

Licence 2 Informatique - Pierre Pompidor Initiation aux scripts Système en Python

Le but de ce TP est de vous initier au langage Python qui est un langage de plus en plus utilisé pour écrire des **scripts système** tout en vous faisant manipuler **emacs**.

Vous allez donc essayer d'écrire dans un premier temps un script qui dit "bonjour" (notamment pour ceux qui n'auraient jamais eu une poupée qui faisait cela (je l'ai toujours pour ma part regretté)).

Puis vous allez améliorer pas à pas ce script.

Et enfin vous allez écrire un "vrai" script qui suivant l'option (a ou p) :

- soit liste les **alias** mémorisés dans le fichier de configuration `.bashrc` ;
- soit indique les répertoires ajoutés à la variable d'environnement `PATH`.

Ecrivez un script Python qui affiche "Bonjour !"

La première étape est d'écrire un "Hello world", donc saisissez le code suivant sous emacs (fichier `bonjour_v1.py`) :

```
#!/usr/bin/python
print "Bonjour !"
```

Pour l'exécuter n'oubliez pas de lui donner le droit d'exécution, puis :

- si vous avez rajouté le répertoire courant (`.`) dans votre `PATH` :

```
bonjour_v1.py
```

- sinon (mais c'est déshonorant) : `./bonjour_v1.py`

Modifiez ce script pour qu'il affiche "Bonjour Machin !"

Maintenant, passons un paramètre au script ; appel :

```
bonjour_v2.py VotrePrenom
```

Ce paramètre sera stocké dans un tableau de paramètre appelé `argv` (accessible à condition que le module `sys` soit inclus), et dont le premier élément (c'est à dire le premier vrai paramètre passé au script) est appelé `sys.argv[1]` (`sys.argv[0]` est le nom du script).

```
#!/usr/bin/python
import sys
print "Bonjour ", sys.argv[1], "!"
```

Maintenant à vous de jouer :

- Nous pourrions vouloir contrôler la présence du paramètre.
→ nous pouvons tester le nombre d'éléments d'une liste par la fonction `len(liste)`
- Nous voulons faire afficher une liste quelconque de paramètres.
→ nous pouvons mettre en place une boucle automatique par `for element in sys.argv :`
(la variable `element` étant instanciée par chaque élément de la liste).

Et puis passons aux choses sérieuses, écrivez le script `waouhh.pl` qui liste vos alias ou indique les répertoires ajoutés au PATH

Attention cela suppose la maîtrise de l'application des expressions régulières en Python... L'algorithme :

```
Si le paramètre est 'a' ou 'p' :
  Si le fichier .bashrc existe
    ouvrir le fichier
    recopier le contenu du fichier dans un tableau en mémoire centrale
  Si le paramètre est 'a'
    Pour toutes les lignes du fichier
      filtrer toutes les lignes contenant alias
      afficher la ligne de commande ... est devenue ...
    FinPour
  Sinon
    Pour toutes les lignes du fichier
      filtrer toutes les lignes contenant PATH=
      afficher les répertoires supplémentaires
    FinPour
  FinSi
  fermer le fichier
FinSi
Finsi
Sinon Message d'alerte
```

Puis pensez à une gestion plus souple des options qui permettrait par exemple d'exécuter `waouhh.pl -ap` ou `waouhh.pl -pa`.