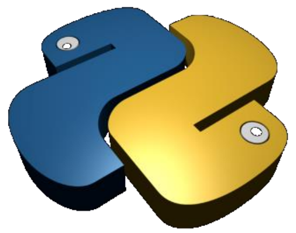
GUI for Python

Session 08

Session Overview

In this session, you will be able to:

- Identify the different Python packages for creating GUIs
- Describe the significance of Tkinter module in Python
- Explain the different Tkinter widgets and geometry managers
- Explain how to manage events in Python



Basics of Graphical User Interface (GUI) in Python

A GUI is a visual aid for a user which is effective in communicating with an application.

These graphical aids eliminate the requirement of typing the commands for communication.

GUIs contain pictorial items through which users inform the application what they expect from it.

Python comes with different options for creating GUIs which are:

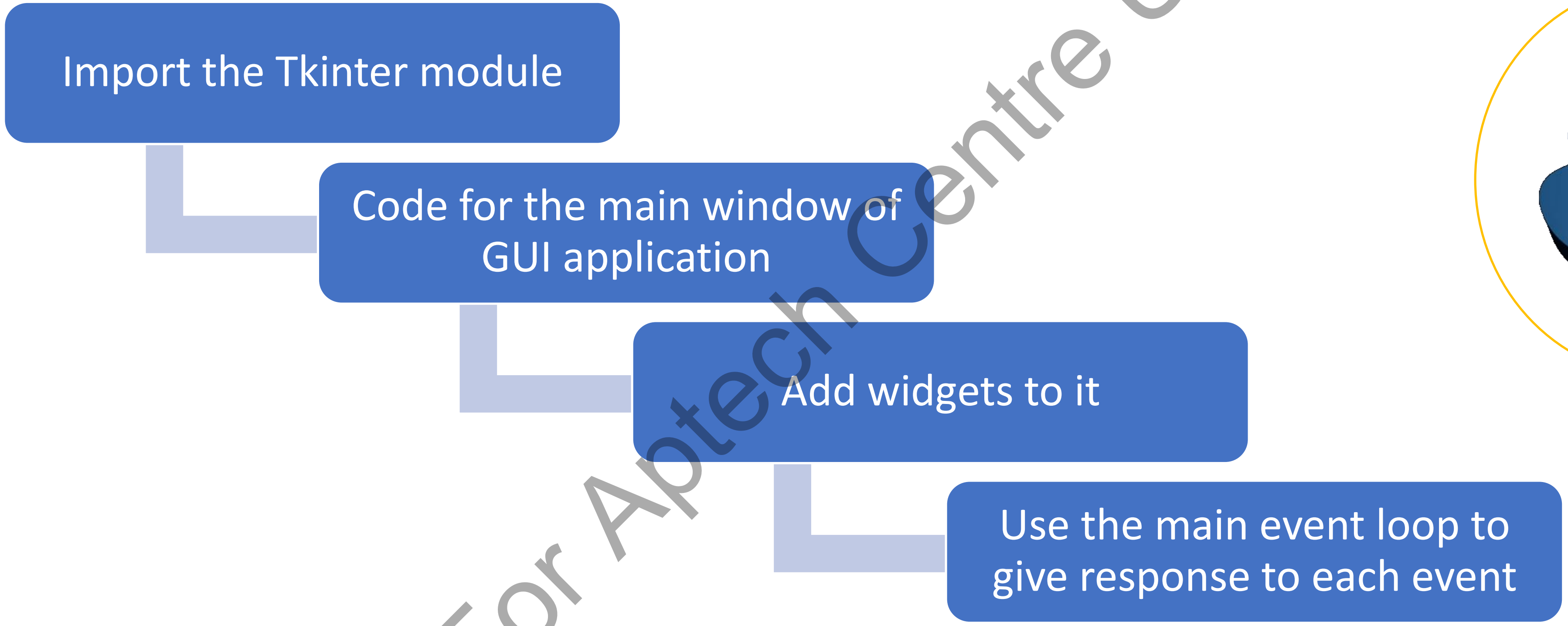
- Tk Interface (Tkinter)
- WxPython
- JPython

Tkinter GUI (1-2)

- Tkinter is the Tk version of GUI for Python.
- Tkinter offers a robust object-oriented interface to the Tk toolkit.
- It offers a variety of dialog boxes and widgets or controls.
- All Tkinter widgets are associated with a few geometry management methods.

Tkinter GUI (2-2)

Steps to be implement by developer:


Import the Tkinter module
Code for the main window of GUI application
Add widgets to it
Use the main event loop to give response to each event

Label Widget (1-2)

The Label widget displays an image or a text in a single or multiple lines.

Adding a label to a Python application:

Syntax

```
wlbl = Label (root,
option1, option2, ...)
```

Label Widget (2-2)

Options that developers can specify for labels:

Anchor	Bg	Bitmap	Bd	Font
Fg	Image	Text	Wraplength	Underline
Justify	Padx	Pady	Width	Textvariable

Message Widget

Options that developers can specify for messages:

Anchor	Aspect	Bg	Bd	Font
Fg	Highlight backgroun d	Text	Relief	Takefocus
Justify	Padx	Pady	Width	Textvaria ble

Grid Geometry Manager/Grid Widget

Grid is a geometry manager that arranges widgets in a table in the containing parent widget.

Options that developers can specify for the grid layout:

- Column
- Column span
- Row
- Rowspan
- ipadx, ipady
- padx, pady
- Sticky

Entry Widget/Entry Box

Entry is a basic Tkinter widget for obtaining a single-line input text from the user.

Unique options for Entry widget:

- Command
- Exportselection
- State
- Show
- Xscrollcommand

Button Widget

- Button is a standard widget for adding different buttons.
- These buttons can contain text or pictures, which indicate their purpose.
- A developer usually attaches a Python function or method with a button, which is called when a user clicks the corresponding button.
- The syntax is same as the Label widget but the `Button` class replaces the `Label` class.

Frame Widget

The Frame widget plays a vital role for grouping and arranging child widgets in a systematic manner.

It acts as a container for arranging the position of these widgets.

A frame is the screen's rectangular area acting as a geometry master that sets the layout and adds padding to these widgets.

Canvas Widget

Canvas is a generic widget that displays drawings and other complex illustrations.

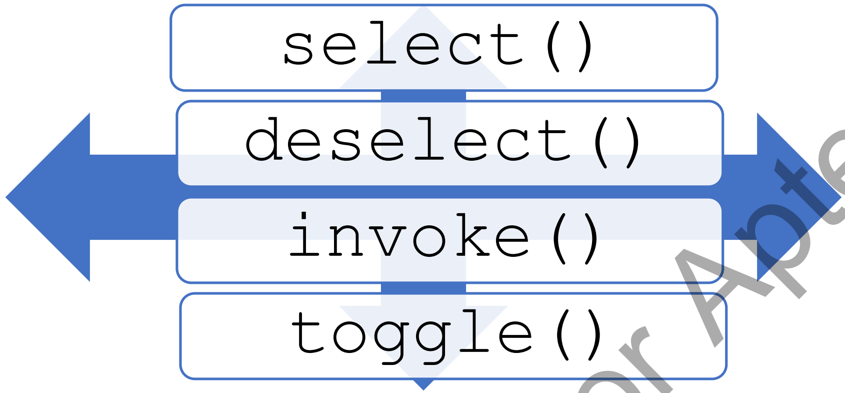
It renders all basic graphical objects such as pictures, lines, and circles.

Through the canvas widget, a developer can draw graphs and plots and create visual editors.

Another common use is to display different custom widgets.

Checkbutton Widget

- The Checkbutton widget is a check box control that allows users to select multiple options from the existing list of options.
- Useful methods of the Checkbutton class:



Radiobutton Widget

A radio button, or an option button, allows selecting only one of the existing options, unlike a check box.

In Tkinter, the Radiobutton widget renders a radio button.

It can hold text in single font or images.

A developer can associate a function or method with a radio button, which is called when a user selects it.

Although the methods of Radiobutton and Checkbutton widgets are same, the former does not have the `toggle()` method.

Listbox Widget

- Listbox is a standard widget showing a list of options from which a user can select one or more options.
- Most useful options that developers can use for Listbox widget:

relief

selectmode

Scrollbar Widget

- The Scrollbar widget adds a scroll bar or a side controller to other compatible widgets.
- It allows rendering vertically scrolled widgets such as Canvas, Listbox, and Text.
- Following are some useful options of the Scrollbar widget:

jump

orient

Scale Widget

- The Scale widget renders a slider for users to select values from a range of values by moving a knob.
- Options of Scale widget:

activebac kground	digits	from_	to	label
length	resolutio n	showvalue	sliderlen gth	state

GUI Events and Binds

A Tkinter Python application runs inside an event loop, which is triggered through the `mainloop()` method.

This is where it waits for events to take place.

An event can be a mouse click or a key press by the user.

To allow developers to handle these events, Tkinter provides a robust mechanism.

For each widget, a developer can bind methods and functions to its corresponding event.

Summary (1-5)

- Tkinter, WxPython, and JPython are the popular packages for developing GUIs in Python.
- Tkinter offers a variety of dialog boxes and widgets all of which are associated with geometry management methods.
- The three geometry managers in Tkinter are `place`, `grid`, and `pack`.

Summary (2-5)

- `Grid` is a more efficient Tkinter geometry manager than pack and arranges all widgets in a tabular format.
- Tkinter is initialized by invoking the `Tk()` method of `Tkinter` module, which also defines the Tk root widget.
- An application can have only root widget.

Summary (3-5)

- All widgets have their class in `Tkinter` module whose constructor accepts the parent window and configuration options as parameters.
- The Python script remains in `mainloop()` method that handles all user events along with Tkinter's queued operations.
- The steps to add a widget to a Python application are importing the Tkinter module, coding for main window, adding the widget to it, and call the main event loop for handling events.

Summary (4-5)

- Checkbutton, Entry, and Radiobutton widgets can connect directly to application variables through special control variables.
- The `textvariable` and `variable` control variables only accept a variable that points to an instance of a child class of `Variable`.
- The Scrollbar widget allows rendering scrolled widgets such as Canvas, Listbox, Entry, and Text.

Summary (5-5)

- If the `event` happens in the widget, Python calls the `handler` function with an event object informing about the event.
- The event sequence is in the form of `<modifier-type-detail>,` which is a string.