# Tkinter

Built-in package

Learning sources:

<https://tkdocs.com/tutorial/index.html>

<https://www.python-course.eu/python_tkinter.php>

<http://zetcode.com/ebooks/tkinter/>

<http://effbot.org/tkinterbook/>

## Tkinter vs Ttkinter

Tkinter (tk) is designed to be highly and easily configurable. You have almost complete control over how they look - border widths, fonts, images, colors, etc.

Ttkinter (ttk) inherits from Tkinter. Its widgets are already designed to look more "native", but at the expense of a loss of configurability (in fact, Ttkinter excludes many customization options from Tkinter).

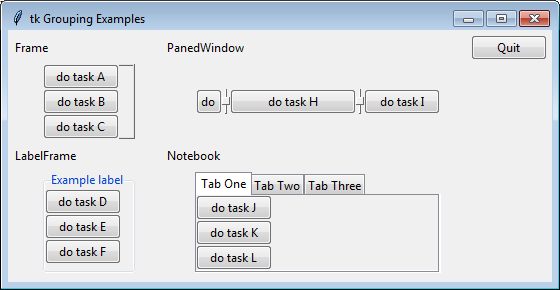
The advice is to **use ttk widgets if you want your GUI to look native and modern**, and **use the tk widgets if you need more configurability**. You can use them both in the same applications.

## Layouts

|  |  |
| --- | --- |
| **Layout Manager** | **Description** |
| place | You specify the exact size and position of each widget. |
| pack | You specify the size and position of each widget relative to each other. |
| grid | You place widgets in a cell of a 2-dimensional table defined by rows and columns. |

More details [here](https://runestone.academy/runestone/books/published/thinkcspy/GUIandEventDrivenProgramming/04_layout_managers.html).

## Widget Grouping



More details [here](https://runestone.academy/runestone/books/published/thinkcspy/GUIandEventDrivenProgramming/05_widget_grouping.html).

## Color Charts

<http://www.science.smith.edu/dftwiki/images/3/3d/TkInterColorCharts.png>

## Multi-Threading

Tkinter doesn't work in a worker thread because it **doesn't support multithreading**. Everything that interacts with a Tkinter widget needs to run in the main thread. If you want to use multithreading, put the GUI in the main thread and your other code in a worker thread, and communicate between them with a thread safe Queue.

## Multi-Processing

All interaction with the GUI must be in a single process. Tkinter itself cannot span processes, nor threads. Your other processes must put work on a queue that the tkinter process periodically polls and acts upon (of course, you could also communicate with any other form of IPC).

The reason for this is that each root window is associated with an embedded TCL interpreter, and this interpreter itself cannot span processes.

## Multiple Instance Initialization

Cause of error "*\_tkinter.TclError: NULL main window*"

You **cannot call Tk() twice**. If you need two windows use TopLevel() to make the second window. Tk() doesn't just make a window, it also does a bunch of initializations that is only supposed to happen once.

# PyQt

<https://pypi.org/project/PyQt5/>

# PySide

<https://doc.qt.io/qtforpython/api.html>

# WxPython

<https://docs.wxpython.org/>

# Extensions

## System Tray

|  |  |  |
| --- | --- | --- |
| **Package** | **Pros** | **Cons** |
| [pystray](https://pystray.readthedocs.io/en/latest/index.html) | Support menu updating (update icon for checkmark, enable/disable menu items, change menu item label dramatically, etc.). | - Have to configure multithreading manually (when started, pystray runs in a separate un-parallel worker thread, so we have to configure a separate parallel thread for it so that it can run concurrently with the main thread or other threads). Check [here](https://www.tutorialspoint.com/python/python_multithreading.htm).  - Cannot know how to add separator for item groups. |
| [infi.systray](https://github.com/Infinidat/infi.systray) | - Easier to use.  - Support multithreading by default. | - Doesn't support menu icon updating (checkmark, etc.)  - Doesn't support [pyinstaller](#_Create_Executable_Python). Fix [here](https://www.reddit.com/r/learnpython/comments/ex44mj/how_to_create_an_executable_python_file_in_windows/fg6n7b9/). |