

## TP9 : dictionnaires et fonctions

Initiation à l'algorithmique et la programmation

L3 TAL — Semestre 5 (2025)

### Partie 1 : Dictionnaires

#### Problème 1: Manipulations

1. Initialiser un dictionnaire, `student`.
2. Assigner un nom et un prénom aux clés `surname`, et `name`.
3. Assigner une liste de notes à la clé `grades`.
4. Calculer la moyenne des notes (`grades`), et l'assigner à la clé `average`.
5. Soit un dictionnaire `id_number` tel que : `id_number = {"id": 220947}`. Fusionner `student` et `id_number`.
6. Supprimer la clé `grades`.

#### Problème 2: Après un partiel...

Soit deux listes :

`students = ["Ava", "Ada", "Ana"]` (liste d'étudiants)

`midterm = [14, 17, 15]` (liste des notes associées à ces élèves)

1. Écrire un programme qui crée un dictionnaire à partir de ces deux listes : les clés du dictionnaire sont les prénoms, les valeurs associées sont les notes !
2. Soit une troisième liste, `final = [12, 11, 16]`. Écrire un programme qui renvoie un dictionnaire, qui associe à chaque élève (clé), un dictionnaire avec ses notes (clés : `midterm`, `final`).

### Partie 2 : Fonctions et Dictionnaires

#### Problème 3: Dictionnaire de fréquences

Écrire une fonction `word_frequencies()` qui prend en entrée une liste de mots et renvoie un dictionnaire où les clés sont les mots uniques et les valeurs sont le nombre d'occurrences de chaque mot.

Exemple :

```
word_frequencies(["chat", "chien", "chat", "oiseau", "chien", "chat"])
```

Suggestion : vous pouvez utiliser la méthode `.split()` (appliquée à une chaîne de caractères, elle renvoie une liste de chaînes, selon un séparateur donné. Par défaut, le séparateur est un espace.)

Exemple : `"le chat est là".split()` renvoie `["le", "chat", "est", "là"]`

#### Problème 4: Inverser un dictionnaire

Écrire une fonction `invert_dict()` qui prend en entrée un dictionnaire et renvoie un nouveau dictionnaire où les clés et valeurs sont inversées. On suppose que toutes les valeurs du dictionnaire original sont uniques.

Exemple :

```
invert_dict({"a": 1, "b": 2, "c": 3}) renvoie {1: "a", 2: "b", 3: "c"}
```

#### Problème 5: Grouper par première lettre

Écrire une fonction `group_by_first_letter()` qui prend en entrée une liste de mots et renvoie un dictionnaire où les clés sont les premières lettres (en minuscules) et les valeurs sont des listes de mots commençant par cette lettre.

Exemple :

```
group_by_first_letter(["apple", "banana", "avocado", "cherry", "apricot"])  
renvoie {"a": ["apple", "avocado", "apricot"], "b": ["banana"], "c":  
["cherry"]}
```

#### Problème 6: Fusionner des dictionnaires

Écrire une fonction `merge_dicts()` qui prend en entrée deux dictionnaires et renvoie un nouveau dictionnaire contenant toutes les clés des deux dictionnaires. Si une clé existe dans les deux dictionnaires, la valeur du second dictionnaire doit être utilisée.

Exemple :

```
merge_dicts({"a": 1, "b": 2}, {"b": 3, "c": 4}) renvoie {"a": 1, "b": 3,  
"c": 4}
```

Attention, sans utiliser la méthode `.update()` ! (Regarder cependant la méthode `.copy()`)

## Partie 3 : Toujours plus de fonctions

### Problème 7: Combien ?!

Écrire une fonction `count_digits()` qui prend en entrée une chaîne de caractère ou un nombre (entier ou flottant) et qui renvoie le nombre de chiffre que l'entrée contient.

Exemple :

- `count_digits("hello45")` renvoie 2.
- `count_digits(3.1415)` renvoie 5.
- `count_digits(42)` renvoie 2.

Utiliser la méthode `.isdigit()` !

### Problème 8: Préfixe commun

Écrire une fonction `common_prefix()`, qui prend en argument deux chaînes de caractères, `s1` et `s2`, et qui détermine la longueur de leur préfixe commun.

Exemples :

- pour `s1 = "maison"`, et `s2 = "masure"`, la fonction renvoie 2.
- pour `s1 = "maison"`, et `s2 = "mais"`, la fonction renvoie 4.
- pour `s1 = "maison"`, et `s2 = "bicoque"`, la fonction renvoie 0.

### Problème 9: Caractères en double

Écrire une fonction `double_car()` qui prend en entrée une chaîne de caractère `s` et qui renvoie une nouvelle chaîne dans laquelle chaque caractère de la chaîne originale est répété 2 fois.

Exemple :

- `double_car("hello")` doit renvoyer `"hheellllloo"`.

### Problème 10: Un caractère pour un autre...

Écrire une fonction `replace(s, old, new)` qui remplace toutes les occurrences de `old` par `new` (on remplace chaque caractère `old` par `new`).

Exemple :

- `replace("hello", "e", "E")` renvoie `"hEllo"`