

Comment faire un notebook avec bouton "fermer"

- Objet : Script Python - Comment faire un notebook avec bouton "fermer"
- Niveau requis :
[débutant](#), [avisé](#)
- Commentaires : *programmer un bouton "fermer" en python avec la bibliothèque graphique GTK*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
[obsolète](#)
 - Création par [syeric](#) le 02/06/2007
 - Testé par [paskal](#) le 25/10/2015
- Commentaires sur le forum : [C'est ici](#)¹⁾

Présentation

Salut !

Alors j'ai mis un moment à le trouver ! Donc, autant que ça serve pour le prochain 😊

Voici le script :

[notebook.py](#)

```
#!/usr/bin/env python
# -*- coding:utf-8 -*-

# notebook.py

import pygtk
pygtk.require('2.0')
import gtk

class NotebookExample:

    ## fonction pour ajouter un bouton fermer
    def add_icon_to_button(self, button):
        # création d'une boîte horizontale
        iconBox = gtk.HBox(False, 0)
        # création d'une image vide
        image = gtk.Image()
        # on récupère l'icone du bouton "fermer"
        image.set_from_stock(gtk.STOCK_CLOSE, gtk.ICON_SIZE_MENU)
        # on enlève le relief au bouton (donné en attribut)
        gtk.Button.set_relief(button, gtk.RELIEF_NONE)
        # on récupère les propriétés du bouton
        settings = gtk.Widget.get_settings(button)
```

```
# on affecte à w et h les dimensions
(w,h) =
gtk.icon_size_lookup_for_settings(settings,gtk.ICON_SIZE_MENU)
# on modifie ces dimensions
gtk.Widget.set_size_request(button, w + 4, h + 4)
image.show()
# on met l'image dans la boite
iconBox.pack_start(image, True, False, 0)
# on ajoute la boite dans le bouton
button.add(iconBox)
iconBox.show()
return

## crée une tab customisée avec un label et un bouton fermer
def create_custom_tab(self,text, notebook, frame):
# on crée une eventbox
eventBox = gtk.EventBox()
# on crée une boite horizontale
tabBox = gtk.HBox(False, 2)
# on crée un label "text" (text donné en attribut)
tabLabel = gtk.Label(text)
# on crée un bouton
tabButton = gtk.Button()
# on lui affecte la méthode remove_book
tabButton.connect('clicked',self.remove_book, notebook, frame)
# on ajoute l'image au bouton en utilisant la méthode
add_icon_to_button
self.add_icon_to_button(tabButton)
eventBox.show()
tabButton.show()
tabLabel.show()
# on attache label et bouton à la boite
tabBox.pack_start(tabLabel, False)
tabBox.pack_start(tabButton, False)
tabBox.show_all()
# on ajoute la boite à l'eventbox
eventBox.add(tabBox)
return eventBox

## fonction de suppression de page
def remove_book(self, button, notebook, frame):
# on supprime la page. Le secret, c'est de donner comme argument le
widget enfant de la page, ici un cadre frame.
notebook.remove(frame)
# on actualise le widget
notebook.queue_draw_area(0,0,-1,-1)

def delete(self, widget, event=None):
gtk.main_quit()
```

```
        return False

def __init__(self):
    window = gtk.Window(gtk.WINDOW_TOPLEVEL)
    window.connect("delete_event", self.delete)
    window.set_border_width(10)

    # on crée un nouveau notebook
    notebook = gtk.Notebook()
    window.add(notebook)
    notebook.show()

    # on ajoute quelques pages
    for i in range(5):
        page_number = i + 1
        frame = gtk.Frame("Frame %d" % page_number)
        frame.set_border_width(10)
        frame.set_size_request(100, 75)
        frame.show()
        label = gtk.Label("Dans la Frame %d" % page_number)
        frame.add(label)
        label.show()

        eventBox = self.create_custom_tab("Tab %d" % page_number,
notebook, frame)
        notebook.append_page(frame, eventBox)
    # page que nous verrons à l'ouverture (page 4)
    notebook.set_current_page(3)
    window.show()

def main():
    gtk.main()
    return 0

if __name__ == "__main__":
    NotebookExample()

    main()
```

Et vous obtenez :



1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:programmation:python:boutonfermer>



Last update: **25/04/2023 21:50**