

# Apprivoiser Python

Cette fiche de travail a été réalisée par mes collègues du Lycée. Merci à eux !

## 1. Les viennoiseries

### La situation

Une boulangerie propose à ses clients de passer leurs commandes du dimanche en ligne pour ainsi bénéficier de réductions intéressantes. Votre travail va consister à écrire un programme qui va établir le montant à payer par un client en fonction du nombre de croissants et de pains au chocolat qu'il souhaite commander. D'un dimanche à l'autre, les réductions proposées sont différentes.

### 1.1. Première situation

Le programme qui permet de calculer le montant de la facture est écrit ci-dessous :

```
def facture_1(croissants : int, pains : int) -> float :
    """
    Fonction calculant le montant de la facture totale pour un achat de croissants et
    de pains au chocolat. Les arguments sont :
    - croissants : le nombre entier de croissants commandés ;
    - pains : le nombre entier de pains au chocolats commandés ;
    """
    total = 0.8 * croissants + 0.9 * pains
    return total
```

1. Créer une arborescence `Maths/Python/boulangerie` dans votre dossier personnel.
2. Ouvrir Thonny.
3. Recopier et sauvegarder le programme ci-dessus dans un fichier un nouveau fichier vide nommé `boulangerie.py` dans le dossier `boulangerie`. **Puis exécutez le programme grâce à la flèche verte ou bien grâce à la touche ++F5++.**
4. Dans la console, exécuter la ligne `facture_1(3,4)` et interpréter le résultat.
5. Quel est le prix d'un croissant ? d'un pain au chocolat ?
6. Compléter alors le tableau ci-dessous :

|                   |   |   |   |   |   |
|-------------------|---|---|---|---|---|
| Croissants        | 3 | 2 | 1 | 2 | 5 |
| Pains au chocolat | 4 | 0 | 2 | 2 | 8 |
| Montant total     |   |   |   |   |   |

### 1.2. Deuxième situation

Si la nombre total de croissant est supérieur ou égal à 10, chaque croissant est facturé 0,50 €.

1. Copier la fonction ci-dessous et **la compléter** dans le fichier `boulangerie.py` (n'oubliez pas de sauvegarder puis relancer) :

```
def facture_2(croissants : int, pains : int) -> float :
    """
    Fonction calculant le montant de la facture totale pour un achat de croissants et
    de pains au chocolat, en prenant en compte une réduction si le nombre
    de croissants dépasse 10. Les arguments sont :
    - croissants : le nombre entier de croissants commandés ;
    - pains : le nombre entier de pains au chocolats commandés ;
    """
```

```

if ... >= 10 :
    total = ...
else :
    total = ...
return total

```

2. Compléter alors le tableau ci-dessous, en utilisant la fonction `facture_2` :

|                   |    |   |    |    |    |
|-------------------|----|---|----|----|----|
| Croissants        | 12 | 8 | 6  | 10 | 15 |
| Pains au chocolat | 5  | 5 | 10 | 2  | 0  |
| Montant total     |    |   |    |    |    |

### 1.3. Troisième situation

Dans cette troisième situation, le montant de la facture se verra attribuer une remise de 20% si le montant dépasse 8 €.

1. Combien devra-t-on payer pour 5 croissants et 8 pains au chocolat ?
2. Recopier et compléter la fonction suivante (n'oubliez pas de sauvegarder puis relancer) :

```

def facture_3(croissants : int, pains : int) -> float :
    """
    Fonction calculant le montant de la facture totale pour un achat de croissants et
    de pains au chocolat, en prenant en compte une réduction de 20% si le montant
    total dépasse 8€. Les arguments sont :
    - croissants : le nombre entier de croissants commandés ;
    - pains : le nombre entier de pains au chocolats commandés ;
    """
    total = ... * croissants + ... * pains
    if ... >= ... :
        total = ...
    return total

```

3. Exécuter dans la console la ligne `facture_3(5, 8)`, et interpréter.
4. Compléter le tableau avec la fonction `facture_3` :

|                   |   |   |    |    |    |
|-------------------|---|---|----|----|----|
| Croissants        | 5 | 3 | 6  | 10 | 0  |
| Pains au chocolat | 8 | 2 | 10 | 0  | 12 |
| Montant total     |   |   |    |    |    |

#### Une faille de sécurité ! (Facultatif)

Dans toutes les fonctions, nous avons commis des erreurs de sécurité. Quelle faille pourrait-être exploitée par un petit malin ?

## 2. Applications

### 2.1. Application 1

Compléter le code de la fonction suivante et tester dans la console pour remplir le tableau suivant.

```

def mystere (x : int) -> None :
    if x >= 18 :
        print( "Tu es un adulte" )
    elif x < 12 :
        print(...)
    else :
        print(...)

```

Saisie dans la console

Affichage obtenu

| Saisie dans la console | Affichage obtenu |
|------------------------|------------------|
| mystere(21)            | ...              |
| mystere(8)             | ...              |
| mystere(15)            | ...              |

## 2.2. Application 2

### 1. Première partie :

a. Compléter le code de la fonction suivante.

```
def test (a : int, b : int) -> None :  
    if a == b :  
        print( "Alice et Bob ont le même âge")  
    if a >= 18 and b >= 18:  
        print("Alice et Bob sont tous les deux adultes.")  
    if ... :  
        print("Alice est plus jeune que Bob")  
    if ... :  
        print("Alice est au moins deux fois plus âgée que Bob")  
    if ... :  
        print("Alice a au moins 10 ans de moins que Bob.")  
    if ... :  
        print("Alice et Bob ont à eux deux plus de 30 ans")
```

- b. Qu'est-il affiché dans la console avec `test(25, 40)` ?
- c. Qu'est-il affiché dans la console avec `test(24, 6)` ?
- d. Qu'est-il affiché dans la console avec `test(15, 15)` ?
- e. Qu'est-il affiché dans la console avec `test(40, 20)` ?
- f. Qu'est-il affiché dans la console avec `test(10, 21)` ?

### 2. Deuxième partie :

- a. Copier-coller le code de la fonction `test` pour créer une fonction `test_2`, en remplaçant tous les `if` par des `elif`, **sauf le premier**.
- b. Effectuez les mêmes tests que dans la question précédente, mais avec la fonction `test_2`. Obtenez-vous les mêmes résultats dans la console ? Pourquoi ?