from tkinter import \* # import at the very start

from tkinter import filedialog #use for file open function

from tkinter import messagebox

from tkinter.colorchooser import askcolor #background and text color functions

import datetime #import for current\_date function

import webbrowser #import for help function

import tkinter.font # import for bold italic underline functions

from tkinter.messagebox import askyesno, showwarning, showinfo

import tkinter.ttk

import sys

def New\_File():

if askyesno('Notepad', 'Save Existing Work?'):

filename = filedialog.asksaveasfilename()

if filename:

alltext = text.get(1.0, END)

open(filename, 'w').write(alltext)

if askyesno('Notepad', 'Open Existing File?'):

text.delete(1.0, END)

file = open(filedialog.askopenfilename(), 'r')

if file != '':

txt = file.read()

text.insert(INSERT, txt)

else:

pass

else:

text.delete(1.0, END)

def Open\_File():

text.delete(1.0, END)

file = open(filedialog.askopenfilename(), 'r')

if file != '':

txt = file.read()

text.insert(INSERT, txt)

else:

pass

def Save\_As():

filename = filedialog.asksaveasfilename()

if filename:

alltext = text.get(1.0, END)

open(filename, 'w').write(alltext)

def Close():

root.destroy()

def Copy():

text.clipboard\_clear()

text.clipboard\_append(text.selection\_get())

def Paste():

try:

teext = text.selection\_get(selection='CLIPBOARD')

text.insert(INSERT, teext)

except:

pass

def Replace():

def replaceall():

findtext = str(find.get(1.0, END))

replacetext = str(replace.get(1.0, END))

alltext = str(text.get(1.0, END))

alltext1 = all.replace(findtext, replacetext)

text.delete(1.0, END)

text.insert('1.0', alltext1)

replacebox = Tk()

replacebox.geometry("230x150")

replacebox.title("Replace..")

find = Text(replacebox, height=2, width=20).pack()

replace = Text(replacebox, height=2, width=20).pack()

replaceallbutton = Button(replacebox, text="Replace..", command=replaceall)

replaceallbutton.pack()

def Erase():

sel = text.get(SEL\_FIRST, SEL\_LAST)

text.delete(SEL\_FIRST, SEL\_LAST)

def Clear\_Screen():

text.delete(2.0,END) #1.0 indicates line no.

#Specifying 2.0 will delete only upto line no 2 not line 1

def Current\_Date():

data = datetime.date.today()

text.insert(INSERT, data)

def Text\_Color():

(triple, color) = askcolor()

if color:

text.config(foreground=color)

def No\_Format():

text.config(font=("Arial", 10))

def Bold():

current\_tags = text.tag\_names("sel.first")

if "bt" in current\_tags:

text.tag\_remove("bt", "sel.first", "sel.last")

else:

text.tag\_add("bt", "sel.first", "sel.last")

text.tag\_config("bt", font=("Arial", 10, "bold"))

def Italic():

text.tag\_add("here", "1.0", "1.4")

text.tag\_add("start", "1.8", "1.13") # use it after explaining 1st tag

text.tag\_config("here", font=("Arial", 10, "italic"))

text.tag\_config("start", background="black", foreground="green") #explain second last

def Underline():

text.tag\_add("here", "1.0", "1.4")

text.tag\_config("here", font=("Arial", 10, "underline"))

def Background():

(triple, color) = askcolor()

if color:

text.config(background=color)

def Online\_Help():

webbrowser.open('https://www.google.com/')

if \_\_name\_\_=="\_\_main\_\_":

root = Tk()

root.title("Notepad")

Main\_Menu = Menu(root)

Commands = Menu(root)

root.config(menu=Main\_Menu)

Main\_Menu.add\_cascade(label="File", menu=Commands)

Commands .add\_command(label="New File", command=New\_File)

Commands .add\_command(label="Open...", command=Open\_File)

Commands .add\_command(label="Save As...", command=Save\_As)

Commands .add\_command(label="Close", command=Close)

Edit\_Menu = Menu(root)

Main\_Menu.add\_cascade(label="Edit", menu=Edit\_Menu)

Edit\_Menu.add\_command(label="Copy", command=Copy)

Edit\_Menu.add\_command(label="Paste", command=Paste)

Edit\_Menu.add\_command(label="Replace", command=Replace)

Edit\_Menu.add\_command(label="Delete", command=Erase)

Edit\_Menu.add\_command(label="Clear Screen", command=Clear\_Screen)

Insert\_in\_Menu = Menu(root)

Main\_Menu.add\_cascade(label="Insert", menu=Insert\_in\_Menu)

Insert\_in\_Menu.add\_command(label="Date", command=Current\_Date)

Change\_Format = Menu(root)

Main\_Menu.add\_cascade(label="Format", menu=Change\_Format)

Change\_Format.add\_cascade(label="Font", command=Text\_Color)

Change\_Format.add\_separator()

Change\_Format.add\_radiobutton(label='No Format', command=No\_Format)

Change\_Format.add\_radiobutton(label='Bold', command=Bold)

Change\_Format.add\_radiobutton(label='Italic', command=Italic)

Change\_Format.add\_radiobutton(label='Underline', command=Underline)

Personalize = Menu(root)

Main\_Menu.add\_cascade(label="Personalize", menu=Personalize)

Personalize.add\_command(label="BackGround", command=Background)

User\_Help = Menu(root)

Main\_Menu.add\_cascade(label="Help", menu=User\_Help)

User\_Help.add\_command(label="Online", command=Online\_Help)

text = Text(root, height=40, width=100, font=("Arial", 10))

scrollbar = Scrollbar(root, command=text.yview)

scrollbar.config(command=text.yview)

text.config(yscrollcommand=scrollbar.set)

scrollbar.pack(side=RIGHT, fill=Y)

text.pack()

root.resizable(0, 0)

root.mainloop()