



# **PYTHON GUI**

**PYTHON TKINTER**

# OBJECTIVES

---

**At the end of this module, you will be able to:**

- Understand and design Python GUI .
- Understand and design a Program using PyAutoGUI module .



# TOPIC AGENDA

---

Python GUI

**PyAutoGui**

# TKINTER

**Tkinter:** Tkinter is the Python interface to the Tk GUI toolkit shipped with Python .

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.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

Import the *Tkinter* module.

Create the GUI application main window.

Add one or more of the above-mentioned widgets to the GUI application.

Enter the main event loop to take action against each event triggered by the user.

# PYTHON TKINTER

- Tk was developed as a GUI extension for the Tcl(tool command language) scripting language by John Ousterhout.
- The first release was in 1991. Tk proved as extremely successful in the 1990's, because it is easier to learn and to use than other toolkits.
- So it is no wonder that many programmers wanted to use Tk independently of Tcl.
- That's why bindings for lots of other programming languages have been developed, including Perl, Ada (called TASH), Python (called Tkinter), Ruby, and Common Lisp.
- Python also uses TK for creating GUI

# TKINTER WIDGETS

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets

Widgets	Description
Button	The Button widget is used to display buttons in your application.
Checkbutton	The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.
Entry	The Entry widget is used to display a single-line text field for accepting values from a user.
Frame	The Frame widget is used as a container widget to organize other widgets.
Label	The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
Listbox	The Listbox widget is used to provide a list of options to a user.
Menu	The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.

# TKINTER WIDGETS

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets

Widgets	Description
Radiobutton	The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.
Text	The Text widget is used to display text in multiple lines.
tkMessageBox	This module is used to display message boxes in your applications.
Scrollbar	The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.

# GEOMETRY MANAGEMENT

All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area.

Tkinter exposes the following geometry manager classes: pack, grid, and place

**The pack() Method - This geometry manager organizes widgets in blocks before placing them in the parent widget.**

**The grid() Method - This geometry manager organizes widgets in a table-like structure in the parent widget.**

**The place() Method - This geometry manager organizes widgets by placing them in a specific position in the parent widget.**

**Please Refer DEMO program**



# Thank You

# PyAutoGUI

# PYAUTOGUI

The pyautogui module has functions for simulating mouse movements, button clicks, and scrolling the mouse wheel.

The purpose of PyAutoGUI is to provide a cross-platform Python module for GUI automation for human beings. The API is designed to be as simple as possible with sensible defaults.

The ultimate tools for automating tasks on your computer are programs you write that directly control the keyboard and mouse.

These programs can control other applications by sending them virtual keystrokes and mouse clicks, just as if you were sitting at your computer and interacting with the applications yourself.

Think of GUI automation as programming a robotic arm. You can program the robotic arm to type at your keyboard and move your mouse for you

PyAutoGUI can be installed from the pip tool or downloaded from PyPI:  
<https://pypi.python.org/pypi/PyAutoGUI>

# MOUSE CONTROL FUNCTIONS

Function Name	Description
position()	The current X and Y coordinates of the mouse cursor are returned by the position() function.
Size()	The screen resolution size is returned by the size() function as a tuple of two integers.
onScreen()	onScreen() function, which will return True if they are within the screen's boundaries and False if not

# MOUSE MOVEMENT FUNCTIONS

Function Name	Description
<code>moveTo(X,Y)</code>	function will move the mouse cursor to the X and Y integer coordinates you pass it. <code>pyautogui.moveTo(100, 200)</code> moves mouse to X of 100, Y of 200. <code>pyautogui.moveTo(100, 200, 2)</code> moves mouse to X of 100, Y of 200 over 2 seconds
<code>dragTo()</code>	<code>dragTo(100, 200, button='left')</code> # drag mouse to X of 100, Y of 200
<code>click()</code>	<code>click()</code> function simulates a single, left-button mouse click at the mouse's current position
<code>mouseDown()</code>	<code>pyautogui.mouseDown(button='right')</code> press the right button down
<code>mouseUp()</code>	<code>pyautogui.mouseUp(button='right', x=100, y=200)</code> move the mouse to 100, 200, then release the right button up

# KEYBOARD FUNCTIONS

Function Name	Description
<code>typewrite()</code> <code>typewrite('Hello world!')</code> <code>typewrite('Hello world!', interval=0.25)</code>	This function will type the characters in the string is passed. prints out "Hello world!" prints out "Hello world!" with a quarter second delay after each character
<code>press()</code> <code>press('enter')</code> <code>press('f1')</code>	To press these keys, call the <code>press()</code> function and pass it a string from the <code>pyautogui.KEYBOARD_KEYS</code> such as <code>enter</code> , <code>esc</code> , <code>f1</code> etc press the Enter key
<code>hotkey()</code> <code>hotkey('ctrl', 'shift', 'esc')</code>	the <code>hotkey()</code> can be passed several key strings which will be pressed down in order, and then released in reverse order
<code>keyUp('esc')</code>	functions, which simulate pressing a key down and then releasing it up.
<code>keyDown('esc')</code>	functions, which simulate pressing a key down .

# SUMMARY

---

**At the end of this module, you are able to:**

- Develop GUI program
- Write program using PyAutoGui module



# Thank You