

Bonjour et bienvenu dans ce petit exemple :
Vous allez ici découvrir les différentes fonctions des styles de notre logiciel.
Premièrement, vous aurez remarqué que le "not" de "notre" est apparu en orange. Tout comme le "or" de "orange". Ceci est dû au fait que notre éditeur est conçu pour de la coloration syntaxique. Il met automatiquement en couleur certains mots, ce qui se règle par le biais du configurateur.
Il vous est aussi possible de créer vos propres styles, à savoir :
Premièrement, tapez votre texte ici. Ensuite, sélectionnez le passage dont vous souhaitez changer la couleur. En faisant clic droit ou bien dans le menu -> Style -> Nouveau style. Choisissez ensuite un nom pour votre style (vous pouvez le laisser vide). Dans la liste des couleurs, prenez-en une pour le texte et une pour le fond. En cliquant sur OK, votre style apparaît !
À chaque fois que vous voudrez réutiliser ce style, utiliser la fonction appliquer un style. Elle vous proposera la liste des styles actuels. Problème détecté, il y aura autant de style identiques que de fois où vous l'appliquez, -> Ce devrait bientôt être résolu :) !

Pour exporter, vous pouvez dès aujourd'hui sortir votre fichier .form en PDF ou bien en Word document. Pour cela, cliquez sur le bouton du menu -> Export -> Exporter en xxxx. Les exports :

- * L'EXPORT WORD : Il est possible d'exporter en word, cependant, les styles ne seront pas encore récupérables !!!

- * L'EXPORT PDF : Il est possible d'exporter en PDF, AVEC les styles. Ils seront conservés et pourront même être imprimés. Vous aurez aussi le choix des marges, inter ligne, inter caractères, sécurité : mot de passe, autorisation, ...

Merci pour votre confiance,

Et, n'hésitez pas à nous faire part de problèmes sur discord :

<https://discord.gg/Tzr7KNHnC3>

```

self.master.overrideredirect(1)
self.master.resizable(width=False, height=False)

self.top = Frame(self.master)
self.top.grid(row=1, column=0)

self.menu = Frame(self.master)
self.menu.grid(row=0, column=0)

name = 'Close '
if is_file:
    name += '"' + self.title + '"'
Button(self.menu, text=name, command=self.Quitter).grid(row=0, column=0)

def Quitter(self):
    self.data = None
    self.master.destroy()

def Geom(self):
    self.master.update()

    sw = self.master.winfo_screenwidth()
    sh = self.master.winfo_screenheight()

    ww = self.master.winfo_reqwidth()
    wh = self.master.winfo_reqheight()

    x = '+' + str(int((sw/2) - (ww/2)))
    y = '+' + str(int((sh/2) - wh))

    self.master.geometry(x + y)

def content(self, disabled = False):
    Label(self.top, text='ExOr Encrypton System', font=('Consolas', 20)).grid(row=0, column=0)
    if self.project:
        Label(self.top, text=self.project.upper(), bg='blue', fg='red', font=('Courier New', 26, 'bold')).grid(row=1, column=0)
        Label(self.top, text='Please, Enter password : ' if not disabled else 'Please, close window...').grid(row=2, column=0)
        self.pwd = StringVar()
        pwd = ttk.Entry(self.top, textvariable=self.pwd, show='$', width=20, font=('Calibri', 16), state='disabled' if disabled else 'normal')
        pwd.grid(row=3, column=0)
        b = ttk.Button(self.top, text='Encrypt/Decrypt', command=self.BtCommand, state='disabled' if disabled else 'normal')
        b.grid(row=4, column=0)
        if not disabled:
            b.bind_all('<Return>', self.BtCommand)

def Generate(self):
    self.top.mainloop()

def BtCommand(self, evt=None):
    self.data = self.pwd.get()
    if self.data == '':
        self.data = None
    self.master.destroy()

class Finished(Graphic):
    def __init__(self):

```

```

        Graphic.__init__(self, '')
        self.window(False)
        self.content(True)
        self.Geom()
        showinfo('ExOr Encrypton System', 'File Encrypted / D
ecrypted\nPlease, use the same password to Encrypt/Decrypt it
!')
        self.Generate()

class Main:
    def __init__(self, filename):
        self.filename = filename
        self.pwd = None
        self.key = None

    def lookProject(self):
        conn = sqlite3.connect(sys.argv[0].replace('main.exe'
, 'files.db'))
        cur = conn.cursor()

        cur.execute('CREATE TABLE IF NOT EXISTS list(id INTEG
ER PRIMARY KEY AUTOINCREMENT, file TEXT, project TEXT)')
        conn.commit()

        for row in cur.execute('SELECT * FROM list'):
            if row[1] == self.filename:
                return row

        return [None for i in range(3)]

    def askKey(self):
        pro = self.lookProject()
        g = Graphic(self.filename, pro[2])
        self.pwd = g.ask()

    def hashKey(self):
        self.key = hashKey(self.pwd)

    def crypt(self):
        e = ExOr()
        e.send(self.filename, self.key)
        e.crypt()

    def end(self):
        Finished()

if __name__ == '__main__':
    try:
        m = Main(sys.argv[1])
        m.askKey()
        m.hashKey()
        m.crypt()
        m.end()

    except IndexError:
        f = open('__init__.log', 'a')
        f.write('Error : no command supplied !\n')
        f.close()

```