

Les listes

Points forts

- . Les listes sont des séquences ordonnées d'objets.
- . Les listes peuvent contenir n'importe quel type d'objets.
- . Les listes sont comme des tableaux.
- . Les listes sont mutables.

```
In [12]: #Exemple: Lorsque l'ont veut pouvoir rajouter ou supprimer des éléments.
liste=[1,2,4, 'a' ]
liste.append( 'b' )
print(liste)
liste.remove( 'b' )
print(liste)

[1, 2, 4, 'a', 'b']
[1, 2, 4, 'a']
```

Les tuples

Points forts

- . Les tuples sont des séquences ordonnées d'objets.
- . Les tuples peuvent contenir n'importe quel type d'objets.
- . Les tuples sont comme des tableaux.

Points faibles

- . Les tuples sont immuables.

```
In [13]: #Exemple: Lorsque l'on veut que la sequence soit inchangeable et ordonné
e.
tuples=(1,2,3,4)
print(tuples)

(1, 2, 3, 4)
```

Les dictionnaires

Points forts

- . Chaque élément est une paire faite d'une clé et d'une valeur.
- . Possibilité d'accéder à la liste des clés ou valeurs indépendamment.

Points faibles

- . La clé doit être unique

```
In [9]: #Exemple: Lorsque l'ont veut accéder à une valeur en utilisant sa clé.
MyData = { 'name': 'Assitan', 'age': 23, 'sexe': 'feminin' }
MyData[ 'name' ]
```

```
Out[9]: 'Aita'
```

Set

Points forts

- . Séquence mutable
- . Séquence non ordonnée

Points faibles

- . Habituellement utilisés pour créer une séquence d'éléments uniques
- . Séquence non ordonnée
- . Set nécessite que les éléments soient hashable

```
In [17]: # Exemple: Lorsque l'ont est intéressé par la présence d'un élément en i
important peu d'importance à la position.
FriendSet = set([ 'Ram', 'Mandar', 'Jessica' ])
Bestfriend = set({ 'Mandar' })
FriendSet & Bestfriend
```

```
Out[17]: { 'Mandar' }
```

Sources:

http://www.thomas-cokelaer.info/tutorials/python/data_structures.html
<http://stackoverflow.com/questions/3489071/in-python-when-to-use-a-dictionary-list-or-set>
<http://www.yourownlinux.com/2016/10/introduction-python-list-tuple-dictionary-string-set-file-object.html>