**TEXT EDITOR**

**Aim-** To create a text editor using GUI in Python.

**Theory-**

**Python GUI – tkinter**

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

**To create a tkinter:**

1. Importing the module – tkinter
2. Create the main window (container)
3. Add any number of widgets to the main window
4. Apply the event Trigger on the widgets.

Importing tkinter is same as importing any other module in the python code. Note that the name of the module in Python 2.x is ‘Tkinter’ and in Python 3.x is ‘tkinter’.

import tkinter

There are two main methods used you the user need to remember while creating the Python application with GUI :-

1. **Tk(screenName=None,  baseName=None,  className=’Tk’,  useTk=1):** To create a main window, tkinter offers a method ‘Tk(screenName=None,  baseName=None,  className=’Tk’,  useTk=1)’. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:

m=tkinter.Tk() where m is the name of the main window object

1. **mainloop():** There is a method known by the name mainloop() is used when you are ready for the application to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event till the window is not closed.

m.mainloop()

|  |
| --- |
| import tkinter  m = tkinter.Tk()  '''  widgets are added here  '''  m.mainloop() |

tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class:-

1. **pack() method:**It organizes the widgets in blocks before placing in the parent widget.
2. **grid() method:**It organizes the widgets in grid (table-like structure) before placing in the parent widget.
3. **place() method:**It organizes the widgets by placing them on specific positions directed by the programmer.

**What is a TEXT EDITOR?**

A text editor is a type of computer program that edits plain text. Such programs are sometimes known as "notepad" software, following the naming of Microsoft Notepad. Text editors are provided with operating systems and software development packages, and can be used to change files such as configuration files, documentation files and programming language source code.

**OUR APPROACH**

1. First, we created a root window (Top-level window). Then we started adding different elements to our GUI such as Text(), Menu(), etc. After creating the menu (Toolbar) section using Menu(), we added different menu options such as “File” and “Edit” using:

* add\_cascade()

1. Under file-menu we added different sections such as:

* New
* Open File
* Save
* Save as

1. Under edit-menu we added different sections such as:

* Cut
* Copy
* Paste

Note:- Different sections were added using:

add\_command()

1. To separate different sections of the menu, we used the function:

* add\_separator()

**Functions used**

|  |  |
| --- | --- |
| **User-defined** | **Built-in** |
| open\_file() | add\_command() |
| save\_file() | add\_separator() |
| saveas\_file() | add\_cascade() |
| new\_file() | askopenfilename() |
| cut() | asksaveasfilename() |
| copy() | delete() |
| paste() | insert() |

**CODE:-**

from tkinter import \*

from tkinter import filedialog

class TextEditor:

clip=" "

filename=" "

def \_\_init\_\_(self,root):

filename="Untitled.txt"

self.t=Text(root, width=80, height=20, wrap=WORD)

self.t.pack()

#BASIC BODY:-

self.menubar=Menu(root) #Creates a menubar

root.config(menu=self.menubar) #Attach menubar to root window, same as pack

#self.s=Scrollbar(root, orient=VERTICAL, command=self.menubar)#.yview)

#self.menubar.configure(yscrollcommand=self.s.set)

#self.s.pack(side=RIGHT, fill=Y)

#FILE SECTION:-

self.filemenu=Menu(root,tearoff=0) #Create file menu

#create menu items in file menu

self.filemenu.add\_command(label="New",command=self.new\_file)

self.filemenu.add\_command(label="Open File",command=self.open\_file)

self.filemenu.add\_separator() #Add a horizontal seperator

self.filemenu.add\_command(label="Save",command=self.save\_file)

self.filemenu.add\_command(label="Save As...",command=self.saveas\_file)

self.filemenu.add\_separator() #Add a horizontal seperator

self.filemenu.add\_command(label="Exit",command=root.destroy)

#Another menu item under seperator

#Add file menu with name "File" to the menubar

self.menubar.add\_cascade(label="File", menu=self.filemenu)

#EDIT SECTION:-

self.editmenu=Menu(root,tearoff=0)

self.editmenu.add\_command(label="Cut",command=self.cut)

self.editmenu.add\_command(label="Copy",command=self.copy)

self.editmenu.add\_command(label="Paste",command=self.paste)

self.menubar.add\_cascade(label="Edit", menu=self.editmenu)

#MAIN CODE:-

def new\_file(self):

self.t.delete(1.0,END)

def save\_file(self):

if(self.filename=='Untitled.txt'):

self.saveas\_file()

else:

s=self.t.get(0.0,END)

f=open(self.filename,'w')

f.write(s)

f.close()

#Method for opening file and displaying it's content in the text box

def open\_file(self):

self.filename=filedialog.askopenfilename(parent=root, title="Select a file",filetypes=( ("Python Files","\*.py"), ("All Files","\*.\*") ) )

if(self.filename!=None):

self.t.delete(1.0,END)

f=open(self.filename,'r')

contents=f.read()

self.t.insert(1.0,contents)

f.close()

#Method to save file that already exists in the text box

def saveas\_file(self):

self.filename=filedialog.asksaveasfilename(parent=root, title="Save the file", defaultextension=".txt")

if(self.filename!=None):

f=open(self.filename,'w')

contents=str(self.t.get(1.0,END))

f.write(contents)

f.close()

#Edit code

def copy(self):

self.clip = self.t.get(SEL\_FIRST,SEL\_LAST)

def cut(self):

self.clip = self.t.get(SEL\_FIRST,SEL\_LAST)

self.t.delete(SEL\_FIRST,SEL\_LAST)

def paste(self):

i=self.t.index(INSERT)

self.t.insert(i,self.clip)

root=Tk() #Creates a blank window.

root.title("TEXT EDITOR")

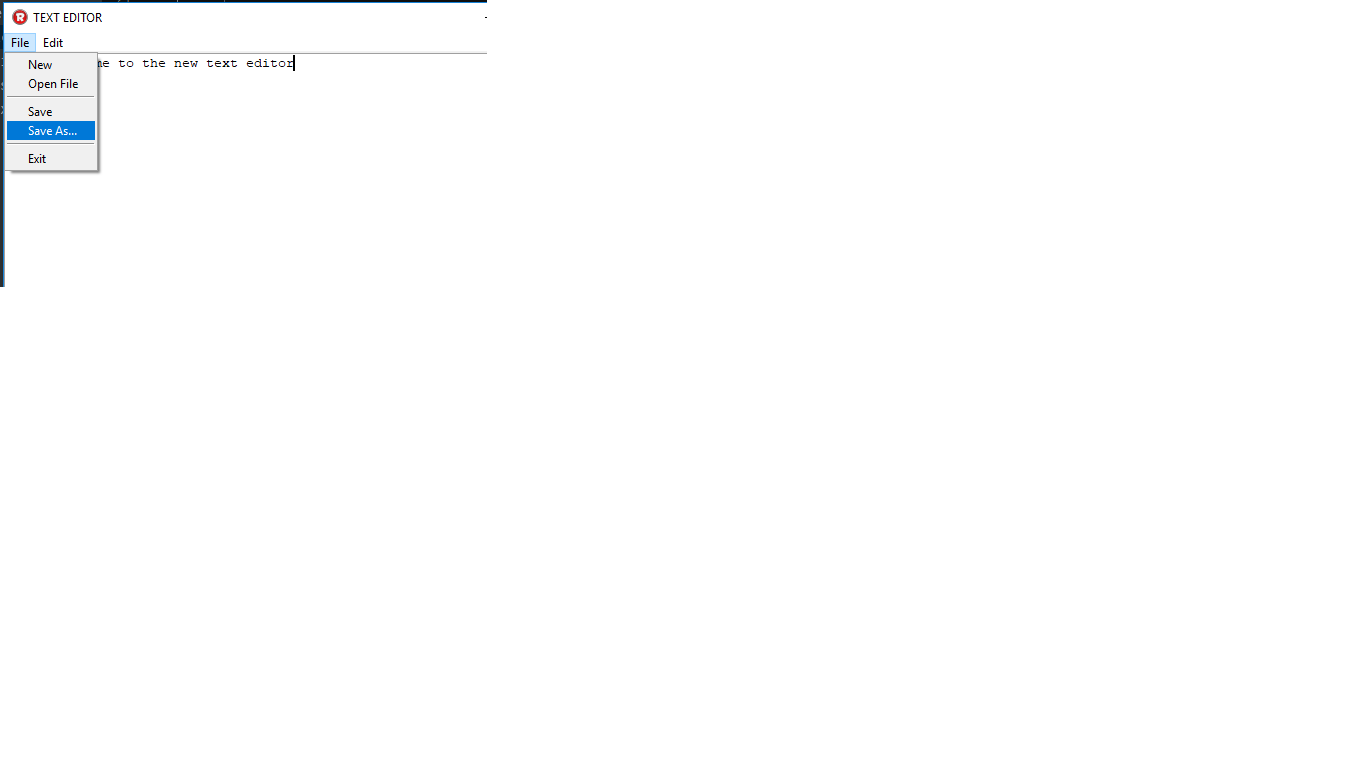
root.geometry("600x400")

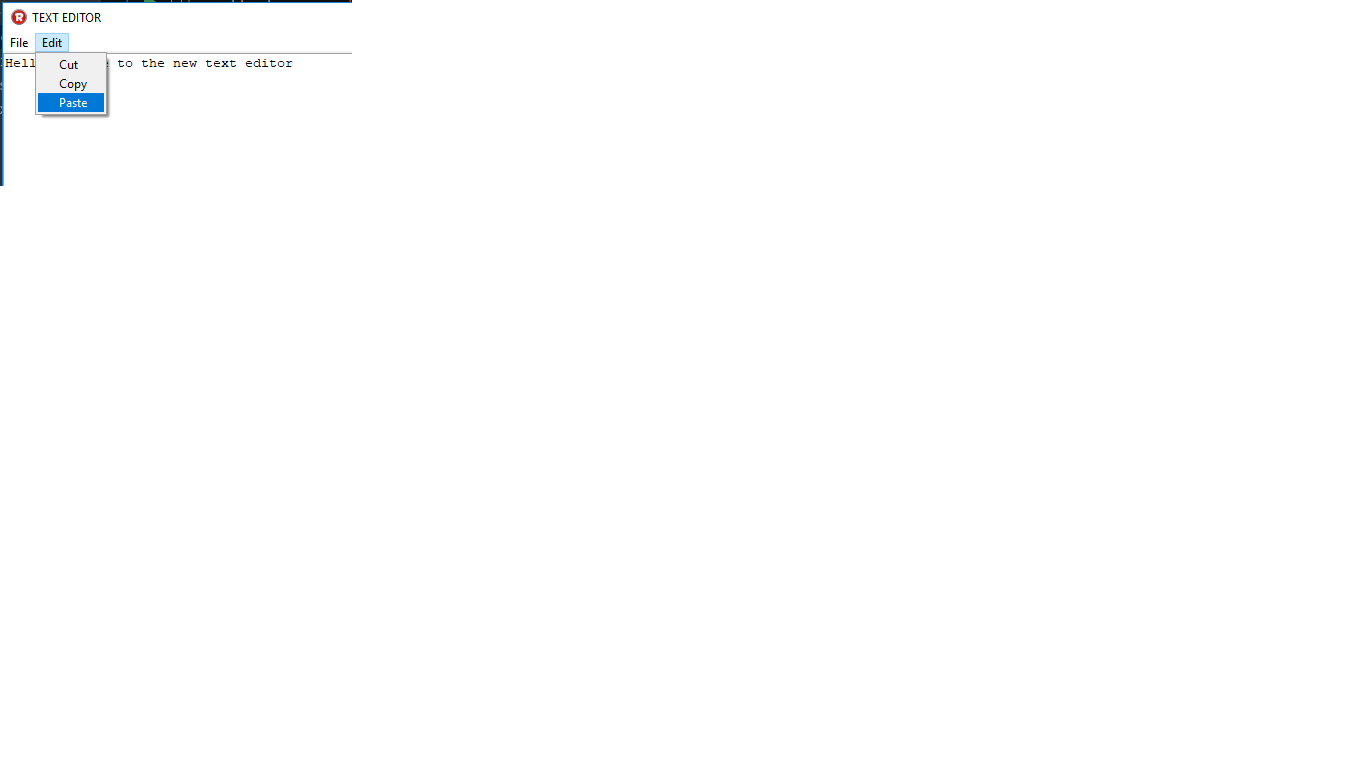
root.wm\_iconbitmap('R.ico')

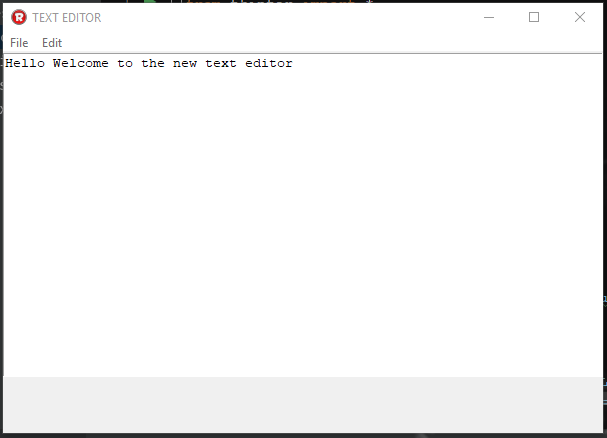
obj=TextEditor(root)

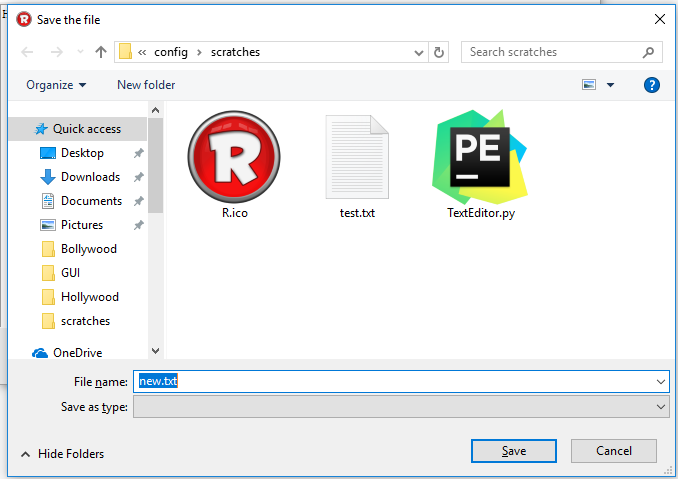
root.mainloop() #Doesn't let the program close after exeuting all stmnts until "x" is clicked.

**SCREENSHOTS :-**

****

****

****

****