

TP5 Listes (NSI_1)

Ex 1: Elements d'une liste

```
s = ['lundi', 'mardi', 'mercredi']
```

Question a1: Quel est l'indice du dernier élément d'une liste? (*on ne connaît pas le nombre total*)

Question a2: Comment modifie-t-on la liste `['lundi', 'mardi', 'mercredi']` pour obtenir `['lundi', 'mardi', 'jeudi']`?

Question a3: Pourquoi l'instruction `s[4] = "samedi"` génère t-elle une erreur?

Avec la liste:

```
s = ['lundi', 'mardi', 'mercredi', 'jeudi', 'vendredi']
```

Question b: Que retourne la proposition `s[1:]`? Découpe t-elle la liste après le premier élément, le 2e élément, ou bien retourne t-elle la liste entière?

Ex 2: Opérations sur les éléments et sur les listes

```
t = [2, 8, 9, 2]
```

```
t[2] = t[2] + 5
```

Question c: Que vaut t à la fin du script `t[2] = t[2] + 5`? La valeur 5 est-elle ajoutée à chaque élément de la liste, ou bien à un seul élément?

```
l1 = [1,2,3]
```

```
l2 = [4,5,6]
```

```
l1 = l1 + l2
```

Question d1: Que fait l'opérateur `+` entre 2 listes?

```
l1 = [1,2,3]
```

```
l2 = l1 * 3
```

Question d2: Que fait l'opération `* 3` sur une liste?

Ex 3: Méthodes de listes

```
s = ['lundi', 'mardi', 'mercredi']
```

Question e1: Pourquoi la valeur renvoyée par `len(s)` évolue t-elle au cours de l'exercice? Quel est le rôle de chacune de ces méthodes (`len`, `pop`, `extend`)?

Question e2: Laquelle de ces méthodes peut-on utiliser pour réaliser l'opération `l1 = l1 + l2` comme dans l'exercice précédent? Ecrire l'instruction équivalente.

```
from random import shuffle
s = ['lundi', 'mardi', 'mercredi', 'jeudi', 'vendredi']
shuffle(s)
```

Question e3: Quel est le rôle de la fonction `shuffle`?

```
>>> s.index('lundi')
```

Question e4: Quel est le rôle de la fonction `index`?

Ex 4: chaîne de caractère comme une liste

```
# script 1
debut = "Bon"
fin = "jour"
debut + fin
```

```
# script 2
debut = "20"
fin = "22"
debut + fin
```

```
# script 3
debut = [2,0]
fin = [2,2]
debut + fin
```

Question f1: Y-a-t-il une ressemblance entre:

- * l'opérateur `+` appliqué à une chaîne de caractères
- * l'opérateur `+` appliqué à une liste?

```
# script 4: transformer un str en une liste
phrase = "une phrase assez longue"
L = list(phrase)
```

```
# script 5: découper un str en une liste
phrase = "mais,pourquoi,autant,de,virgules?"
L = phrase.split(',')
```

Question f2: à partir de ces exemples, comment pourrait-on obtenir `['une', 'phrase', 'assez', 'longue']` à partir de la chaîne `phrase`?

Ex 5: utiliser l'indice (liste) ou une clé (dictionnaire)

> Compléter l'instruction `print(...)` qui, selon le numéro `n` du jour de la semaine (1-7) affiche le nom du jour de la semaine.

```
semaine = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'vendredi', 'samedi', 'dimanche']
```

```
for n in range(len(semaine)):
    print(semaine[... ])
```

> Compléter l'instruction `print(...)` pour afficher chaque jour de la semaine:

```
semaine =
{1:'Lundi',2:'Mardi',3:'Mercredi',4:'Jeudi',5:'vendredi',6:'samedi',7:'dimanche'
}
for n in semaine.keys():
    print(semaine[... ])
```

Question g: Compléter la phrase: Pour parcourir les éléments de liste, on fait une boucle avec un variant sur les `[indices/clés]`. Pour les éléments d'un dictionnaire, on utilise un variant sur les `[indices/clés]`.

Traiter les éléments d'un dictionnaire

> Compléter le programme. A chaque itération, vous devrez:

- * lire le jour de la semaine associé à la clé `n`
- * ajouter le jour dans une liste `L`

```
L = []
semaine =
{1:'Lundi',2:'Mardi',3:'Mercredi',4:'Jeudi',5:'vendredi',6:'samedi',7:'dimanche'
}
for n in semaine.keys():
    jour = semaine[... ]
    L.append(... )
```

Question h1: Quelle instruction est utilisée pour parcourir les clés d'un dictionnaire?

Question h2: Quelle instruction est utilisée pour placer une nouvelle valeur dans une liste (à la fin de cette liste)?

> Compléter le script ci-dessous. Celui-ci doit:

- * parcourir les éléments de liste (parcours par indice)
- * si l'indice est pair, c'est une clé, mémoriser cette clé
- * si l'indice est impair, c'est une valeur, mémoriser cette valeur
- * placer dans un dictionnaire tous les couples clé:valeur

```
L = ['nom', 'Alice', 'classe', 'Premiere', 'activité', 'Judo']
D = {}
for i in range(len(L)):
    if ... % 2 == ...:
        cle = L[...]
```

```
else:  
    valeur = L[...]  
    D[....] = valeur  
print(D)
```

Question h3: Quelle instruction est utilisée pour tester la parité de l'indice `i`?

Question h4: Quelle instruction est utilisée pour placer un nouveau couple clé:valeur dans le dictionnaire?