

Dictionnaires

On utilisera pour ce TP l'éditeur en ligne : Future Coder

Notions travaillées :

- Dictionnaires
- Compréhension, implémentation et production d'algorithmes

I Manipulation de Dictionnaires

Exercice 1. Création de dictionnaire

- Créer un dictionnaire `d0` avec les associations suivante :
 - "nadine": 12
 - "glycerine": 21
 - 12: "nadine"
 - "velociraptor": 12
- Créer un dictionnaire `d1` vide.
- Ajouter dans `d1` l'association "albumine": 42
- Ajouter dans `d1` l'association ("nadine", 12): 12
- Essayer d'ajouter à `d1` l'association ["nadine", 12]: 12
- Essayer d'ajouter à `d1` l'association `d0`: 12
- Ajouter dans `d1` l'association "nadine": ["nadine", 12]

Exercice 2. Modification et opérations

- Associer à `n` le nombre d'association de `d0`.
- Changer dans `d0` la valeur associée à "nadine" par [1, 2, 3].
- Associer dans `d1` la clé "allan" à la valeur associée à "velociraptor" dans `d0`.
- Vérifier si "nadine" est une clé de `d0`.
- Vérifier si "12" est une clé de `d0`.
- Associer à la variable `cles` la liste des clés de `d0`.
- Associer à la variable `assoc` la liste des couples (clé, valeur) de `d1`.

II Algorithmes

Exercice 3. Recherche linéaire.

Implémenter une fonction `recherche(d, x)` qui prend en entrée un dictionnaire, `d`, ne contenant pas de clé `None` et un objet, `x`, et qui renvoie `None` si `x` n'est associé à aucune clé dans `d` et qui renvoie la clé `k` tel que `k` est associée à `x` sinon.

On ajoutera une *assertion* au début de la fonction qui vérifie si `None` est clé de `d`.

Exercice 4. Nombre d'occurrences.

Implémenter une fonction `occurrences(s)` qui prend en entrée une chaîne de caractère, `s`, et renvoie un dictionnaire dont les clés sont les caractères apparaissant au moins une fois dans `s`, et la valeur associée à la clé `k` est le nombre d'occurrences de `k` dans `s`.

Par exemple, `occurrences("nadine_est_magnanime")` renvoie

```
{ 'n': 4,  
  'a': 3,  
  'e': 3,  
  'i': 2,  
  ' ': 2,  
  'm': 2,  
  'd': 1,  
  's': 1,  
  't': 1,  
  'g': 1 }
```

III Pour s'entraîner

Exercice 5. *Clé du maximum.*

Implémenter une fonction `cmax(d)` qui prend en entrée un dictionnaire, `d`, dont toutes les valeurs sont des entiers et qui renvoie la valeur la plus grande de `d`. L'algorithme renvoie `None` si `d` est vide.

On pourra utiliser `-float("inf")` qui a les même propriété que $-\infty$ pour les comparaisons.

Exercice 6. *Somme*

Implémenter une fonction `somme(d)` qui prend en entrée un dictionnaire, `d`, dont toutes les valeurs sont des entiers et qui renvoie la somme des valeurs de `d`.