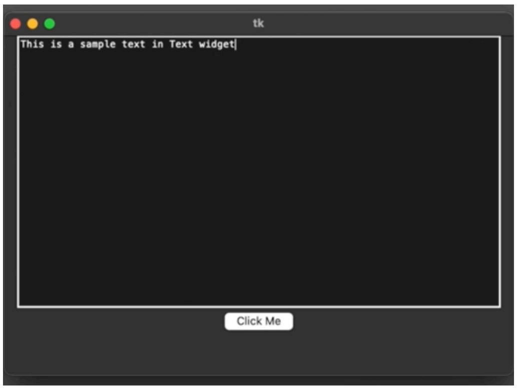
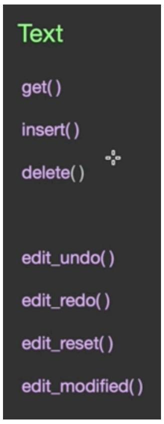
Text Widget

- It is almost same as the Entry widget, that is it takes the input from the user
- However Text widget allows the user to type multiline text where as Entry widget allows only one line of text entry
- In text widget we can also have a scroll bar



This is a sample text in Text widget

- The methods for a text widget are



Text

get( )

insert( )

delete( )

edit_undo( )

edit_redo( )

edit_reset( )

edit_modified( )

- We already know the working of get( ), insert( ) and delete( ), however when you call these methods we have to mention the index value
- When mentioning index we have to give 2 things, the line number and character index

- The other given methods are for undo and redo operations

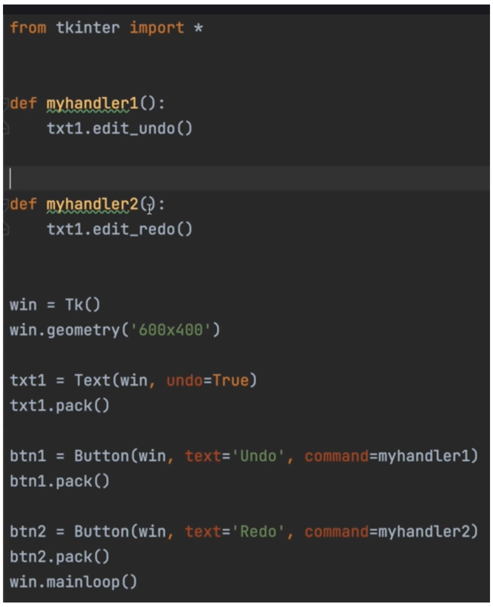
- **edit_undo( )** is used for undoing the sequence of action which we have done and **edit_redo( )** is the vice vera of it

- The sequence of operations which we perform are maintained in a stack and it will be piling up upon on another and when undo operation is performed the stack comes down in size and when we redo it the operation is performed once again. therefore the stack is maintained

- **edit_reset( )** will clear the stack

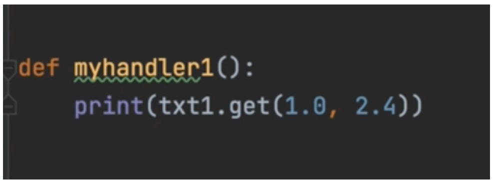
- **edit_modified( )** is used to know if any modifications have been performed on the text widget

- Example of undo and redo method



```
from tkinter import *


def myhandler1():
    txt1.edit_undo()


def myhandler2():
    txt1.edit_redo()


win = Tk()
win.geometry('600x400')

txt1 = Text(win, undo=True)
txt1.pack()

btn1 = Button(win, text='Undo', command=myhandler1)
btn1.pack()

btn2 = Button(win, text='Redo', command=myhandler2)
btn2.pack()
win.mainloop()
```

- Example of get( ) option



```
def myhandler1():
    print(txt1.get(1.0, 2.4))
```