# SYSTEM SOFTWARE

**Mini project**

# File Explorer in Python using Tkinter

Python offers various modules to create graphics programs. Out of these Tkinter provides the fastest and easiest way to create GUI applications.

In this mini-project we had made a simple python GUI using the Tkinter module. The tkinter package (“Tk interface”) is the standard Python interface to the Tk GUI toolkit. Both Tk and tkinter  are available on most Unix platforms, as well as on Windows systems.

Running python -m tkinter from the command line should open a window demonstrating a simple Tk interface, letting you know that tkinter  is properly installed on your system, and also showing what version of Tcl/Tk is installed, so you can read the Tcl/Tk documentation specific to that version.

Most of the time, tkinter is all you really need, but a number of additional modules are available as well. The Tk interface is located in a binary module named \_tkinter. This module contains the low-level interface to Tk, and should never be used directly by application programmers. It is usually a shared library (or DLL), but might in some cases be statically linked with the Python interpreter.

In addition to the Tk interface module, tkinter includes a number of Python modules, tkinter.constants being one of the most important. Importing tkinter will automatically import tkinter.constants, so, usually, to use Tkinter all you need is a simple import statement:

**from** **tkinter** **import** \*

The [tkinter.filedialog](https://docs.python.org/3/library/dialog.html" \l "module-tkinter.filedialog" \o "tkinter.filedialog: Dialog classes for file selection (Tk)) module provides classes and factory functions for creating file/directory selection windows.

The below functions when called create a modal, native look-and-feel dialog, wait for the user’s selection, then return the selected value(s) or None to the caller.

# tkinter.filedialog.**askopenfile**(*mode="r"*, *\*\*options*)

# tkinter.filedialog.**askopenfiles**(*mode="r"*, *\*\*options*)

The above two functions create an [Open](https://docs.python.org/3/library/dialog.html#tkinter.filedialog.Open) dialog and return the opened file object(s) in read-only mode.

# tkinter.filedialog.**asksaveasfile**(*mode="w"*, *\*\*options*)

Create a [SaveAs](https://docs.python.org/3/library/dialog.html" \l "tkinter.filedialog.SaveAs" \o "tkinter.filedialog.SaveAs) dialog and return a file object opened in write-only mode.

# tkinter.filedialog.**askopenfilename**(*\*\*options*)

# tkinter.filedialog.**askopenfilenames**(*\*\*options*)

The above two functions create an [Open](https://docs.python.org/3/library/dialog.html#tkinter.filedialog.Open) dialog and return the selected filename(s) that correspond to existing file(s).

# tkinter.filedialog.**asksaveasfilename**(*\*\*options*)

Create a [SaveAs](https://docs.python.org/3/library/dialog.html" \l "tkinter.filedialog.SaveAs" \o "tkinter.filedialog.SaveAs) dialog and return the selected filename.

# tkinter.filedialog.**askdirectory**(*\*\*options*)

Prompt user to select a directory.

Additional keyword option:

*mustexist* - determines if selection must be an existing directory.

*# class*tkinter.filedialog.**Open**(*master=None*, *\*\*options*)

*# class*tkinter.filedialog.**SaveAs**(*master=None*, *\*\*options*)

The above two classes provide native dialog windows for saving and loading files.

Getting started with the program:

**Prerequisites:** Python offers various modules to create graphics programs. Out of these Tkinter provides the fastest and easiest way to create GUI applications.

**The following steps are involved in creating a tkinter application:**

* Importing the tkinter module.
* Creation of the main window (container).
* Addition of widgets to the main window
* Applying the event Trigger on widgets like buttons, etc.

#### Creating the File Explorer

In order to do so, we have to import the **filedialog** module from Tkinter. The Filedialog module will help you open, save files or directories.

In order to open a file explorer, we have to use the method, askopenfilename(). This function creates a file dialog object.

**PROGRAM STEPS:**

1. import all components from the tkinter library.
2. import filedialog module.
3. Create a Function for opening the file explorer window.
4. Name the function as browsefile and change label contents.
5. Now create a root window as tk()
6. Set window title
7. Then set window site
8. We can also change the background color.
9. Create a File Explorer label
10. Grid method is chosen for placing the widgets at respective positions in a table like structure by specifying rows and columns
11. Let the window wait for any events

