

Dans la série
Les tutoriels libres
présentés par le site FRAMASOFT

Internationalisation d'une application Python

Exemple avec SudokuBan

Logiciel:	Python
Auteur(s):	Python Software Foundation
Plateforme(s):	Indépendant
Version:	2.4
Licence:	Python Licence
Site:	http://www.python.org

Par

Leblanc Simon

20 mai 2006



Publié sous licence Art Libre

<http://artlibre.org/licence/lal/>

Framasoft

« Partir de Windows pour découvrir le libre... »

<http://www.framasoft.net>

Table des matières

<u>Pré-requis</u>	3
Généraux.....	3
Pour notre exemple.....	3
<u>Mise en place de l'internationalisation</u>	4
Bref mise en route.....	4
Extraction des chaînes de caractères.....	4
Traduction des chaînes de caractères.....	5
Modification de l'application.....	7
Création du fichier binaire de langue.....	9
<u>Remerciements et contact</u>	11
Remerciements.....	11
Contact.....	11

Pré-requis

Généraux

Étant donné que je n'ai jamais réussi à trouver les outils d'internationalisation de Python, je vais décrire la procédure avec les outils généralistes. Cela n'aura aucune incidence sur le bon fonctionnement de votre programme. Bien entendu, si vous souhaitez utiliser les outils Python, la méthode sera identique et ce tutoriel est donc également valable.

Il vous faut donc installer avant toutes choses l'utilitaire gettext qui comprend une suite d'applications destinées à l'internationalisation des applications (quelque soit le langage de programmation utilisé).

- Pour les personnes utilisant MS Windows, vous pouvez vous rendre sur ce [site internet](#).
- Pour les personnes utilisant GNU/Linux, vous pouvez télécharger gettext sur le [site officiel de GNU](#). Mais pensez également qu'il y a de grandes chances pour qu'il soit disponible dans les dépôt de votre distribution (via apt-get ou yum par exemple)

Pour notre exemple

Dans notre exemple nous traduirons le logiciel SudokuBan. Nous allons donc le récupérer sur le [site officiel](#).

Il faudra ensuite décompresser l'archive téléchargée.

On considérera dans ce tutoriel que nous avons décompressé l'archive de façon à obtenir la structure suivante:

- Pour GNU/Linux: /home/leviathan/sudokuban
- Pour MS Windows: C:\Documents and Settings\leviathan\sudokuban

Mise en place de l'internationalisation

Bref mise en route

L'internationalisation d'une application passe par quelques étapes indispensables. Nous devons tout d'abord extraire les chaînes de caractères à traduire, ensuite traduire les chaînes pour enfin modifier légèrement notre application afin qu'elle prenne en compte l'internationalisation.

Extraction des chaînes de caractères

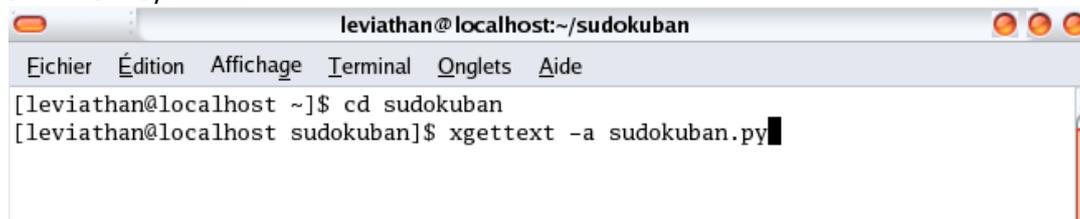
Nous devons tout d'abord extraire les chaînes de caractères dans un fichier messages.po afin de les traduire. Pour ce faire, nous allons utiliser l'utilitaire xgettext fourni grâce à la suite gettext.

Sous GNU/Linux et sous MS Windows la ligne de commande est identique:

```
xgettext -a sudokuban.py
```

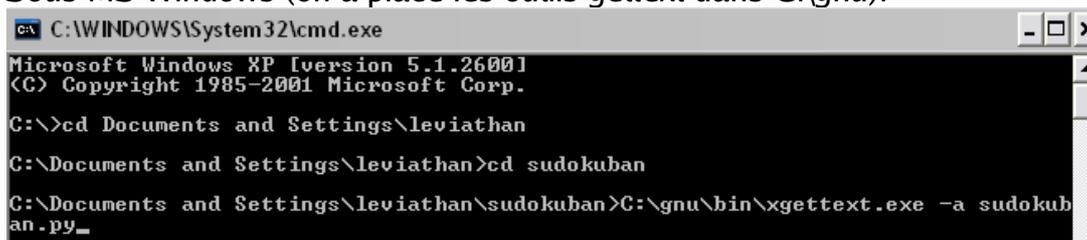
L'option « a » permet d'extraire toutes les chaînes de caractères sans exception. Je vous conseille de consulter l'aide de xgettext si vous souhaitez affiner l'extraction des chaînes de caractères.

- Sous GNU/Linux:



```
leviathan@localhost:~/sudokuban
Fichier Édition Affichage Terminal Onglets Aide
[leviathan@localhost ~]$ cd sudokuban
[leviathan@localhost sudokuban]$ xgettext -a sudokuban.py
```

- Sous MS Windows (on a placé les outils gettext dans C:\gnu):



```
C:\WINDOWS\System32\cmd.exe
Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\>cd Documents and Settings\leviathan
C:\Documents and Settings\leviathan>cd sudokuban
C:\Documents and Settings\leviathan\sudokuban>C:\gnu\bin\xgettext.exe -a sudokuban.py_
```

Cela a pour conséquence de vous créer un fichier nommé messages.po dans votre dossier « sudokuban ». Nous allons déplacer ce fichier, mais avant cela il faut créer l'arborescence qui va accueillir notre fichier. Cette arborescence est un standard, il ne faut donc pas la modifier¹.

Voici donc l'arborescence à créer dans notre dossier sudokuban:

- sous GNU/Linux:
/home/leviathan/sudokuban/locale/fr_FR/LC_MESSAGES/
- sous MS Windows:
C:\Documents and Settings\leviathan\sudokuban\locale\fr_FR\LC_MESSAGES\

Le dossier « fr_FR » indique que la langue est le français de France, il faudra donc adapter ce dossier en fonction de la langue et du pays que vous souhaitez ajouter.

Vous placerez donc le fichier messages.po dans le dossier nouvellement créé « LC_MESSAGES »

Traduction des chaînes de caractères

Voici en fait la partie la plus complexe du travail car il vous faut:

- Premièrement ne sélectionner que les chaînes de caractères utiles. Pour cela, soit vous regardez la source de l'application, soit vous vous rapprochez de l'auteur de l'application.
- Deuxièmement, bien il faut traduire correctement :-)

Pour cela ouvrez, votre fichier messages.po (soit avec un éditeur de texte tel que Kwrite ou Notepad2, soit avec un outil spécialisé tel que Kbabel ou PoEdit)

Je ne détaillerai ici que la procédure lors de l'ouverture avec un éditeur de texte. Votre fichier a cette apparence:

¹ Du moins à ma connaissance.



```

# SOME DESCRIPTIVE TITLE.
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT HOLDER
# This file is distributed under the same license as the PACKAGE package.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
#
#: sudokuban.py:391 sudokuban.py:474 sudokuban.py:547
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2006-05-20 20:59+0200\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: 8bit\n"

#: sudokuban.py:30
msgid "1.1pre"
msgstr ""

#: sudokuban.py:44
msgid "Create a new empty or random puzzle."
msgstr ""

```

Comment comprendre ce fichier?

Toute la première partie concerne le Copyright et les détails concernant le(s) traducteur(s), nous n'allons pas nous en occuper ici.

Ce qui nous intéresse ici c'est la partie à partir de la ligne « #: sudokuban.py:30 », voici donc la signification de ces lignes:

- « #: sudokuban.py:30 » : indique le numéro de ligne où est situé la chaîne de caractères qui a été extraite.
- « msgid "1.1pre" » : c'est le message qui a été extrait et qui sera utilisé par le programme lors de l'internationalisation. Il ne faut donc pas le modifier.
- « msgstr "" » : c'est ici que l'on va entrer la traduction du message contenu dans la ligne juste au dessus.

Voici donc un exemple de ce qui faudrait mettre dans ce fichier:

```
# SOME DESCRIPTIVE TITLE.
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT HOLDER
# This file is distributed under the same license as the PACKAGE package.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
#
#: sudokuban.py:391 sudokuban.py:474 sudokuban.py:547
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: SudokuBan\n"
"Report-Msgid-Bugs-To: contact@leblanc-simon.eu\n"
"POT-Creation-Date: 2006-05-20 20:59+0200\n"
"PO-Revision-Date: 2006-05-20 20:59+0200\n"
"Last-Translator: Leblanc Simon <contact@leblanc-simon.eu>\n"
"Language-Team: français <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Content-Transfer-Encoding: 8bit\n"

#: sudokuban.py:44
msgid "Create a new empty or random puzzle."
msgstr "Créer une nouvelle grille vide ou aléatoire."
```

On peut remarquer que le bloc correspondant à « msgid "1.1pre" » a disparu. Il a été supprimé car après étude du code de l'application, j'ai estimé qu'il ne fallait pas le traduire et donc je l'ai supprimé du fichier de traduction.

Il faut donc traduire tous les textes utiles pour ensuite passer à la partie suivante.

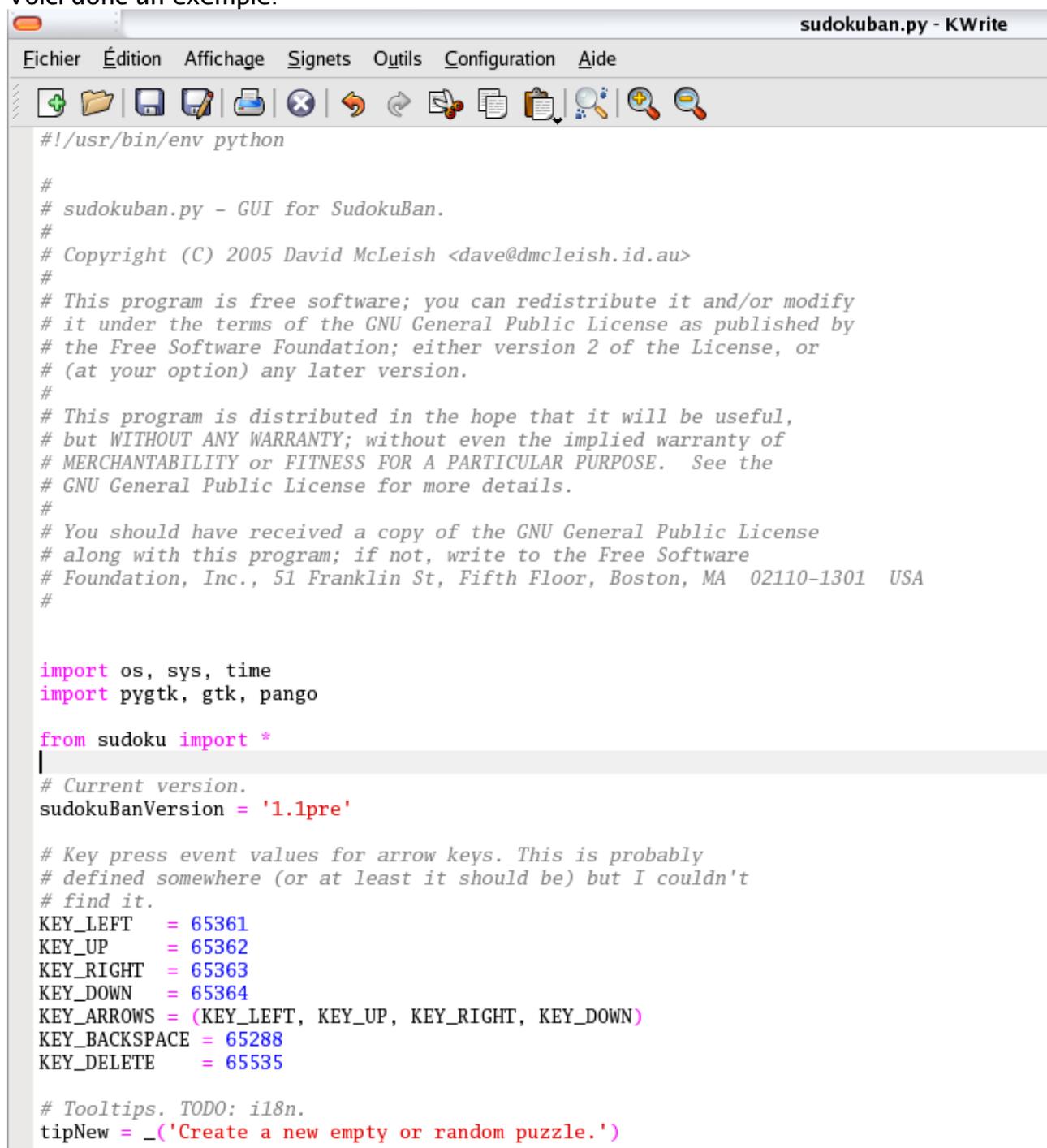
Modification de l'application

Nous avons fini la traduction du fichier « messages.po », il va donc falloir modifier légèrement notre application afin qu'elle prenne en compte nos traductions.

Pour ce faire, ouvrons nos deux fichiers « messages.po » et « sudokuban.py ». Le fichier « messages.po » nous indique les lignes où nous devons faire les changements dans notre fichier « sudokuban.py ». Eh oui, souvenez vous de la ligne « #: sudokuban.py:30 », le 30 indique que le message a été pris à la ligne 30 du fichier « sudokuban.py »

Quel est donc cette modification? Elle est très simple, elle consiste à ajouter les caractères `_()` autour des chaînes de caractères que l'on a traduit.

Voici donc un exemple:



```
#!/usr/bin/env python

#
# sudokuban.py - GUI for SudokuBan.
#
# Copyright (C) 2005 David McLeish <dave@dmcleish.id.au>
#
# 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.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
#

import os, sys, time
import pygtk, gtk, pango

from sudoku import *

# Current version.
sudokuBanVersion = '1.1pre'

# Key press event values for arrow keys. This is probably
# defined somewhere (or at least it should be) but I couldn't
# find it.
KEY_LEFT    = 65361
KEY_UP      = 65362
KEY_RIGHT   = 65363
KEY_DOWN    = 65364
KEY_ARROWS = (KEY_LEFT, KEY_UP, KEY_RIGHT, KEY_DOWN)
KEY_BACKSPACE = 65288
KEY_DELETE  = 65535

# Tooltips. TODO: i18n.
tipNew = _('Create a new empty or random puzzle.')
```

Nous n'avons pas traduit le « 1.1pre », nous ne devons donc pas ajouter les caractères spécifiques autour, mais la chaîne « Create a new empty or random puzzle. » a été traduite, il nous faut donc ajouter `_()` autour de la chaîne en anglais.

Une fois que vous avez fait toutes les modifications concernant les chaînes de caractères que vous avez traduit, il faudra ajouter un petit bout de code afin d'indiquer au programme

de prendre en compte l'internationalisation.

Voici ce code:

```
import gettext
pathname = os.path.dirname(sys.argv[0])
localdir = os.path.abspath(pathname) + "/locale"
gettext.install("messages", localdir)
```

Voici le code dans notre exemple:

```
import os, sys, time
import pygtk, gtk, pango
import gettext

from sudoku import *

#i18n
pathname = os.path.dirname(sys.argv[0])
localdir = os.path.abspath(pathname) + "/locale"
gettext.install("messages", localdir)

# Current version.
sudokuBanVersion = '1.1pre'

# Key press event values for arrow keys. This is probably
```

Création du fichier binaire de langue

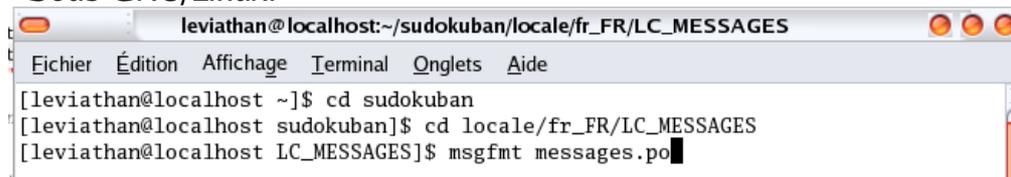
Voici enfin la dernière étape de notre parcours. Afin que le logiciel puisse lire le fichier de langue, il est nécessaire de transformer notre fichier « messages.po » en fichier binaire.

Pour ce faire, nous utiliserons un programme présent dans la suite gettext: msgfmt

Identique sous GNU/Linux et MS Windows, voici la ligne de commande:

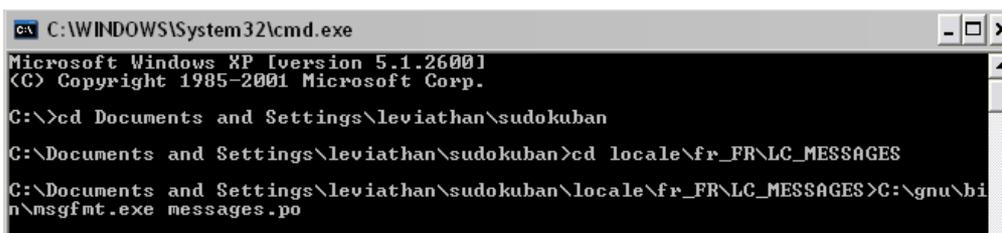
`msgfmt messages.po`

- Sous GNU/Linux:



```
leviathan@localhost:~/sudokuban/locale/fr_FR/LC_MESSAGES
Fichier Édition Affichage Terminal Onglets Aide
[leviathan@localhost ~]$ cd sudokuban
[leviathan@localhost sudokuban]$ cd locale/fr_FR/LC_MESSAGES
[leviathan@localhost LC_MESSAGES]$ msgfmt messages.po
```

- Sous MS Windows



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>cd Documents and Settings\leviathan\sudokuban
C:\Documents and Settings\leviathan\sudokuban>cd locale\fr_FR\LC_MESSAGES
C:\Documents and Settings\leviathan\sudokuban\locale\fr_FR\LC_MESSAGES>C:\gnu\bin\msgfmt.exe messages.po
```

Cette commande a pour conséquence la création d'un fichier « messages.mo » qui est l'équivalent binaire de « messages.po ». Vous pouvez donc maintenant lancer votre programme qui sera en français (ou dans la langue que vous avez choisi et qui est la langue de votre système)

Remerciements et contact

Remerciements

Je tiens tout d'abord à remercier David McLeish, créateur de l'excellent SudokuBan et qui a eu la bonne idée de placer son programme sous licence GNU/GPL.

J'adresse également mes remerciements à François Schnell qui m'a plus que guidé dans ma tentative d'internationalisation d'une application en Python.

Le logo illustrant la licence Art Libre est une oeuvre sous licence Art Libre que vous pouvez retrouver sur le site: <http://artlibre.org>

Contact

Vous pouvez me contacter à l'adresse suivante: contact@leblanc-simon.eu