To develop GUI application in Python, there are multiple options in terms of python packages. The most generally used package is `tkinter`.

In this Python GUI Tutorial, we will use tkinter to learn how to develop GUI applications. You may be wondering on why we are using tkinter. The answer is quite simple. There is a large tkinter community online that can help you, through forums and other websites.

While tkinter provides the widgets with all the functionality and behavior aspects, there is another module named `tkinter.ttk` which provides themed widget set.

### Getting Started with Tkinter

Tkinter is an inbuilt python package. You can import the package and start using the package functions and classes.

```python
import tkinter as tk
import tkinter as tk

or you can use the other variation of importing the package

from tkinter import *
from tkinter import *
```

### Create a Simple GUI Window

To create a GUI Window, tkinter provides `Tk()` class. The syntax of `Tk()` class is:

```python
Tk(screenName=None, baseName=None, className='Tk', useTk=1)
```

Following is a simple example to create a GUI Window.

**Python Program**
import tkinter as tk

main_window = tk.Tk()
main_window.mainloop()

Output

You can change the title of the window by using title function on the root or main window widget.

Python Program

tkinter.py

import tkinter as tk

import tkinter as tk

main_window = tk.Tk()
main_window.title('Python GUI Tutorial - by TutorialKart')
main_window.mainloop()

Output
You can add widgets into the window. Also note that there are a wide variety of widgets you can use from tkinter. In this Tkinter Tutorial, we will cover all these widgets. Following are the list of Tkinter widgets.

- Button
- Canvas
- Checkbutton
- Radiobutton
- Entry
- Frame
- Label
- Listbox
- Menu
- MenuButton
- Message
- Scale
- Scrollbar
- Text
- TopLevel
- SpinBox
- PannedWindow

After creating a GUI window using Tk() and before calling the mainloop() function on the window, you can add as many widgets as required.

```python
from tkinter import *
```
from tkinter import *

gui = Tk()
# add widgets here
gui.mainloop()

Example – Tkinter Button Widget

To add a button to the Python Window, use the following syntax

button = Button(master, option=value)
button.pack()

where master is the window to which you would like to add this button, and you may provide different options to the button constructor. The options available for button are:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>activebackground</td>
<td>button’s background color when button is under the cursor</td>
</tr>
<tr>
<td>activeforeground</td>
<td>button’s foreground color when button is under the cursor</td>
</tr>
<tr>
<td>bg</td>
<td>background color of button</td>
</tr>
<tr>
<td>command</td>
<td>function to be called on click</td>
</tr>
<tr>
<td>font</td>
<td>font on the button label</td>
</tr>
<tr>
<td>image</td>
<td>image on the button</td>
</tr>
<tr>
<td>width</td>
<td>width of the button</td>
</tr>
<tr>
<td>height</td>
<td>height of the button</td>
</tr>
<tr>
<td>text</td>
<td>text of the button label</td>
</tr>
</tbody>
</table>

In this example, we will create a simple button with values provided for some of the options,

Python Program

```python
from tkinter import *
```
from tkinter import *

# create gui window
gui = Tk(className='Python GUI Tutorial - by TutorialKart')

#widgets start

button = Button(gui, text='Submit', width=50, height=4, bg='#33CC33', fg='#FFFFFF', activebackground='#44DD44', activeforeground='bg
button.pack()

#widgets end

gui.mainloop()

Output

Tkinter Problems – Solved

While working with Tkinter, you may come across some of the following issues.

- Python Tkinter Frame Width Height Not Working

Conclusion

In this Python Tutorial, we learned about Tkinter library and the widgets it provides to build a GUI application in Python.