Software Engineering Laboratory CS29006

Abir Das

Computer Science and Engineering Department
Indian Institute of Technology Kharagpur

Agenda

- Getting familiar with some GUI programing concepts in Python

- Assumption: You are already familiar with basics of Python e.g., conditions, loops, functions, different containers.

Sources

- Materials for these slides are taken majorly from the following websites.

  - https://anzeljg.github.io/rin2/book2/2405/docs/tkinter/index.html
  - https://python-course.eu/tkinter/

- A good youtube playlist to learn Tkinter from codemy.com

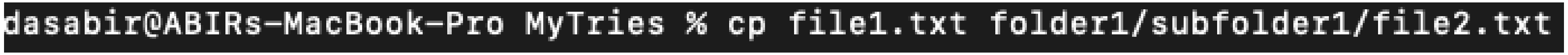
  - https://www.youtube.com/playlist?list=PLCC34OHNcOtoC6GglhF3ncJ5rLwQrLGnV

Graphical User Interface (GUI)

- Wikipedia says – '*The graphical user interface (GUI) is a form of user interface that **allows users to interact** with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation.*'

Pic source: www.computerhope.com

- Why GUI:
  - CLI (command line interface) can be intimidating to a beginner.



```
dasabir@ABIRs-MacBook-Pro MyTries % cp file1.txt folder1/subfolder1/file2.txt
```

  - A picture is worth a thousand words!

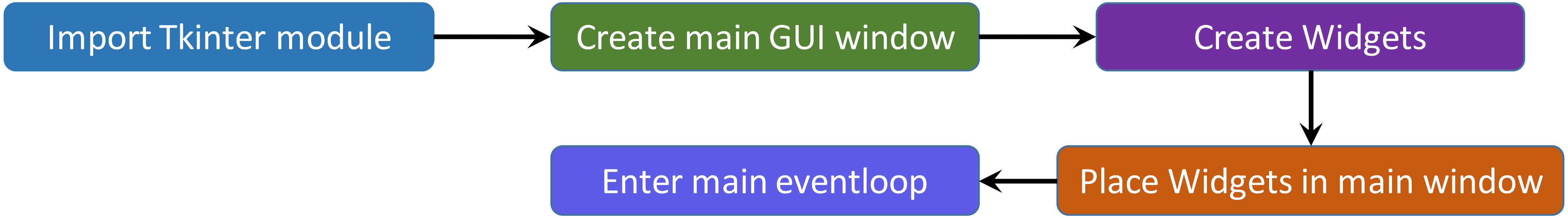
Python GUIs

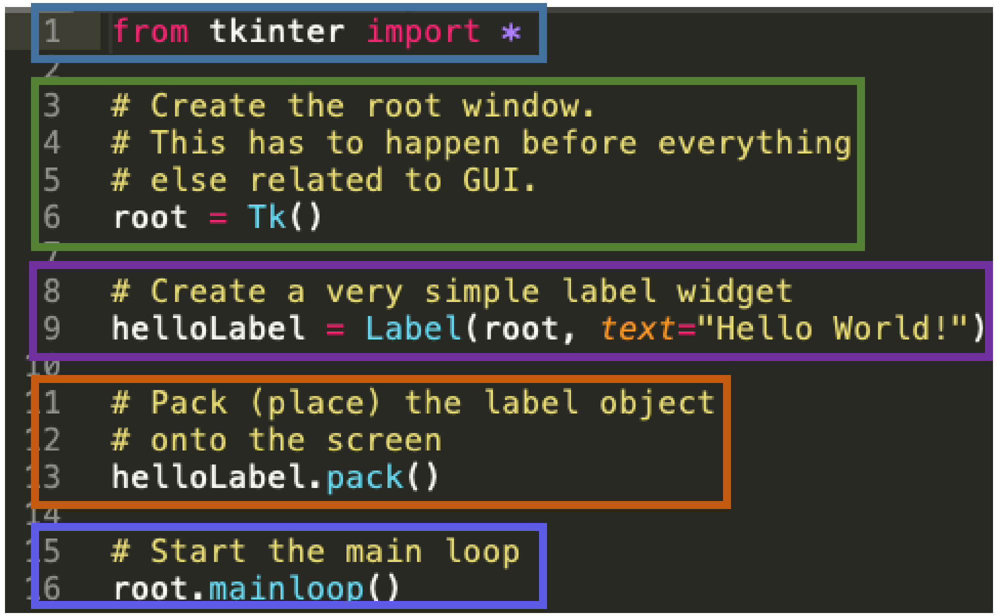
- **Tkinter**: standard built-in Python GUI library. built on top of the Tcl/Tk widget set.
- **wxPython**: wxWidgets is C++ based GUI library. wxPython is Python bindings for it.
- **PyQT**: wraps Qt toolkit facilities.

- **PyObjc**: Mac OS specific library.
- **PythonWin**: set of wrappings used to create Windows based GUIs.

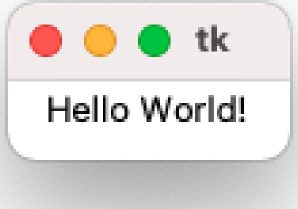
Fundamentals of Tkinter

- General workflow:


Import Tkinter module → Create main GUI window → Create Widgets → Place Widgets in main window → Enter main eventloop

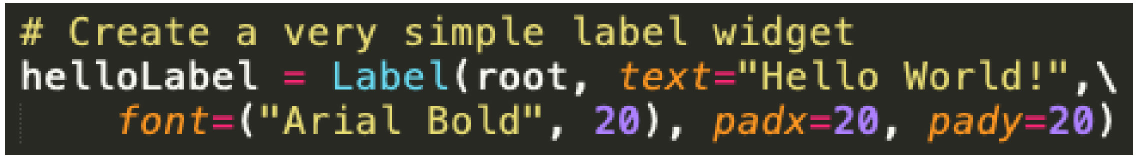


```python
from tkinter import *

# Create the root window.
# This has to happen before everything
# else related to GUI.
root = Tk()

# Create a very simple label widget
helloLabel = Label(root, text="Hello World!")

# Pack (place) the label object
# onto the screen
helloLabel.pack()

# Start the main loop
root.mainloop()
```

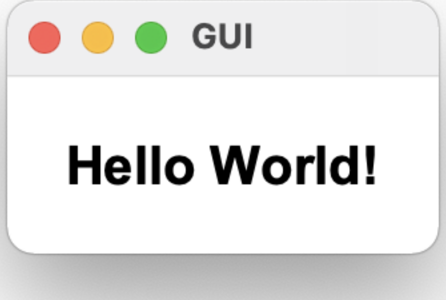


Label Widget

- A label is used to display text or an image. The label widget is for just viewing, not for interacting with.



```
# Create a very simple label widget
helloLabel = Label(root, text="Hello World!",\
    font=("Arial Bold", 20), padx=20, pady=20)
```



Hello World!

- Many options to play with: ‘bg’, ‘fg’, ‘font’, ‘bd’, ‘image’ etc.
- Comprehensive list: Tkinter reference. Link.

Tkinter 8.5 reference: a GUI for Python

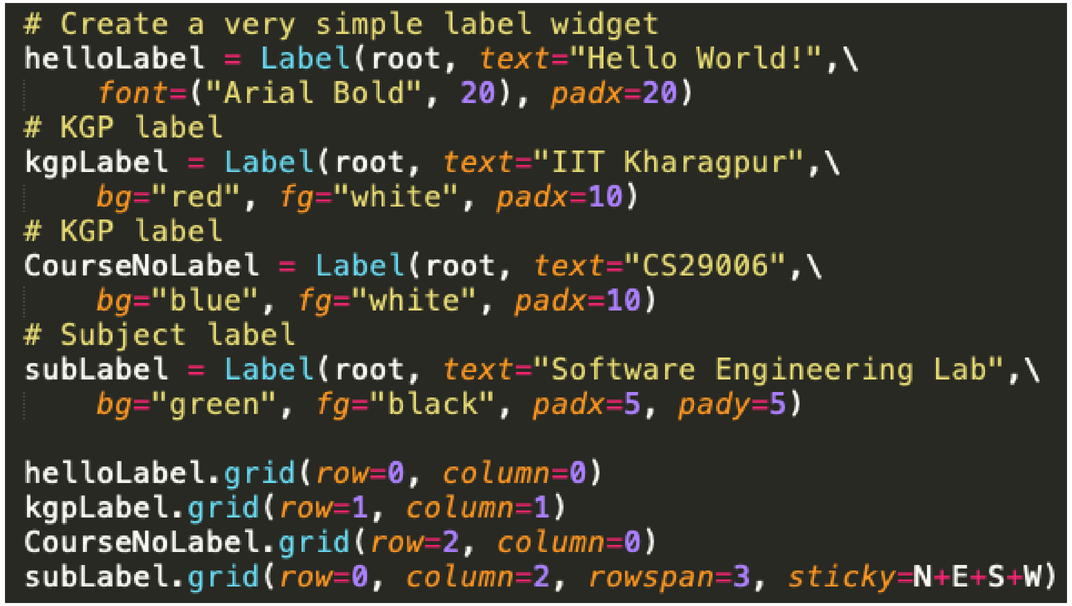
NEW MEXICO TECH
Computer Center www.nmt.edu/tcc

Layout/Geometry Manager in Tkinter

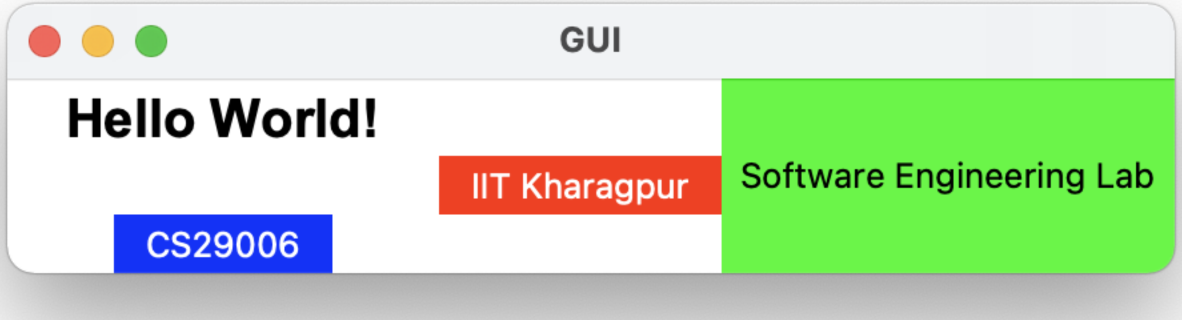
- Arranges widgets in the main GUI window.
  - pack
  - grid
  - place
- These three should never be mixed in the same master window!
- ‘pack’ is easiest to use but limited in capabilities. ‘packing’ widgets by default puts them below the other.
- ‘place’ allows explicit setting of position and size of widgets, either in absolute terms, or relative to another widget.

Layout/Geometry Manager in Tkinter

- ‘grid’ treats every window or frame as a table - a gridwork of rows and columns.
- A *cell* is the area at the intersection of one row and one column.
- The width of column is the width of widest cell in that column.
- The height of row is the height of tallest cell in that row.

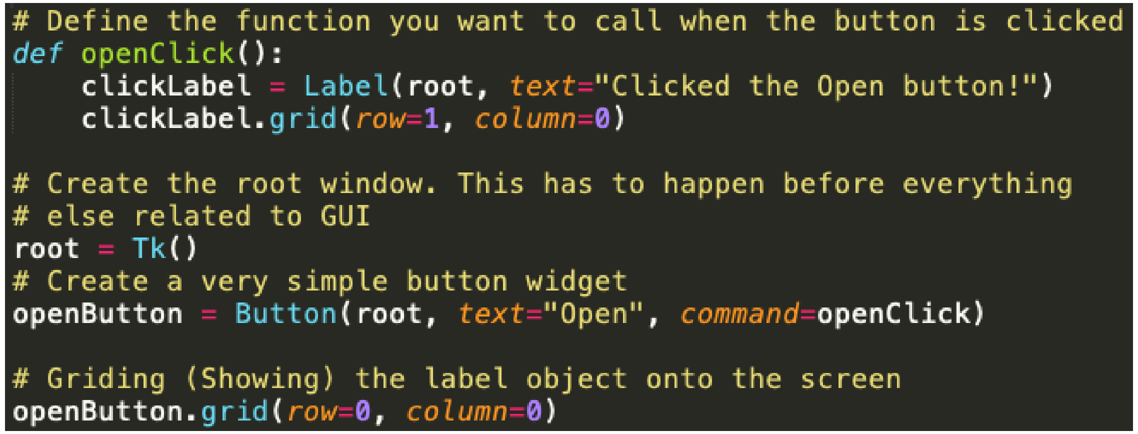


```
# Create a very simple label widget
helloLabel = Label(root, text="Hello World!",\
    font=("Arial Bold", 20), padx=20)
# KGP label
kgpLabel = Label(root, text="IIT Kharagpur",\
    bg="red", fg="white", padx=10)
# KGP label
CourseNoLabel = Label(root, text="CS29006",\
    bg="blue", fg="white", padx=10)
# Subject label
subLabel = Label(root, text="Software Engineering Lab",\
    bg="green", fg="black", padx=5, pady=5)

helloLabel.grid(row=0, column=0)
kgpLabel.grid(row=1, column=1)
CourseNoLabel.grid(row=2, column=0)
subLabel.grid(row=0, column=2, rowspan=3, sticky=N+E+S+W)
```

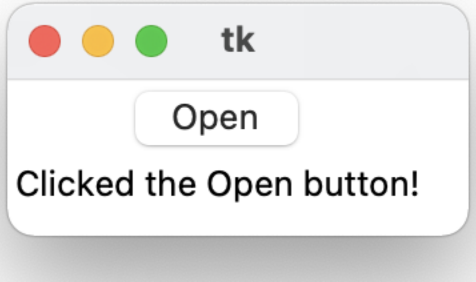


Button Widget

- Button press initiates a event.
- ‘command’ action takes the function we need to execute on buttonpress as input.



```
# Define the function you want to call when the button is clicked
def openClick():
    clickLabel = Label(root, text="Clicked the Open button!")
    clickLabel.grid(row=1, column=0)

# Create the root window. This has to happen before everything
# else related to GUI
root = Tk()
# Create a very simple button widget
openButton = Button(root, text="Open", command=openClick)

# Griding (Showing) the label object onto the screen
openButton.grid(row=0, column=0)
```

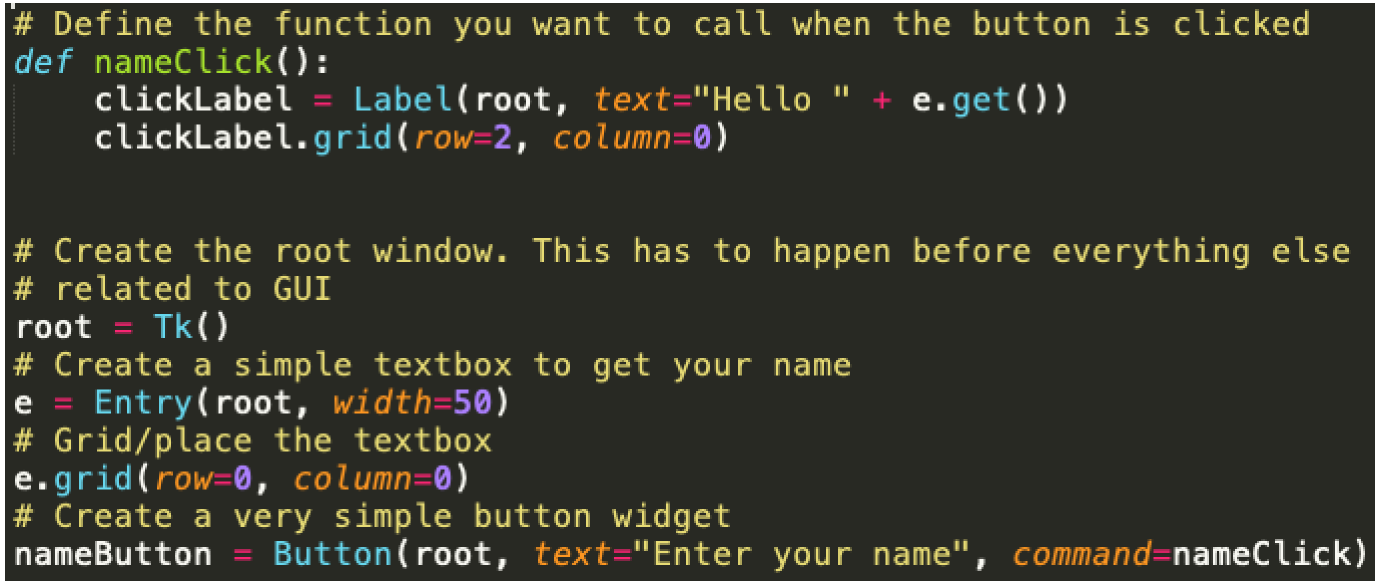


Clicked the Open button!

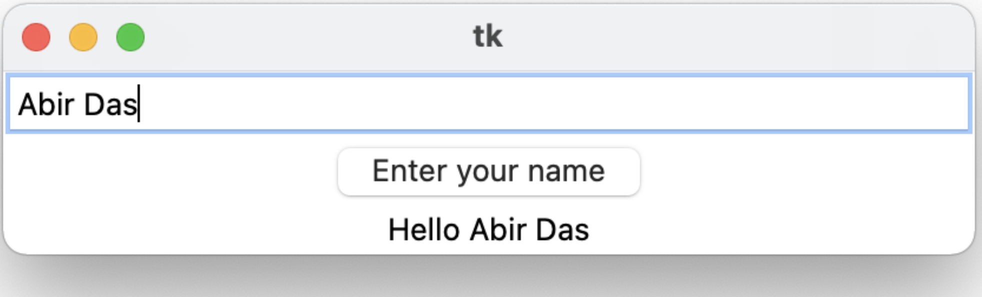
- Comprehensive list: Tkinter reference. Link

Textbox/Entry Widget

- Entry widgets are used to get input, i.e. text strings, from user.
- Some important methods of Entry object:
  - *.get()* – Get current text in 'Entry' as string
  - .insert(index, s) - Insert string 's' before the character at the given index.
  - .delete(first, last=None) - Delete characters starting at first, up to but not including position last.



```
# Define the function you want to call when the button is clicked
def nameClick():
    clickLabel = Label(root, text="Hello " + e.get())
    clickLabel.grid(row=2, column=0)


# Create the root window. This has to happen before everything else
# related to GUI
root = Tk()
# Create a simple textbox to get your name
e = Entry(root, width=50)
# Grid/place the textbox
e.grid(row=0, column=0)
# Create a very simple button widget
nameButton = Button(root, text="Enter your name", command=nameClick)
```



Abir Das

Hello Abir Das

For Assignment

- You will need to play with Tkinter FileDialog.
- You will need to play with Tkinter Combobox.
- Display images in Label.
- The assignment is based on the previous Python Data Science assignment.

Thank You