Calendar Generator Project

The Calendar Generator is a simple project developed using python. It is a simple interesting calendar generating mini project. This project is a GUI based, which helps in creating calendar from the year of users input. This Calendar application is designed for creating calendar. In this application, the user can input the year of which the user wants to generate, then press enter. Then the user will get the calendar for that year. Also, the design of this system is simple so that the user won’t get any difficulties while working on it. The library I am using in this project is tkinter.

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task. Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Tkinter provides a variety of common GUI elements which I can use to build the interface – such as buttons, menus and various kinds of entry fields and display areas. These elements are called widgets. Each widget has a parent widget, all the way up to the root window of the application. For example, a button or a text field needs to be inside some kind of containing window.

The widget classes provide a lot of default functionality. They have methods for configuring the GUI’s appearance – for example, arranging the elements according to some kind of layout – and for handling various kinds of user-driven events.

There are two main methods used which the user needs to remember while creating the Python application with GUI.

1. **Tk(screenName=None,  baseName=None,  className=’Tk’,  useTk=1)-** To create a main window, tkinter offers a method ‘Tk(screenName=None, baseName=None, className=’Tk’,  useTk=1)’. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:

*m=tkinter.Tk() where m is the name of the main window object*

1. **mainloop()-** There is a method known by the name mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.

Tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

1. **pack() method-** It organizes the widgets in blocks before placing in the parent widget.
2. **grid() method-** It organizes the widgets in grid (table-like structure) before placing in the parent widget.
3. **place() method-** It organizes the widgets by placing them on specific positions directed by the programmer.

Link: <https://docs.python.org/3/library/tkinter.html>

Link: <http://msdl.cs.mcgill.ca/presentations/02.02.Tkinter/python-intro.pdf>