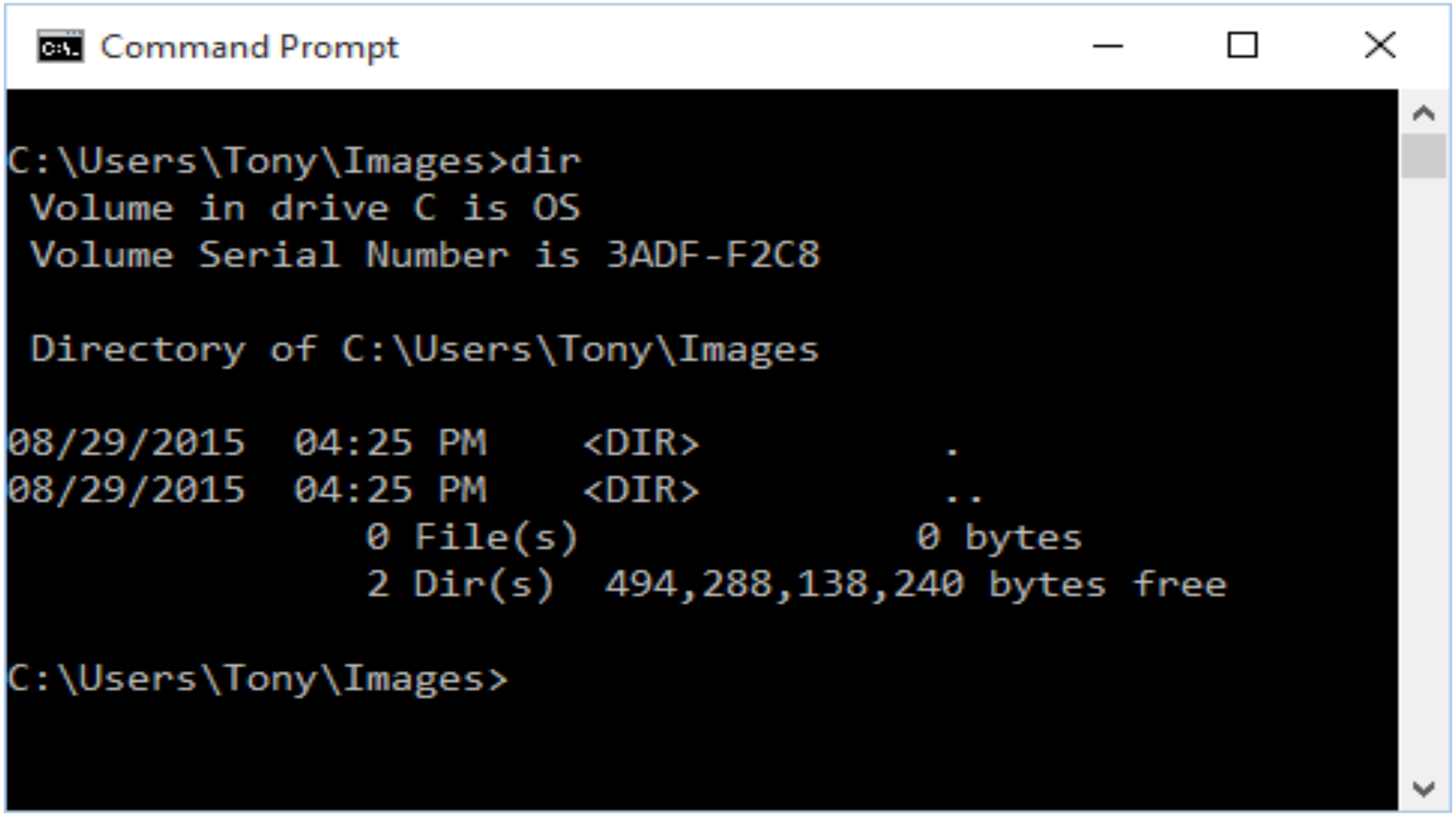
Intro to Computer Science
CS-UH 1001, Spring 2022

Lecture 23 – GUIs using tkinter

GUI using tkinter

- Using the `tkinter` Module
- Display Text with `Label` Widgets
- Organizing Widgets with Frames
- `Button` Widgets and Info Dialog Boxes
- Getting Input with the `Entry` Widget
- Using Labels as Output Fields

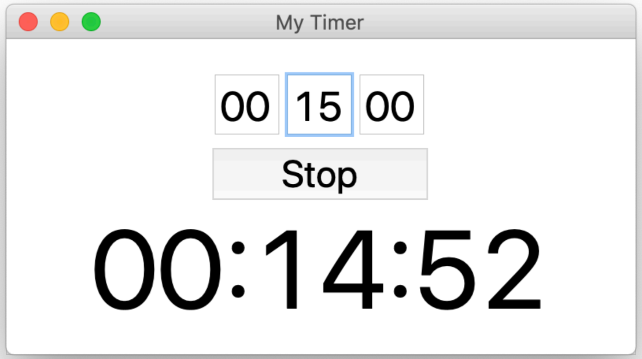
Command Line Interface

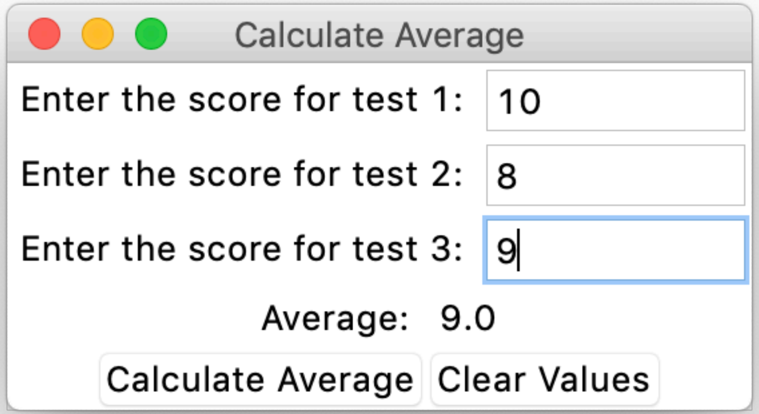


```
C:\Users\Tony\Images>dir
 Volume in drive C is OS
 Volume Serial Number is 3ADF-F2C8

 Directory of C:\Users\Tony\Images

08/29/2015  04:25 PM    <DIR>          .
08/29/2015  04:25 PM    <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)  494,288,138,240 bytes free

C:\Users\Tony\Images>
```

Graphical User Interface

- Much nicer, ehhh!
- Much more work too!





Enter the score for test 1: 10
Enter the score for test 2: 8
Enter the score for test 3: 9
Average: 9.0

GUI Programs are Event-Driven

- **In text-based environments, programs determine the order in which things happen**
  - The user can only enter data in the order requested by the program
- **GUI environment is event-driven**
  - The user determines the order in which things happen

Using the `tkinter` Module

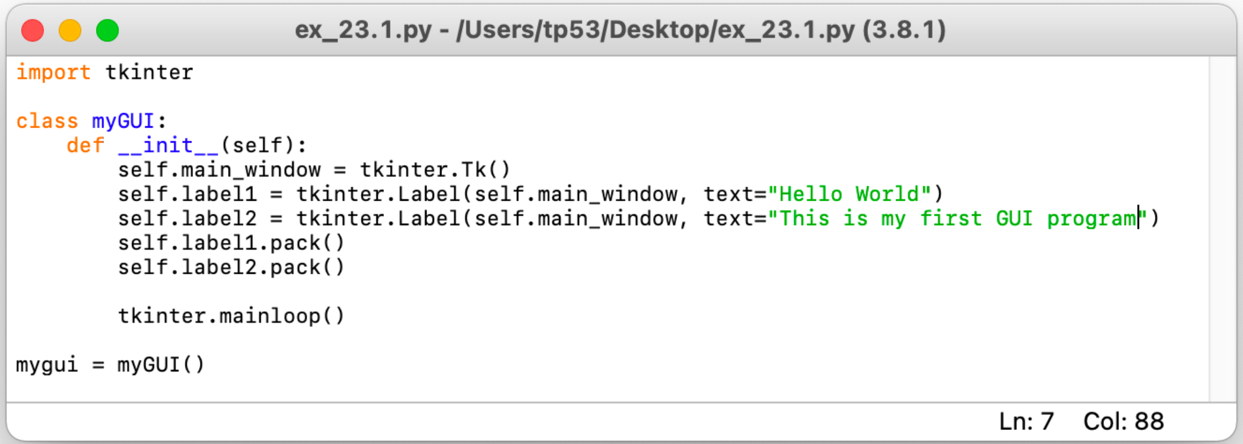
- `tkinter` module: allows you to create simple GUI programs
- Widget: graphical element that the user can interact with or view

- Check if it is installed:

import tkinter

tkinter on Windows

- Download and install Python IDLE from https://www.python.org/downloads/
- Open up IDLE
- File -> New File
- When you are done writing the code, hit F5, or go to Run -> Run Module

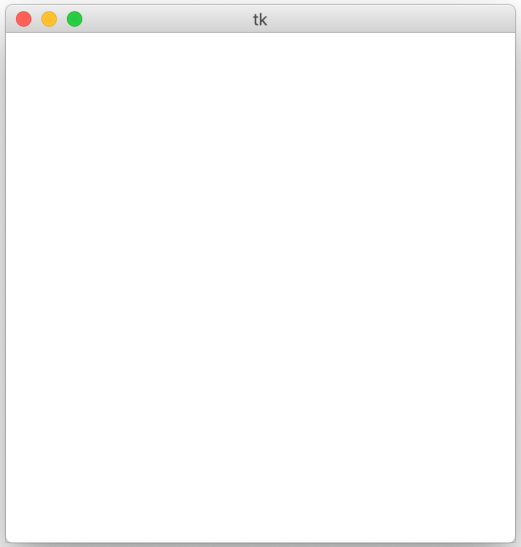

ex_23.1.py - /Users/tp53/Desktop/ex_23.1.py (3.8.1)

```
import tkinter

class myGUI:
    def __init__(self):
        self.main_window = tkinter.Tk()
        self.label1 = tkinter.Label(self.main_window, text="Hello World")
        self.label2 = tkinter.Label(self.main_window, text="This is my first GUI program")
        self.label1.pack()
        self.label2.pack()

        tkinter.mainloop()

mygui = myGUI()
```

Ln: 7   Col: 88

Using the `tkinter` Module

- **Most programmers take an object-oriented approach when writing GUI programs:**
  - When an instance is created, the GUI appears on the screen
    - `main_window = tkinter.Tk()`
  - Enter the tkinter main loop
    - `tkinter.mainloop()`



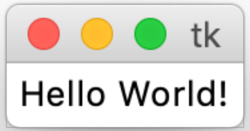
Using the `tkinter` Module

- **Methods to customize the window:**
  - `.title("My GUI")` sets the title of the GUI
  - `.geometry("400x400")` sets the size of the GUI
  - `.maxsize(width=500, height=500)` sets the max size of the GUI when resizing
  - `.minsize(width=100, height=100)` sets the min size of the GUI when resizing

Display Text with `Label` Widgets

- **`Label` widget: displays text in a window**
  - Made by creating an instance of `tkinter` module’s `Label` class

    -
      ```
      tkinter.Label(self.main_window,
                    text = ‘Hello World!’)
      ```
    - First argument references the root widget, second argument shows text that should appear in the label

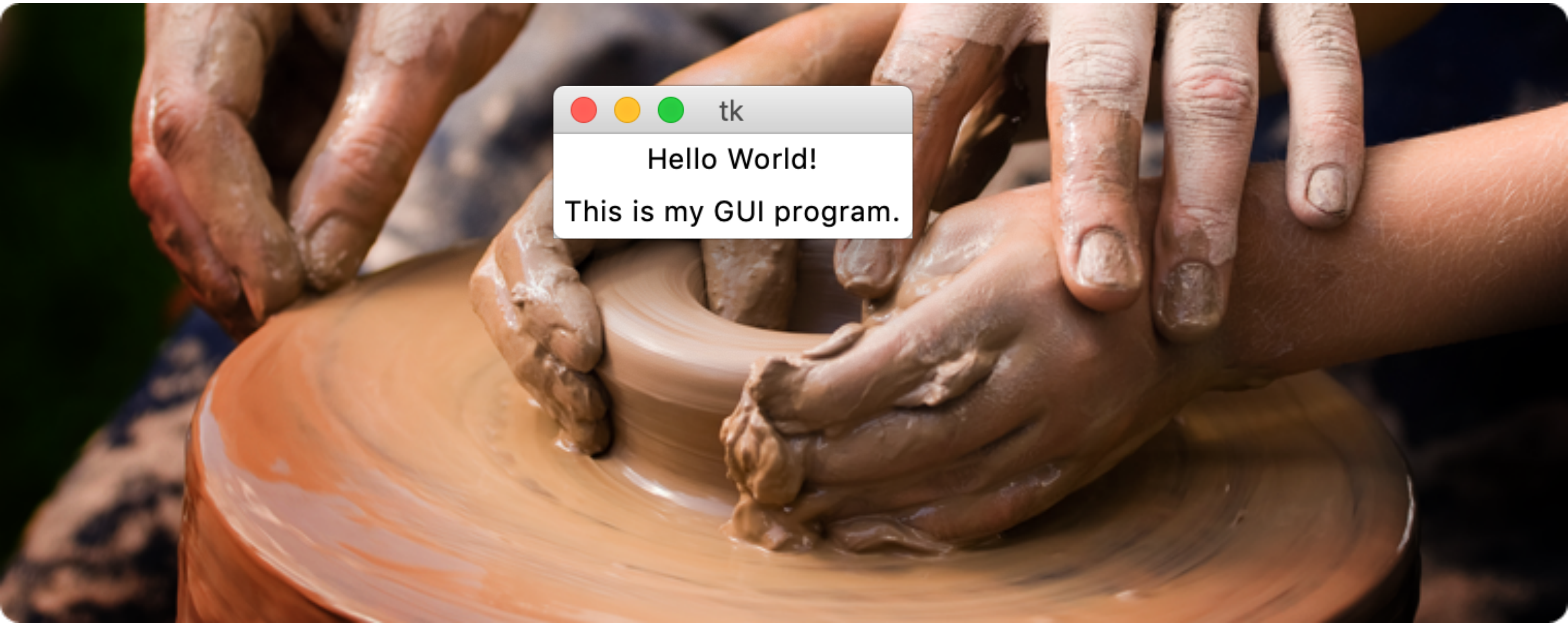


Hello World!

Display Text with `Label` Widgets

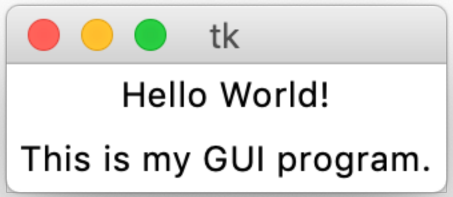
- **<u>`pack` method</u>: determines where a widget should be positioned and makes it visible when the main window is displayed**
  - Receives an argument to specify positioning
    - Positioning depends on the order in which widgets were added (packed) to the window
    - Position arguments: `side='top', side='left', side='right', side='bottom'`
  - Center the widget, if `.geometry()` is used:
    - Argument: `expand=True`

Hands on warm-up (ex_23.1.py):
Hello World



```
Hello World!
This is my GUI program.
```

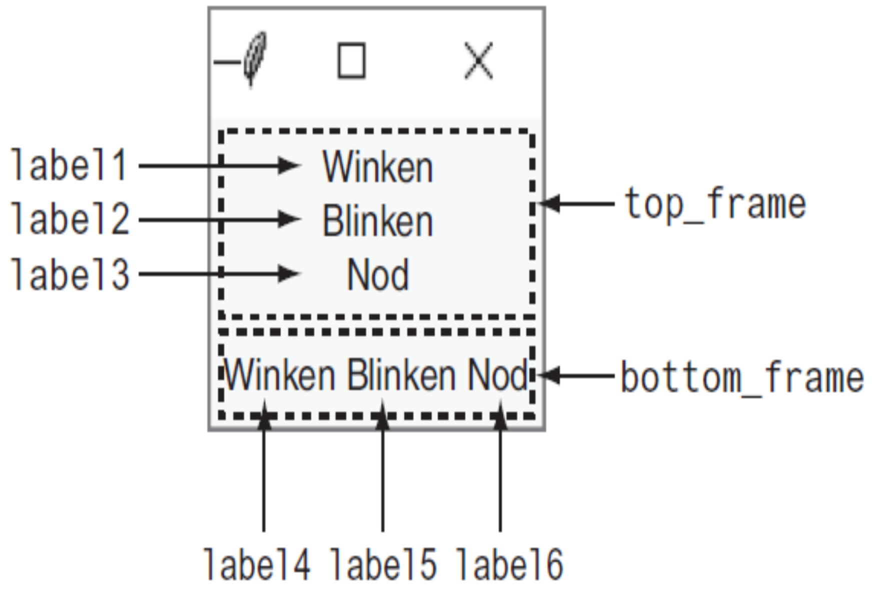
```
import tkinter
class MyGUI:
    def __init__(self):
        # Create the main window widgets
        self.main_window = tkinter.Tk()

        # Create two Label widgets
        self.label1 = tkinter.Label(self.main_window,
                                    text='Hello World!')
        self.label2 = tkinter.Label(self.main_window,
                            text='This is my GUI program.')

        # Call both Label widgets' pack method.
        self.label1.pack()
        self.label2.pack()

        # Enter the tkinter main loop.
        tkinter.mainloop()

my_gui = MyGUI()
```



```
Hello World!
This is my GUI program.
```

Organizing `Label` Widgets with Frames

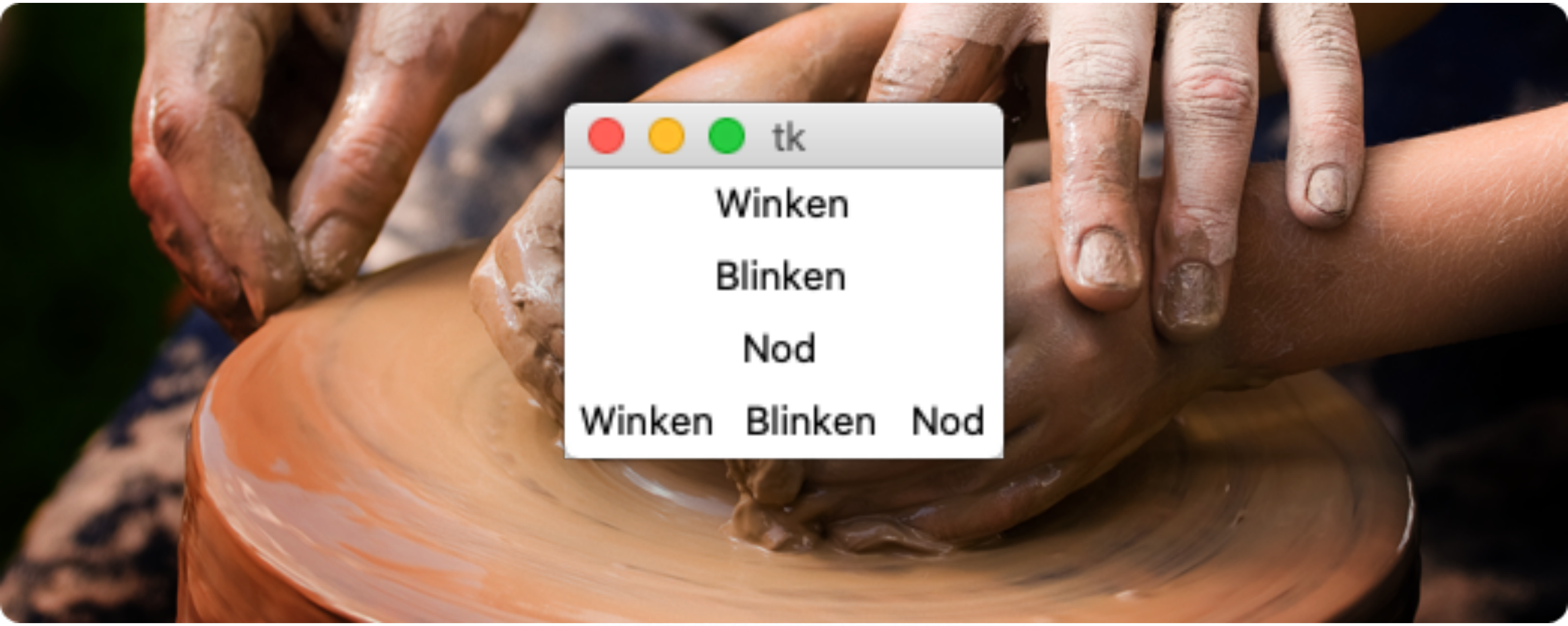
Figure 13-9 Arrangement of widgets



Organizing Widgets with Frames

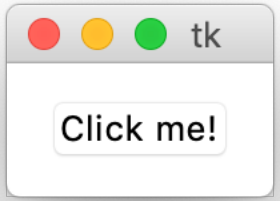
- `Frame` **widget: container that holds and organizes widgets in a window**
  - Defining a Frame:
    - `self.top_frame = tkinter.Frame(self.main_window)`
  - The contained widgets are added to the frame widget
    - `tkinter.Label(self.top_frame,text='Winken')`
    - `tkinter.Label(self.top_frame,text='Blinken')`
  - The Frame widget also has to be packed to make it visible:
    - `self.top_frame.pack()`

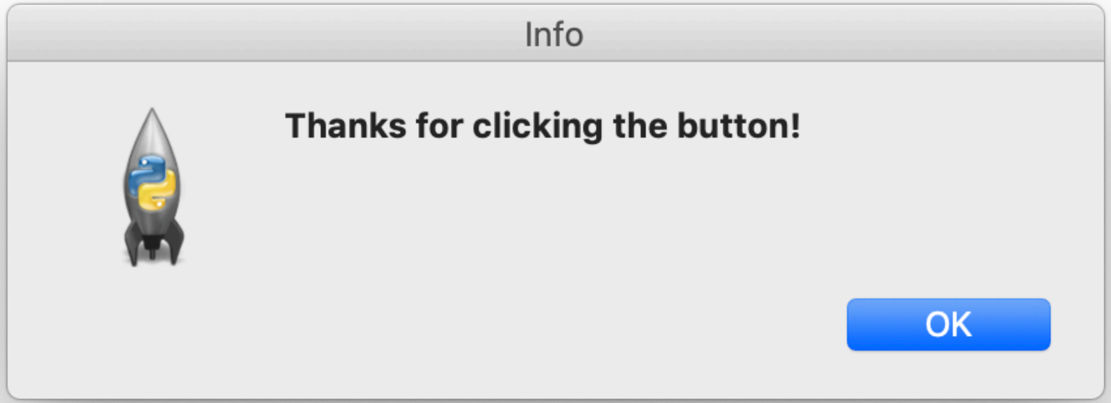
Hands on warm-up II (ex_23.2.py):
Frames



```
Winken
Blinken
Nod
Winken  Blinken  Nod
```

`Button` Widgets and Info Dialog Boxes





Thanks for clicking the button!

`Button` Widgets

- **<u>`Button` widget</u>: widget that the user can click to cause an action to take place**
  - Text to appear on the face of the button
  - A callback function
- **<u>Event handler or callback function</u>: function or method that executes when the user clicks the button**
- **Example:**
  - `self.my_button = tkinter.Button(self.main_window, text='Click Me!', command = self.do_something)`
- **Don't forget to pack the Button**

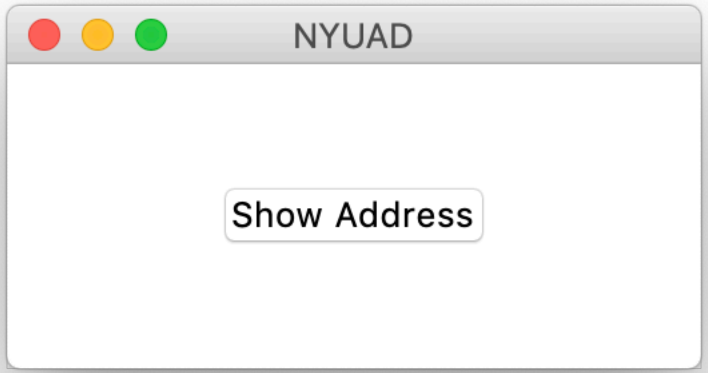
Info Dialog Boxes

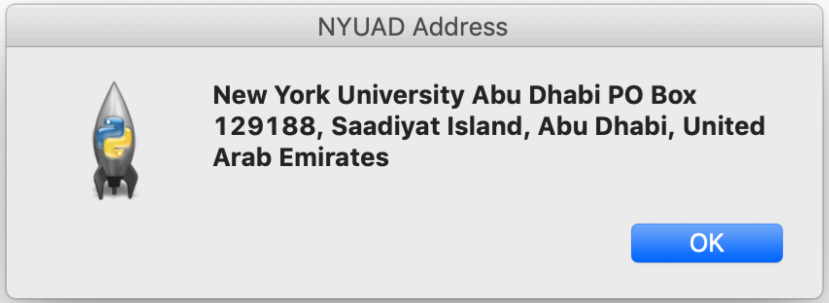
- **<u>Info dialog box</u>: a dialog box that shows information to the user**
  - Import `tkinter.messagebox` module
  - `tkinter.messagebox.showinfo(`*`title`*`,` *`message`*`)`
    - *`title`* is displayed in dialog box's title bar
    - *`message`* is an informational string displayed in the main part of the dialog box

Breakout session I (ex_23.3.py):
NYUAD address

Hands-on I: NYUAD address

- Write a GUI program that displays the NYUAD address when a button is clicked
- The information should be displayed in a Dialog Box





New York University Abu Dhabi PO Box 129188, Saadiyat Island, Abu Dhabi, United Arab Emirates

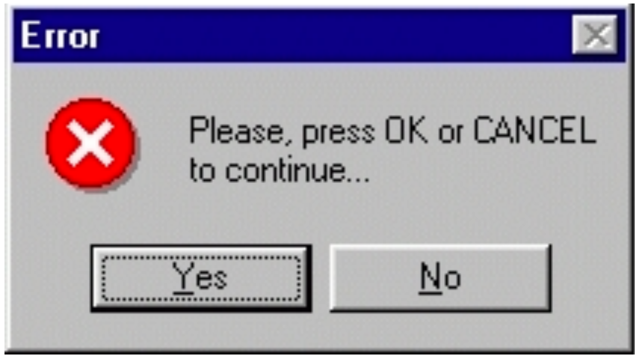
Use:

```
import tkinter.messagebox
self.my_button = tkinter.Button(self.main_window, text='Click
                                Me!', command = self.do_something)
tkinter.messagebox.showinfo(title, message)
```

Let's take a break from all these windows…

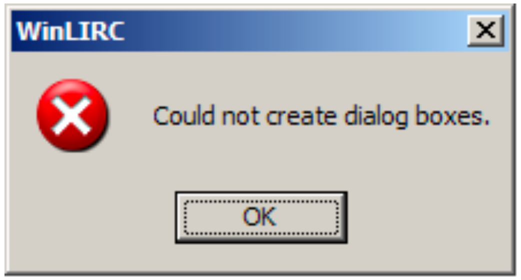
Keyboard not plugged

Windows 95 was unable to detect your keyboard. Press F1 to retry or F2 to abort.



Error

Please, press OK or CANCEL to continue...



WinLIRC

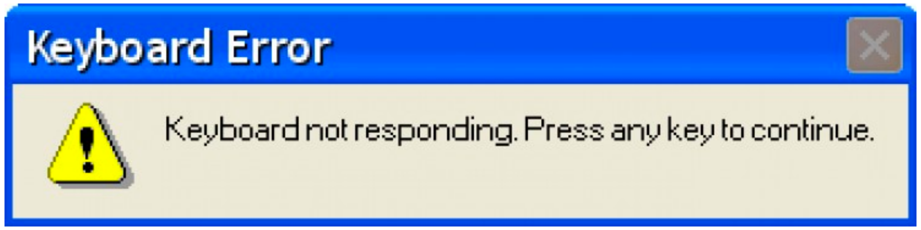
Could not create dialog boxes.

Windows XP

Task failed successfully.

Windows Internet Explorer

Stack overflow at line: 7



Keyboard Error

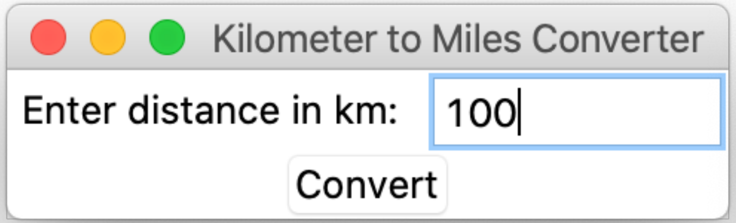
Keyboard not responding. Press any key to continue.

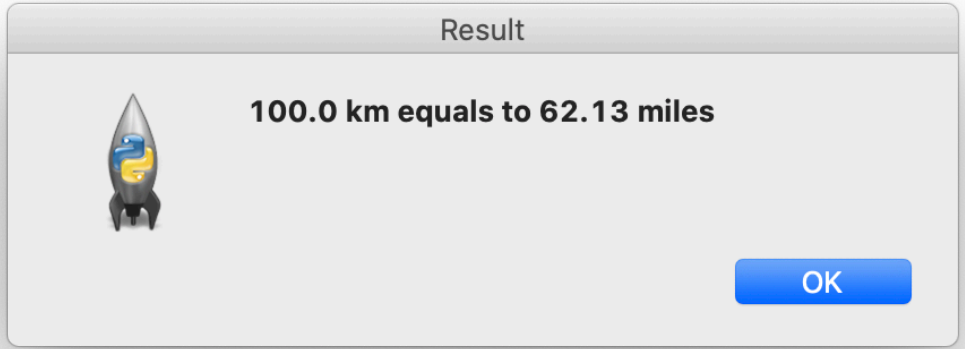
Getting Input with the `Entry` Widget

- **`Entry` widget: rectangular area that the user can type text into**
  - Used to gather input in a GUI program
    - `self.entry = tkinter.Entry(self.top_frame, width=10)`
  - `Entry` widget's `get` method: used to retrieve the data from an `Entry` widget
    - `string = self.entry.get()`

Hands II (ex_23.4.py):
Kilometers to Miles converter

Kilometer to Miles Converter





100.0 km equals to 62.13 miles

Hint: 1 km = 0.6214 miles

Use:

```
self.entry = tkinter.Entry(self.top_frame, width=10)
string = self.entry.get()
```

Using Labels as Output Fields

- **Can use `Label` widgets to dynamically display output**
  - Used to replace info dialog box
  - Create empty `Label` widget in main window
  - Write code that displays desired data in the label when a button is clicked

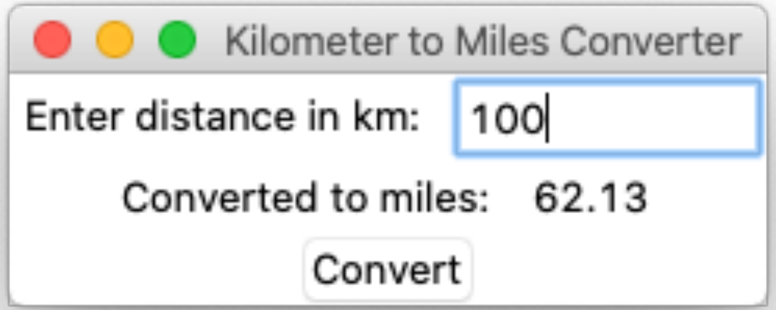
Using Labels as Output Fields

- **`StringVar` class: `tkinter` module class that can be used along with `Label` widget to display data**
  - Create `StringVar` object and then create `Label` widget and associate it with the `StringVar` object
  - Subsequently, any value stored in the `StringVar` object with automatically be displayed in the `Label` widget:
    - `self.result = tkinter.StringVar()`
    - `self.label = tkinter.Label(self.top_frame, textvariable = self.result)`
    - `self.result.set("string")`

Breakout session III (ex_23.5.py):
Miles converter using StringVar

Miles converter using StringVar

- **Modify the previous code so that the result is shown within the window**



Use:

```
self.result = tkinter.StringVar()
self.label = tkinter.Label(self.top_frame,
                   textvariable = self.result)
self.result.set(“string”)
```