

**Commentaires sur le DST d'informatique décembre 2023**

- Dans le sujet est noté : *Des rappels sur les chaînes de caractères sont donnés en Annexe.*

Si on regarde dans l'annexe, on voit toutes les fonctions ou méthodes susceptibles d'être utilisées pour traiter le DST.

Il n'est donc pas nécessaire d'utiliser les méthodes plus complexes comme `.split()` ou `.strip()`. De plus, on utilise ces méthodes que lorsqu'on les connaît parfaitement, c'est-à-dire lorsqu'on connaît exactement ce que cela renvoie.

Dans la question 1, le sujet explique exactement ce que l'on attend : **Le programme devra parcourir la chaîne de caractères représentant le numéro de sécurité sociale en supprimant les caractères d'espace, puis transformer chaque caractère en entier et stocker chaque entier dans une liste.** Donc on code exactement ce que l'on nous dit de coder. Pas d'inventivité!!!

Pourquoi? `int("2 123 4")` donne une erreur car la chaîne de caractère contenant à la fois des chiffres et des espaces ne peut être convertie en `int`.

- En annexe, on rappelle comment convertir un `int` en `str`. On dit par exemple que `str(290)` vaut `"290"`. Il faut bien stocker cette valeur dans une variable!!! Par exemple `a = str(290)` : dans la variable `a` il y a `"290"`.
- Certains élèves ne savent pas appeler une fonction. Il faut revoir le TP sur les fonctions avec tous les premiers exemples.

Je rappelle que si une fonction renvoie un résultat, celui-ci doit être stocké dans une variable.

*Exemple :*

```
1 def f(a):  
2     return a + 10  
3  
4 b = f(a) #je stocke le résultat de la fonction dans la variable b
```

Je rappelle qu'une fonction qui renvoie un résultat doit contenir le mot clé `return` dans le corps de la fonction.

- On ne réécrit pas le code d'une fonction déjà écrite précédemment. On appelle directement la fonction : c'est une des utilités d'une fonction!!!

Par exemple, dans la question 5 où on veut le numéro de sécurité sociale complet, on va utiliser la fonction `clef` écrite à la question 4.

- Il est important de savoir le type des variables qu'on utilise. Suivant le type, on peut faire certaines opérations et uniquement certaines opérations.

Par exemple, la méthode `pop` s'applique à une liste et uniquement à une liste.

Ainsi, vous devez toujours vous interroger sur le type des variables que vous utilisez.

- Pour enlever un élément d'une liste, il y a deux méthodes `pop` et `remove`.

La méthode `remove` prend en argument l'élément que l'on veut enlever. Mais que se passe-t-il s'il y a 2 fois le même élément dans la liste? Cette méthode enlève la première occurrence de cet élément. Donc si on veut enlever un élément bien particulier, il est nettement préférable d'utiliser la méthode `pop` qui prend en argument l'indice de l'élément à enlever. Ainsi, il n'y a pas de confusion possible.

- Pour modifier un élément d'une liste, par exemple le 2e élément d'une liste en 0, on utilise l'instruction `liste[1] = 0`.

On peut écrire :

```
1 liste.pop(1)  
2 liste.insert(1, 0)
```

Mais **on ne fera jamais cela**. Pourquoi ? Parce que derrière la méthode `insert` se cachent plusieurs opérations de décalage pour insérer la valeur 0. Ces opérations prennent plus de temps que d'écraser la valeur grâce à l'instruction `liste[1]=0`.