

bmarkmidori

- Objet : script python pour midori¹⁾ sous Openbox²⁾
- Niveau requis :
[débutant, avisé](#)
- Commentaires : *Script mis à disposition par **Thuban** afin d'améliorer le menu de midori.*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
[à-tester](#)
 - Création par [thuban](#) le 31/12/2011
 - Testé par le
- Commentaires sur le forum : [C'est ici](#)³⁾

Aspect

Une fois le script en action, le menu de midori ressemblera à :



Script

Ajouter⁴⁾ à votre fichier :

```
~/ .config/openbox/menu.xml
```

Cette ligne :

[Extrait de .config/openbox/menu.xml](#)

```
menu execute="/usr/bin/python
~/Documents/Scripts/Openbox/ob_bmarkmidori.py" id="midori-menu"
label="Marques-Pages"
```

Et éditer le fichier :

```
~/Documents/Scripts/Openbox/ob_bmarkmidori.py
```

pour y écrire le script suivant :

"code python" [ob_bmarkmidori.py](#)

```
#!/usr/bin/env python
# -*- coding:Utf-8 -*-
# thuban <thuban@singularity.fr>
```

```
#
# add this line to menu.xml
#     <menu execute="/usr/bin/python
~/Documents/Scripts/Openbox/ob_bmarkmidori.py" id="midori-menu"
label="Marques-Pages"/>
#
# be careful not to use the id 2 in another menu to avoid conflicts
#####
#     This program is free software; you can redistribute it and/or
modify
#     it under the terms of the GNU General Public License as
published by
#     the Free Software Foundation; either version 2 of the License,
or
#     (at your option) any later version.
#####
#
import os
from xml.etree import ElementTree as ET

browser = 'midori'
bookmarks = os.path.expanduser('~/.config/midori/bookmarks.xbel')

def print_label(title, url):
    print('<item label="{0}">'.format(title.encode('utf-8')))
    print(' <action name="Execute">')
    print(' <execute>{0} {1}</execute>'.format(browser, url))
    print(' </action>')
    print(' </item>')

def print_submenu(title, titleList, urlList):
    print('<menu id="{0}" \
label="{1}">'.format(str(title.encode('utf-8'))+'6666', str(title).encod
e('utf-8')))
    if len(titleList) > 0:
        for i in range(len(titleList)):
            print_label(titleList[i], urlList[i])
    print('</menu>')

def print_bookmarks(elem):
    '''print les bookmarks avec leur url'''
    t = elem.getiterator('title')
    u = elem.getiterator('bookmark')

    for i in range(len(u)):
        print_label(t[i].text, u[i].get('href'))

def print_folder_menu(folder):
    folderTitle = folder[0].getiterator('title')[0].text
```

```
urls, titles = [], []
if len(folder[1:]) > 0:
    for i in folder[1:]:
        u = i.getiterator('bookmark')
        t = i.getiterator('title')
        for i in t:
            titles.append(i.text)
        for i in u:
            urls.append(i.get('href'))

print_submenu(folderTitle, titles, urls)

def rbuild_tree(id, bookmarks):
    '''Crée le menu openbox à partir d'une liste d'éléments'''

    tree = ET.parse(bookmarks).getroot()
    folderList = tree.getiterator('folder')

    print('<openbox_pipe_menu>')

    for folder in folderList:
        tree.remove(folder)
        print_folder_menu(folder)

    print_bookmarks(tree)

    print('</openbox_pipe_menu>')

def main():

    rbuild_tree(2, bookmarks)

if __name__ == '__main__':
    main()
```

1)

Voir : [midori](#)

2)

[Gestionnaire Openbox](#)

3)

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

4)

[vim](#)

From:

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

Permanent link:

http://debian-facile.org/doc:programmation:python:ob_bmarkmidori



Last update: **17/12/2015 19:52**