


<p>TP</p> <p>TSIN</p>	<p>Programmation en langage structuré</p> <p>TP1-3 : Python – Initiation (les listes)</p>	
-----------------------	---	--

1 Les listes

Une liste est une **collection d'éléments** séparés par des **virgules** ; l'ensemble étant entre crochets.

Elle peut contenir des éléments de types différents : entiers, flottants, chaînes de caractères, ...

L'intérêt d'une liste est qu'on appelle un élément par sa position (**indice**, ou *index*).

liste : ['girafe', 'tigre', 125, 'souris']

indice : 0 1 2 3

Les éléments d'une liste ont un ordre défini. Les indices d'une liste débutent à zéro.

1.1 Exercices

Ecrire, tester en commentant chaque ligne les syntaxes suivantes

1. Accès aux éléments d'une liste

```
# Les listes

# Déclaration des variables
s = ['lundi', 'mardi', 230, 'mercredi', 'dimanche']

# Accès aux éléments d'une liste
# Accès simple
print(s)
print(s[0])
print(s[2])
print(s[-1])
print(s[-2])

print("-----")
# Découpage d'une liste
print(s[1:3])
print(s[1:-1])
print(s[2:])
```

2. Opérations sur les listes

```
# Opérations sur les listes
print(len(s))
del(s[2])
print(s)
s = s+['jeudi', 'vendredi']
print(s)
print(s*2)
s[2] = 'bonjour'
print(s)
s[0:2] = ['jeudi', 38]
print(s)
```

3. Fonctions avancées

```
# Fonctions avancées
print("----- Fonctions avancées")
s.append('samedi')
print(s)
s.insert(3, 'tyty')
print(s)
s.remove('bonjour')
print(s)
s.pop(1)
print(s)
s.pop()
print(s)
print(s.index('tyty'))
s.reverse()
print(s)
s.sort()
print(s)
```

1.2 Exercices de synthèse

1. Liste de 10 entiers

Écrire un programme qui crée une liste de 10 entiers sélectionnés de manière aléatoire par l'ordinateur et qui réalise ensuite les actions suivantes :

- affiche la valeur de L[4]
- modifie la liste en remplaçant L[1] par 17 et L[3] par la somme des cases voisines L[2] et L[4]
- affiche la nouvelle liste

Remarque : Pour générer de manière aléatoire des nombres par l'ordinateur il faut utiliser la fonction **randint**(debut, fin) en ayant importé la librairie **random**.

2. Échange de valeurs

Écrire un programme qui demande à l'utilisateur les indices des deux cases qui doivent être permutées.

3. Modifications d'une liste

Écrire un programme qui crée une liste de 10 entiers aléatoires puis augmente de 1 la valeur de chaque élément de la liste.

4. Parcours d'une liste

Écrire un programme qui crée une liste de 10 entiers, puis à l'aide de parcours successifs, effectue les actions suivantes :

- a) Afficher la liste en colonne
- b) Compter le nombre de multiples de 3 présents dans la liste
- c) Calculer la somme de toutes les valeurs paires de la liste
- d) Trouver le maximum et le minimum des éléments de la liste

5. Chaîne de caractères

Écrire un programme qui demande à l'utilisateur d'entrer une chaîne de caractère et qui la transforme en liste.

On n'utilisera pas la syntaxe Python suivante : `Liste = list(chaine)`

6. Conversion binaire / décimal

Écrire un programme qui convertit un nombre binaire en nombre décimal ; la démarche est identique à celle utilisée quand on fait l'opération manuellement.

Aide :

- Convertir la chaîne de caractère en liste
- Inverser la liste
- Parcourir la liste et faire les opérations nécessaires en prenant en compte le poids (2^{indice})