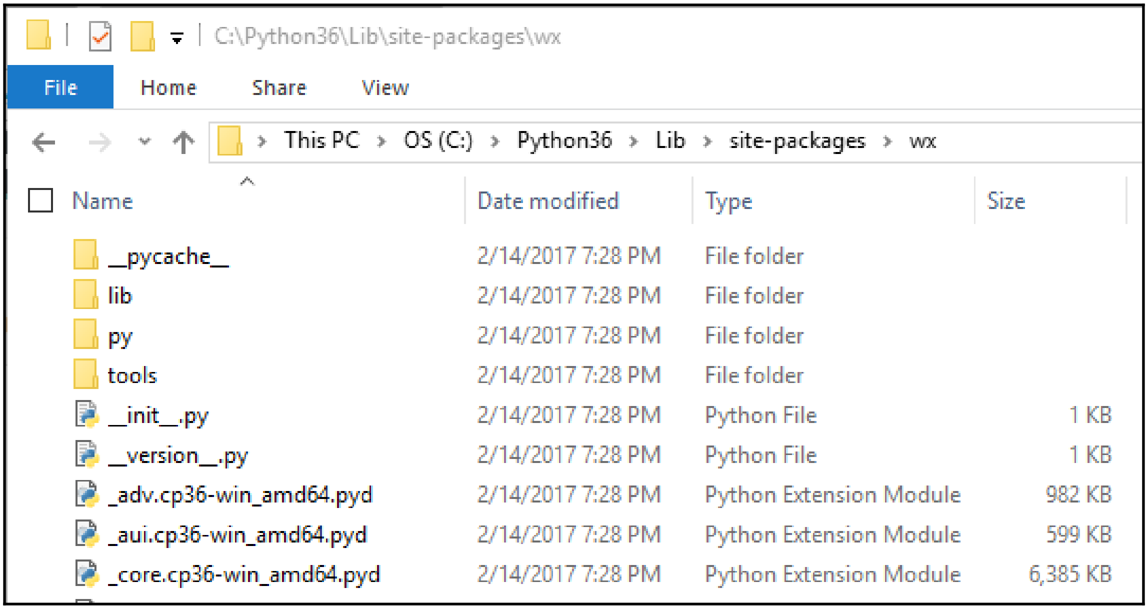
wxPython3.0-win32-docs-demos.exe



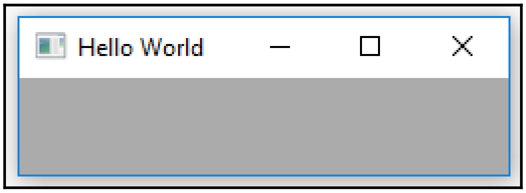
wxPython_Phoenix-3.0.3.dev2808+717fdf9-cp36-cp36m-win_amd64.whl 2017-02-08 02:28 20M

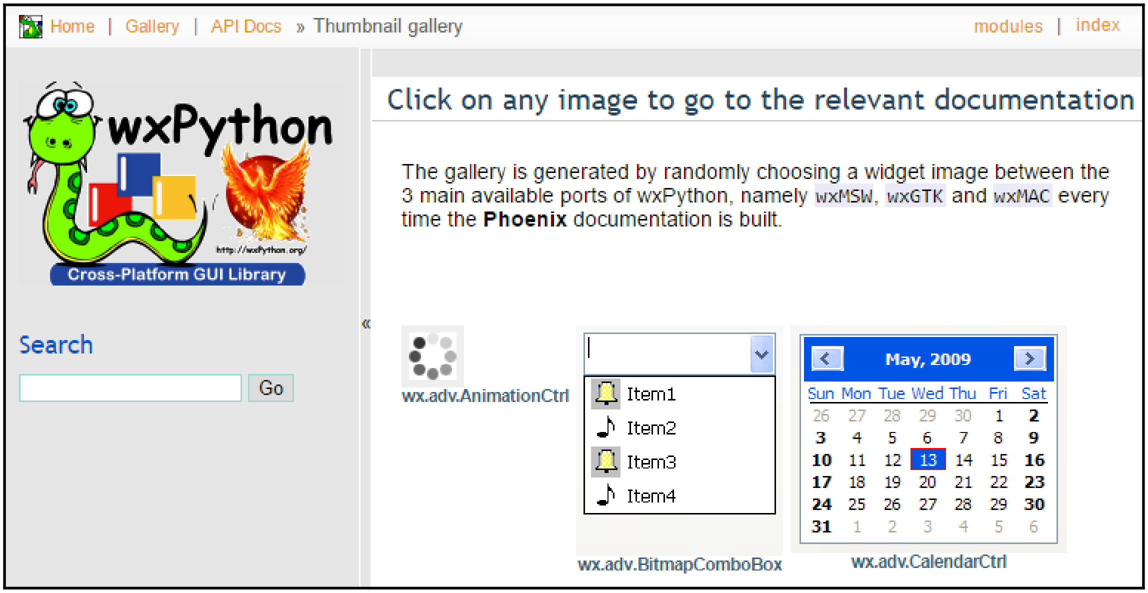


```
C:\>pip install "wxPython_Phoenix-3.0.3.dev2808+717fdf9-cp36-cp36m-win_amd64.whl"
Processing c:\wxpython_phoenix-3.0.3.dev2808+717fdf9-cp36-cp36m-win_amd64.whl
Requirement already satisfied: six in c:\python36\lib\site-packages (from wxPython-Phoenix==3.0.3.dev2808+717fdf9)
Installing collected packages: wxPython-Phoenix
Successfully installed wxPython-Phoenix-3.0.3.dev2808+717fdf9

C:\>
```


C:\Python36\Lib\site-packages\wx
File
Home
Share
View
This PC › OS (C:) › Python36 › Lib › site-packages › wx

Name	Date modified	Type	Size
__pycache__	2/14/2017 7:28 PM	File folder	
lib	2/14/2017 7:28 PM	File folder	
py	2/14/2017 7:28 PM	File folder	
tools	2/14/2017 7:28 PM	File folder	
__init__.py	2/14/2017 7:28 PM	Python File	1 KB
__version__.py	2/14/2017 7:28 PM	Python File	1 KB
_adv.cp36-win_amd64.pyd	2/14/2017 7:28 PM	Python Extension Module	982 KB
_aui.cp36-win_amd64.pyd	2/14/2017 7:28 PM	Python Extension Module	599 KB
_core.cp36-win_amd64.pyd	2/14/2017 7:28 PM	Python Extension Module	6,385 KB




Home | Gallery | API Docs » Thumbnail gallery
modules | index
wxPython
http://wxPython.org/
Cross-Platform GUI Library
Search
Go
«

Click on any image to go to the relevant documentation

The gallery is generated by randomly choosing a widget image between the 3 main available ports of wxPython, namely `wxMSW`, `wxGTK` and `wxMAC` every time the **Phoenix** documentation is built.

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

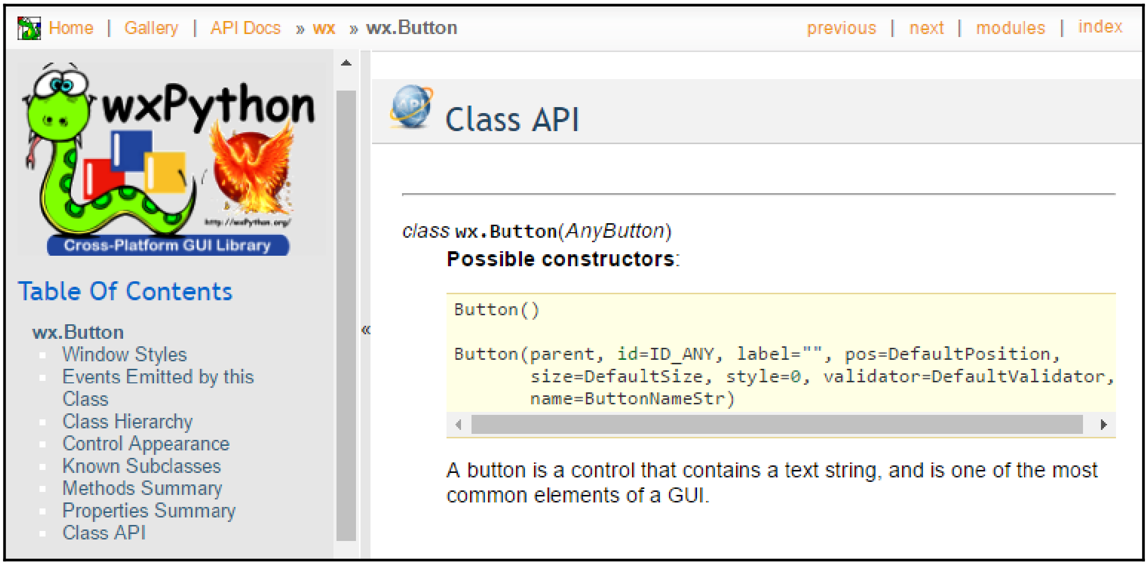

Home | Gallery | API Docs » wx » wx.Button
previous | next | modules | index
wxPython
http://wxPython.org/
Cross-Platform GUI Library
«

Table Of Contents

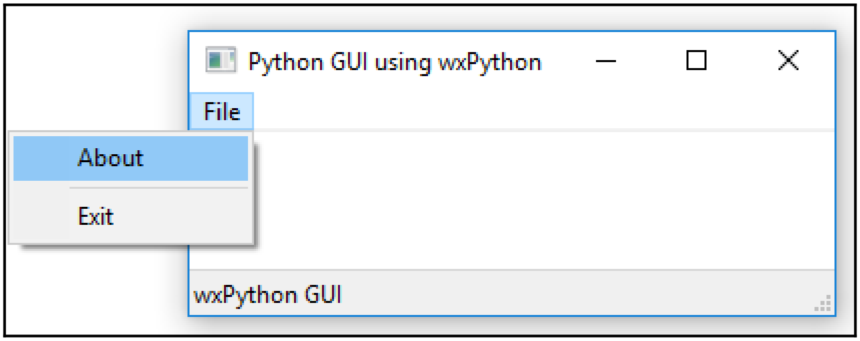
Class API

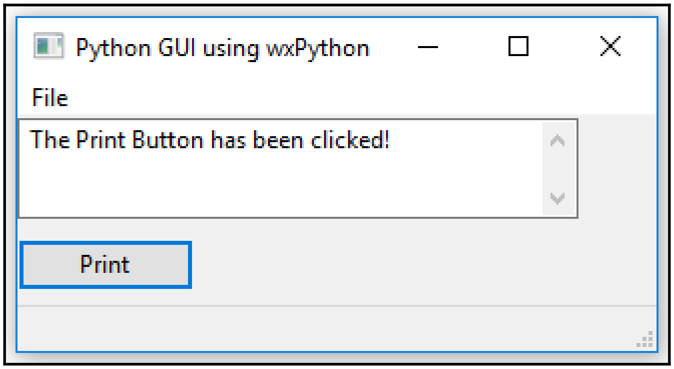
class **`wx.Button`**(*AnyButton*)

Possible constructors:

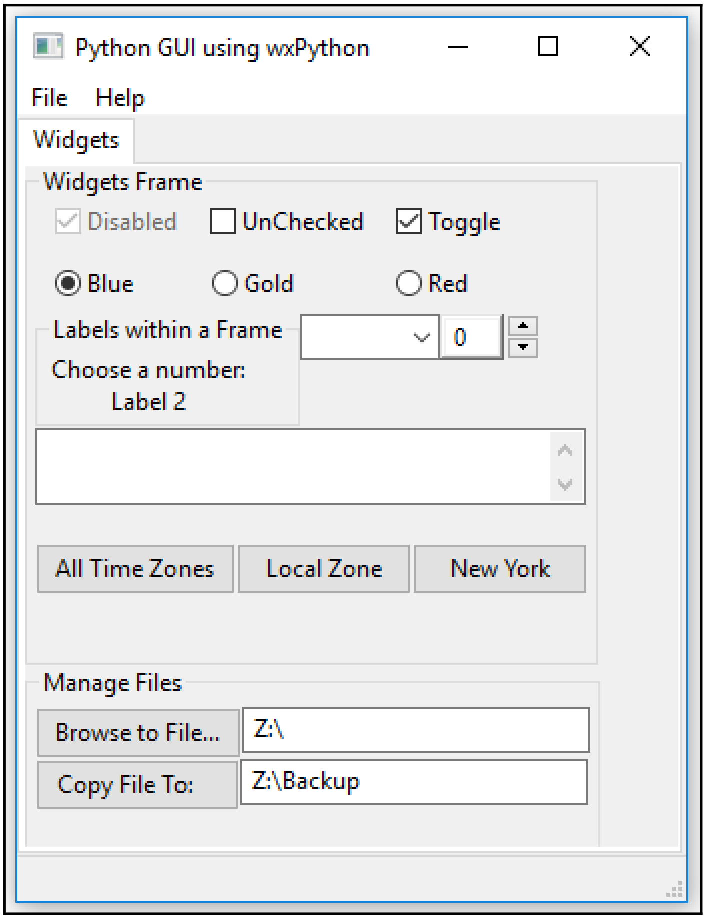
```
Button()

Button(parent, id=ID_ANY, label="", pos=DefaultPosition,
       size=DefaultSize, style=0, validator=DefaultValidator,
       name=ButtonNameStr)
```

A button is a control that contains a text string, and is one of the most common elements of a GUI.





The Print Button has been clicked!



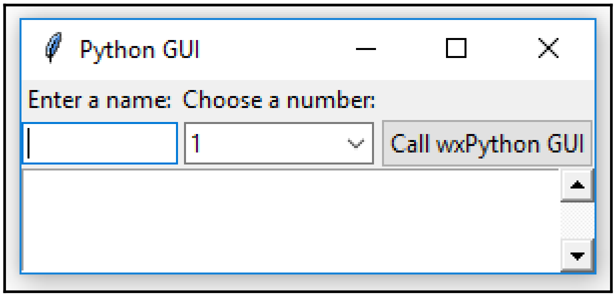
☑ Disabled □ UnChecked ☑ Toggle

● Blue ○ Gold ○ Red

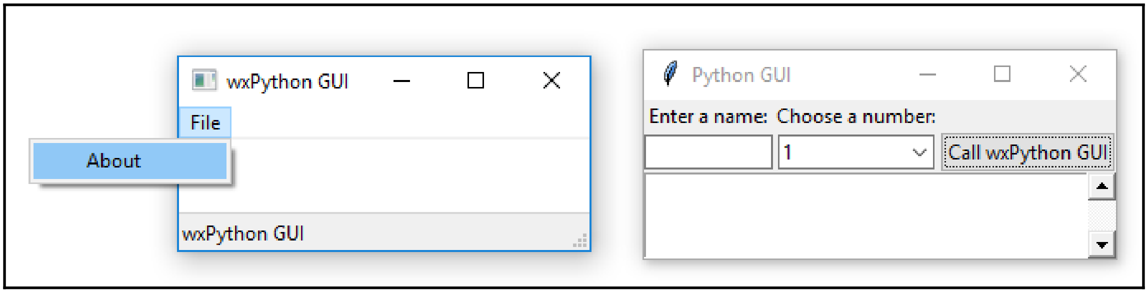
0

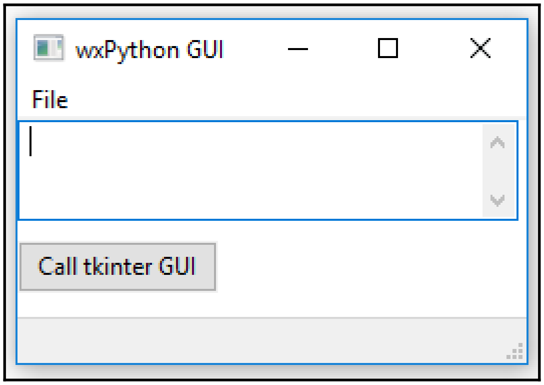
Z:\

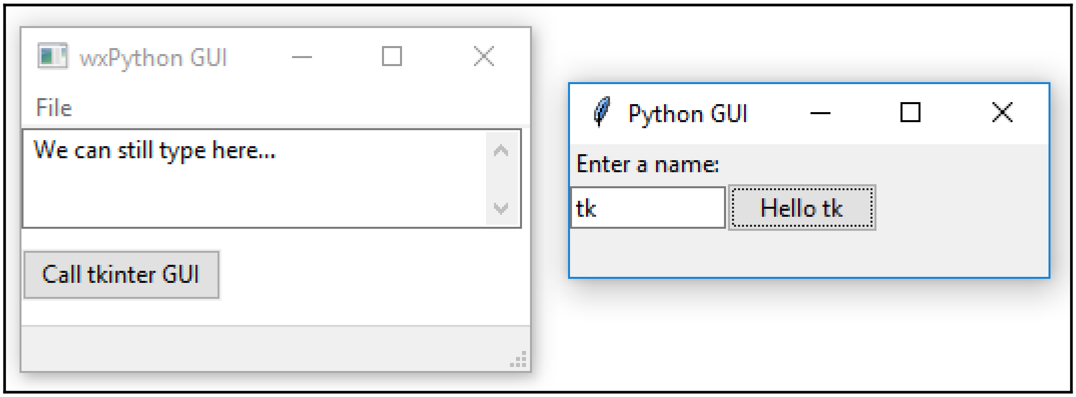
Z:\Backup



1

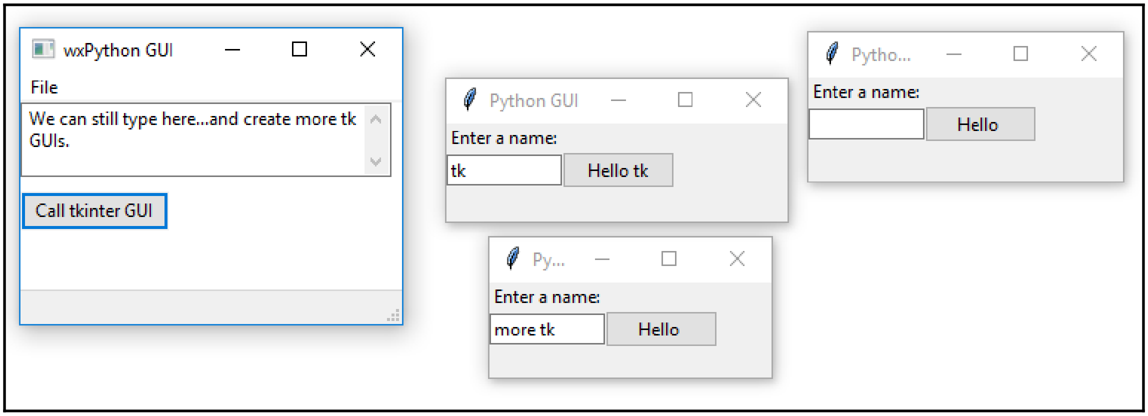






We can still type here...

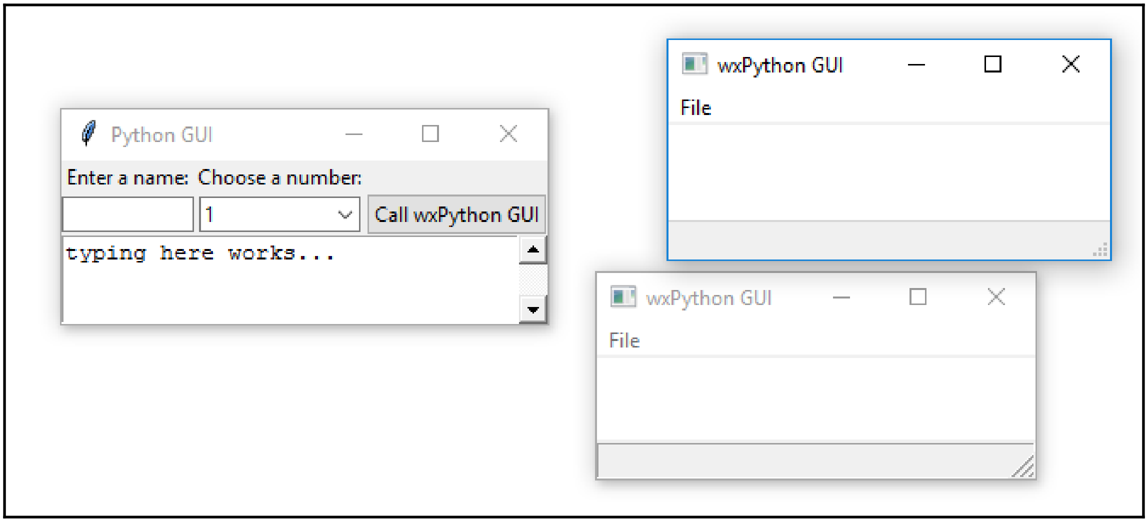
tk



We can still type here...and create more tk GUIs.

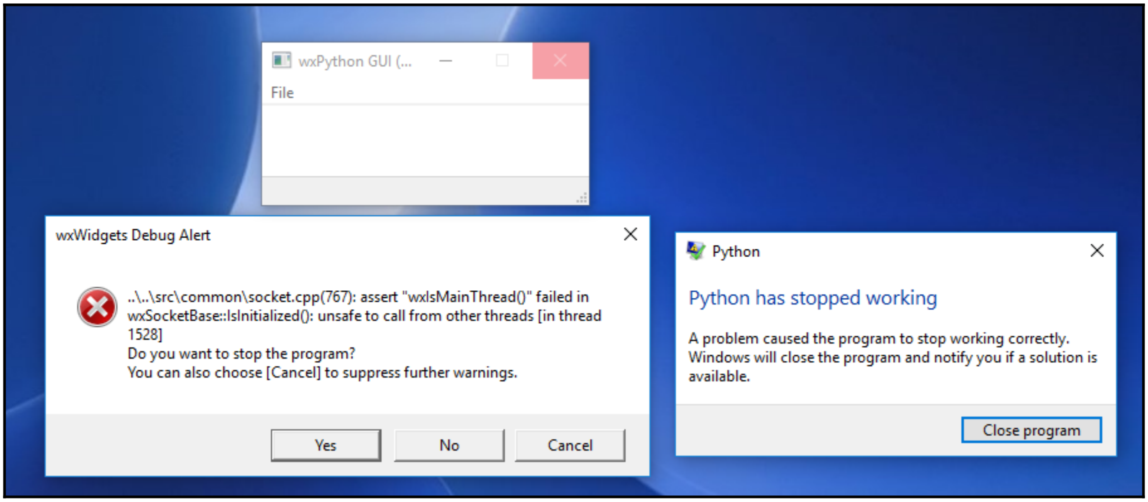
Enter a name: tk

Enter a name: more tk



```
typing here works...
```

Choose a number: 1



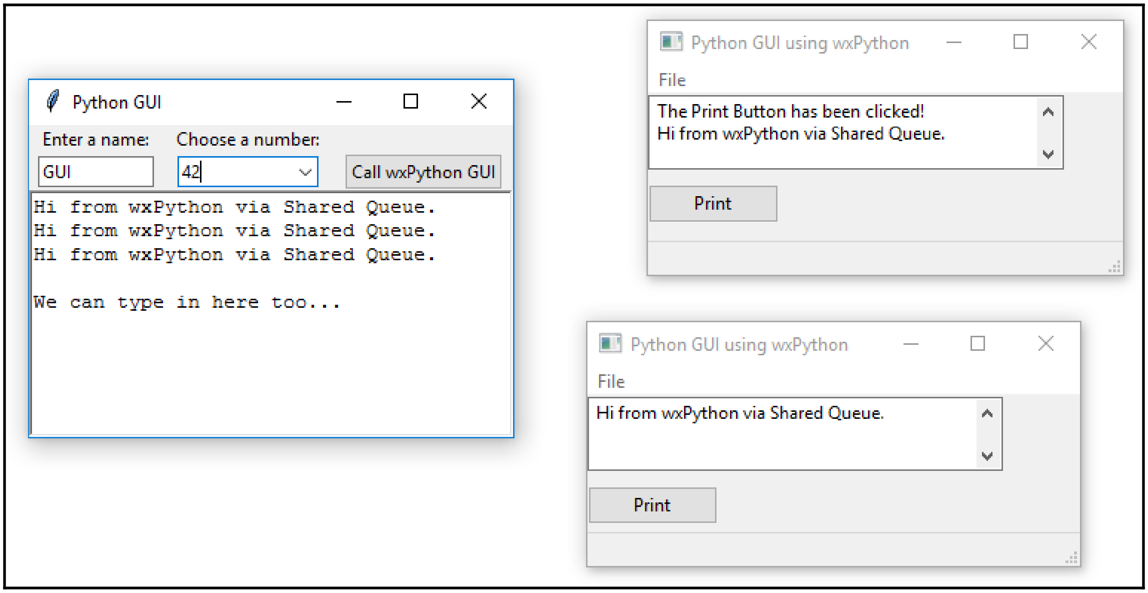
wxWidgets Debug Alert

```
..\..\src\common\socket.cpp(767): assert "wxIsMainThread()" failed in
wxSocketBase::IsInitialized(): unsafe to call from other threads [in thread
1528]
Do you want to stop the program?
You can also choose [Cancel] to suppress further warnings.
```

Python

Python has stopped working

A problem caused the program to stop working correctly. Windows will close the program and notify you if a solution is available.



Enter a name: GUI

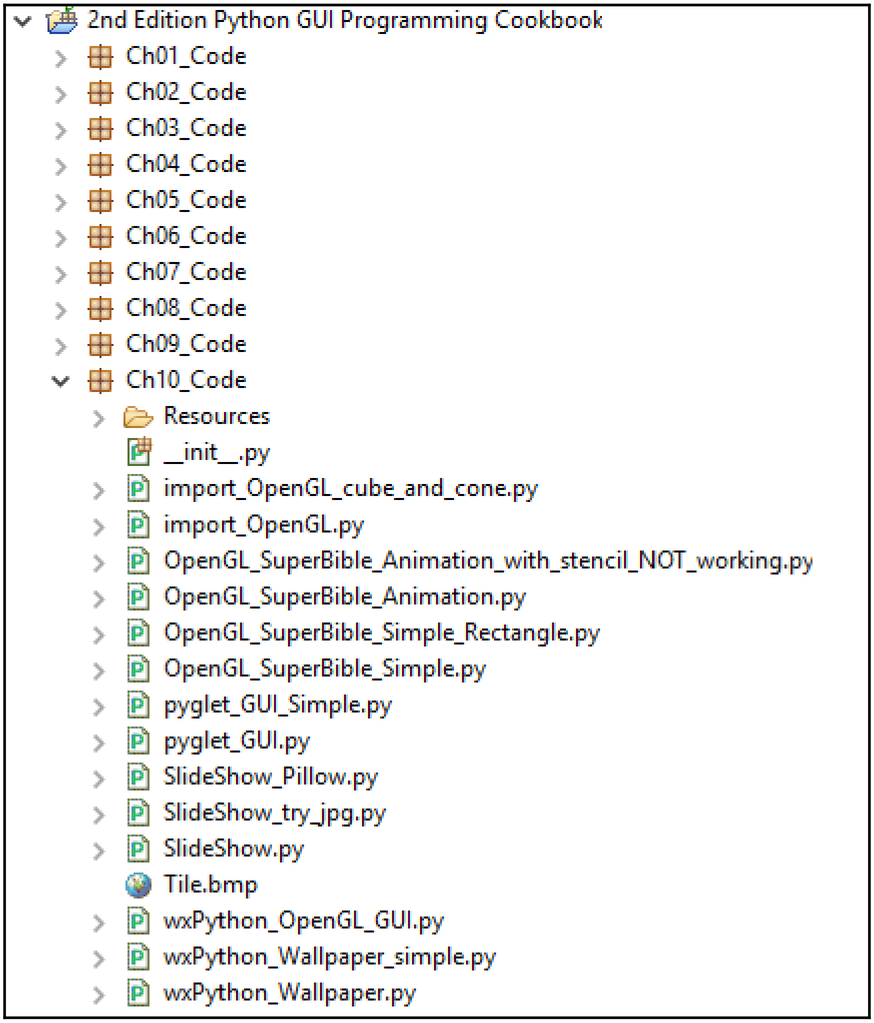
Choose a number: 42

```
Hi from wxPython via Shared Queue.
Hi from wxPython via Shared Queue.
Hi from wxPython via Shared Queue.

We can type in here too...
```

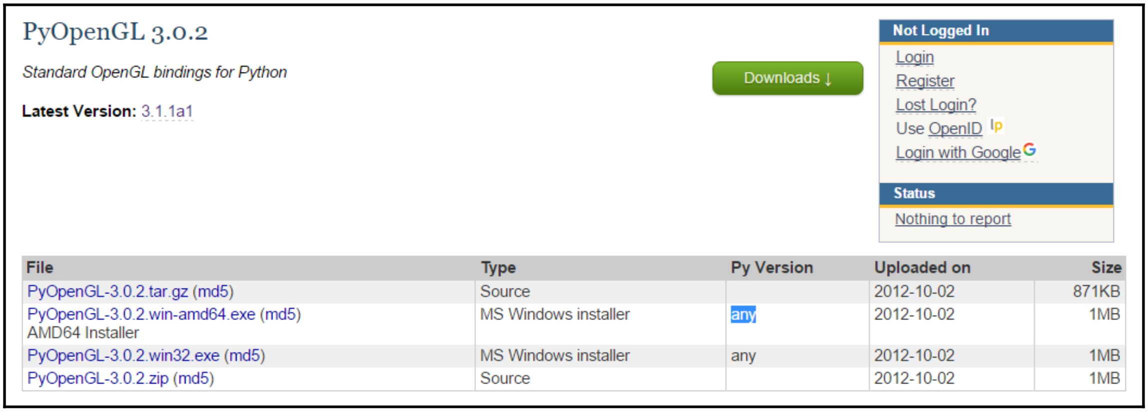
```
The Print Button has been clicked!
Hi from wxPython via Shared Queue.
```

```
Hi from wxPython via Shared Queue.
```

Chapter 10:
Creating Amazing 3D GUIs with PyOpenGL and PyGLet



- 2nd Edition Python GUI Programming Cookbook
  - Ch01_Code
  - Ch02_Code
  - Ch03_Code
  - Ch04_Code
  - Ch05_Code
  - Ch06_Code
  - Ch07_Code
  - Ch08_Code
  - Ch09_Code
  - Ch10_Code
    - Resources
    - __init__.py
    - import_OpenGL_cube_and_cone.py
    - import_OpenGL.py
    - OpenGL_SuperBible_Animation_with_stencil_NOT_working.py
    - OpenGL_SuperBible_Animation.py
    - OpenGL_SuperBible_Simple_Rectangle.py
    - OpenGL_SuperBible_Simple.py
    - pyglet_GUI_Simple.py
    - pyglet_GUI.py
    - SlideShow_Pillow.py
    - SlideShow_try_jpg.py
    - SlideShow.py
    - Tile.bmp
    - wxPython_OpenGL_GUI.py
    - wxPython_Wallpaper_simple.py
    - wxPython_Wallpaper.py

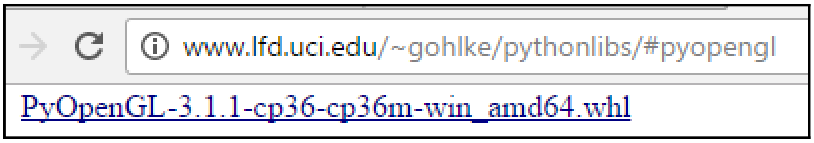


PyOpenGL 3.0.2

Standard OpenGL bindings for Python

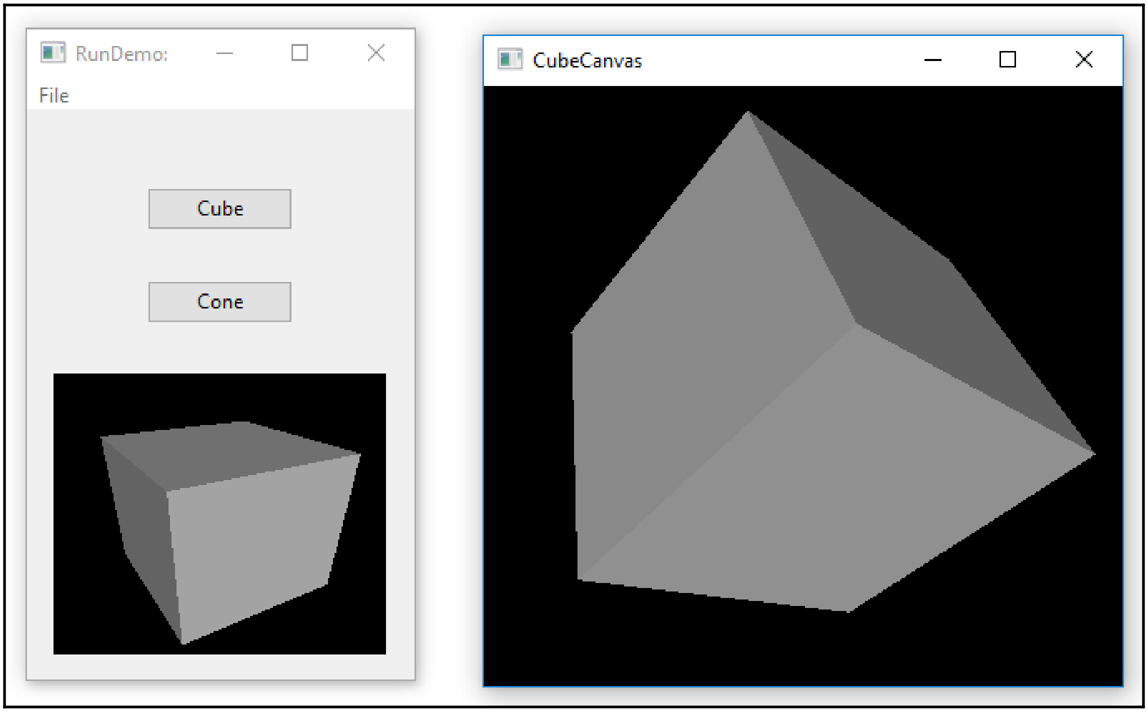
Latest Version: 3.1.1a1

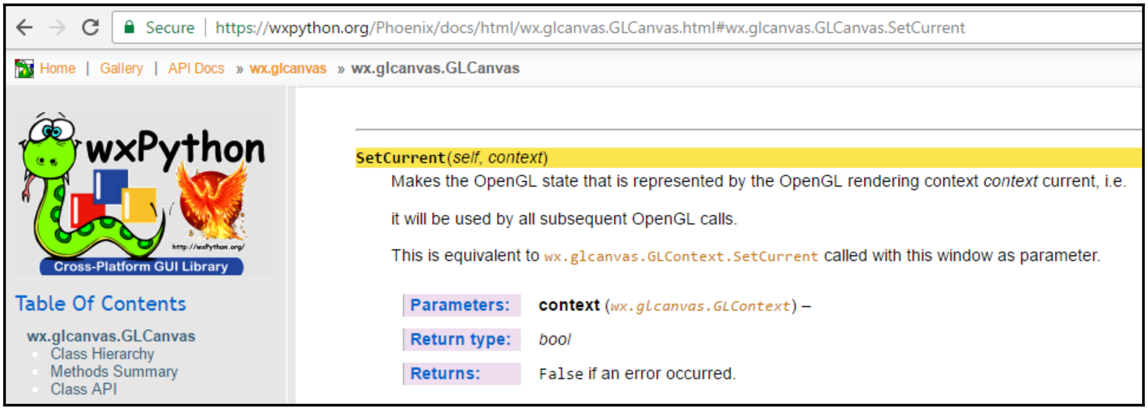
File	Type	Py Version	Uploaded on	Size
PyOpenGL-3.0.2.tar.gz (md5)	Source		2012-10-02	871KB
PyOpenGL-3.0.2.win-amd64.exe (md5) AMD64 Installer	MS Windows installer	any	2012-10-02	1MB
PyOpenGL-3.0.2.win32.exe (md5)	MS Windows installer	any	2012-10-02	1MB
PyOpenGL-3.0.2.zip (md5)	Source		2012-10-02	1MB



www.lfd.uci.edu/~gohlke/pythonlibs/#pyopengl

PyOpenGL-3.1.1-cp36-cp36m-win_amd64.whl





Secure | https://wxpython.org/Phoenix/docs/html/wx.glcanvas.GLCanvas.html#wx.glcanvas.GLCanvas.SetCurrent

Home | Gallery | API Docs » wx.glcanvas » wx.glcanvas.GLCanvas

wxPython

Cross-Platform GUI Library

Table Of Contents

wx.glcanvas.GLCanvas
- Class Hierarchy
- Methods Summary
- Class API

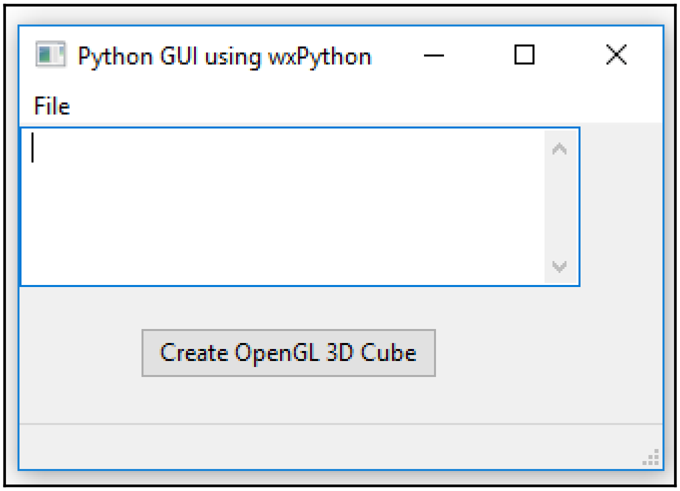
`SetCurrent`(*self*, *context*)

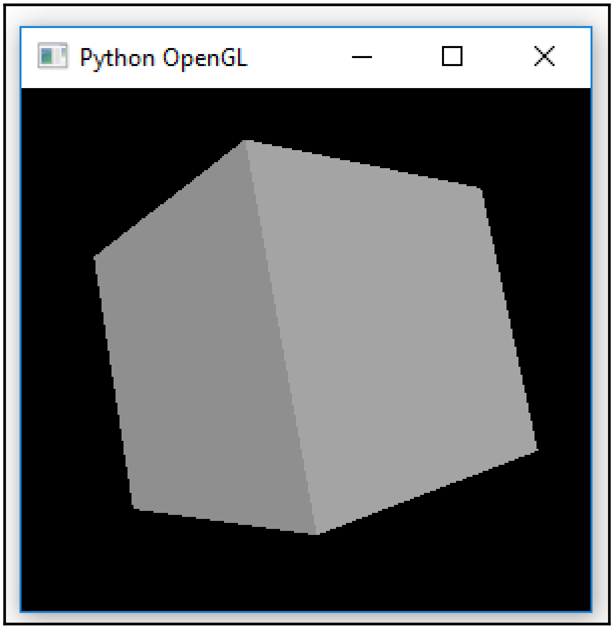
Makes the OpenGL state that is represented by the OpenGL rendering context *context* current, i.e.

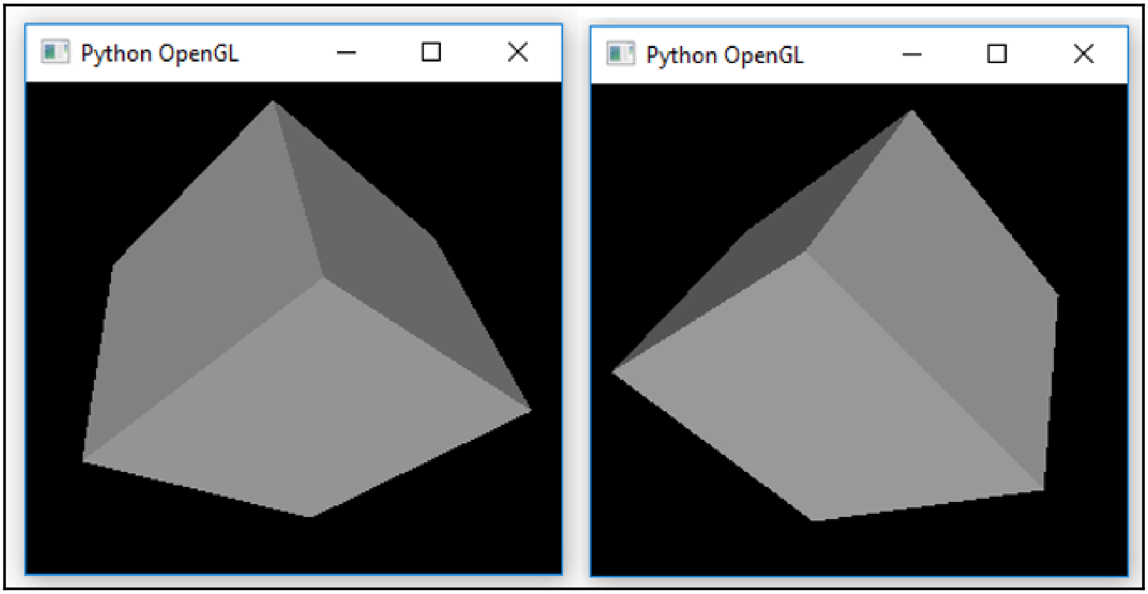
it will be used by all subsequent OpenGL calls.

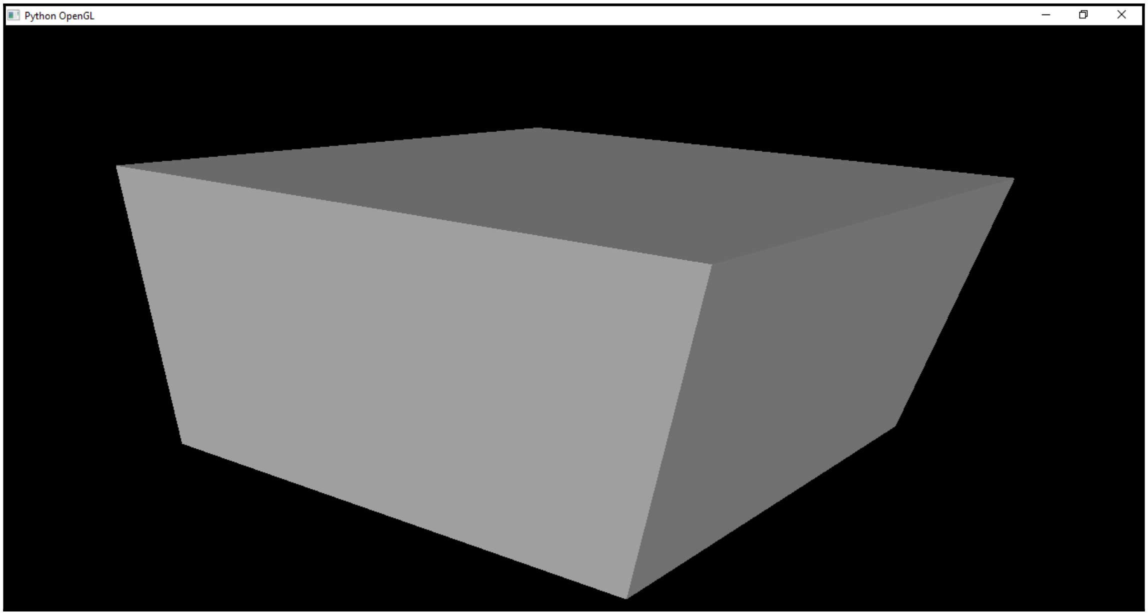
This is equivalent to `wx.glcanvas.GLContext.SetCurrent` called with this window as parameter.

Parameters:	**context** (*`wx.glcanvas.GLContext`*) –
Return type:	*bool*
Returns:	`False` if an error occurred.

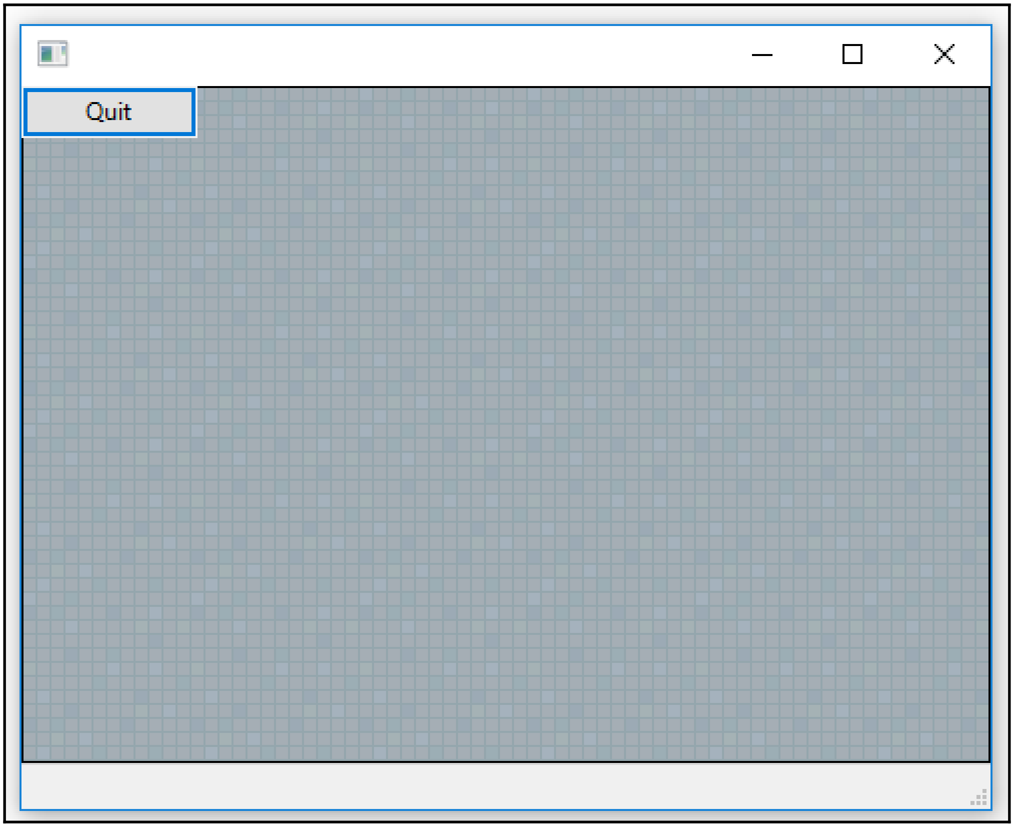


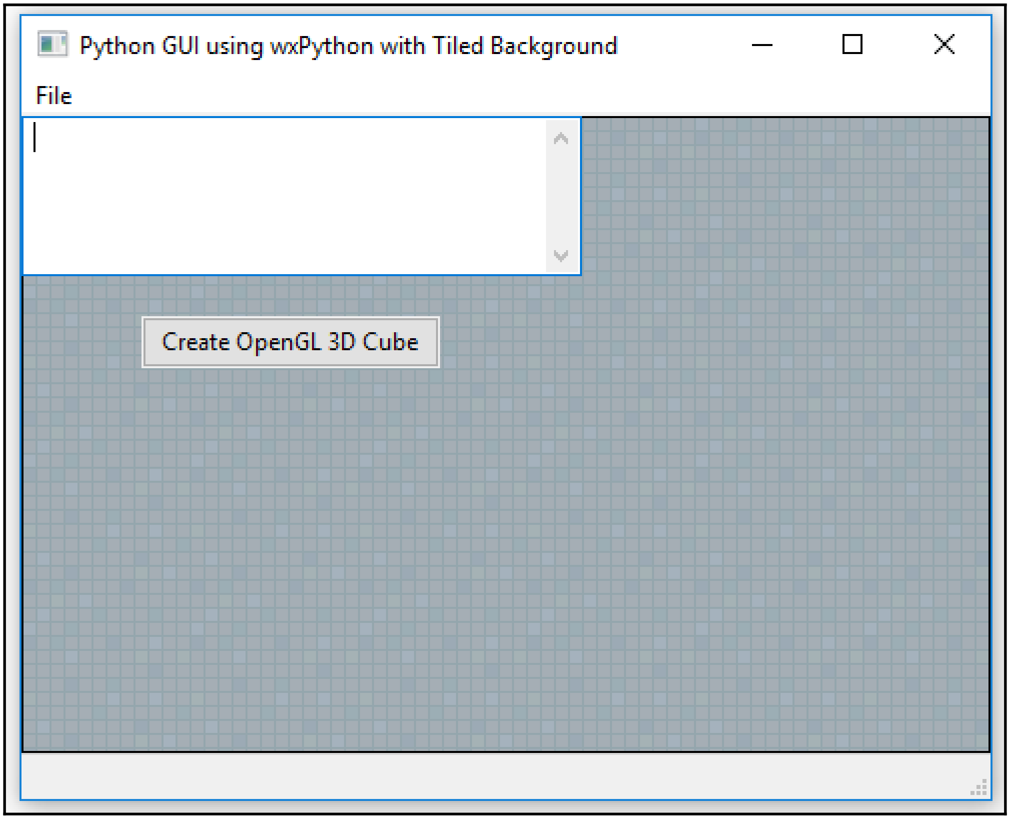

Python OpenGL

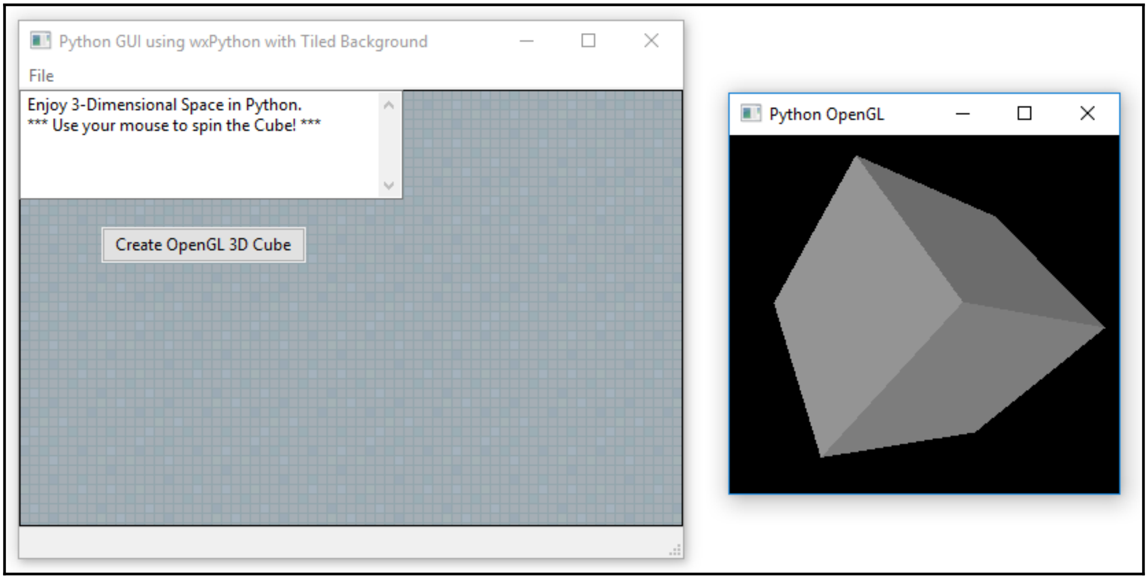

Python OpenGL
Python OpenGL



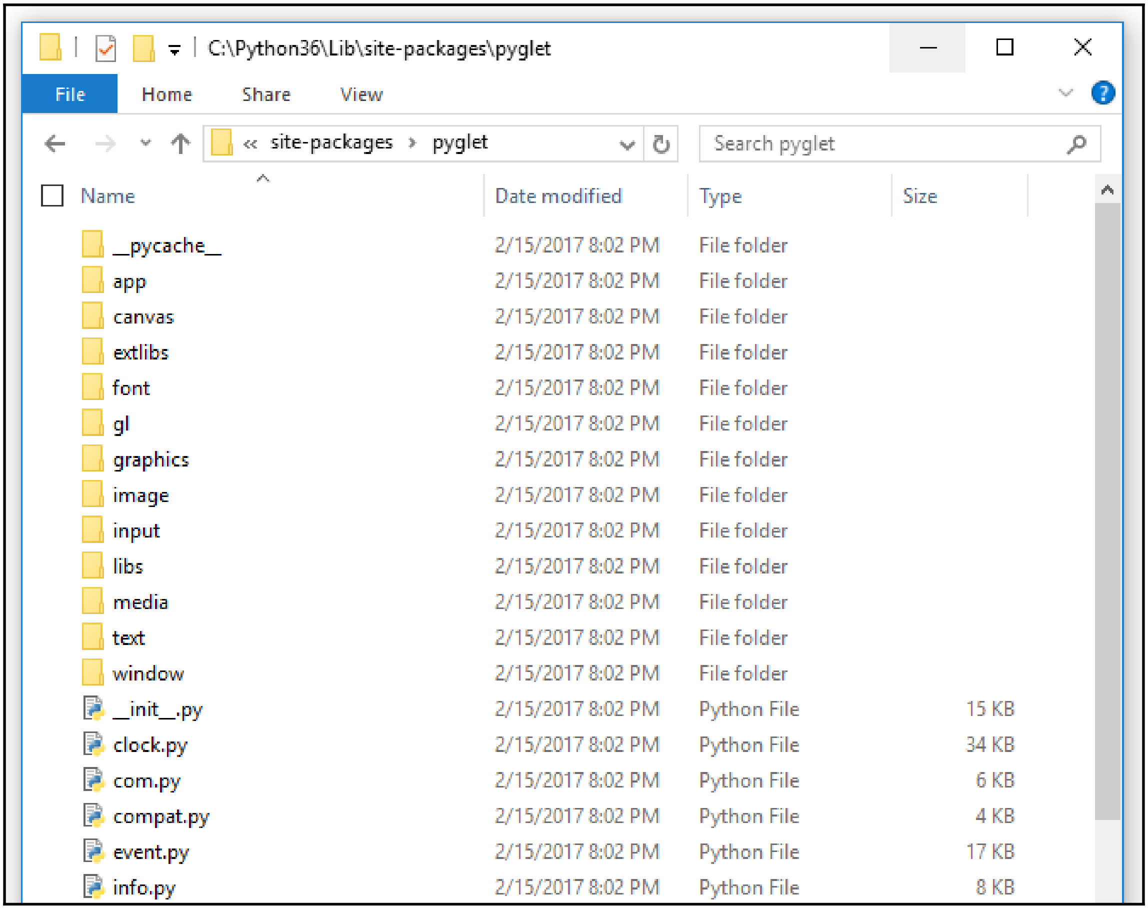
Python OpenGL


Quit


Python GUI using wxPython with Tiled Background
File
Create OpenGL 3D Cube



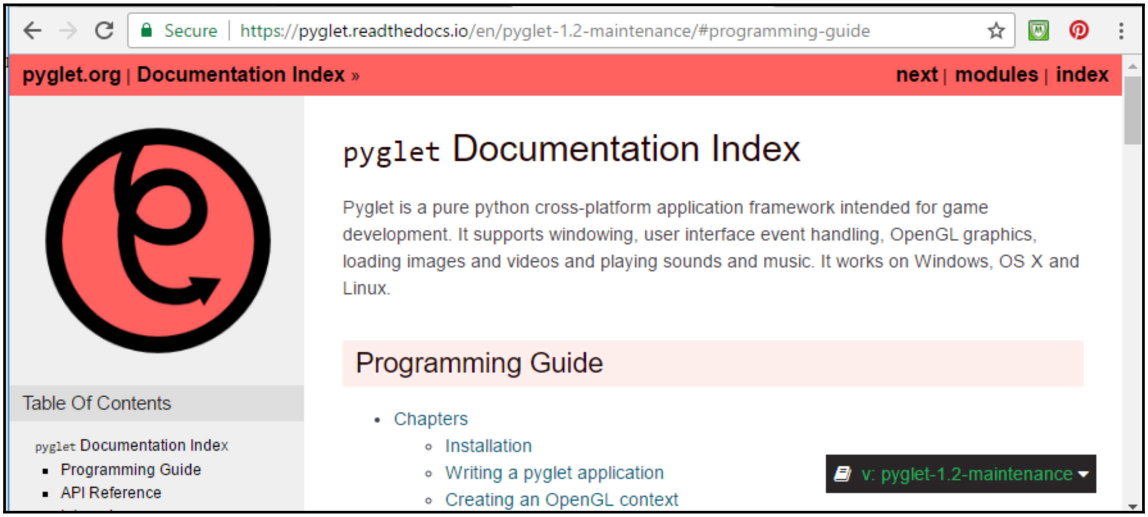
Enjoy 3-Dimensional Space in Python.
*** Use your mouse to spin the Cube! ***



C:\Python36\Lib\site-packages\pyglet

« site-packages › pyglet

Name	Date modified	Type	Size
__pycache__	2/15/2017 8:02 PM	File folder	
app	2/15/2017 8:02 PM	File folder	
canvas	2/15/2017 8:02 PM	File folder	
extlibs	2/15/2017 8:02 PM	File folder	
font	2/15/2017 8:02 PM	File folder	
gl	2/15/2017 8:02 PM	File folder	
graphics	2/15/2017 8:02 PM	File folder	
image	2/15/2017 8:02 PM	File folder	
input	2/15/2017 8:02 PM	File folder	
libs	2/15/2017 8:02 PM	File folder	
media	2/15/2017 8:02 PM	File folder	
text	2/15/2017 8:02 PM	File folder	
window	2/15/2017 8:02 PM	File folder	
__init__.py	2/15/2017 8:02 PM	Python File	15 KB
clock.py	2/15/2017 8:02 PM	Python File	34 KB
com.py	2/15/2017 8:02 PM	Python File	6 KB
compat.py	2/15/2017 8:02 PM	Python File	4 KB
event.py	2/15/2017 8:02 PM	Python File	17 KB
info.py	2/15/2017 8:02 PM	Python File	8 KB


Secure
https://pyglet.readthedocs.io/en/pyglet-1.2-maintenance/#programming-guide
pyglet.org | Documentation Index »
next | modules | index
v: pyglet-1.2-maintenance

pyglet Documentation Index

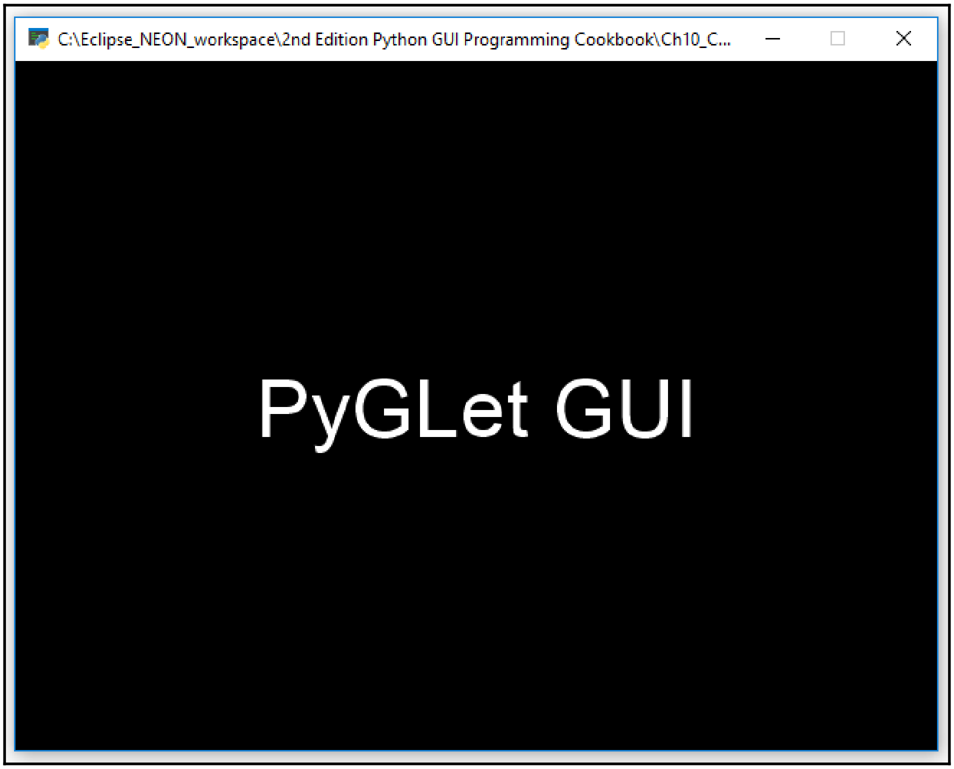
Pyglet is a pure python cross-platform application framework intended for game development. It supports windowing, user interface event handling, OpenGL graphics, loading images and videos and playing sounds and music. It works on Windows, OS X and Linux.

Programming Guide

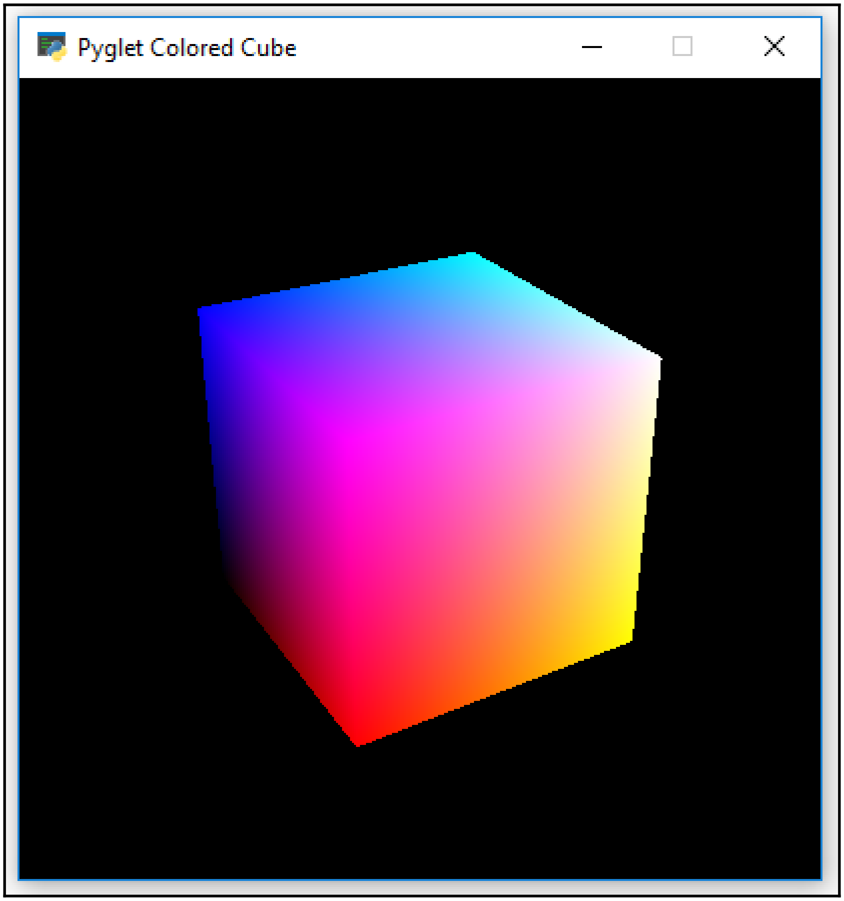
- Chapters
  - Installation
  - Writing a pyglet application
  - Creating an OpenGL context

Table Of Contents

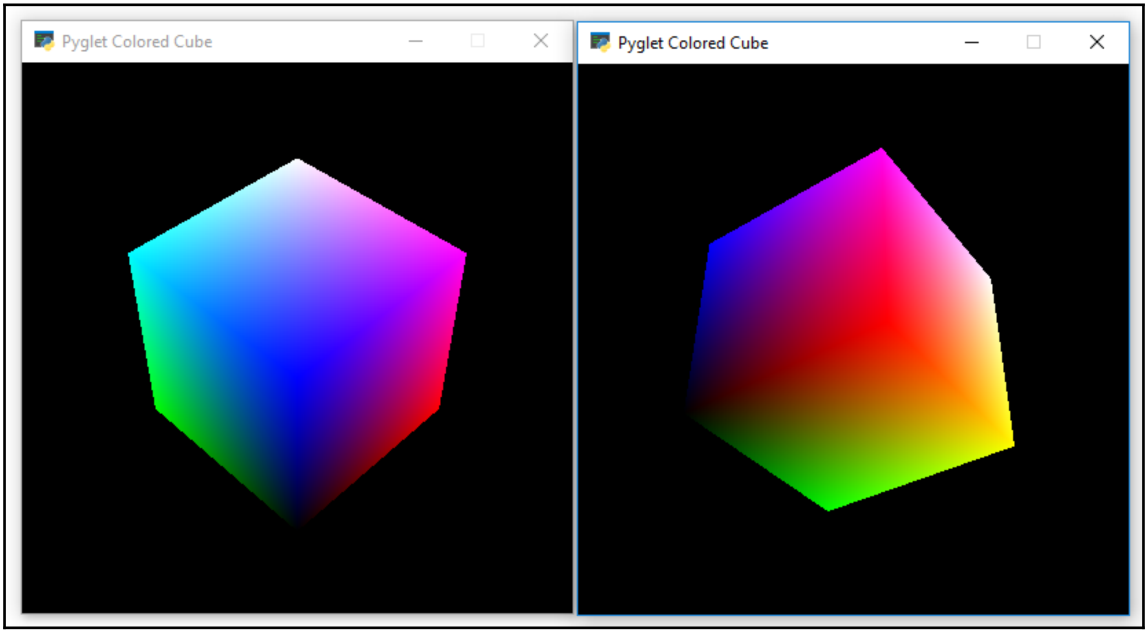
pyglet Documentation Index
- Programming Guide
- API Reference

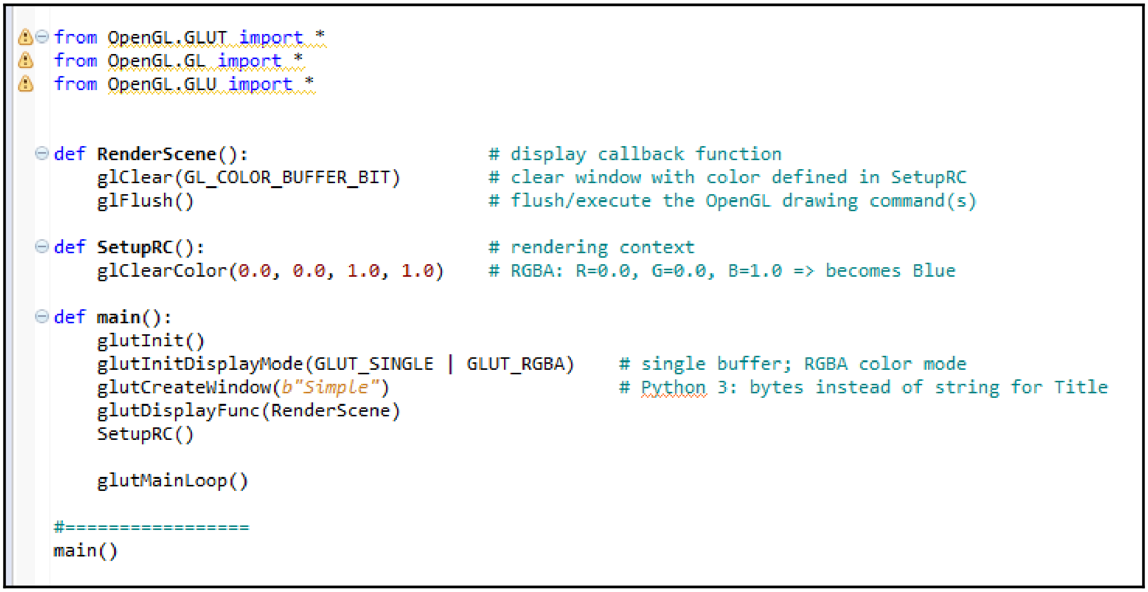

C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch10_C...

PyGLet GUI

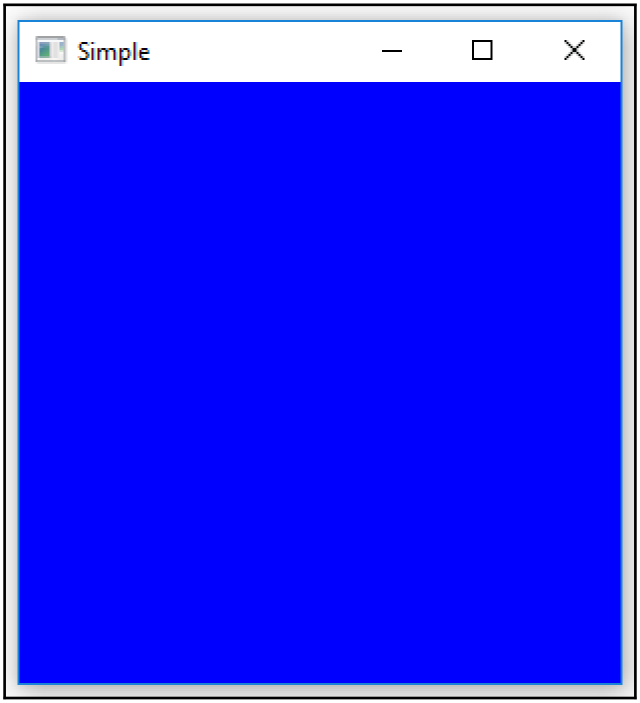


Pyglet Colored Cube

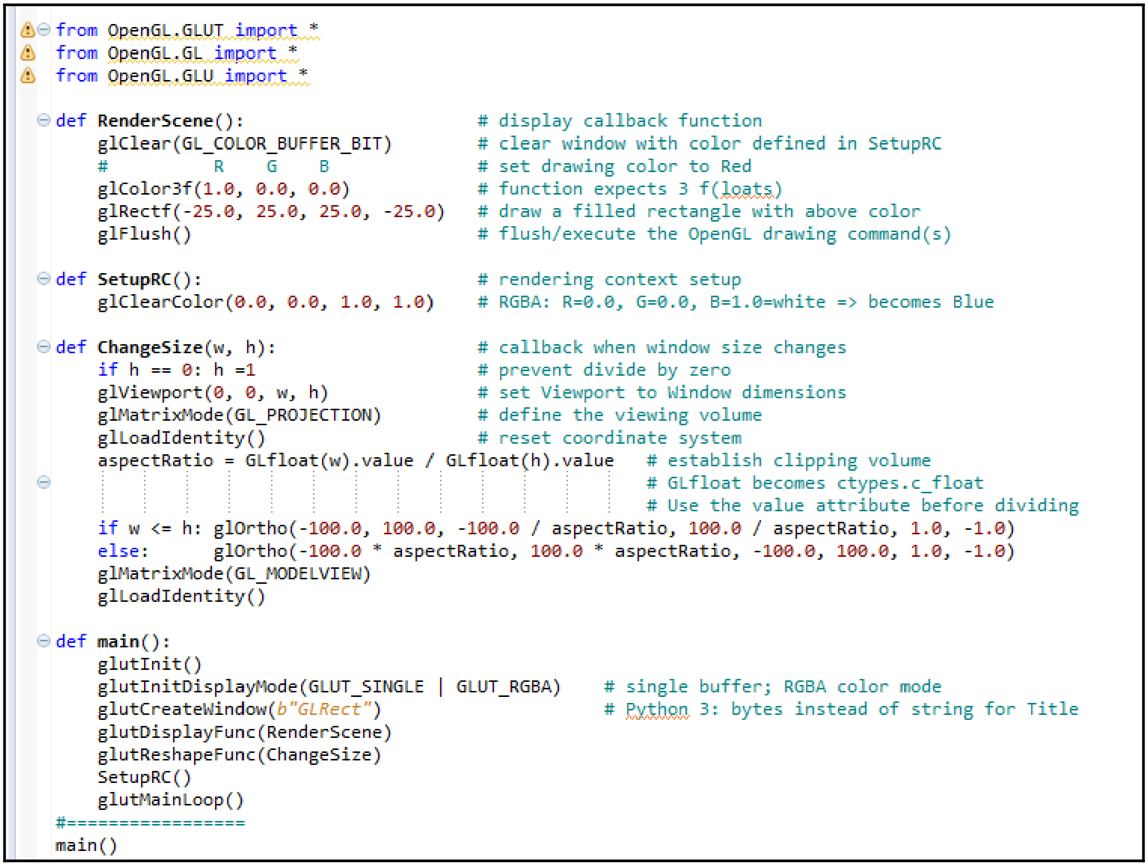




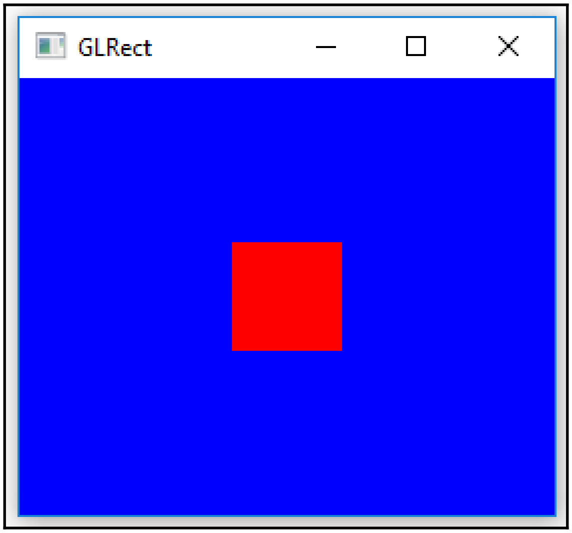
```
from OpenGL.GLUT import *
from OpenGL.GL import *
from OpenGL.GLU import *


def RenderScene():                      # display callback function
    glClear(GL_COLOR_BUFFER_BIT)        # clear window with color defined in SetupRC
    glFlush()                           # flush/execute the OpenGL drawing command(s)

def SetupRC():                          # rendering context
    glClearColor(0.0, 0.0, 1.0, 1.0)    # RGBA: R=0.0, G=0.0, B=1.0 => becomes Blue

def main():
    glutInit()
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA)    # single buffer; RGBA color mode
    glutCreateWindow(b"Simple")                     # Python 3: bytes instead of string for Title
    glutDisplayFunc(RenderScene)
    SetupRC()

    glutMainLoop()

#=================
main()
```

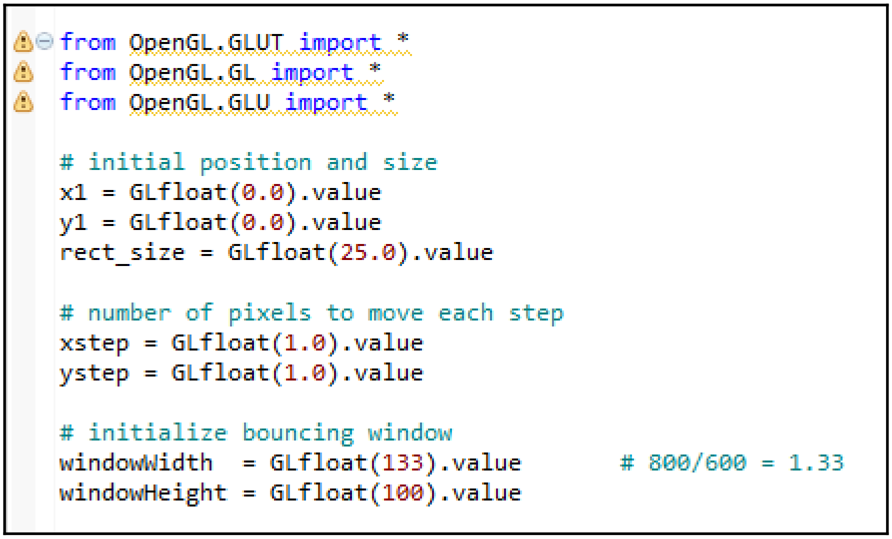


Simple

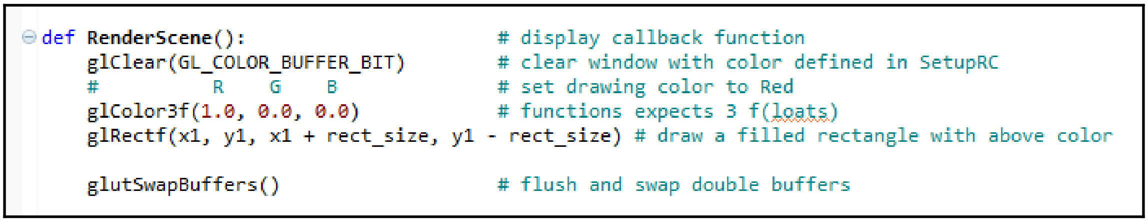


```
from OpenGL.GLUT import *
from OpenGL.GL import *
from OpenGL.GLU import *

def RenderScene():                      # display callback function
    glClear(GL_COLOR_BUFFER_BIT)        # clear window with color defined in SetupRC
    #          R    G    B              # set drawing color to Red
    glColor3f(1.0, 0.0, 0.0)            # function expects 3 f(loats)
    glRectf(-25.0, 25.0, 25.0, -25.0)   # draw a filled rectangle with above color
    glFlush()                           # flush/execute the OpenGL drawing command(s)

def SetupRC():                          # rendering context setup
    glClearColor(0.0, 0.0, 1.0, 1.0)    # RGBA: R=0.0, G=0.0, B=1.0=white => becomes Blue

def ChangeSize(w, h):                   # callback when window size changes
    if h == 0: h =1                     # prevent divide by zero
    glViewport(0, 0, w, h)              # set Viewport to Window dimensions
    glMatrixMode(GL_PROJECTION)         # define the viewing volume
    glLoadIdentity()                    # reset coordinate system
    aspectRatio = GLfloat(w).value / GLfloat(h).value   # establish clipping volume
                                                        # GLfloat becomes ctypes.c_float
                                                        # Use the value attribute before dividing
    if w <= h: glOrtho(-100.0, 100.0, -100.0 / aspectRatio, 100.0 / aspectRatio, 1.0, -1.0)
    else:      glOrtho(-100.0 * aspectRatio, 100.0 * aspectRatio, -100.0, 100.0, 1.0, -1.0)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()

def main():
    glutInit()
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA)    # single buffer; RGBA color mode
    glutCreateWindow(b"GLRect")                     # Python 3: bytes instead of string for Title
    glutDisplayFunc(RenderScene)
    glutReshapeFunc(ChangeSize)
    SetupRC()
    glutMainLoop()
#=================
main()
```

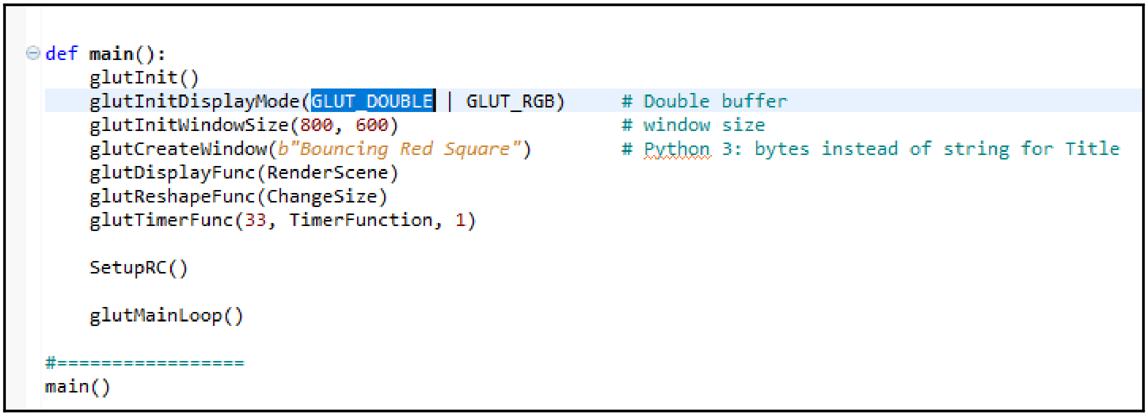




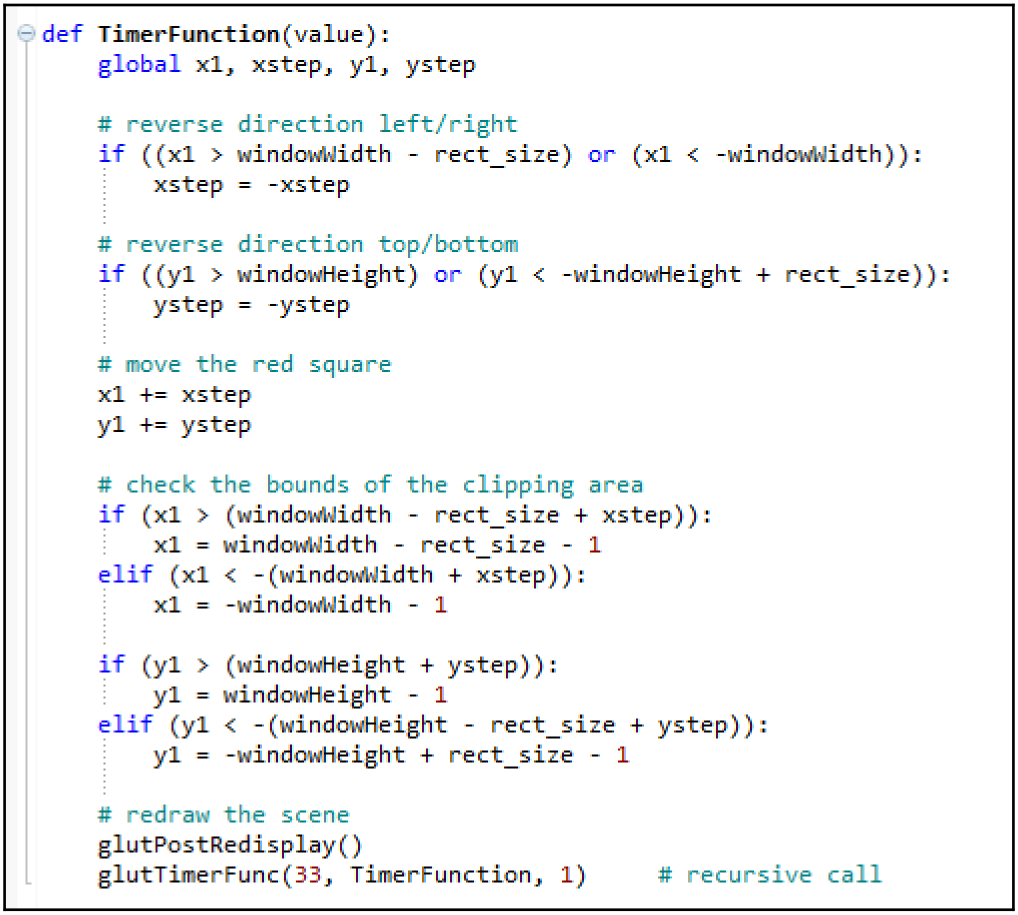
```
from OpenGL.GLUT import *
from OpenGL.GL import *
from OpenGL.GLU import *

# initial position and size
x1 = GLfloat(0.0).value
y1 = GLfloat(0.0).value
rect_size = GLfloat(25.0).value

# number of pixels to move each step
xstep = GLfloat(1.0).value
ystep = GLfloat(1.0).value

# initialize bouncing window
windowWidth  = GLfloat(133).value       # 800/600 = 1.33
windowHeight = GLfloat(100).value
```



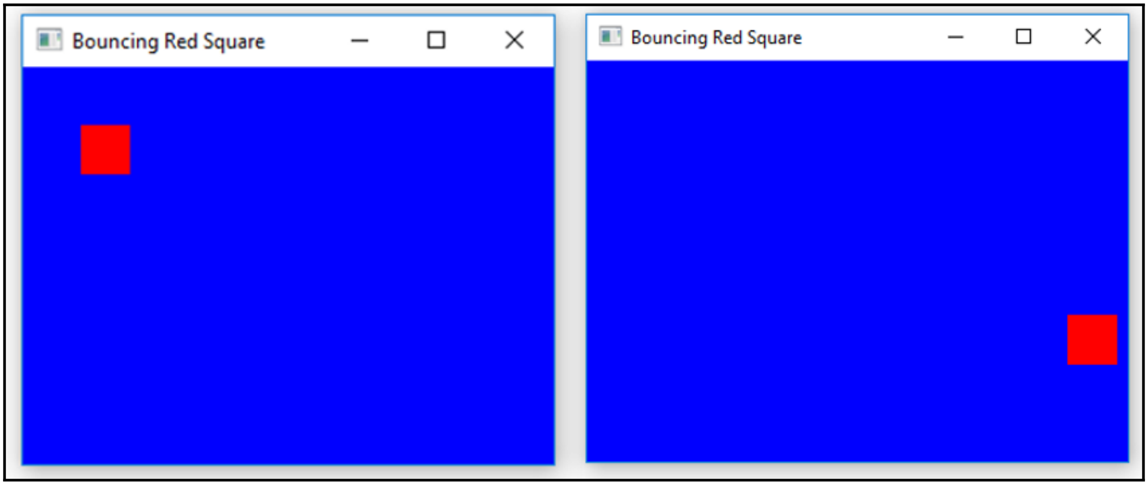
```
def RenderScene():                      # display callback function
    glClear(GL_COLOR_BUFFER_BIT)        # clear window with color defined in SetupRC
    #          R    G    B              # set drawing color to Red
    glColor3f(1.0, 0.0, 0.0)            # functions expects 3 f(loats)
    glRectf(x1, y1, x1 + rect_size, y1 - rect_size) # draw a filled rectangle with above color

    glutSwapBuffers()                   # flush and swap double buffers
```



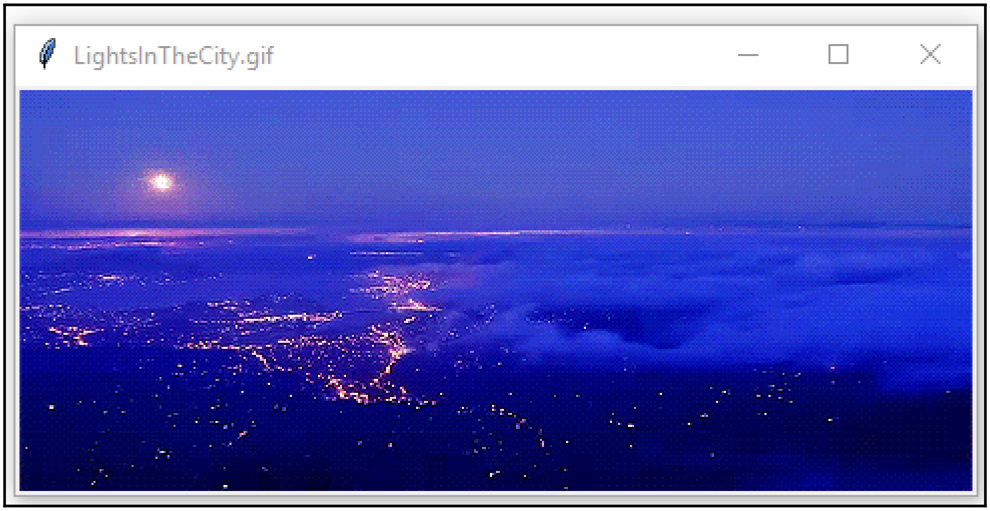
```
def main():
    glutInit()
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB)     # Double buffer
    glutInitWindowSize(800, 600)                    # window size
    glutCreateWindow(b"Bouncing Red Square")        # Python 3: bytes instead of string for Title
    glutDisplayFunc(RenderScene)
    glutReshapeFunc(ChangeSize)
    glutTimerFunc(33, TimerFunction, 1)

    SetupRC()

    glutMainLoop()

#==================
main()
```

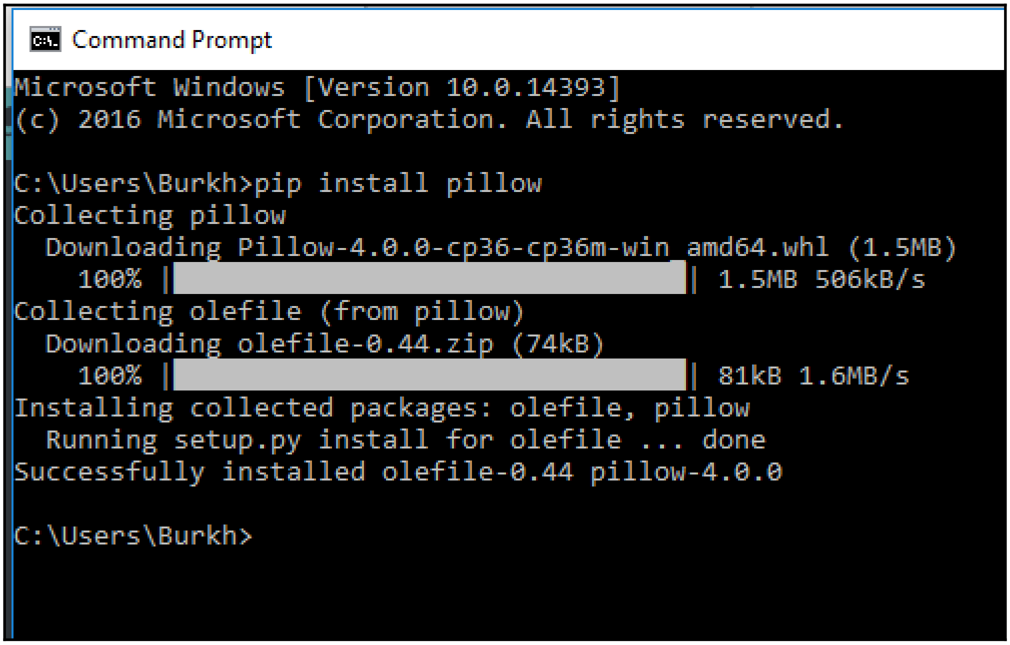


```
def TimerFunction(value):
    global x1, xstep, y1, ystep

    # reverse direction left/right
    if ((x1 > windowWidth - rect_size) or (x1 < -windowWidth)):
        xstep = -xstep

    # reverse direction top/bottom
    if ((y1 > windowHeight) or (y1 < -windowHeight + rect_size)):
        ystep = -ystep

    # move the red square
    x1 += xstep
    y1 += ystep

    # check the bounds of the clipping area
    if (x1 > (windowWidth - rect_size + xstep)):
        x1 = windowWidth - rect_size - 1
    elif (x1 < -(windowWidth + xstep)):
        x1 = -windowWidth - 1

    if (y1 > (windowHeight + ystep)):
        y1 = windowHeight - 1
    elif (y1 < -(windowHeight - rect_size + ystep)):
        y1 = -windowHeight + rect_size - 1

    # redraw the scene
    glutPostRedisplay()
    glutTimerFunc(33, TimerFunction, 1)     # recursive call
```

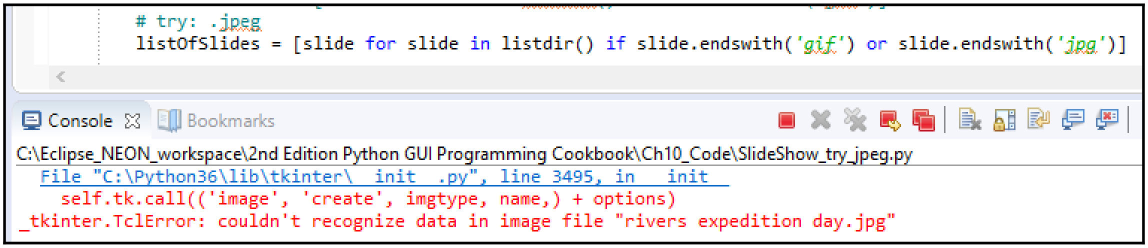

Bouncing Red Square
Bouncing Red Square

Galaxy.gif



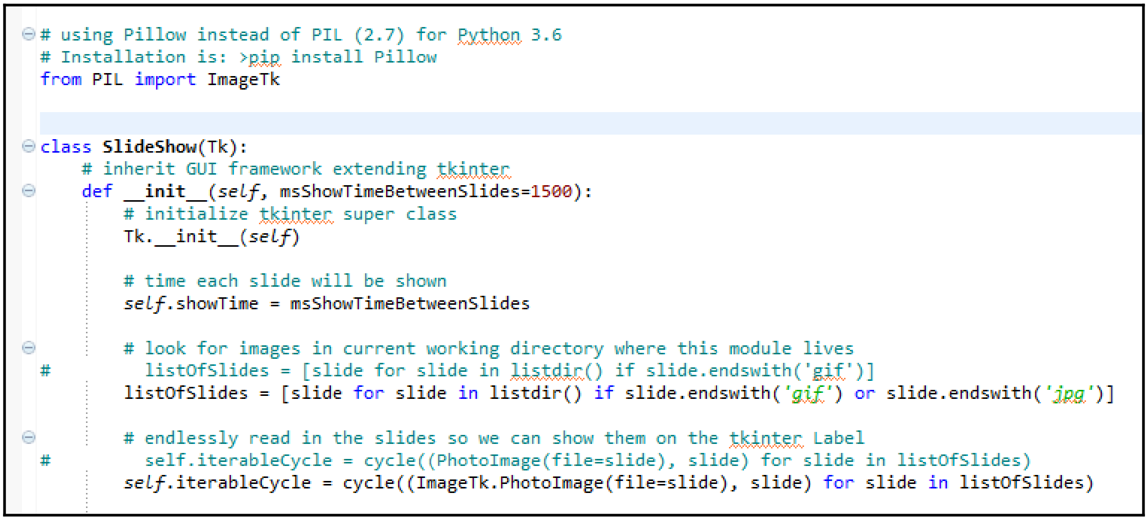


```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Burkh>pip install pillow
Collecting pillow
  Downloading Pillow-4.0.0-cp36-cp36m-win_amd64.whl (1.5MB)
    100% |████████████████████████████████| 1.5MB 506kB/s
Collecting olefile (from pillow)
  Downloading olefile-0.44.zip (74kB)
    100% |████████████████████████████████| 81kB 1.6MB/s
Installing collected packages: olefile, pillow
  Running setup.py install for olefile ... done
Successfully installed olefile-0.44 pillow-4.0.0

C:\Users\Burkh>
```



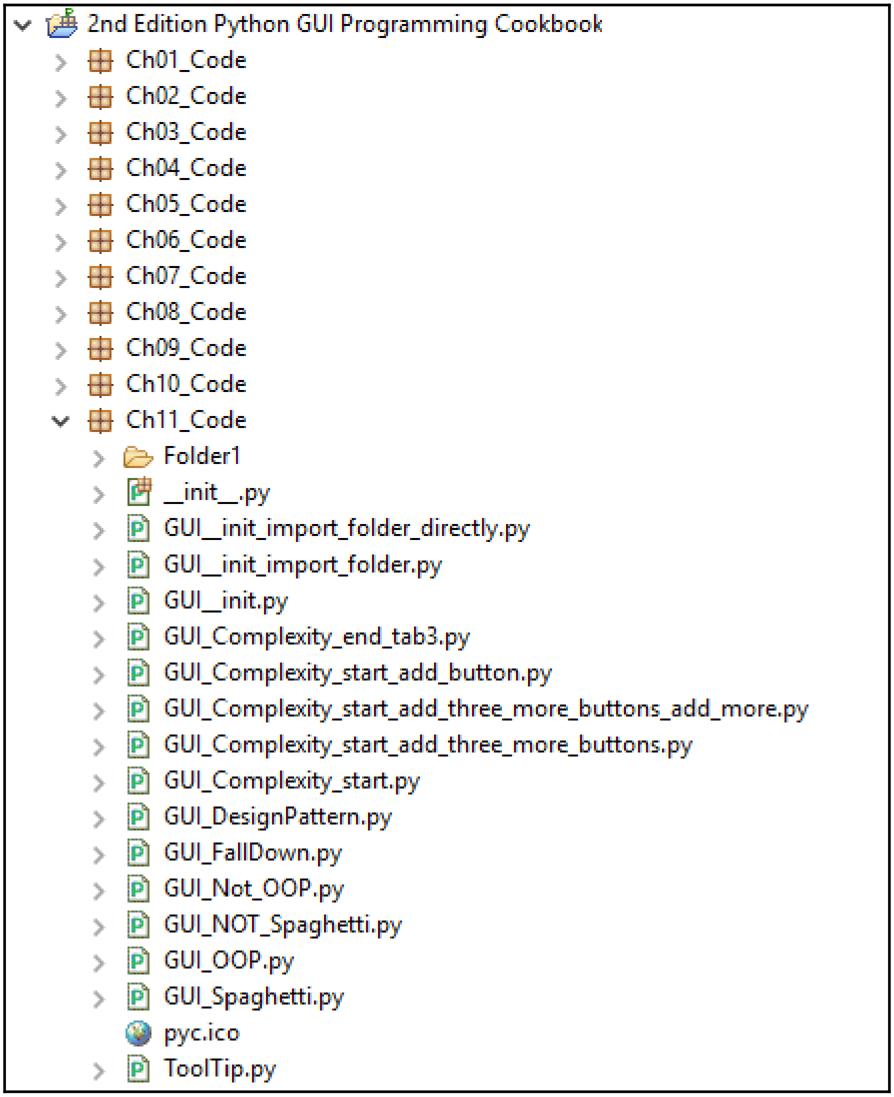
```
        # try: .jpeg
        listOfSlides = [slide for slide in listdir() if slide.endswith('gif') or slide.endswith('jpg')]
```

```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch10_Code\SlideShow_try_jpeg.py
  File "C:\Python36\lib\tkinter\__init__.py", line 3495, in __init__
    self.tk.call(('image', 'create', imgtype, name,) + options)
_tkinter.TclError: couldn't recognize data in image file "rivers expedition day.jpg"
```

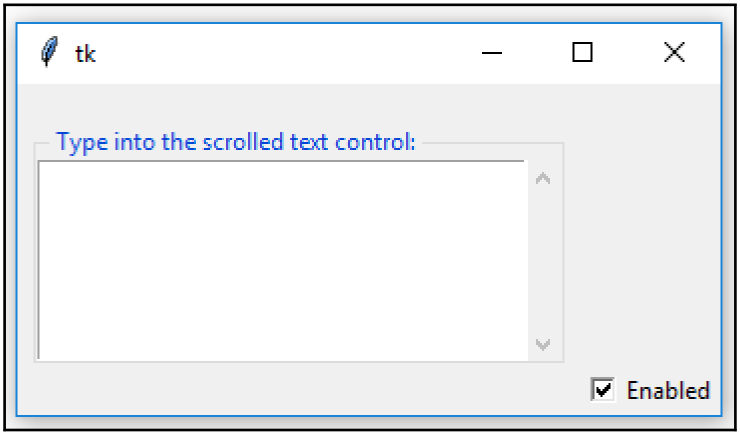


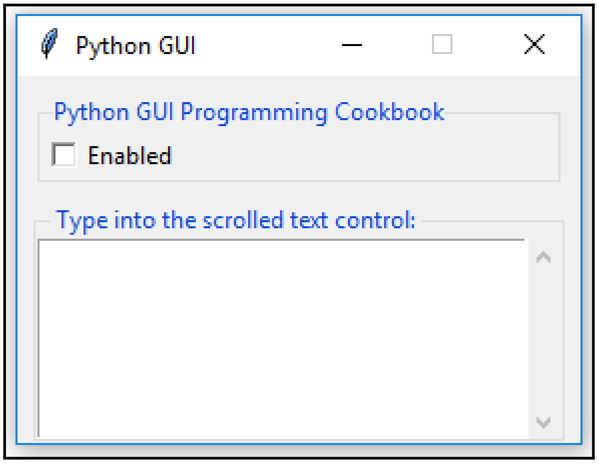
```
# using Pillow instead of PIL (2.7) for Python 3.6
# Installation is: >pip install Pillow
from PIL import ImageTk


class SlideShow(Tk):
    # inherit GUI framework extending tkinter
    def __init__(self, msShowTimeBetweenSlides=1500):
        # initialize tkinter super class
        Tk.__init__(self)

        # time each slide will be shown
        self.showTime = msShowTimeBetweenSlides

        # look for images in current working directory where this module lives
#         listOfSlides = [slide for slide in listdir() if slide.endswith('gif')]
        listOfSlides = [slide for slide in listdir() if slide.endswith('gif') or slide.endswith('jpg')]

        # endlessly read in the slides so we can show them on the tkinter Label
#         self.iterableCycle = cycle((PhotoImage(file=slide), slide) for slide in listOfSlides)
        self.iterableCycle = cycle((ImageTk.PhotoImage(file=slide), slide) for slide in listOfSlides)
```

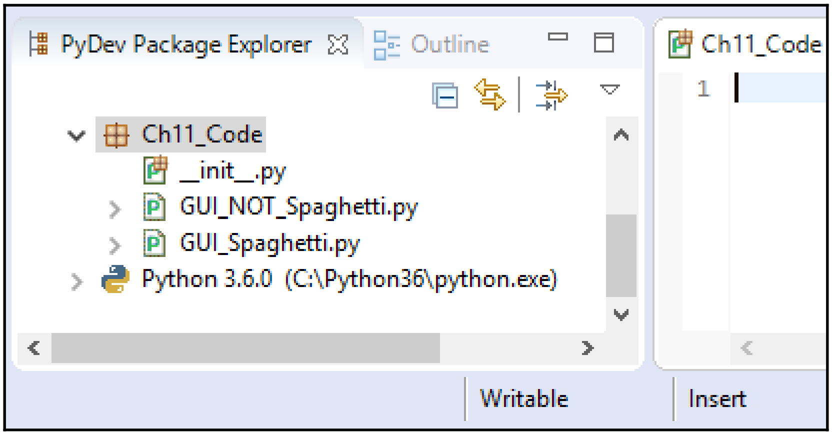
Chapter11: Best Practices

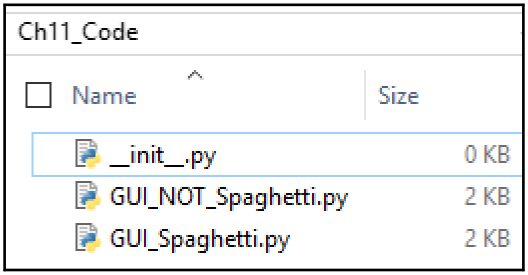


```
2nd Edition Python GUI Programming Cookbook
    Ch01_Code
    Ch02_Code
    Ch03_Code
    Ch04_Code
    Ch05_Code
    Ch06_Code
    Ch07_Code
    Ch08_Code
    Ch09_Code
    Ch10_Code
    Ch11_Code
        Folder1
        __init__.py
        GUI__init_import_folder_directly.py
        GUI__init_import_folder.py
        GUI__init.py
        GUI_Complexity_end_tab3.py
        GUI_Complexity_start_add_button.py
        GUI_Complexity_start_add_three_more_buttons_add_more.py
        GUI_Complexity_start_add_three_more_buttons.py
        GUI_Complexity_start.py
        GUI_DesignPattern.py
        GUI_FallDown.py
        GUI_Not_OOP.py
        GUI_NOT_Spaghetti.py
        GUI_OOP.py
        GUI_Spaghetti.py
        pyc.ico
        ToolTip.py
```


tk
Type into the scrolled text control:
☑ Enabled

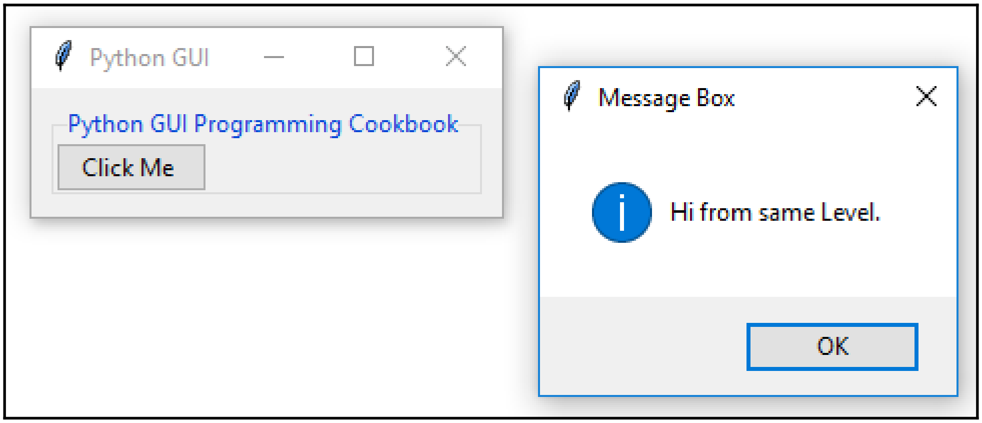

Python GUI
Python GUI Programming Cookbook
□ Enabled
Type into the scrolled text control:


PyDev Package Explorer
Outline
Ch11_Code
1
Ch11_Code
__init__.py
GUI_NOT_Spaghetti.py
GUI_Spaghetti.py
Python 3.6.0 (C:\Python36\python.exe)
Writable
Insert

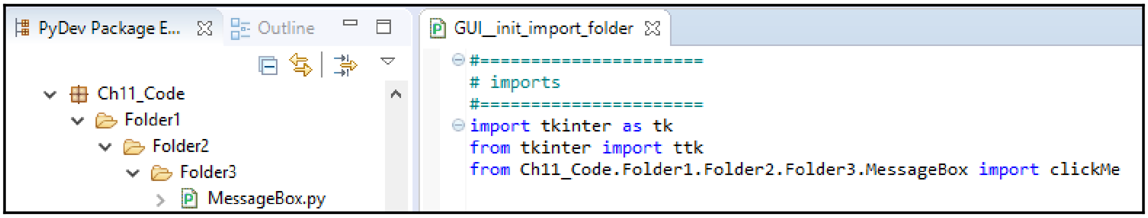


Ch11_Code

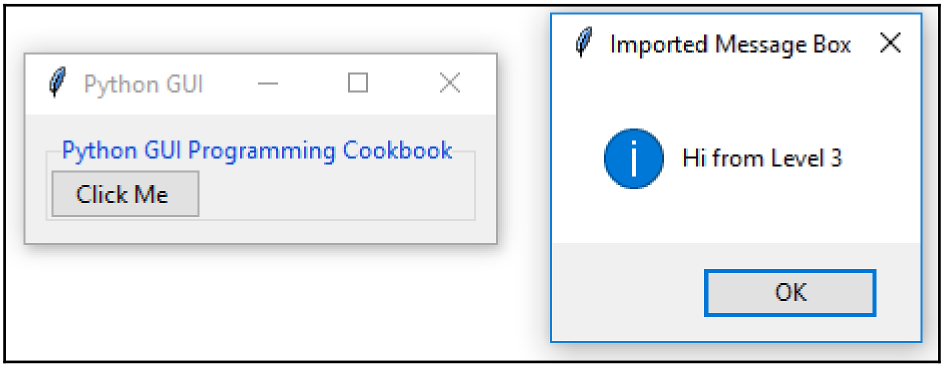
Name	Size
__init__.py	0 KB
GUI_NOT_Spaghetti.py	2 KB
GUI_Spaghetti.py	2 KB



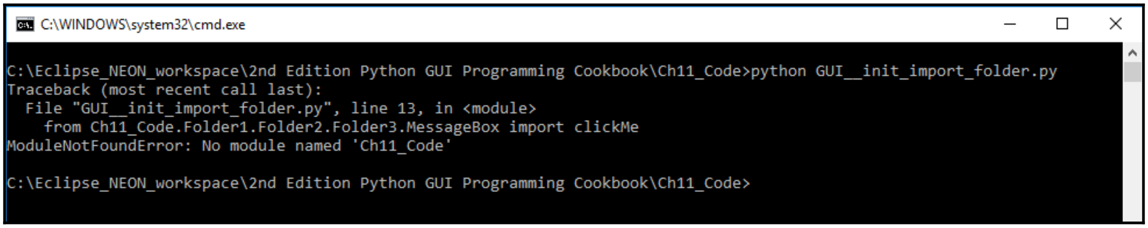
Hi from same Level.



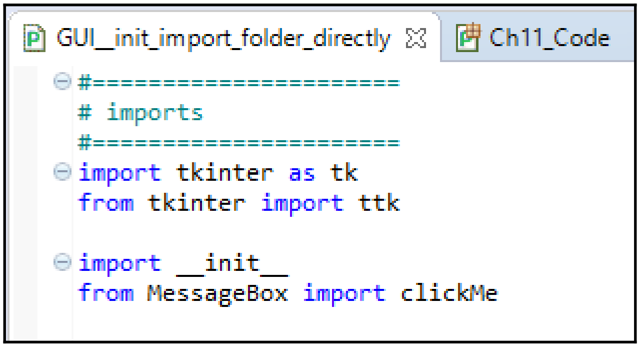
```
#======================
# imports
#======================
import tkinter as tk
from tkinter import ttk
from Ch11_Code.Folder1.Folder2.Folder3.MessageBox import clickMe
```



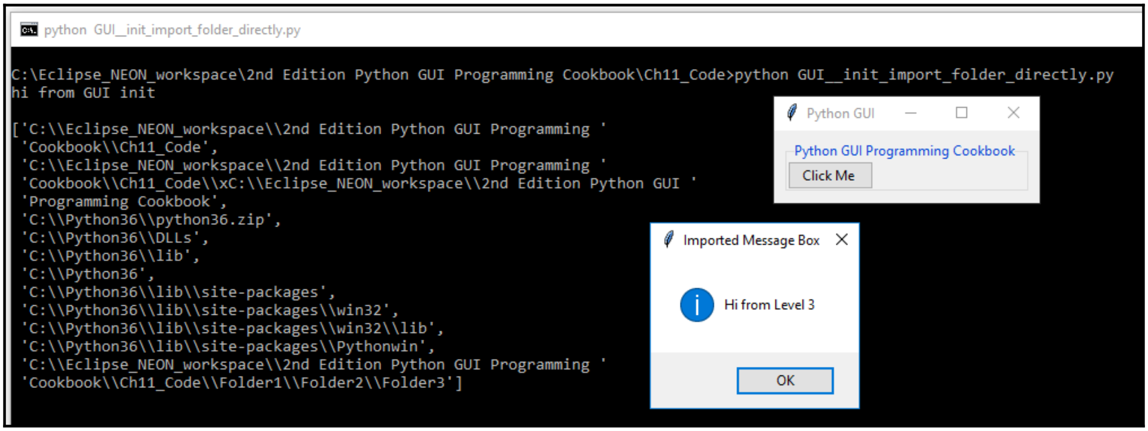
Hi from Level 3



```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch11_Code>python GUI__init_import_folder.py
Traceback (most recent call last):
  File "GUI__init_import_folder.py", line 13, in <module>
    from Ch11_Code.Folder1.Folder2.Folder3.MessageBox import clickMe
ModuleNotFoundError: No module named 'Ch11_Code'

C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch11_Code>
```

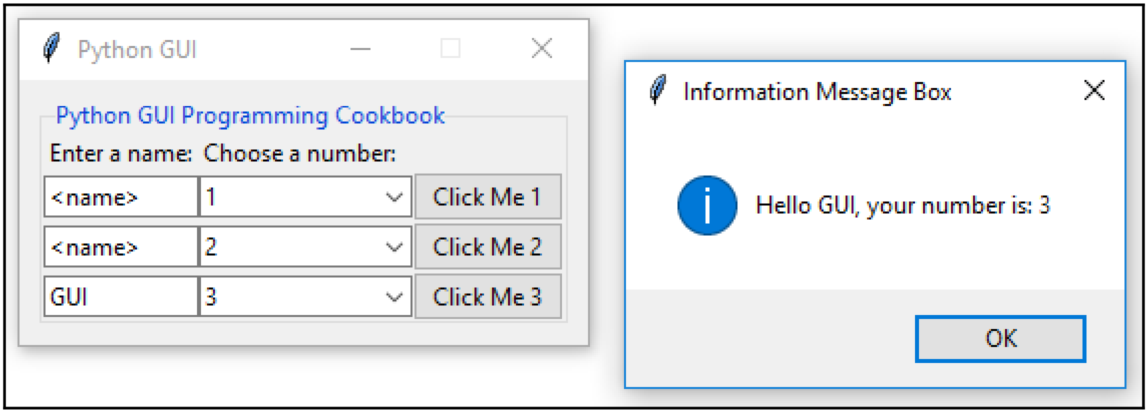


```
#======================
# imports
#======================
import tkinter as tk
from tkinter import ttk

import __init__
from MessageBox import clickMe
```

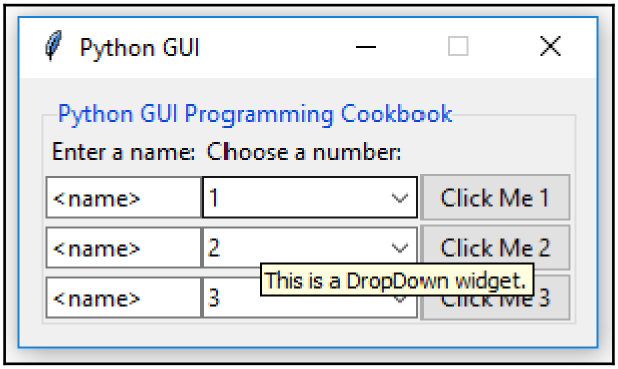

python GUI__init_import_folder_directly.py
Python GUI
Python GUI Programming Cookbook
Click Me
Imported Message Box
OK

```
C:\Eclipse_NEON_workspace\2nd Edition Python GUI Programming Cookbook\Ch11_Code>python GUI__init_import_folder_directly.py
hi from GUI init

['C:\\Eclipse_NEON_workspace\\2nd Edition Python GUI Programming '
 'Cookbook\\Ch11_Code',
 'C:\\Eclipse_NEON_workspace\\2nd Edition Python GUI Programming '
 'Cookbook\\Ch11_Code\\xC:\\Eclipse_NEON_workspace\\2nd Edition Python GUI '
 'Programming Cookbook',
 'C:\\Python36\\python36.zip',
 'C:\\Python36\\DLLs',
 'C:\\Python36\\lib',
 'C:\\Python36',
 'C:\\Python36\\lib\\site-packages',
 'C:\\Python36\\lib\\site-packages\\win32',
 'C:\\Python36\\lib\\site-packages\\win32\\lib',
 'C:\\Python36\\lib\\site-packages\\Pythonwin',
 'C:\\Eclipse_NEON_workspace\\2nd Edition Python GUI Programming '
 'Cookbook\\Ch11_Code\\Folder1\\Folder2\\Folder3']
```

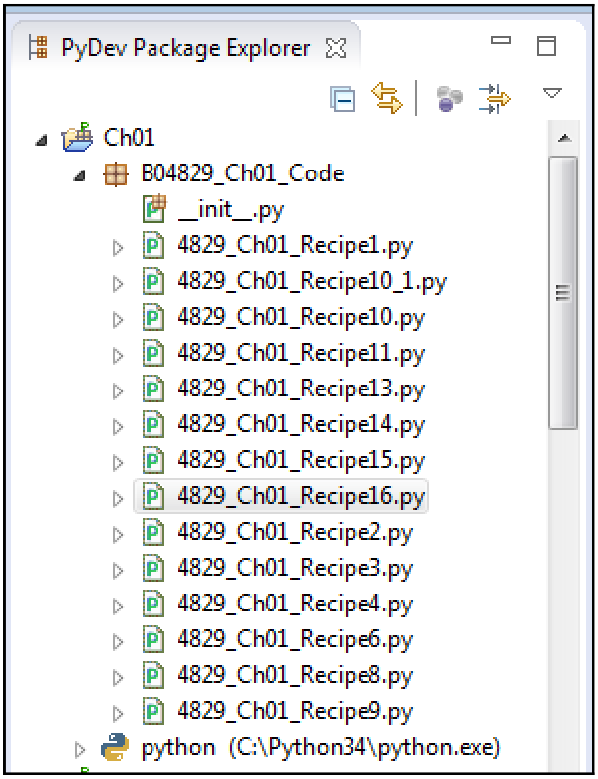
Hi from Level 3



Hello GUI, your number is: 3

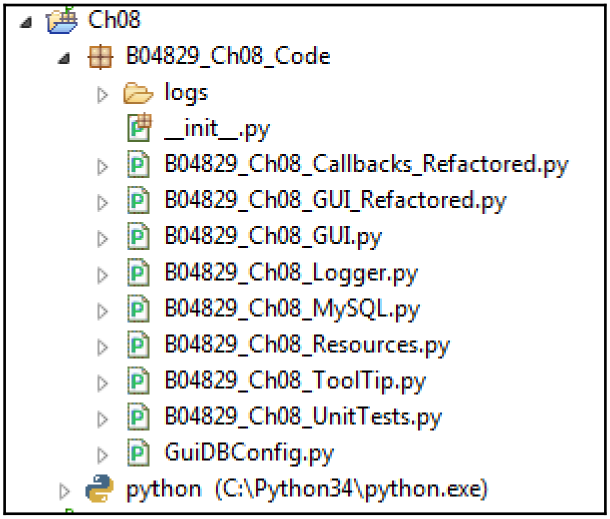


This is a DropDown widget.

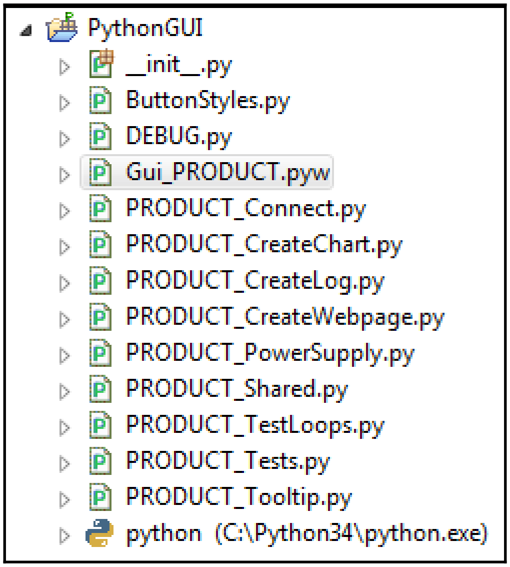


PyDev Package Explorer

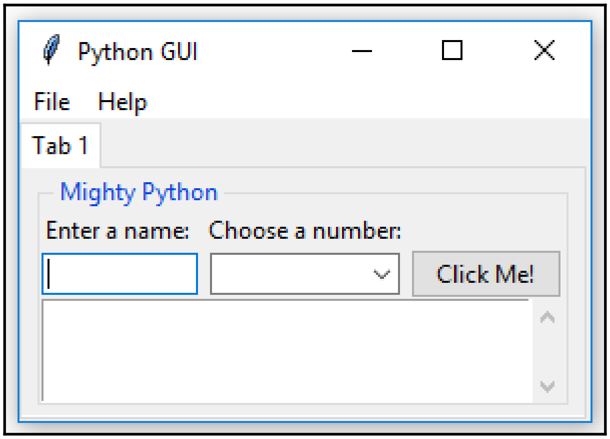
```
Ch01
    B04829_Ch01_Code
        __init__.py
        4829_Ch01_Recipe1.py
        4829_Ch01_Recipe10_1.py
        4829_Ch01_Recipe10.py
        4829_Ch01_Recipe11.py
        4829_Ch01_Recipe13.py
        4829_Ch01_Recipe14.py
        4829_Ch01_Recipe15.py
        4829_Ch01_Recipe16.py
        4829_Ch01_Recipe2.py
        4829_Ch01_Recipe3.py
        4829_Ch01_Recipe4.py
        4829_Ch01_Recipe6.py
        4829_Ch01_Recipe8.py
        4829_Ch01_Recipe9.py
    python  (C:\Python34\python.exe)
```

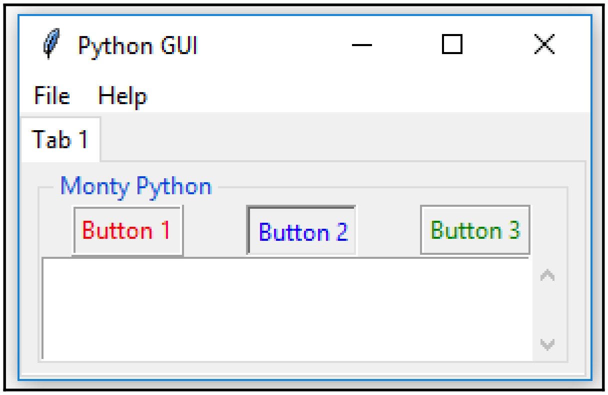


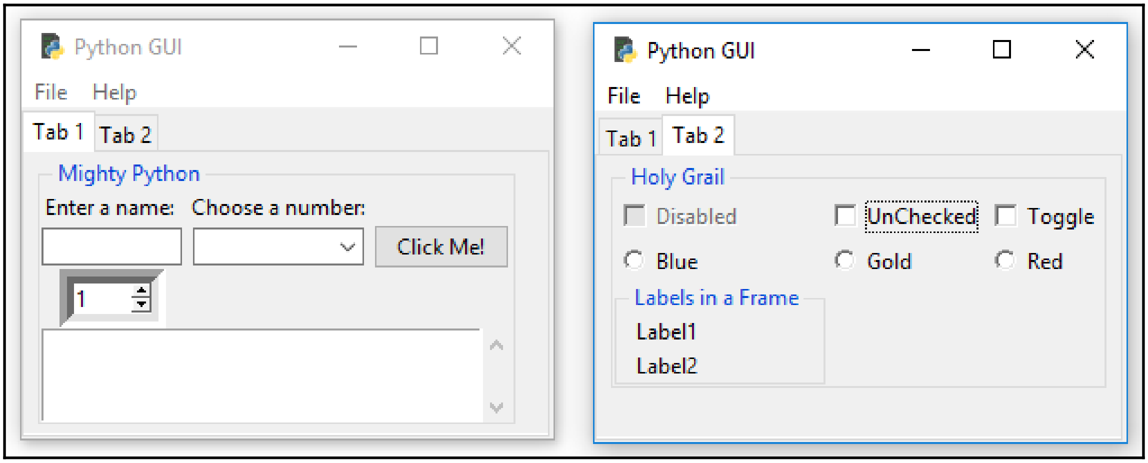
```
Ch08
    B04829_Ch08_Code
        logs
        __init__.py
        B04829_Ch08_Callbacks_Refactored.py
        B04829_Ch08_GUI_Refactored.py
        B04829_Ch08_GUI.py
        B04829_Ch08_Logger.py
        B04829_Ch08_MySQL.py
        B04829_Ch08_Resources.py
        B04829_Ch08_ToolTip.py
        B04829_Ch08_UnitTests.py
        GuiDBConfig.py
    python  (C:\Python34\python.exe)
```

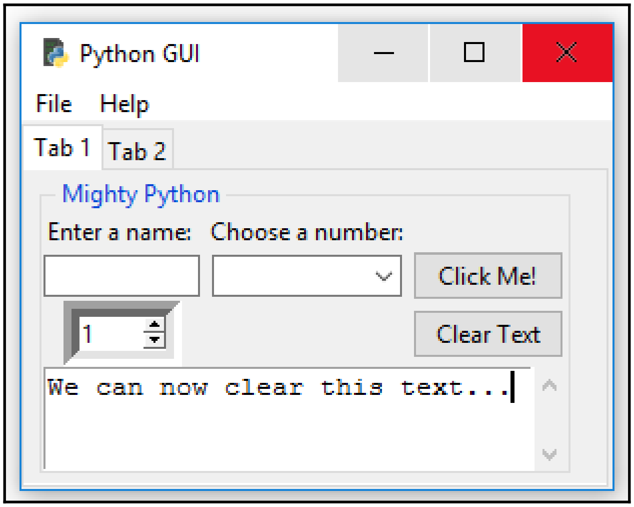


PythonGUI
- __init__.py
- ButtonStyles.py
- DEBUG.py
- Gui_PRODUCT.pyw
- PRODUCT_Connect.py
- PRODUCT_CreateChart.py
- PRODUCT_CreateLog.py
- PRODUCT_CreateWebpage.py
- PRODUCT_PowerSupply.py
- PRODUCT_Shared.py
- PRODUCT_TestLoops.py
- PRODUCT_Tests.py
- PRODUCT_Tooltip.py
- python (C:\Python34\python.exe)

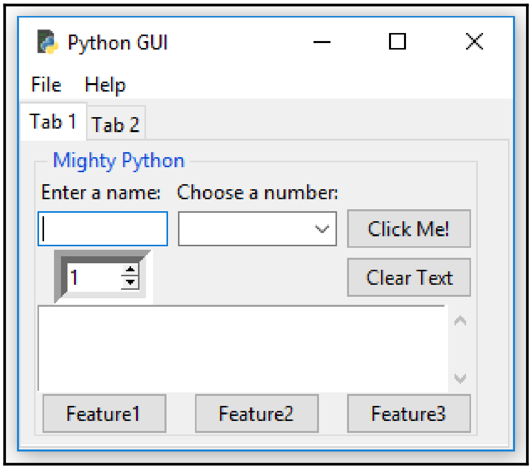


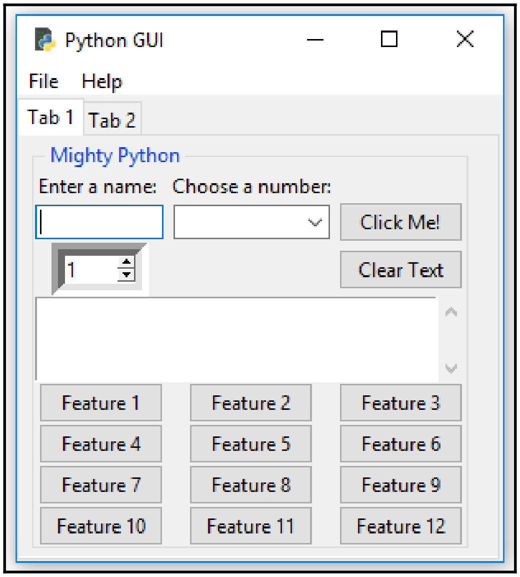

Python GUI
File
Help
Tab 1
Monty Python
Button 1
Button 2
Button 3

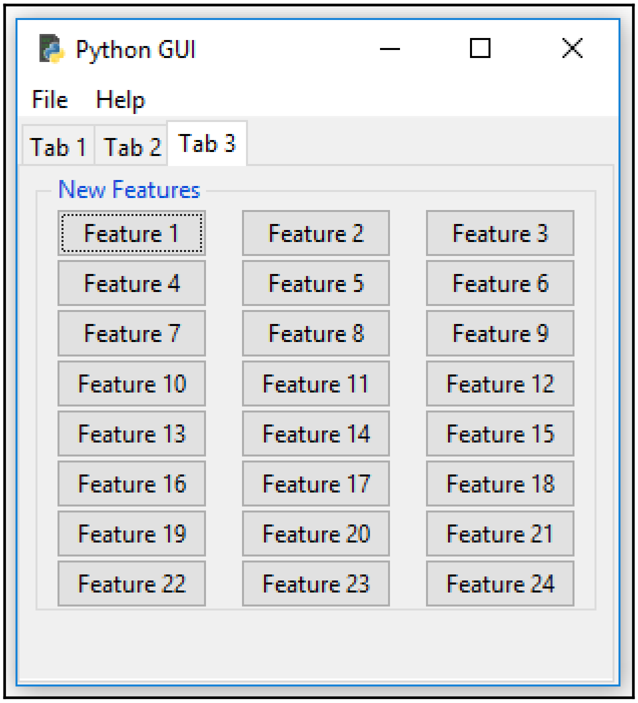

Python GUI
File
Help
Tab 1
Tab 2
Mighty Python
Enter a name:
Choose a number:
Click Me!
1
Python GUI
File
Help
Tab 1
Tab 2
Holy Grail
□ Disabled
□ UnChecked
□ Toggle
○ Blue
○ Gold
○ Red
Labels in a Frame
Label1
Label2

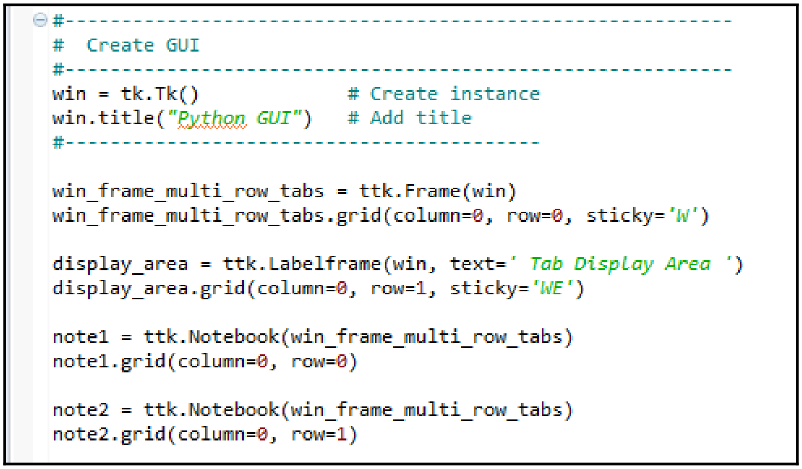


```
We can now clear this text...
```

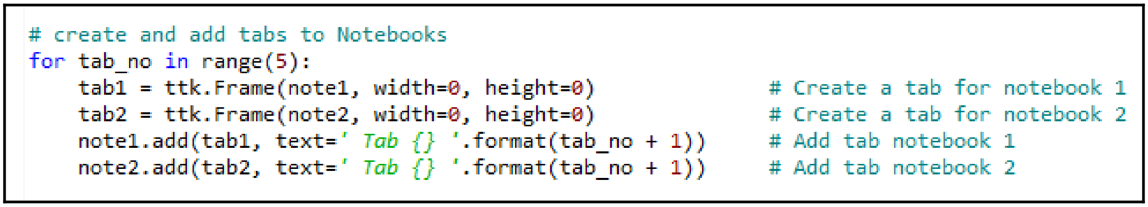



Python GUI
File
Help
Tab 1
Tab 2
Mighty Python
Enter a name:
Choose a number:
Click Me!
1
Clear Text
Feature 1
Feature 2
Feature 3
Feature 4
Feature 5
Feature 6
Feature 7
Feature 8
Feature 9
Feature 10
Feature 11
Feature 12

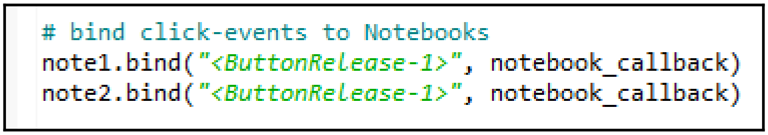

Python GUI
File
Help
Tab 1
Tab 2
Tab 3
New Features
Feature 1
Feature 2
Feature 3
Feature 4
Feature 5
Feature 6
Feature 7
Feature 8
Feature 9
Feature 10
Feature 11
Feature 12
Feature 13
Feature 14
Feature 15
Feature 16
Feature 17
Feature 18
Feature 19
Feature 20
Feature 21
Feature 22
Feature 23
Feature 24



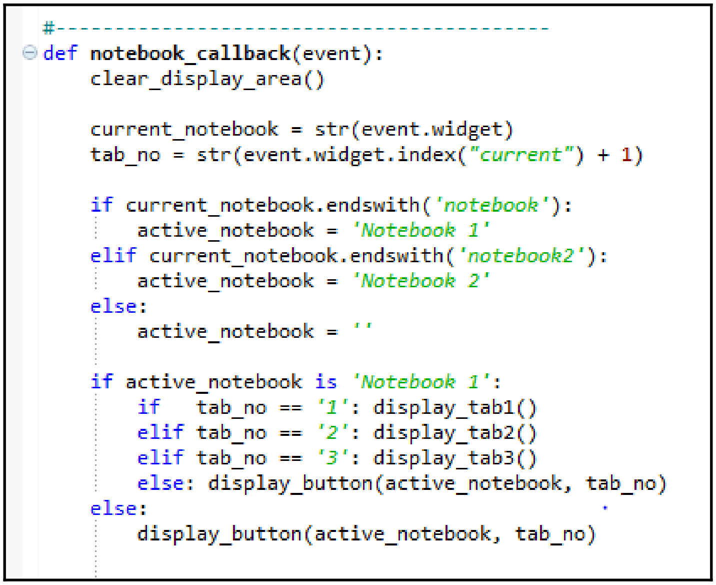
```python
#-----------------------------------------------------------
#  Create GUI
#-----------------------------------------------------------
win = tk.Tk()              # Create instance
win.title("Python GUI")    # Add title
#------------------------------------------

win_frame_multi_row_tabs = ttk.Frame(win)
win_frame_multi_row_tabs.grid(column=0, row=0, sticky='W')

display_area = ttk.Labelframe(win, text=' Tab Display Area ')
display_area.grid(column=0, row=1, sticky='WE')

note1 = ttk.Notebook(win_frame_multi_row_tabs)
note1.grid(column=0, row=0)

note2 = ttk.Notebook(win_frame_multi_row_tabs)
note2.grid(column=0, row=1)
```



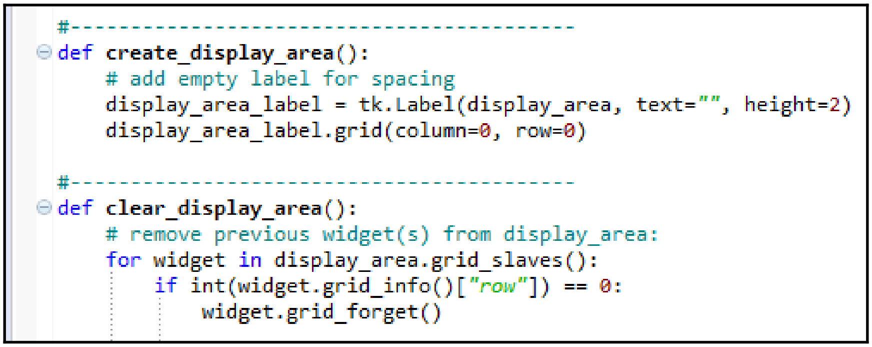
```python
# create and add tabs to Notebooks
for tab_no in range(5):
    tab1 = ttk.Frame(note1, width=0, height=0)             # Create a tab for notebook 1
    tab2 = ttk.Frame(note2, width=0, height=0)             # Create a tab for notebook 2
    note1.add(tab1, text=' Tab {} '.format(tab_no + 1))    # Add tab notebook 1
    note2.add(tab2, text=' Tab {} '.format(tab_no + 1))    # Add tab notebook 2
```



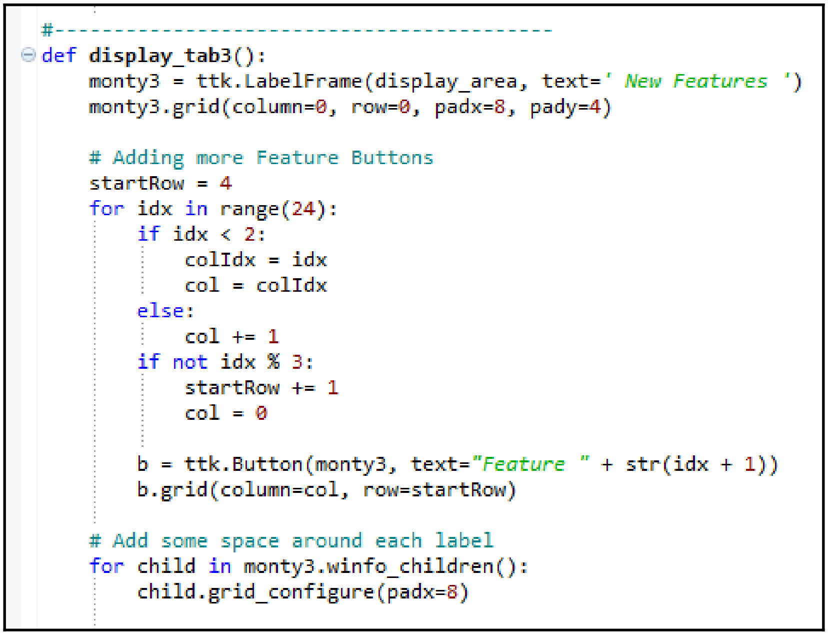
```python
# bind click-events to Notebooks
note1.bind("<ButtonRelease-1>", notebook_callback)
note2.bind("<ButtonRelease-1>", notebook_callback)
```



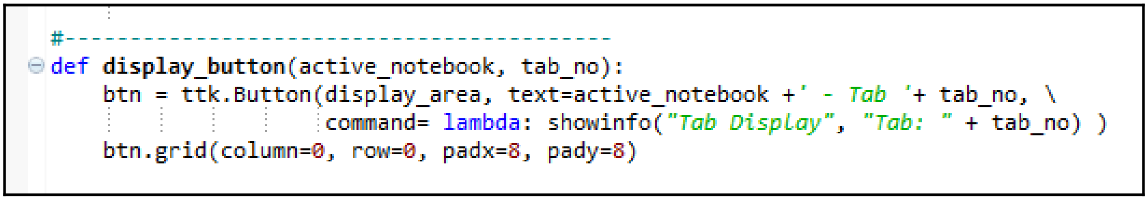
```
#-------------------------------------------
def notebook_callback(event):
    clear_display_area()

    current_notebook = str(event.widget)
    tab_no = str(event.widget.index("current") + 1)

    if current_notebook.endswith('notebook'):
        active_notebook = 'Notebook 1'
    elif current_notebook.endswith('notebook2'):
        active_notebook = 'Notebook 2'
    else:
        active_notebook = ''

    if active_notebook is 'Notebook 1':
        if   tab_no == '1': display_tab1()
        elif tab_no == '2': display_tab2()
        elif tab_no == '3': display_tab3()
        else: display_button(active_notebook, tab_no)
    else:
        display_button(active_notebook, tab_no)
```



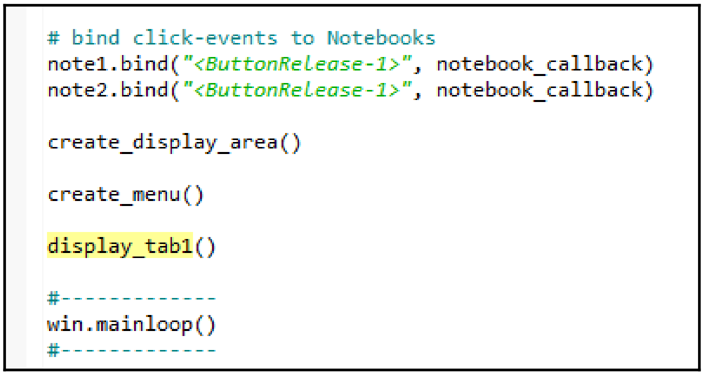
```
#-------------------------------------------
def create_display_area():
    # add empty label for spacing
    display_area_label = tk.Label(display_area, text="", height=2)
    display_area_label.grid(column=0, row=0)

#-------------------------------------------
def clear_display_area():
    # remove previous widget(s) from display_area:
    for widget in display_area.grid_slaves():
        if int(widget.grid_info()["row"]) == 0:
            widget.grid_forget()
```



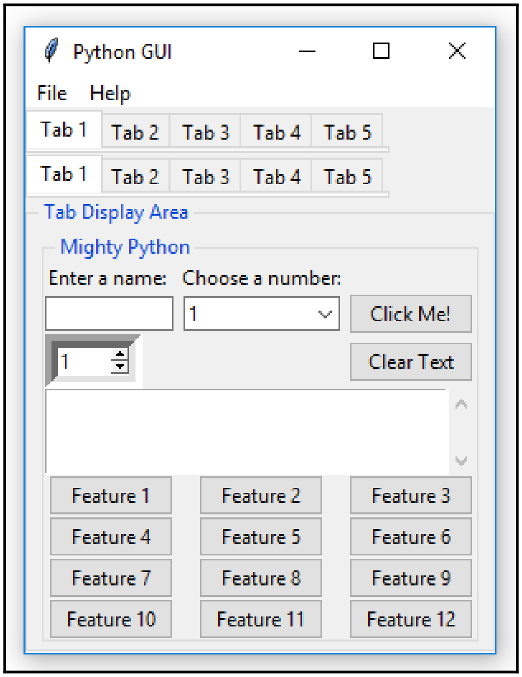
```
#-------------------------------------------
def display_tab3():
    monty3 = ttk.LabelFrame(display_area, text=' New Features ')
    monty3.grid(column=0, row=0, padx=8, pady=4)

    # Adding more Feature Buttons
    startRow = 4
    for idx in range(24):
        if idx < 2:
            colIdx = idx
            col = colIdx
        else:
            col += 1
        if not idx % 3:
            startRow += 1
            col = 0

        b = ttk.Button(monty3, text="Feature " + str(idx + 1))
        b.grid(column=col, row=startRow)

    # Add some space around each label
    for child in monty3.winfo_children():
        child.grid_configure(padx=8)
```

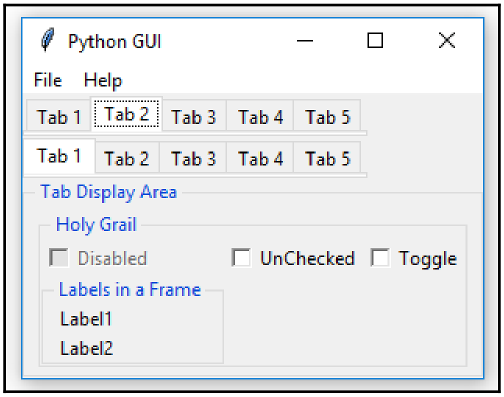


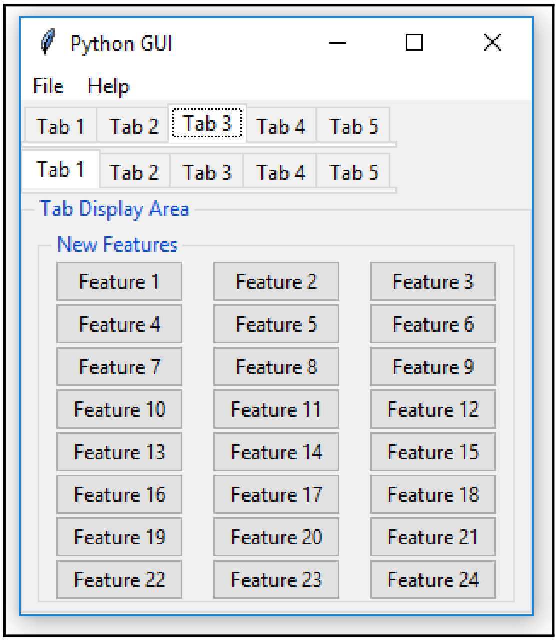
```
#-----------------------------------------
def display_button(active_notebook, tab_no):
    btn = ttk.Button(display_area, text=active_notebook +' - Tab '+ tab_no, \
                     command= lambda: showinfo("Tab Display", "Tab: " + tab_no) )
    btn.grid(column=0, row=0, padx=8, pady=8)
```

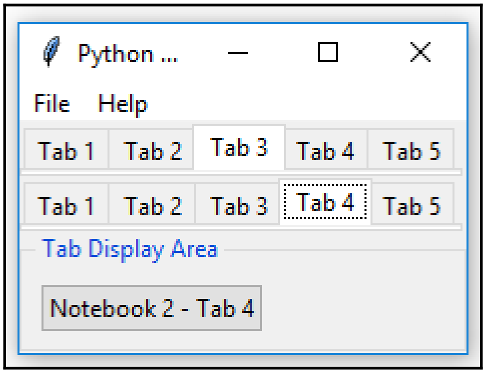


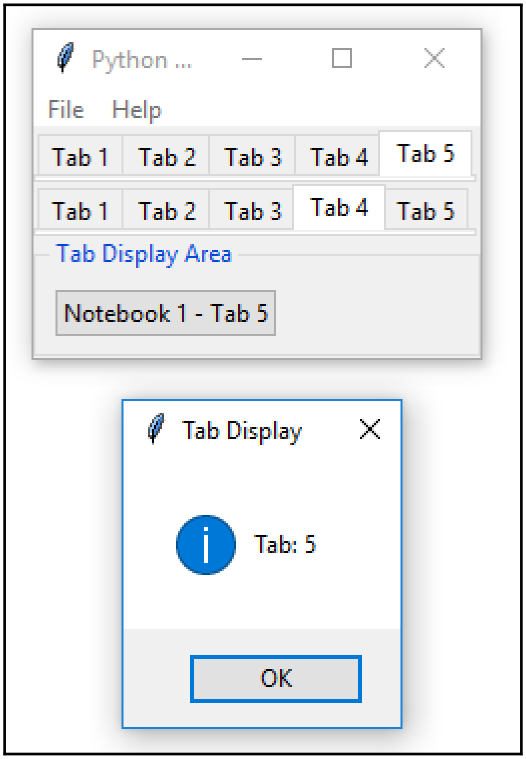
```
# bind click-events to Notebooks
note1.bind("<ButtonRelease-1>", notebook_callback)
note2.bind("<ButtonRelease-1>", notebook_callback)

create_display_area()

create_menu()

display_tab1()

#-------------
win.mainloop()
#-------------
```


Python GUI
File
Help
Tab 1
Tab 2
Tab 3
Tab 4
Tab 5
Tab 1
Tab 2
Tab 3
Tab 4
Tab 5
Tab Display Area
Mighty Python
Enter a name:
Choose a number:
1
Click Me!
1
Clear Text
Feature 1
Feature 2
Feature 3
Feature 4
Feature 5
Feature 6
Feature 7
Feature 8
Feature 9
Feature 10
Feature 11
Feature 12


Python GUI
File
Help
Tab 1
Tab 2
Tab 3
Tab 4
Tab 5
Tab 1
Tab 2
Tab 3
Tab 4
Tab 5
Tab Display Area
Holy Grail
Disabled
UnChecked
Toggle
Labels in a Frame
Label1
Label2


Python GUI
File
Help
Tab 1
Tab 2
Tab 3
Tab 4
Tab 5
Tab 1
Tab 2
Tab 3
Tab 4
Tab 5
Tab Display Area
New Features
Feature 1
Feature 2
Feature 3
Feature 4
Feature 5
Feature 6
Feature 7
Feature 8
Feature 9
Feature 10
Feature 11
Feature 12
Feature 13
Feature 14
Feature 15
Feature 16
Feature 17
Feature 18
Feature 19
Feature 20
Feature 21
Feature 22
Feature 23
Feature 24


Python ...
File
Help
Tab 1
Tab 2
Tab 3
Tab 4
Tab 5
Tab 1
Tab 2
Tab 3
Tab 4
Tab 5
Tab Display Area
Notebook 2 - Tab 4


Python ...
File
Help
Tab 1
Tab 2
Tab 3
Tab 4
Tab 5
Tab 1
Tab 2
Tab 3
Tab 4
Tab 5
Tab Display Area
Notebook 1 - Tab 5
Tab Display
OK

Tab: 5