ENG 346
Data Structures and Algorithms for Artificial Intelligence
Tkinter

Dr. Kürşat İnce
kince@gtu.edu.tr

What widgets TKinter provide?

- tk_optionMenu
- panedwindow
- progressbar
- radiobutton
- scale
- scrollbar
- separator
- sizegrip
- spinbox
- text
- treeview
- button
- canvas
- checkbutton
- combobox
- entry
- frame
- label
- labelframe
- listbox
- menu
- menubutton
- message
- notebook

Also…

- **tk_chooseColor** - pops up a dialog box for the user to select a color.
- **tk_chooseDirectory** - pops up a dialog box for the user to select a directory.
- **tk_dialog** - creates a modal dialog and waits for a response.
- **tk_getOpenFile** - pops up a dialog box for the user to select a file to open.
- **tk_getSaveFile** - pops up a dialog box for the user to select a file to save.
- **tk_messageBox** - pops up a message window and waits for a user response.
- **tk_popup** - posts a popup menu.
- **toplevel** - creates and manipulates toplevel widgets.

Tk has three geometry managers:

- **place** - which positions widgets at absolute locations
- **grid** - which arranges widgets in a grid
- **pack** - which packs widgets into a space

To start

- import tkinter as tk

or:

- from tkinter import *

import tkinter

- The first widget (window object) that must be set up is the 'root' widget

```
root = Tk()
```

- There can only be one root widget in a program.
- A top level window is sometimes referred to as a 'master'

Then…

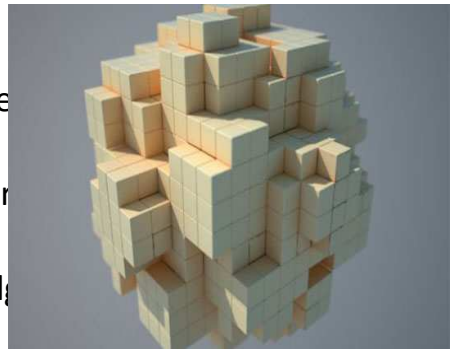
- Once the root is in place we can use other widgets each has to be instantiated as a child of the root

```
introlabel = Label(root, text="Hello Tkinter!")
```

- Widgets inside the window are sometimes referred to as ‘slaves’.

Geometry…

- We need to place the widget in the window.
- TK has 3 built-in geometry managers



- **place()** – This organizes widgets by placing the
in the parent.
- **pack()** – This organizes widgets in blocks befor
parent.
- **grid(***column***,** *row, sticky***)** – This organizes wid
structure in the parent.

Place

widget.place(place_options)

•**anchor** − The exact spot of widget other options refer to: may be N, E, S, W, NE, NW, SE, or SW, compass directions indicating the corners and sides of widget; default is NW (the upper left corner of widget)
•**bordermode** − INSIDE (the default) to indicate that other options refer to the parent's inside (ignoring the parent's border); OUTSIDE otherwise.
•**height, width** − Height and width in pixels.
•**relheight, relwidth** − Height and width as a float between 0.0 and 1.0, as a fraction of the height and width of the parent widget.
•**relx, rely** − Horizontal and vertical offset as a float between 0.0 and 1.0, as a fraction of the height and width of the parent widget.
•**x, y** − Horizontal and vertical offset in pixels

Pack

widget.pack(pack_options)

•**expand** − When set to true, widget expands to fill any space not otherwise used in widget's parent.
•**fill** − Determines whether widget fills any extra space allocated to it by the packer, or keeps its own minimal dimensions: NONE (default), X (fill only horizontally), Y (fill only vertically), or BOTH (fill both horizontally and vertically).
•**side** − Determines which side of the parent widget packs against: TOP (default), BOTTOM, LEFT, or RIGHT.

Grid

widget.grid(grid_options)

•**column** − The column to put widget in; default 0 (leftmost column).
•**columnspan** − How many columns the widget occupies; default 1.
•**padx, pady** − How many pixels to pad widget, horizontally and vertically, outside a widget's borders.
•**ipadx, ipady** −How many pixels to pad widget, horizontally and vertically, inside a widget's borders.
•**row** − The row to put awidget in; default the first row that is still empty.
•**rowspan** − How many rows widget occupies; default 1.
•**sticky** − What to do if the cell is larger than widget. By default, a widget is centered in its cell. Sticky may be the string concatenation of zero or more of N, E, S, W, NE, NW, SE, and SW, compass directions indicating the sides and corners of the cell t which widget sticks.

Then What?

- To run the program we must create enter the event loop

```
root.mainloop()
```

- This will keep the window open until we close it.

Let's try this!

- https://www.tutorialspoint.com/python3/python_gui_programming.htm