Bonne Pratique 2 : Assertion : Valider les données

1/ Présentation

L'assertion est un mécanisme de programmation défensive qui vise à se protéger contre des données inadaptées à un programme.

Un programme est prévu pour obtenir un certain résultat à partir de certaines données. La description des **données nécessaires** est faite dans la **spécification**.

- Un utilisateur indélicat pourrait proposer des données inadaptées au programme, entrainant des comportements non prévus, et donc non gérées. Un programme bien construit doit intégrer la gestion de ce cas de figure.
- Une « assertion informatique » est une instruction qui permet au programme d'indiquer une erreur dès que les données sont inadéquates, avant d'exécuter la première instruction ;

2/ Mise en place

- Une fois la spécification faite proprement (les données sont donc bien caractérisées, leur type et leurs propriétés aussi), le programmeur peut rédiger les assertions à vérifier pour une bonne exécution du programme.
- Ces assertions doivent donc être rédigée après le « doc string », et avant les instructions nécessaires au programme.
- Si l'assertion n'est pas vérifiée, une erreur « AssertionError » est indiquée à l'utilisateur.

(une assertion n'est utile que si le programme doit gérer des données non maitrisées)

3/ Rédaction en langage python

Les assertions doivent être rédigées en langage python sous la forme de condition logique (expression aboutissant à un booléen).
Elles sont précédées du mot clé `assert`.
Elles peuvent être suivies d'une chaine de caractères qui sera affiché à l'utilisateur (faisant office de message d'erreur).

`assert` expression
ou
`assert` expression, `'commentaire sur l'erreur'`

Si l'expression est fausse (`False`), le programme s'arrête et est affiché le message d'erreur : `AssertionError: commentaire sur l'erreur`

4/ Exemple :

```
"""
Calcul de l'avancement d'un trajet en pourcentage: l'utilsateur définit la
distance totale (totale) à parcourir, et la distance parcourue (dist)
"""
totale=float(input("distance à parcourir ?"))
dist=float(input("distance parcourue ?"))
assert totale > 0, 'La distance à parcourir doit etre strictement positive '
assert 0 <= dist , 'La distance parcourue doit etre positive '
assert dist <= totale , 'La distance parcourue doit etre inferieure à celle à parcourir'
print(dist / totale * 100)
```