NSI · TERMINALE / EXERCICES CORRIGÉS

Modularité et mise au point

Astrolab Cours · astrolabcours.fr

Ex. 1 Écrire des tests ●○○ Application

Écrire trois tests avec `assert` pour une fonction `maximum(t)` qui renvoie le plus grand élément d'une liste non vide, dont un cas limite.

Indice

Penser à une liste d'un seul élément et à des nombres négatifs.

Des tests

```
assert maximum([3, 8, 1]) == 8
assert maximum([5]) == 5
assert maximum([-4, -2, -9]) == -2
```

Le dernier test attrape une erreur classique : initialiser le maximum à 0.

Ex. 2 Spécifier une fonction ●●○ Méthode

Écrire la docstring de la fonction `indice(t, x)` qui renvoie l'indice de la première occurrence de `x` dans la liste `t`, ou -1 si `x` est absent.

Indice

Préciser les entrées, la sortie et le cas particulier.

La docstring

```
def indice(t, x):
    """Renvoie l'indice de la première occurrence de x dans la liste t.
    Renvoie -1 si x n'est pas dans t.
    t n'est pas modifiée."""
```

Ex. 3 Le test trompeur ●●○ Piège

Un élève teste sa fonction :

© Astrolab Cours · astrolabcours.fr/nsi/terminale/modularite-et-mise-au-point/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

```
def maximum(t):
    m = 0
    for x in t:
        if x > m:
            m = x
    return m

assert maximum([2, 7, 4]) == 7
```

Le test passe. La fonction est-elle juste ?

Indice

Essayer avec une liste de nombres négatifs.

L'erreur

`maximum([-3, -1])` renvoie 0, qui n'est même pas dans la liste.

La correction

Initialiser `m = t[0]` et ajouter un test avec des nombres négatifs.

Ex. 4 Une saisie robuste ●●● Type bac

Écrire une fonction `demander_note()` qui redemande la saisie tant que l'utilisateur n'a pas tapé un entier entre 0 et 20, en utilisant `try` et `except`.

Indice

Une boucle while True, un return quand la saisie est bonne.

L'idée

`int()` lève une `ValueError` si le texte n'est pas un entier ; on vérifie ensuite l'intervalle.

La fonction

```
def demander_note():
    while True:
        try:
            n = int(input("Note sur 20 : "))
            if 0 <= n <= 20:
                return n
            print("Entre 0 et 20.")
        except ValueError:
            print("Il faut un nombre entier.")
```

© Astrolab Cours · astrolabcours.fr/nsi/terminale/modularite-et-mise-au-point/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

© Astrolab Cours · astrolabcours.fr/nsi/terminale/modularite-et-mise-au-point/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits