



- ▶ Présentation générale
- ▶ QUESTIONNAIRE : VOS ATTENTES ET MOTIVATIONS
- ▶ Semaine 1. Introduction au MOOC et aux outils Python
- ▶ Semaine 2. Notions de base pour écrire son premier programme en Python
- ▼ **Semaine 3. Renforcement des notions de base, références partagées**
 - 1. Les fichiers**
Quiz Échéance le janv 25, 2018 at 23:30 UTC
 - 2. Les tuples**
Quiz Échéance le janv 25, 2018 at 23:30 UTC
 - 3. Tables de hash**
Quiz Échéance le janv 25, 2018 at 23:59 UTC
 - 4. Les dictionnaires**
Quiz Échéance le janv 25, 2018 at 23:59 UTC
 - 5. Les ensembles**

QUIZ 15 - LES ENSEMBLES (2/3 points)

Éléments dans un ensemble

Parmi les objets suivants, lesquels peut-on insérer comme élément dans un ensemble de type `set` ? :

☐ Un objet (de type) `set`

☒ Un objet `frozenset` ✓

☐ Un tuple de `set` 's

☒ Un tuple de `frozenset` 's ✓

☐ Une liste de `frozenset` 's



EXPLANATION

Pour pouvoir être inséré dans un objet `set`, un objet doit être globalement immuable.

option 1: ce qui n'est évidemment pas le cas pour un `set`, dans lequel on peut insérer et enlever des éléments.

option 2: le type `frozenset` est immuable; de plus comme un `frozenset` est un ensemble, tous ses éléments sont aussi globalement immuables, donc l'objet lui-même est globalement immuable.

option3: le tuple est immuable mais ses éléments ne le sont pas, on ne peut donc pas insérer ce tuple dans un ensemble.

option 4: un tuple d'objets globalement immuables est immuable.

option 5: la liste elle-même est mutable.



5. Les exceptions

Quiz Echéance le janv
25, 2018 at 23:59 UTC

7. Les références

partagées

Quiz Echéance le janv
25, 2018 at 23:59 UTC

8. Introduction
aux classes

Quiz Echéance le janv
25, 2018 at 23:59 UTC

- ▶ Semaine 4.
Fonctions et
portée des
variables
- ▶ Semaine 5.
Itération,
importation et
espace de
nommage
- ▶ Semaine 6.
Conception
des classes
- ▶ Semaine 7.
L'écosystème
data science
Python
- ▶ Semaine 8.
Programmation
asynchrone -
asyncio

Pour créer un ensemble `set` via

Rechercher un cours



☒ `{}`

☐ `set()` ✓

☐ `set([])` ✓



EXPLANATION

La première forme renvoie un objet de type `dict`

Les deux autres formes fonctionnent très bien; la logique ici est qu'on appelle la fonction `set` qui est un constructeur d'ensemble, ou si on préfère, un convertisseur vers le type `set`.

La fonction `set` accepte donc en argument un objet de type quelconque et s'efforce de le traduire dans un ensemble. Lorsqu'on ne passe aucun argument à `set` (option 2), ou qu'on lui passe une liste vide, `set` renvoie naturellement un ensemble vide; on aurait pu utiliser aussi bien, de la même manière, `set(())`, `set({})`, ou même `set('')` pour arriver au même résultat.

Test d'appartenance à un ensemble

On veut écrire un test d'appartenance de l'objet `x` dans un ensemble `s`. Quelle affirmation est correcte ? :

☐ Le test peut s'écrire `x in s` et il prend un temps qui grandit avec la taille de l'ensemble

☒ Le test peut s'écrire `x in s` et il prend un temps constant ✓



Il faut balayer tous les éléments de `s` et les comparer à `x`

EXPLANATION

Les ensembles sont implémentés comme des tables de hachage, au même titre que les dictionnaires. Que `s` soit un `set` ou un `frozenset`, on peut donc tester l'appartenance de `x` avec `x in s`, et cette opération prend un temps qui ne dépend pas de la taille de l'ensemble.

Vous avez utilisé 3 essais sur 3

[A propos](#)

[Aide](#)

[Contact](#)

[Conditions générales d'utilisation](#)

[Charte utilisateurs](#)

[Politique de confidentialité](#)

[Mentions légales](#)



POWERED BY
OPENedX