

Partie Pierre Pompidor

mercredi 3 juin 2009 - 8h30-10h30

Compréhension d'un script système écrit en Python : 10 points

Pour expliquer ce que fait le script suivant, **spécifiez un fichier d'exemple sur lequel ce script va fonctionner et donnez son résultat sur celui-ci**. Par ailleurs pour **une ligne du fichier**, donnez également la signification de toutes les expressions régulières.

- Remarque 1 : `\\` permet la création d'un anti-slash dans une chaîne de caractères ;
- Remarque 2 : `.?*?` signifie la plus petite chaîne de caractères avant ce qui suit ;
- Remarque 3 : la méthode `rstrip('\n')` supprime `\n` à la fin de la chaîne sur laquelle elle s'applique ;
- Remarque 4 : `break` provoque la sortie de la boucle la plus proche.

```
#!/usr/bin/python
import sys, re
radicaux = ['if', 'for', 'while', 'def']
motifs = {'(':')', '[':']', '{':'}'}

def analyse(ligne) :
    for motif in motifs.keys() :
        if len(motif) == 1:
            chaine = "(\\\\"+motif+".*?\\\\"+motifs[motif]+")(.*)"
            chaine2 = "\\\\"+motif
        else :
            chaine = "("+motif+".*?"+"motifs[motif]+")(.*)"
            chaine2 = motif
    copie_ligne = ligne.rstrip('\n')
    while 1 :
        res = re.search(chaine, copie_ligne)
        if res :
            if len(res.group(2)) == 0 :
                copie_ligne = ""
                break
            copie_ligne = res.group(2)
        else :
            break
    res2 = re.search(chaine2, copie_ligne)
    if res2 : print ligne

if len(sys.argv) > 1 :
    fd = open(sys.argv[1])
    for radical in radicaux : motifs["^\\s*" + radical + "\\s"] = ':'
    lignes = fd.readlines()
    for ligne in lignes : analyse(ligne)
```