15.8. Tkinter Standard Dialog Boxes

There are many common programming tasks that can be performed using pre-defined GUI dialog boxes. The following discussion describes these dialog boxes and provides some simple examples. You can refer to the Python documentation for additional optional parameters.

<





15.8.1. Messages

A messagebox can display information to a user. There are three variations on these dialog boxes based on the type of message you want to display. The functions' first parameter gives a name for the dialog box which is displayed in the window's header. The second parameter is the text of the message. The functions return a string which is typically ignored.

```
from tkinter import messagebox

messagebox.showinfo("Information","Informative message")
messagebox.showerror("Error", "Error message")
messagebox.showwarning("Warning","Warning message")
```

15.8.2. Yes/No Questions

The tkinter messagebox object also allows you to ask a user simple yes/no type questions and varies the button names based on the type of question. These functions are:

```
from tkinter import messagebox

answer = messagebox.askokcancel("Question", "Do you want to open this file?")
answer = messagebox.askretrycancel("Question", "Do you want to try that again?")
answer = messagebox.askyesno("Question", "Do you like Python?")
answer = messagebox.askyesnocancel("Question", "Continue playing?")
```

The return value is a Boolean, True or False, answer to the question. If "cancel" is an option and the user selects the "cancel" button, None is returned.

15.8.3. Single Value Data Entry

If you want to ask the user for a single data value, either a string, integer, or floating point value, you can use a simpledialog object. A user can enter the requested value and hit "OK", which will return the entered value. If the user hits "Cancel," then None is returned.

```
import tkinter as tk
from tkinter import simpledialog
application window = tk.Tk()
answer = simpledialog.askstring("Input", "What is your first name?",
                                parent=application window)
if answer is not None:
    print("Your first name is ", answer)
else:
    print("You don't have a first name?")
answer = simpledialog.askinteger("Input", "What is your age?",
                                 parent=application window,
                                 minvalue=0, maxvalue=100)
if answer is not None:
    print("Your age is ", answer)
else:
    print("You don't have an age?")
answer = simpledialog.askfloat("Input", "What is your salary?",
                               parent=application_window,
                               minvalue=0.0, maxvalue=100000.0)
if answer is not None:
    print("Your salary is ", answer)
else:
    print("You don't have a salary?")
```

15.8.4. File Chooser

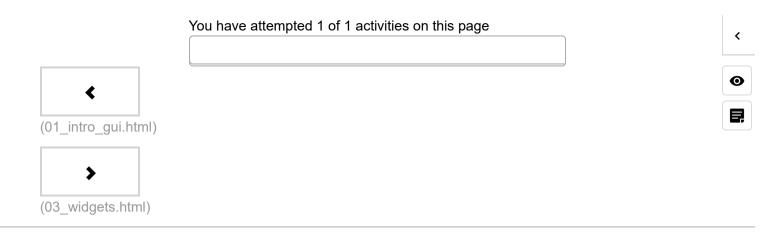
A common task is to select the names of folders and files on a storage device. This can be accomplished using a filedialog object. Note that these commands do not save or load a file. They simply allow a user to select a file. Once you have the file name, you can open, process, and close the file using appropriate Python code. These dialog boxes always return you a "fully qualified file name" that includes a full path to the file. Also note that if a user is allowed to select multiple files, the return value is a tuple that contains all of the selected files. If a user cancels the dialog box, the returned value is an empty string.

o gui.html)

```
import tkinter as tk
from tkinter import filedialog
import os
application window = tk.Tk()
# Build a list of tuples for each file type the file dialog should display
my_filetypes = [('all files', '.*'), ('text files', '.txt')]
# Ask the user to select a folder.
answer = filedialog.askdirectory(parent=application_window,
                                 initialdir=os.getcwd(),
                                 title="Please select a folder:")
# Ask the user to select a single file name.
answer = filedialog.askopenfilename(parent=application window,
                                    initialdir=os.getcwd(),
                                    title="Please select a file:",
                                    filetypes=my filetypes)
# Ask the user to select a one or more file names.
answer = filedialog.askopenfilenames(parent=application_window,
                                     initialdir=os.getcwd(),
                                     title="Please select one or more files:",
                                     filetypes=my_filetypes)
# Ask the user to select a single file name for saving.
answer = filedialog.asksaveasfilename(parent=application window,
                                      initialdir=os.getcwd(),
                                      title="Please select a file name for saving:",
                                      filetypes=my_filetypes)
```

15.8.5. Color Chooser

Tkinter includes a nice dialog box for choosing colors. You provide it with a parent window and an initial color, and it returns a color in two different specifications: 1) a RGB value as a tuple, such as (255, 0, 0) which represents red, and 2) a hexadecimal string used in web pages, such as "#FF0000" which also represents red. If the user cancels the operation, the return values are None and None.



user not logged in

o_gui.html)