

```
#####
# Thomas Guldentops #
# TFE - Programmation#
# Juin 2018          #
#####

from tkinter import *
from tkinter import filedialog
from tkinter.filedialog import askopenfilename

import getpass
from os import path

class Notepad(object):
    ''' Création d'une application qui permet d'écrire et d'enregistrer du texte.'''
    def __init__(self, fen_largeur=400, fen_hauteur=210):
        self.fen = Tk()
        self.fen.geometry("{0}x{1}".format(str(fen_largeur), str(fen_hauteur)))
        self.fen.title("Bloc Note")
        self.fen.iconbitmap("images/image1.ico")
        self.fen.resizable(width=False, height=False)
        self.fen_largeur = fen_largeur
        self.fen_hauteur = fen_hauteur

        self.user = getpass.getuser() #Permet, sous window, d'avoir l'utilisateur actif, pratique pour définir le chemin à ouvrir lors

        self.scrollbar = Scrollbar(self.fen)
        self.scrollbar.pack(side=RIGHT, fill=Y)

        # Zone de texte
        self.var = StringVar()
        self.zoneText = Text(self.fen, bg="#282828", fg="#FDF1B8", wrap=WORD, yscrollcommand = self.scrollbar.set, insertbackground =
        self.zoneText.pack()
        # Sauvegarder ctrl + shift +s
        self.zoneText.bind("<Control-Shift-S>", self.saveCtrlShiftS) #création de raccourcis pour sauvegarder / ouvrir un fichier plus

        self.zoneText.bind("<Control-s>", self.saveCtrlS)
        # Ouvrir un fichier
        self.zoneText.bind("<Control-o>", self.openFile)
        # Barre de défilement
        self.scrollbar.config(command = self.zoneText.yview)

        # Menus de l'application
        self.barreMenu = Menu(self.fen)
        self.fen.config(menu = self.barreMenu)

        # Option de la barre de menu
        self.optionMenu = Menu(self.barreMenu, tearoff = 0)
        self.barreMenu.add_cascade(label="File", menu = self.optionMenu)
        self.optionMenu.add_command(label="Open (Ctrl - o)", command = self.openFile)
        self.optionMenu.add_command(label="Save (Ctrl - s)", command = self.saveCtrlS)
        self.optionMenu.add_command(label = "Save as (Ctrl - Shift - s)", command =self.saveCtrlShiftS )

        self.filename = None

        # Onglets

        self.fen.mainloop()

    def changeTitle(self, filename):
        ''' Permet de changer le titre de la fenêtre selon le fichier ouvert.'''
        if filename:
            self.fen.title("{} - Bloc Note".format(path.basename(filename)))

    def saveCtrlS(self, *event):
        ''' Permet de sauvegarder son avancé. Si le fichier existe déjà il va l'enregistrer sans demander à nouveau où l'enregistrer'
        try:
            if self.filename:
                if path.exists(self.filename):
                    with open(self.filename, 'w') as f:
                        f.write(self.zoneText.get('1.0', END))

            else:
                self.filename = filedialog.asksaveasfile(initialdir = "C:/Users/%s/Documents" % self.user,
                                                            mode = "w",
                                                            defaultextension=".txt",
                                                            filetypes = (("Text File (*.txt)", "*.txt"), ("All Files", "*.*")))

                self.filename.write(self.zoneText.get("1.0",END))
                self.filename.close()
                self.changeTitle(self.filename.name)
        except IOError as e :
            print(e)

    def saveCtrlShiftS(self, *event):
        ''' Permet de sauvegarder son avancé. Va toujours demander où enregistrer le fichier.'''
        try:
            self.filename = filedialog.asksaveasfile(initialdir = "C:/Users/%s/Documents" % self.user,
                                                        mode = "w",
                                                        defaultextension=".txt",
                                                        filetypes = (("Text File (*.txt)", "*.txt"), ("All Files", "*.*")))
```

```
        with open(self.filename, "w") as f:
            f.write(self.zoneText.get('1.0', END))
            self.changeTitle(self.filename.name)

    except IOError as e:
        print(e)

def openFile(self, *event):
    ''' Permet d'ouvrir un fichier texte existant sur l'ordinateur'''
    try:
        self.filename = askopenfilename(initialdir="C:/Users/%s/Documents" % self.user,
                                         filetypes=(("Text File", "*.txt"), ("All Files", "*.*")))
        self.zoneText.delete('1.0', END)

        with open(self.filename, 'r') as UseFile:
            self.zoneText.insert('1.0', UseFile.read())
            self.changeTitle(self.filename)

    except:
        print("File doesn't exist.")

if __name__ == "__main__":
    Notepad()
```