

Correction des exercices bonus 2

Voici une proposition de correctif pour les exercices bonus 2. Le code proposé ici n'est pas l'unique façon de faire, donc si vous n'avez pas exactement la même chose, ce n'est pas grave !

1. Le but ici était d'utiliser `def` pour définir une fonction, et l'appeler ensuite.

```
def print_integer():  
    print(5)  
  
print_integer()
```

Le résultat affiché devrait être 15 dans ce cas-ci.

En effet, $0 + 1 + 2 + 3 + 4 + 5 = 15$

2. La subtilité dans l'exercice était qu'il fallait non seulement introduire une boucle, mais aussi une variable auxiliaire pour conserver le résultat, tout en faisant attention aux bornes de la boucle.

```
def print_integer_sum():  
    result = 0  
    for i in range(5+1):  
        result = result + i  
    print(result)  
  
print_integer_sum()
```

Faites bien attention au `range(5+1)`, le `+1` veillant bien à ce que 5 soit compris dans le range utilisé ! Souvenez-vous, `range(5)` aurait généré la liste `[0, 1, 2, 3, 4]`.

3. Ici, le changement était moins important, il fallait donc simplement ajouter l'argument à la fonction, l'utiliser dans le code, et le passer comme paramètre.

```
def print_integer_sum(number):
    result = 0
    for i in range(number+1):
        result = result + i
    print(result)

print_integer_sum(5)
```

4. Vous vous souvenez de `input` ? Et des conversions de données ?

```
def print_integer_sum(number):
    result = 0
    for i in range(number+1):
        result = result + i
    print(result)

n = input("Entrez un nombre : ")
print_integer_sum(int(n))
```

Il fallait donc non seulement garder dans une variable la réponse de l'utilisateur, mais aussi transformer le string en entier via `int()`. Notez que, si vous essayez d'entrer autre chose qu'un nombre entier, vous obtiendrez une `ValueError`. Nous verrons dans le cours de Python plus avancé comment prendre en charge ce type de cas.

5. La solution consistait ici à introduire un `return` au lieu du `print`, et de déplacer le `print` dans la partie "code de test" du script.

```
def integer_sum(number):
    result = 0
    for i in range(number+1):
        result = result + i
    return result

n = input("Entrez un nombre : ")
res = integer_sum(int(n))
print(res)
```

6. **BONUS** : Voici 2 exemples de façons dont vous pouvez résoudre l'exercice. Il en existe de nombreuses autres, plus ou moins concises, et plus ou moins efficaces.

```
def integer_sum_1(number):
    result = 0
    if number > 0:
        for i in range(number+1):
            result = result + i
    else:
        for i in range(number, 0):
            result = result + i
    return result

def integer_sum_2(number):
    result = 0
    for i in range(min(number, 0), max(0, number+1)):
        result = result + i
    return result

n = input("Entrez un nombre : ")

res1 = integer_sum_1(int(n))
print(res1)

res2 = integer_sum_2(int(n))
print(res2)
```

Voilà, en espérant que le reste de votre formation se passe bien en attendant de réattaquer un peu de Python :)