

Rapport

NDIONE Mouhamed, HADJAB Hanane

Scripting : Python vs Bash

1 Introduction :

Un script est un fichier contenant une série d'instructions à exécuter par un programme externe. Ce programme externe est en l'occurrence un interpréteur de commande. Les scripts permettent d'automatiser des tâches.

Il existe plusieurs types de langage de script, Bash et python font partie des plus utilisés du fait de leurs caractéristiques :

Bash est l'interpréteur par défaut dans la plupart des distributions linux.

Python, un langage portable et à usage généraliste.

Nous allons voir dans ce qui suit les points forts ainsi que les points faibles d'utiliser Bash au lieu de python et vice versa.

2 Le langage Bash

2.1 Historique

Bash est l'acronyme de bourne again shell faisant référence ainsi au premier shell qui a été développé par Steve Bourne.

Il s'agit d'un langage de programmation shell et est l'interpréteur de commande par défaut dans la plupart des distributions linux.

Créé par Brian fox en 1989, Bash apporte des améliorations du k-shell et du c-shell.

2.2 Utilisation

Bash est présent dans pas mal de domaines, notamment dans la sécurité informatique, dans l'administration système, dans le déploiement d'applications, etc.

Sa syntaxe permet facilement de coupler des commandes et utilitaires systèmes (tels que sed, awk, cut, grep, ...) pour faire des scripts complexes tout en n'ayant pas besoin d'installer des bibliothèques tierces.

3 Le langage Python

3.1 Historique

Python est un langage de programmation de haut niveau à usage général largement utilisé.

Il a été conçu par Guido van Rossum en 1991 et développé par la Python Software

Foundation. Il a été développé pour mettre l'accent sur la lisibilité du code, et sa syntaxe permet aux programmeurs d'exprimer des concepts en quelques lignes de code.

3.2 Librairies et modules pour le scripting sous python

Le langage python dispose d'une grande communauté, beaucoup de librairies sont disponibles pour réaliser des tâches complexes. Ce qui fait sans doute de python, un des langages les plus préférés des programmeurs lorsqu'ils font des scripts.

Ci dessous, une petite liste de librairies utilisées dans le "scripting" :

- **Argparse:**
Analyseur d'options de ligne de commande, d'arguments et de sous-commandes, elle génère des message d'aide et d'utilisation et génère des erreurs dans le cas où les arguments ne sont pas valides.
- **BeautifulSoup:**
Chercher des éléments dans un fichier html par exemple peut parfois être fastidieux lorsqu'on souhaite utiliser les regex. La bibliothèque **BeautifulSoup** réponds à ce problème.
Bibliothèque permettant l'extraction des données de fichier HTML et XML et elle fournit plusieurs méthode de navigation de modification de l'arbre d'analyse
- **OS modules:**
Permet d'effectuer des opérations lié au Système d'exploitation et elle fonction sur n'importe quel système d'exploitation, si on veut par exemple avoir le nom de répertoire courant on a qu'a faire `os.getcwd()`, et `os.chdir()` pour changé le répertoire courant ... etc
- **Sys module:**
Ce module donne accès à certaines variables utilisées ou maintenues par l'interpréteur et à des fonctions qui interagissent fortement avec l'interpréteur. (`sys.stdin`, `sys.stdout`, `sys.stderr`. etc)
- **Subprocess:**
Ce module exécute des commandes système ainsi d'exécuter des programmes externe sur notre code python exp: `subprocess.run(["ls", "-l"])`

4 Les points forts et faibles de Bash et de Python

4.1 Bash :

Les points forts:

L'utilisation des pipes ("|") est sans doute le point le plus important sous **Bash**.

En effet, les pipes permettent de combiner des commandes et les sorties précédentes sont les entrées de ceux qui en suivent.

```
# Entrée
:
Area:          547,030 square kilometres
Capital:       Paris
Country (District): France
Population:    64,768,389
Tld:           .fr
'

sed "s/[^$$(tail -1 pays.txt)]$/;/g" pays.txt | awk -F ':' '{print $NF}' | xargs >>result.csv

#sortie dans le fichier result.csv
# 547,030 square kilometre; Pari; Franc; 64,768,38; .fr
```

La **syntaxe de bash** permet de combiner par exemples des commandes, des fonctions, des expressions régulières etc. Ce qui permet en **quelques lignes de codes seulement**, d'écrire un programme assez complexe.

Utilisation directe des **commandes** et **utilitaires systèmes limite** l'appel à des librairies tierces. Et enfin Bash est l'**interpréteur de commande par défaut** dans la plupart des distributions linux

Les points faibles:

Un code écrit en Bash est assez **difficile à déboguer** parfois et **n'est portable que** sur certaines distributions linux.

un espace dans la déclaration d'une variable fait planter tout votre programme, sa syntaxe requiert du temps à maîtriser et **nécessite beaucoup de rigueur** dans le code à défaut de se tromper, ce qui **complique aussi la maintenance**.

Souvent aussi un script bash est utilisé pour **un usage spécifique** à faire une chose seulement, le code est assez brute dans ce cas.

4.2 Python:

Les points forts:

Python est un langage à **usage généraliste** et **portable**. Sa syntaxe et les noms de ses fonctions font qu'un utilisateur peut **intuitivement** savoir ce que le code fait.

Il dispose aussi de **plusieurs librairies** qui simplifient des tâches complexes aux programmeurs. Et enfin il est aussi un langage **multi-paradigme** et **maintenable facilement** grâce aux nombreux outils qu'ils intègrent

Les points faibles:

Un programme écrit sous python peut nécessiter l'**appel à plusieurs librairies tierces** ce qui peut rendre son **exécution parfois lente** et **consomme beaucoup de mémoires**.

Ses bibliothèques externes peuvent aussi contenir **des failles de sécurité** qui peuvent passer sous les radars de ses outils d'analyse.

Comparé à Bash, le code écrit en python est souvent beaucoup plus long.

Et enfin la gestion des dépendances peut être compliquée d'où l'expression "**Dependency hell**" (l'enfer des dépendances) peut se sentir facilement dans un programme python.

Conclusion:

Aucun langage de programmation n'est parfait. Il n'existe même pas un langage meilleur que d'autres ; il n'y a que des langages en adéquation ou peu conseillés pour des buts particuliers (herbert mayer)

Il existe plein d'autres langages de scripting. On peut citer par exemple JavaScript, Php, Ruby, Perl, Powershell et Groovy.

Ce qui fait la grande différence entre Python et Bash réside dans le fait que Bash est un langage à usage spécifique alors que Python est un langage à usage général et dispose de plusieurs bibliothèques qui peuvent faciliter des tâches complexes.

Références :

- <https://docs.python.org/fr/3.5/howto/argparse.html>
- <https://github.com/geekcomputers/Python>
- <https://askanydifference.com/difference-between-python-and-bash/>
- https://frederic-lang.developpez.com/tutoriels/linux/prog-shell/?page=page_2
- <https://abs.traduc.org/abs-fr/>
- <https://devhints.io/bash>
- <https://tiswww.case.edu/php/chet/bash/bashtop.html>