TP d'introduction à Python
Webgalerie

Pierre-Marie de Rodat (LSE)

20 novembre 2012

Table des matières

1 Introduction

Envie de monter un site Web pour montrer vos photos de vacances et épater la galerie ? Python peut vous aider !

Génerer un site Web statique (avec de bêtes pages HTML et des images) est une tâche longue et répétitive… tout ce qu'il faut pour un script ! Le déroulement est très simple :

- Lister les images, par exemple toutes celles présentes dans un dossier spécifique.
- Générer une page Web par image.
- Générer une page d'index pour présenter l'ensemble des images.

C'est parti !

2 Lister les fichiers d'un dossier

Le prérequis ici est de lister les fichiers d'un dossier. Ensuite, il suffira de filtrer cette liste pour n'en garder que les images (c'est-à-dire les fichiers dont le nom se termine par `png`, `jpg`, etc.).

Pour lister les fichiers d'un dossier, on utilise le module `os` de Python [1] :

```
import os

# This is going to print every filename in the
# C:/ directory... straightforward!
for filename in os.listdir('C:/'):
    print filename
```

[2]

Une autre tâche utile est de se déplacer dans l'arborescence des fichiers : si votre programme tourne dans le dossier `machin`, alors créer un fichier `truc.txt` se fera à l'emplacement `machin/truc.txt`. Pour se déplacer, on utilise aussi le module `os` :

```
import os

# Move to the C:/foo directory
os.chdir('C:/foo')

# Create the file C:/foo/bar.txt and write
#'Hello world!' in it, then close it.
f = open('bar.txt', 'w')
f.write('Hello world!')
f.close()
```

[3]

3 Filtrer les images

Généralement, les personnes commençant le Python appliquent l'algorithme que l'on retrouve dans beaucoup de langages impératifs :

1. `http://docs.python.org/library/os.html`
2. Voir le fichier source tp-webgalerie-listdir.py
3. Voir le fichier source tp-webgalerie-chdir.py

```
filenames = [
    'webpage.html', 'document.odt', 'song.wav',
    'picture.jpeg', 'picture2.JPG',
    'icon.gif', 'logo.png',
]

# First, initialize an empty list
images = []

# Add only the interesting filenames
for filename in filenames :
    # Get the extension
    extension = filename.split('.')[-1].lower()
    # Append it if it is an image
    if extension in ('png', 'jpg', 'jpeg'):
        images.append(filename)

# Done!
```

4

Avec Python, il est possible de faire plus élégant (et accessoirement plus efficace) en utilisant une *list comprehension* :

```
filenames = [
    'webpage.html', 'document.odt', 'song.wav',
    'picture.jpeg', 'picture2.JPG',
    'icon.gif', 'logo.png',
]
extensions = ('png', 'jpg', 'jpeg')

images = [
    filename
    for filename in filenames
    if filename.split('.')[-1].lower() in extensions
]

# 'images' contains 'picture.jpeg', 'picture2.JPG',
# 'icon.gif', 'logo.png'
```

5

4. Voir le fichier source tp-webgalerie-nofilter.py
5. Voir le fichier source tp-webgalerie-listcomp.py

4 Générer les pages Web

On arrive au centre du sujet : générer le site Web ! Rien de bien compliqué : il suffit à chaque fois d'ouvrir un fichier, d'écrire ce qui nous intéresse (un peu d'HTML), de fermer le fichier… et de passer au suivant !

Comme dit plus haut, il suffit tout d'abord de créer les pages Web contenant une image, puis de créer celle qui liste les images. Voyez plutôt la structure :

```
images = []
# ... Fill the images ...

def get_name(image):
    # Get the name of the image (stripping the
    # file extension)
    # name = TODO
    return name

# Then for every image, create its own Web page...
for (i, image) in enumerate(images):
    name = get_name(image)

    f = open('page-%s.html', 'w')
    f.write('''\
<html>
<body>
    <h1>%(name)s</h1>
    <img src="%(image)s"><br>
''' % {'name' : name, 'image' : image})

    # Add links for the previous page, the next
    # page, and the home page.
    def add_link(number):
        name = get_name(images[number])
        f.write('<a href="page-%s.html">%s</a><br>' % \
            (num, name)
        )

    if i > 0 :
        add_link(i - 1)
    if i < len(image) - 1 :
```

```
        add_link(i + 1)

    f.write('''\
</body>
</html>
''')
    f.close()

# And finally create a home page to list every image.
f = open('index.html', 'w')

# TODO: write the list of the images

f.close()
```

6

Et *voilà* !

5 Bonus : et après ?

Le site que vous obtenez n'est peut-être pas très follichon. Qu'à cela ne tienne ! De nombreuses améliorations sont possibles, chacune d'entre elle étant à votre portée, n'ayant de limite que votre imagination :

- Refaire le design du site (si tant est qu'il y en avait un !) : mettre de la couleur, de la mise en forme avec des CSS ;
- Chercher une bibliothèque Python permettant de lire les données EXIF[7] de vos photos. Vous pourrez ainsi avoir un titre plus agréable, géolocaliser vos prises de vues (et les placer sur un plan du monde), etc.
- Rajouter un système de commentaires (par exemple en associant à chaque photo un fichier texte la décrivant : `l-hotel.jpg` et `l-hotel.txt` par exemple.

À vos éditeurs !

6. Voir le fichier source tp-webgalerie-generation.py
7. `http://en.wikipedia.org/wiki/Exchangeable_image_file_formath`