NSI · PREMIÈRE / EXERCICES CORRIGÉS

# Fonctions, spécification et tests

**Astrolab Cours** · astrolabcours.fr

## Ex. 1 Suivre une boucle ●○○ Application

Que vaut `s` à la fin ?

```
s = 0
for i in range(1, 5):
    s = s + i
```

*Indice*

`range(1, 5)` donne 1, 2, 3, 4.

**Les tours**

s vaut 1, puis 3, puis 6, puis 10.

**La réponse**

`s` vaut 10.

## Ex. 2 Écrire une fonction ●●○ Méthode

Écrire une fonction `est_bissextile(a)` qui renvoie True si l'année `a` est bissextile : divisible par 4 mais pas par 100, sauf si elle est divisible par 400. Donner trois tests.

*Indice*

Traduire la règle avec and, or et %.

**La fonction**

```
def est_bissextile(a):
    """Renvoie True si l'année a est bissextile."""
    return (a % 4 == 0 and a % 100 != 0) or a % 400 == 0
```

© Astrolab Cours · astrolabcours.fr/nsi/premiere/bases-de-la-programmation/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

**Les tests**

Un cas de chaque sorte :

```
assert est_bissextile(2024)
assert not est_bissextile(1900)
assert est_bissextile(2000)
```

## Ex. 3 Le None mystérieux ●●○ Piège

Pourquoi `total` vaut-il `None` ?

```
def double(x):
    print(2 * x)

total = double(5) + 1
```

*Indice*

Que renvoie `double` ?

**L'erreur**

`double` affiche 10 mais ne renvoie rien, donc `None`. Et `None + 1` provoque même une erreur.

**La correction**

Remplacer `print(2 * x)` par `return 2 * x`.

## Ex. 4 Une boucle while ●●● Type bac

Écrire une fonction `nb_divisions(n)` qui renvoie combien de fois on peut diviser l'entier `n ≥ 1` par 2 (division entière) avant d'arriver à 1. Justifier qu'elle termine.

*Indice*

Une boucle while et un compteur.

**La fonction**

```
def nb_divisions(n):
    c = 0
    while n > 1:
        n = n // 2
        c = c + 1
    return c
```

© Astrolab Cours · astrolabcours.fr/nsi/premiere/bases-de-la-programmation/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

**La terminaison**

`n` est un entier qui diminue strictement à chaque tour tant que $n > 1$ : il finit par valoir 1, la boucle s'arrête.

© Astrolab Cours · astrolabcours.fr/nsi/premiere/bases-de-la-programmation/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits