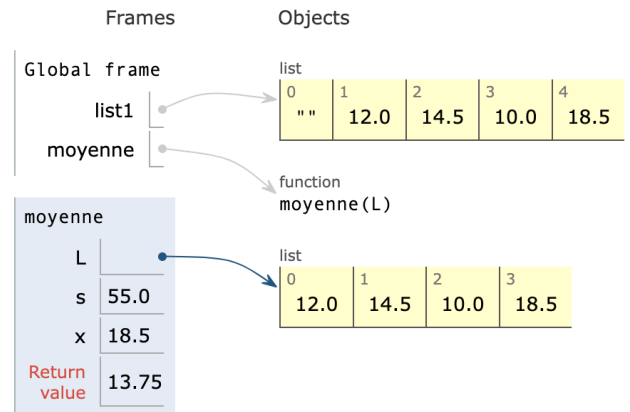


# 1. Traitement d'une liste

## Script 1:

```

1. list1 = ["",12.0,14.5,10.0,18.5]
2.
3. def moyenne(L):
4.     s = 0
5.     for x in L:
6.         s+=x
7.     return s/len(L)
8.
9. m = moyenne(list1[1:])
10. print(m)
    
```



### Questions:

1a. Commenter le schéma ci-dessus: que signifient les flèches? Que signifient les cases bleues et jaunes?

1b. Pourquoi écrit-on l'instruction `m = moyenne(list1[1:])` et non `m = moyenne(list1)`?

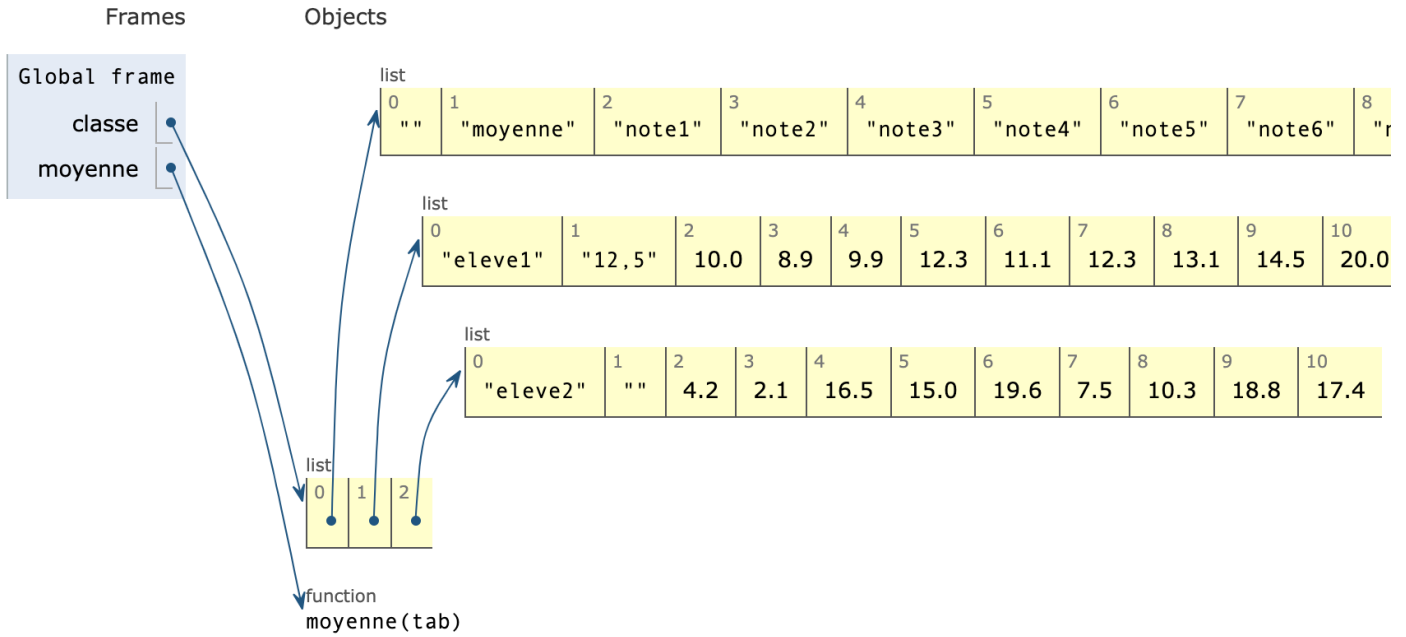
1c Modifier le script (faire \*edit\* dans pythontutor) pour que le programme place la valeur `m` dans la case `list1[0]`. Noter ici l'instruction utilisée. Comment voit-on la modification dans pythontutor?

# 2. Traitement sur une table

## Script 2:

```

1. classe = [['\u00e9', 'moyenne', 'note1', 'note2', 'note3', 'note4', 'note5',
2.           'note6', 'note7', 'note8', 'note9'],
3.           ['eleve1', '12,5', 10.0, 8.9, 9.9, 12.3, 11.1, 12.3, 13.1,
4.           14.5, 20.0],
5.           ['eleve2', '', 4.2, 2.1, 16.5, 15.0, 19.6, 7.5, 10.3, 18.8,
6.           17.4]]
7.
8. def moyenne(tab):
9.     s = 0
10.    for note in tab:
11.        s += note
12.    return s / len(tab)
13.
14. eleve = classe[2]
15. m = round(moyenne(eleve[2:]),2)
16. print(m)
    
```



2a. Commenter le schéma ci-dessus: que signifient les flèches? Que signifient les cases bleues et jaunes

2b. Que contient la liste `eleve`? Recopier ici son contenu.

2c. Pourquoi écrit-on l'instruction `m = moyenne(eleve[2:])` et non `m = moyenne(eleve)`?

2d. Modifier le script pour que le programme place la valeur m dans la case `eleve[1]`. Noter ici l'instruction utilisée. Comment voit-on la modification dans pythontutor?

2e. Modifier maintenant le programme pour calculer la moyenne de l'élève 1. Comment doit-on modifier le programme?

### 3. Traitement sur un fichier de notes

Le tableau réalisé sous Excel, `classe.csv`, est chargé dans le programme python avec le script 3 (ci-dessous)

#### A. TABLEAU EXCEL

classe

	moyenne	note1	note2	note3	note4	note5	note6	note7	note8	note9
eleve1	12,5	10	8,9	9,9	12,3	11,1	12,3	13,1	14,5	20
eleve2		4,2	2,1	16,5	15	19,6	7,5	10,3	18,8	17,4
eleve3		8,4	18,4	6,3	19,7	6,4	11,3	4	18,4	18
eleve4		16,4	6,4	8,9	5,4	6,5	13,9	1,4	9,8	9,7
eleve5		16,7	7,8	2	18,4	11,9	18	12,1	3,9	12,9
eleve6		13,6	6,7	14,8	2,2	19,3	5,2	3,1	14,8	3,3
eleve7		18,8	3,9	8,5	4,9	8,4	13,5	8,9	1,1	19,9
eleve8		5,4	14,7	17,6	13,7	4,9	4,1	11,5	4,6	6,1
eleve9		8,9	5,3	1,2	5,8	1	17,9	13,9	7,4	5,3
eleve10		1,5	19,2	6,1	16,1	7,4	18,3	5	13,1	7,6

#### B. OUVERTURE ET LECTURE DU FICHIER:

Script 3:

```
1. import csv
2. with open('datas/classe.csv', newline='') as csvfile:
3.     spamreader = csv.reader(csvfile, delimiter = ";")
4.     classe = []
5.     for row in spamreader:
6.         classe.append(row)
```

Après lecture du fichier: les données sont mises dans une table python (une liste de listes):

```
7. classe = [['\ufeff', 'moyenne', 'note1', 'note2', 'note3', 'note4', 'note5',
8.     'note6', 'note7', 'note8', 'note9'],
9.     ['eleve1', '12,5', 10.0, 8.9, 9.9, 12.3, 11.1, 12.3, 13.1, 14.5, 20.0],
10.    ['eleve2', '', 4.2, 2.1, 16.5, 15.0, 19.6, 7.5, 10.3, 18.8, 17.4],
11.    ['eleve3', '', 8.4, 18.4, 6.3, 19.7, 6.4, 11.3, 4.0, 18.4, 18.0],
12.    ['eleve4', '', 16.4, 6.4, 8.9, 5.4, 6.5, 13.9, 1.4, 9.8, 9.7]]
```

B1. Quelle partie de la table correspond à `classe[0]` ? (surligner)

B2. Quelle partie de la table correspond à `classe[1]` ? (surligner d'une autre couleur)

B3. Colorier à l'aide d'une couleur jaune la partie de la table qui correspond à `classe[2][2:]`

#### C. EXPLOITER LE TABLEAU

Pour parcourir les éléments de la liste `classe`, on utilise une boucle qui parcourt chaque élément `eleve`:

```
12. for eleve in classe:
13.     print(eleve)
```

On obtient alors:

```
['\uffeff', 'moyenne', 'note1', 'note2', 'note3', 'note4', 'note5', 'note6',
'note7', 'note8', 'note9']
['eleve1', '12,5', 10.0, 8.9, 9.9, 12.3, 11.1, 12.3, 13.1, 14.5, 20.0]
...
```

C3. Adapter le script aux lignes 12.13. pour parcourir la liste depuis `classe[1]` jusqu'à la fin du tableau (on ne souhaite pas afficher ou traiter la première ligne). Ecrire ce nouveau script:

```
12. for eleve in .. ..
13.     print(eleve)
```

C4. Ecrire l'instruction qui calcule la moyenne des notes de la ligne `classe[2]`, entre les indices 2 et plus, c'est à dire pour la table `classe[2][2:]`

C5. Quelle est l'instruction python qui permet d'atteindre la cellule de la colonne 1 (celle de la moyenne de l'élève), pour le 2e élève de la liste `classe`?

C6. Ecrire une boucle bornée sur `classe[1:]` qui parcourt chaque élément `eleve` et qui stocke dans la première colonne le résultat de la moyenne de ses notes `eleve[2:]`

```
12. for eleve in .. ..
13.     .. ..
14.     .. ..
15.     .. ..
```

## 4. Exercice: Fichier des effectifs du lycée

### A. TABLEAU EXCEL DES EFFECTIFS DANS UN LYCÉE, PAR NIVEAU:

	2020	2021	2022	2023	2024	total
seconde	495	495	495	495	525	
premiere	450	510	450	480	450	
terminale	525	495	435	480	480	

### B. QUESTIONS

B1. Ecrire le script python qui ouvre le fichier `effectifs.csv` et qui stocke les valeurs dans un tableau `effectif`

B2. Colorier en gris la partie du tableau qui correspond à `effectif[1]`

B3. Colorier à l'aide d'une couleur jaune la partie du tableau qui correspond à `effectif[1][1:6]`

B4. Quelle est l'instruction python qui permet d'atteindre la cellule de la colonne 6, pour la ligne du niveau `seconde`?

B5. Quelle autre instruction permet d'atteindre l'effectif de `premiere` en 2024.

B6. Ecrire l'instruction qui calcule la somme des effectifs de la ligne `effectif[1]`, entre les indices 1 et 5, c'est à dire pour la table `effectif[1][1:6]`. Utiliser la fonction `somme`.

B7. Quelle instruction place la somme des eleves de `terminale` dans la dernière colonne du niveau `terminale`?

B8. Ecrire une boucle bornée sur `effectif[1:]` qui parcourt chaque élément `niveau` et qui stocke dans la dernière colonne le résultat de la somme des effectifs (2020 à 2024).