

Spécialité NSI Première	DEVOIR SURVEILLE DE NSI N° 6	Jeudi 16 mars 2023
Lycée d'Avesnières		Durée : 2 h
Année scolaire 2022-2023		Calculatrice interdite

L'énoncé complet est à rendre avec la copie.

NOM : **Prénom :**

Exercice 1 (5 points)

Une liste Python est composée de valeurs. Chaque valeur est repérée par son indice.

Indice	0	1	2	3
Valeur	35	35	42	39

On peut parcourir une liste par valeurs ou par indice.

Soit la liste nombres = [35, 35, 42, 39]

- Exemple de parcours de la liste nombres **par valeurs** en utilisant une boucle for :

```
for element in nombres:
    print(element)
```

- Exemple de parcours de la liste nombres **par indices** en utilisant une boucle for :

```
for i in range(len(nombres)):
    print(nombres[i])
```

1) On dispose d'un tableau contenant des nombres entiers dont certains sont peut-être en double. Dans chacun des cas suivants dire, en justifiant, si on peut-on utiliser un **parcours par valeurs** du tableau :

- Rechercher le plus grand nombre
- Déterminer si 0 est présent dans le tableau
- Déterminer la position du minimum
- Calculer la somme de tous les nombres.
- Créer un nouveau tableau ne contenant que les valeurs de la première moitié du tableau.
- Mettre au carré toutes les valeurs du tableau.

2) Alice veut créer la liste des notes qu'elle a obtenues en NSI. Ces notes sont des nombres entiers ou à virgule flottante.

La fonction `creation_liste(n)` prend le paramètre `n` de type entier.

La fonction doit renvoyer la variable `liste_notes` de type liste de flottants. C'est la liste des `n` notes saisies par l'utilisateur.

Recopier toutes les lignes du code suivant sur la copie en les complétant.

```
def creation_liste(n):
    """
    n est un entier donnant la taille de la liste à saisir.
    La fonction renvoie la liste saisie.

    """
    liste_notes = ...
    for i in range(...):
        note = float(input("saisir votre note : "))
        ...
    return ...
```

Exemple :

```
>>> creation_liste(5)
saisir votre note : 10
saisir votre note : 12.5
saisir votre note : 17
saisir votre note : 8
saisir votre note : 11
```

La valeur renvoyée est :

```
[10.0, 12.5, 17.0, 8.0, 11.0]
```

3) Écrire une fonction `statistiques(liste)` qui prend en argument la variable `liste` contenant les notes de l'élève et qui renvoie le triplet de flottants `mini`, `moyenne`, `maxi` qui sont respectivement la valeur minimale, la valeur moyenne, la valeur maximale de la liste.

Exemple :

```
>>> statistiques([10.0, 12.5, 17.0, 8.0, 11.0])
(8.0, 11.7, 17.0)
```

- 4) Les cinq notes d’Alice sont respectivement les notes obtenues aux DS1, DS2, DS3, DS4 et DS5. Écrivez une fonction `transforme(notes, titres)` qui prend en paramètres une liste de flottants (les notes) et une liste de chaînes de caractères (les titres des devoirs) et qui renvoie un dictionnaire d’items dont les clés sont respectivement les titres des devoirs et les valeurs sont les notes d’Alice.

Exemple :

```
notes_alice = [10.0, 12.5, 17.0, 8.0, 11.0]
titres_ds = ['DS1', 'DS2', 'DS3', 'DS4', 'DS5']

alice = transforme(notes_alice, titres_ds)

>>> alice

{'DS1': 10.0, 'DS2': 12.5, 'DS3': 17.0, 'DS4': 8.0, 'DS5': 11.0}
```

- 5) Le professeur a dans sa classe de NSI trois élèves Alice, Bob et Charlie. Les listes de leurs notes sont respectivement :

```
[10.0, 12.5, 17.0, 8.0, 11.0],
[8.0, 11.0, 12.0, 13.0, 16.0],
[4.0, 8.5, 10.0, 6.0, 2.0].
```

Les cinq notes sont respectivement les notes obtenues aux DS1, DS2, DS3, DS4 et DS5.

Ainsi, le professeur a constitué une liste de listes de notes :

```
ma_liste = [[10.0, 12.5, 17.0, 8.0, 11.0],
            [8.0, 11.0, 12.0, 13.0, 16.0],
            [4.0, 8.5, 10.0, 6.0, 2.0]]
```

Le professeur a écrit cette fonction :

```
def mystere(notes_classe, titres):
    classe = []
    for i in range(len(notes_classe)):
        x = transforme(notes_classe[i], titres)
        classe.append(x)
    return classe
```

Écrire la valeur renvoyée par l’appel `mystere(ma_liste, titres_ds)`

Exercice 2 (6 points)

On dispose d'un fichier csv nommé `table1.csv` :

nom	prenom	age	ville	pays	email
Dupont	Marie	32	Paris	France	mdupont@gmail.com
Durand	Pierre	45	Lyon	France	pdurand@gmail.com
Smith	John	28	Londres	Royaume-Uni	johnsmith@gmail.com
Lee	Hyun-Joo	31	Séoul	Corée du Sud	hyunlee@gmail.com
Gonzalez	Carlos	40	Madrid	Espagne	cgonzalez@gmail.com

et d'un deuxième fichier nommé `table2.csv` :

nom	prenom	age	ville	pays	email
Lee	Hyun-Joo	31	Séoul	Corée du Sud	hyunlee@gmail.com
Gonzalez	Carlos	40	Madrid	Espagne	cgonzalez@gmail.com
Dupont	Marie	32	Paris	France	mdupont@gmail.com
Hernandez	Isabel	26	Barcelone	Espagne	ihernandez@gmail.com

Ces tables contiennent uniquement des chaînes de caractères. On dispose également d'un fichier `exercice.py` placé dans le même répertoire que `table1.csv` et `table2.csv` contenant le code :

```
import csv
#ouverture du fichier csv
def lecture_fichier(nom_fichier):
    with open(nom_fichier, mode='r', encoding='utf-8-sig') as fichier_ouvert:
        return [ligne for ligne in csv.reader(fichier_ouvert)]
```

1) On souhaite récupérer le contenu des fichiers csv dans le programme Python.

Quelle instruction Python faut-il écrire afin de lire les fichiers `table1.csv` et `table2.csv` et d'affecter leurs contenus respectivement dans les variables appelées `table_1` et `table_2` ?

2) Quel est le type des variables `table_1` et `table_2` ?

3) Quels sont les descripteurs communs aux deux tables ?

4) On souhaite fusionner les deux tables en une seule, de manière à obtenir la `table_3` :

nom	prenom	age	ville	pays	email
Dupont	Marie	32	Paris	France	mdupont@gmail.com
Durand	Pierre	45	Lyon	France	pdurand@gmail.com
Smith	John	28	Londres	Royaume-Uni	johnsmith@gmail.com
Lee	Hyun-Joo	31	Séoul	Corée du Sud	hyunlee@gmail.com
Gonzalez	Carlos	40	Madrid	Espagne	cgonzalez@gmail.com
Lee	Hyun-Joo	31	Séoul	Corée du Sud	hyunlee@gmail.com
Gonzalez	Carlos	40	Madrid	Espagne	cgonzalez@gmail.com
Dupont	Marie	32	Paris	France	mdupont@gmail.com
Hernandez	Isabel	26	Barcelone	Espagne	ihernandez@gmail.com

Donner une instruction Python qui permet d'obtenir `table_3` à partir de `table_1` et `table_2`.

5) On remarque que la `table_3` contient des doublons. On veut obtenir la variable `table_4` qui contient les données suivantes :

nom	prenom	age	ville	pays	email
Dupont	Marie	32	Paris	France	mdupont@gmail.com
Durand	Pierre	45	Lyon	France	pdurand@gmail.com
Smith	John	28	Londres	Royaume-Uni	johnsmith@gmail.com
Lee	Hyun-Joo	31	Séoul	Corée du Sud	hyunlee@gmail.com
Gonzalez	Carlos	40	Madrid	Espagne	cgonzalez@gmail.com
Hernandez	Isabel	26	Barcelone	Espagne	ihernandez@gmail.com

Écrire une fonction `supprimer_doublons(liste)` qui prend en argument la variable `liste` contenant des doublons et qui renvoie une liste avec les mêmes enregistrements, sans les doublons.

Exemple :

```
table_3 vaut [['nom', 'prenom', 'age', 'ville', 'pays', 'email'], ['Dupont', 'Marie', '32', 'Paris', 'France', 'mdupont@gmail.com'], ['Durand', 'Pierre', '45', 'Lyon', 'France', 'pdurand@gmail.com'], ['Smith', 'John', '28', 'Londres', 'Royaume-Uni', 'johnsmith@gmail.com'], ['Lee', 'Hyun-Joo', '31', 'Séoul', 'Corée du Sud', 'hyunlee@gmail.com'], ['Gonzalez', 'Carlos', '40', 'Madrid', 'Espagne', 'cgonzalez@gmail.com'], ['Lee', 'Hyun-Joo', '31', 'Séoul', 'Corée du Sud', 'hyunlee@gmail.com'], ['Gonzalez', 'Carlos', '40', 'Madrid', 'Espagne', 'cgonzalez@gmail.com'], ['Dupont', 'Marie', '32', 'Paris', 'France', 'mdupont@gmail.com'], ['Hernandez', 'Isabel', '26', 'Barcelone', 'Espagne', 'ihernandez@gmail.com']]
```

```
>>> table_4 = supprimer_doublons(table_3)
```

```
table_4 vaut [['nom', 'prenom', 'age', 'ville', 'pays', 'email'], ['Dupont', 'Marie', '32', 'Paris', 'France', 'mdupont@gmail.com'], ['Durand', 'Pierre', '45', 'Lyon', 'France', 'pdurand@gmail.com'], ['Smith', 'John', '28', 'Londres', 'Royaume-Uni', 'johnsmith@gmail.com'], ['Lee', 'Hyun-Joo', '31', 'Séoul', 'Corée du Sud', 'hyunlee@gmail.com'], ['Gonzalez', 'Carlos', '40', 'Madrid', 'Espagne', 'cgonzalez@gmail.com'], ['Hernandez', 'Isabel', '26', 'Barcelone', 'Espagne', 'ihernandez@gmail.com']]
```

6) Ecrire une fonction `age_moyen(table)` qui prend en argument la variable `liste` contenant une ligne d'en-tête et des lignes de données sans doublon et qui renvoie l'âge moyen des individus présents dans la table.

Exemple :

```
>>> age_moyen(table_4)
33.666666666666664
```

7) On saisit maintenant l'instruction Python

```
table_5 = sorted(table_4[1:], key=lambda ligne: ligne[0])
```

Que contient la variable `table_5` ?

Exercice 3 (3 points)

Alice a décidé d'envoyer un message en plusieurs sous-messages à Bob en utilisant le protocole du bit alterné.

Bob intègre une méthode de validation des sous-messages reçus. Il pourra décider de les garder ou de les écarter. Le but est d'éviter les doublons.

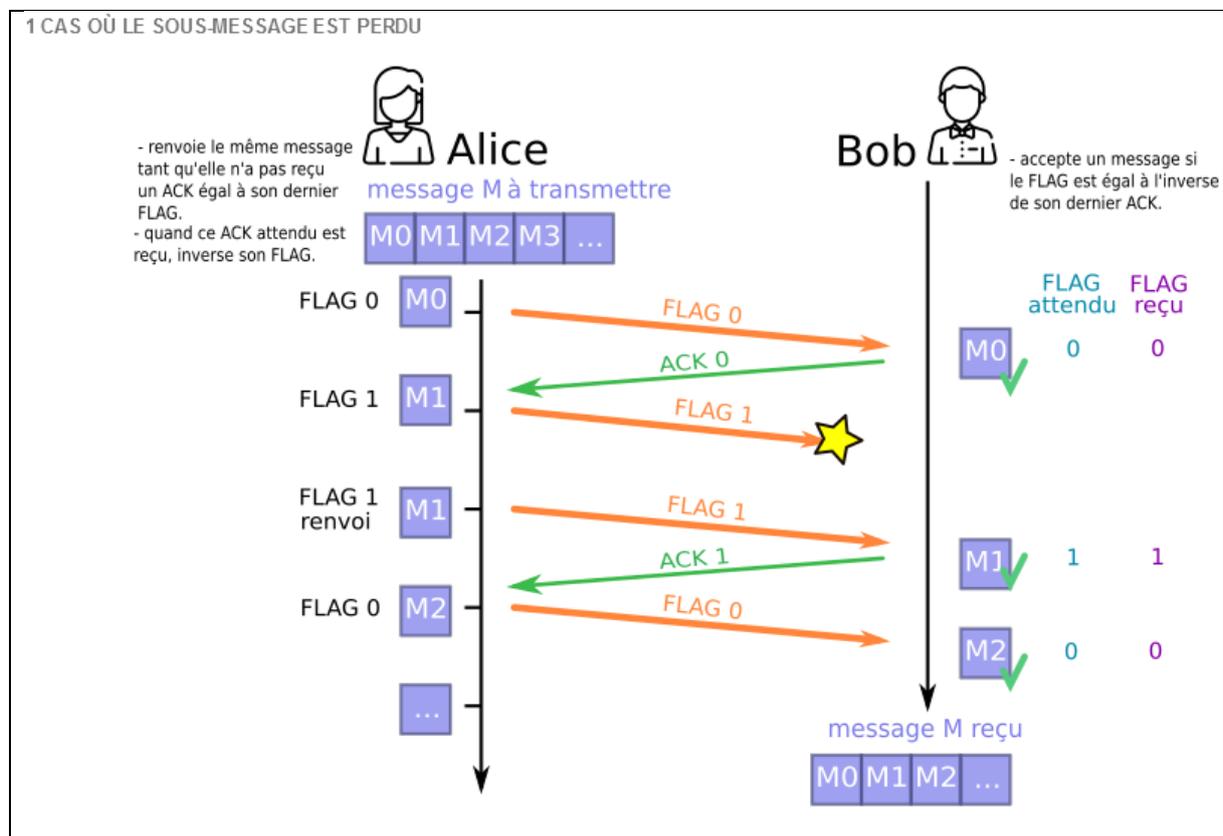
Pour réaliser ceci, Alice va ajouter à chacun de ses sous-messages un bit de contrôle, que nous appellerons FLAG (drapeau). Au départ, ce FLAG vaut 0. Quand Bob reçoit un FLAG, il renvoie un ACK (accusé de réception) **égal au FLAG reçu**.

Alice va attendre ce ACK contenant le même bit que son dernier FLAG envoyé :

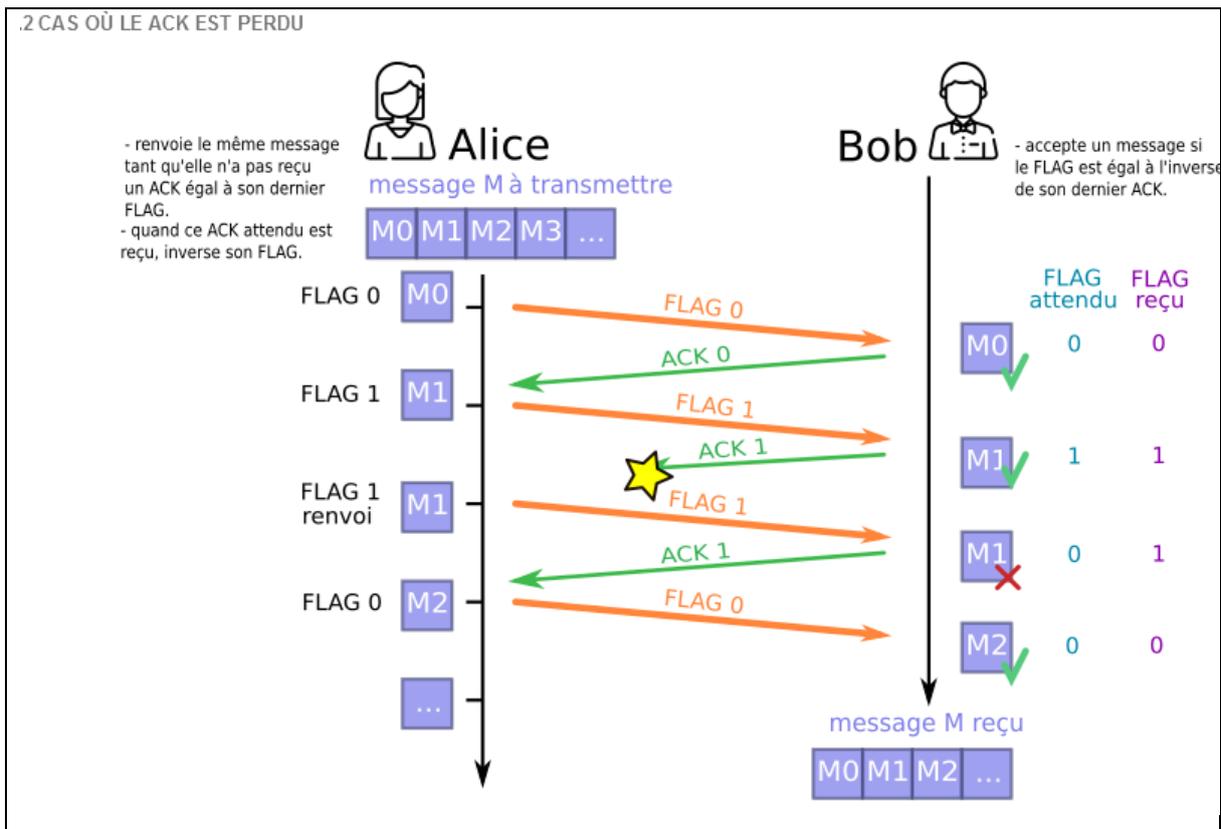
- tant qu'elle ne l'aura pas reçu, elle continuera à envoyer **le même sous-message, avec le même FLAG**.
- dès qu'elle l'a reçu, elle peut envoyer un nouveau sous-message en **inversant** («alternant») **le bit de son dernier FLAG** (d'où le nom de ce protocole).

Bob, de son côté, va contrôler la validité de ce qu'il reçoit : il ne gardera que **les sous-messages dont le FLAG est égal à l'inverse de son dernier ACK**. C'est cette méthode qui lui permettra d'écarter les doublons.

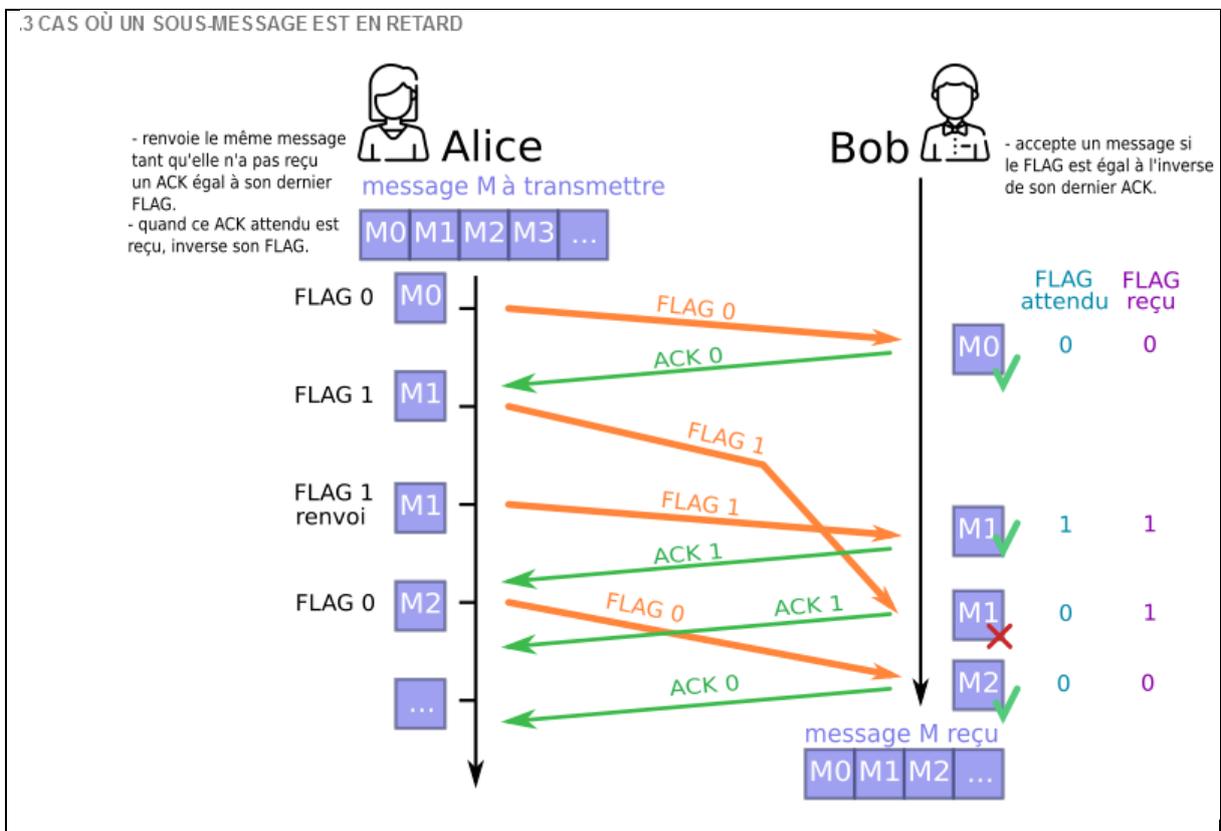
1) Cas n°1 : la transmission est-elle correcte ? Justifier la réponse



2) Cas n°2 : la transmission est-elle correcte ? Justifier la réponse



3) Cas n°3 : la transmission est-elle correcte ? Justifier la réponse



Exercice 4 (6 points)

Document 1 : Objectif à atteindre

- On veut réaliser un convertisseur de la devise Euro vers le Dollar ayant cette apparence :

Convertisseur d'euros en dollars

Nom	Taux	Valeur
Entrez une valeur en €	1,00000	<input type="text"/>
Dollar US \$	1,17000	<input type="button" value="Convertir"/>

- Utilisation
 - L'utilisateur entre une valeur en Euro dans la zone de texte.
 - L'utilisateur clique sur le bouton Convertir.
 - Dès que l'utilisateur a cliqué sur le bouton Convertir, la valeur en Dollars s'affiche sur le bouton (à la place de "Convertir").

Exemple

Nom	Taux	Valeur
Entrez une valeur en €	1,00000	<input type="text" value="2"/>
Dollar US \$	1,17000	<input type="button" value="Convertir"/>

Nom	Taux	Valeur
Entrez une valeur en €	1,00000	<input type="text" value="2"/>
Dollar US \$	1,17000	<input type="button" value="2.34"/>

Document 2 : Fichiers (incomplets) déjà écrits au départ

- devises.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Convertisseur en Dollars</title>
  <link href="style_corrige.css" type="text/css" rel="stylesheet">
</head>
<body>
  <h3>Convertisseur d'Euros en Dollars</h3>
  <table id="conversion">
    <thead>
      <tr>
        <th>Nom</th>
        <th>Taux</th>
        <th>Valeur</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>
          <p>Entrez une valeur en &#x20ac</p>
        </td>
        <td>
          <p>1,00000</p>
        </td>
        <td>
          <p><input type="text" id="valeur" size="5" class="nombre"></p>
        </td>
      </tr>
      <tr>
        <td>
          <p>Dollar US &#x0024</p>
        </td>
        <td>
          <p>1,17000</p>
        </td>
        <td>
          <p><button id="convertir" onclick="convertir()"> Convertir </button></p>
        </td>
      </tr>
    </tbody>
  </table>
  <script src="script.js"></script>
</body>
</html>
```

- script.js

```
function arrondi(x) {
  return Math.round(x * ...) / ...;
}

function Conversion() {
  ... = Number(document.getElementById("valeurE").value);
  let valeurD = arrondi(... * ...);
  document.getElementById("boutonD").innerHTML = ...;
}
```

- style.css

```
body {
  margin: 0 20%;
  background-color: #f0f0f0;
}
h3 {
  text-align: center;
  border: 1px solid #aaaaaa;
  border-radius: 10px;
  padding: 20px;
  background-color: #dadada;
}
.nombre {
  text-align: right;
}
#conversion {
  margin: 0 auto;
}
table {
  border-collapse: collapse;
  border: 1px solid black;
}
th {
  background-color: #dadada;
}
td, th {
  border-top: 1px solid black;
  padding: 1em;
  text-align: center;
}
button {
  width: 100px;
  height: 30px;
  font-size: 20px;
}
```

1) Fichier HTML

- a. La ligne "Entrez une valeur en €" du tableau HTML est codée par le bloc de code suivant :

```
<tr>
  <td>
    <p>Entrez une valeur en €</p>
  </td>
  <td>
    <p>1,00000</p>
  </td>
  <td>
    <p><input type=... id=... size="5" class="nombre"></p>
  </td>
</tr>
```

Recopiez ce bloc de code en complétant les pointillés pour que s'affiche dans la page HTML le composant graphique *zone de texte* . L'identifiant unique id="..." est utilisé par la fonction JavaScript Conversion() pour qu'elle puisse prendre en entrée la valeur que l'utilisateur a saisi dans cette zone de texte.

- b. La ligne "Dollar US \$" du tableau HTML est codée par le bloc de code suivant :

```
<tr>
  <td>
    <p>Dollar US $</p>
  </td>
  <td>
    <p>1,17000</p>
  </td>
  <td>
    <p><button id=... onclick=...> Convertir </button></p>
  </td>
</tr>
```

On remarque que cette ligne du tableau contient un composant graphique *bouton* et qu'à la place de la balise `<input type="button" id=... onclick=... value="Convertir">` on a utilisé la balise `<button id=... onclick=...> Convertir </button>`. **L'avantage est qu'on peut modifier le texte situé sur le bouton à l'aide de la propriété innerHTML de l'objet bouton.**

Recopiez ce bloc de code en complétant les pointillés pour que s'affiche dans la page HTML le composant graphique *bouton* . L'identifiant unique id="..." est utilisé par la fonction JavaScript Conversion() pour qu'elle puisse afficher sur le bouton le résultat de la conversion en Dollars.

2) Fichier JavaScript

a. Dans le fichier JavaScript la fonction arrondi(x) est codée par le bloc de code suivant :

```
function arrondi(x) {  
    return Math.round(x * ...)/...;  
}
```

Recopiez en complétant les pointillés pour qu'elle renvoie la valeur arrondie de x à deux décimales.

b. Dans le fichier JavaScript est écrite une deuxième fonction nommée Conversion()

```
function Conversion() {  
    ... = Number(document.getElementById("valeurE").value);  
    let valeurD = arrondi(... * ...);  
    document.getElementById("boutonD").innerHTML = ...;  
}
```

Recopiez ce bloc de code en complétant les pointillés pour qu'elle fasse les tâches suivantes :

- Affecter à la variable entree la valeur que l'utilisateur saisit comme valeur en Euros.
- A partir de la variable entree, calculer valeurD qui est la valeur Dollars en utilisant le taux de conversion de 1,17 \$ pour 1 € tout en l'arrondissant à deux décimales.
- Affecter au texte situé sur le bouton "Convertir" la valeur Dollars ainsi calculée.

3) On veut maintenant compléter le convertisseur de la devise Euro vers le Dollar ou la Livre sterling ou le Yen ayant cette apparence :

Convertisseur d'euros en dollars, en livres et en yens

Nom	Taux	Valeur
Entrez une valeur en €	1,00000	<input type="text"/>
Dollar US \$	1,17000	<input type="button" value="Convertir"/>
Livre Sterling £	0,86000	<input type="button" value="Convertir"/>
Yen ¥	128,43000	<input type="button" value="Convertir"/>

- Utilisation
 1. Entrer une valeur en Euro dans la zone de texte.
 2. Cliquer sur le bouton Convertir sur une des trois lignes. C'est le bouton cliqué qui affiche le résultat.
 3. Dès que l'utilisateur a cliqué sur un des boutons Convertir, la valeur s'affiche sur ce bouton.
 - a. Écrivez les modifications à faire dans le fichier devises.html

Indication : En HTML pour les symboles monétaires, utilisez le n° de point de code Unicode en écriture hexadécimale.

Par exemple, le symbole € a pour point de code U+20ac donc en HTML on aura € etc.

x signifie que le nombre qui suit est en écriture hexadécimale.

- Le point de code Unicode du symbole de la Livre sterling £ est U+00a3.
- Le point de code Unicode du symbole du Yen ¥ est U+00a5.

- b. Écrivez les modifications à faire dans le fichier script.js