
TP n°2 - Structures de données natives en Python

Notions abordées

- Liste
- Chaînes de caractères
- Ensemble
- Dictionnaire

 Toutes les fonctions doivent être commentées et testées. Elles doivent notamment être munies d'une description faisant suite à d'éventuelles hypothèses sur leurs arguments placée entre triple guillemets sous la signature de la fonction, (`""" documentation sur plusieurs lignes """`). Les autres commentaires s'écrivent sur des lignes commençant par dièse : `# ligne de commentaire`.

Exercice 1 Quelques listes

Question 1

Définir une fonction `entiers_pairs` qui prend en argument deux entiers `a` et `b`, et qui retourne la liste des entiers pairs compris entre `a` et `b`, ces deux valeurs étant potentiellement incluses.

Question 2 puis

Définir une fonction `indique_diviseurs` qui prend en argument un entier naturel non nul `n`, et qui retourne la liste de taille `n` dont l'élément d'indice `i` indique si `i+1` est un diviseur de `n`.

En déduire une fonction `nb_diviseurs` puis une fonction `est_premier`.

Est-ce un procédé efficace pour tester la primalité ? Justifier par des arguments précis de complexité.

Question 3

Définir une fonction `longueurs_chaines` qui prend en argument une liste de chaînes de caractères et qui retourne la liste de leurs longueurs.

Par exemple `longueurs_chaines(["ohé", "ohé", "matelot", ""])` retourne `[3,3,7,0]`.

Question 4 *

Définir une fonction `groupe_par_3` qui prend en argument une liste de chaînes de caractères et qui retourne la liste de chaînes de 3 caractères, formés des caractères des différentes chaînes de la liste, sans les intervertir, éventuellement complétés par `"_"`.

Par exemple `groupe_par_3(["ohé", "matelot", ""])` retourne `["ohé", "mat", "elo", "t__"]`.

Exercice 2 Sondage

Question 1

Quel type de dictionnaire permettrait de modéliser les réponses à un sondage dont les options de réponses sont oui ou non, et dont les répondants sont identifiés par leur prénom seulement ?

Question 2

Quel type de dictionnaire permettrait de modéliser les réponses à un sondage dont les options de réponses sont oui ou non, et dont les répondant sont identifiés par leur prénom et leur nom de famille ?

Question 3

Quel type de dictionnaire permettrait de modéliser les réponses à un sondage dont les options de réponses sont "info" (encodé par 1), "SI" (encodé par 2), ou "ne sais pas" (encodé par 0) et où les répondants sont identifiés par leur prénom et leur nom de famille ?

Dans la suite de cet exercice les dictionnaires seront de ce type.

Question 4

Définir en Python un tel dictionnaire (ou plusieurs) en vue de réaliser des jeux de tests pour les fonctions suivantes.

Question 5

Définir une fonction `nb_repondants` qui prend en argument un dictionnaire, et qui retourne le nombre de réponses enregistrées.

Question 6

Définir une fonction `nb_votants` qui prend en argument un dictionnaire, et qui retourne le nombre de personnes ayant donné une réponse différente de 0.

Question 7

Définir une fonction `prenoms_votants_pour` qui prend en argument un dictionnaire, et une réponse (possible ou non, mais du bon type) et qui retourne la liste des prénoms des personnes ayant donné cette réponse.

Question 8

Définir une fonction `nouvelle_reponse` qui prend en argument un dictionnaire `D`, un nom, un prénom et une réponse (0, 1 ou 2 seulement) et qui modifie `D` de manière à prendre en compte cette réponse (ajout de réponse ou modification selon le cas).

Question 9

Définir une fonction `affiche_resultats` qui prend en argument un dictionnaire `D`, et qui quelle réponse, hors 0, a obtenu une majorité stricte sur le nombre de votants (pas le nombre de participants). En cas d'égalité on affiche "égalité". On affichera à la suite de cette conclusion le détail du nombre de réponses 0, 1 et 2.

Exercice 3 Ensemble

Dans cet exercice on recode les opérations ensemblistes à savoir l'union `|`, l'intersection `&` et l'intersection avec le complémentaire `-` ("*privé de*"). Ces opérateurs ne doivent donc pas être utilisés dans cet exercice.

Question 1

Définir une fonction `union` qui prend en argument deux ensembles de même type, et qui retourne leur union.

Question 2

Définir une fonction `intersection` qui prend en argument deux ensembles de même type, et qui retourne leur intersection.

Question 3

Définir une fonction `union` qui prend en argument deux ensembles de même type A et B, et qui retourne A privé B.

Question 4

Définir une fonction `union` qui prend en argument deux ensembles de même type A et B, et qui retourne le complémentaire de A dans B.

Exercice 4 Multi-ensemble

Un multi-ensemble peut-être vu comme une fonction d'un ensemble dans \mathbb{N} , la valeur associée à chaque élément étant sa multiplicité dans le multi-ensemble. Dans cet exercice on travaille sur les multi-ensembles d'entiers seulement, et on les note entre double accolades, (*i.e.* entre `{ { }`).

Question 1

Définir une fonction `cardinal` qui prend en entrée un multi-ensemble codé par un dictionnaire et qui retourne le cardinal de ce multi-ensemble, sachant que les éléments doivent être comptés avec multiplicité.

Question 2

Définir une fonction `union` qui prend en entrée deux multi-ensembles codés par des dictionnaires et qui retourne le dictionnaire codant pour leur union. On rappelle que la multiplicité d'un élément dans l'union est le maximum de ses multiplicités dans les ensembles initiaux. Par exemple, l'union de `{ { 1, 1, 2, 3, 1, 2 } }` et `{ { 1, 1, 2, 2, 2, 3 } }` est `{ { 1, 1, 1, 2, 2, 2, 3 } }`.

Question 3

Définir une fonction `intersection` qui prend en entrée deux multi-ensembles codés par des dictionnaires et qui retourne le dictionnaire codant pour leur intersection. On rappelle que la multiplicité d'un élément dans l'intersection est le minimum de ses multiplicités dans les ensembles initiaux. Par exemple, l'intersection de `{ { 1, 1, 2, 3, 1, 2 } }` et `{ { 1, 1, 2, 2, 2, 3 } }` est `{ { 1, 1, 2, 2, 3 } }`.

Question 4

Définir une fonction `ensemble` qui prend en argument un multi-ensemble codé par un dictionnaire et qui retourne l'*ensemble* de ses éléments. (*Cette fonction revient à oublier la multiplicité.*)

Exercice 5 Traduction avec un dictionnaire

Le but de cet exercice est de procéder à une traduction sommaire de phrase en traduisant mot à mot, mais en respectant la ponctuation et les majuscules du texte initial.

Question 1

Définir une fonction `traduit_mot` qui prend en argument un dictionnaire de type `Dict[str, str]`, et un mot, et qui retourne la traduction de ce mot s'il est dans le dictionnaire, et le mot lui-même sinon.

Question 2

Définir une fonction `decoupe_phrase` qui prend en argument une chaîne de caractères `phrase`, qui la découpe au niveau des espaces, et qui retourne la liste des sous-chaîne obtenues (sans espaces).

Par exemple

```
decoupe_phrase("Le lion est mort ce soir.")
retourne ["Le", "lion", "est", "mort", "ce", "soir."],
et decoupe_phrase("Pleut-il ?")
retourne ["Pleut-il", "?"].
```

Question 3

Définir une fonction `est_ponctuation` qui prend en argument une chaîne de caractères `chaine` et teste si cette chaîne est réduite à un symbole de ponctuation, c'est-à-dire un point, une virgule, deux-points, un point-virgule, point d'interrogation ou point d'exclamation.

Question 4

Définir une fonction `extrait_ponctuation` qui prend en argument une liste de chaînes de caractères et qui retourne la liste de chaînes dans laquelle les symboles de ponctuation placés en fin de chaîne (à la suite d'au moins une lettre) sont séparés de cette chaîne et stockés dans une autre chaîne, placée immédiatement après dans la liste.

Par exemple

```
extrait_ponctuation(["Le", "lion", "est", "mort", "ce", "soir."])
retourne ["Le", "lion", "est", "mort", "ce", "soir", "."],
et extrait_ponctuation("Pleut-il ?")
retourne ["Pleut-il", "?"].
```

Question 5

Définir une fonction `initiale_majuscule` qui prend en argument une chaîne de caractères non vide `mot`, et qui teste si le premier caractère de cette chaîne est une lettre majuscule.

Question 6

Grâce aux fonctions précédemment définies, définir une fonction `traduit_phrase` qui prend en entrée une phrase et qui la traduit mot à mot, en conservant la ponctuation et les majuscules. *Cette fonction prend donc en entrée un chaîne de caractères, et retourne une chaîne de caractères.*