

# Dates et dictionnaires en Python

Planche de TD pour un cours dispensé à l'université de Rennes 2

Romain Tavenard

## 1 Organisation de votre code

Pour ce TD, vous créerez un nouveau fichier `td5.py` dans le répertoire que vous avez créé à la première séance. Dans ce fichier, votre code sera organisé de la manière suivante :

```
# Section 1 : les imports (optionnel)

# Section 2 : les fonctions
def ma_premiere_fonction_qui_a_un_meilleur_nom_que_celui_la(n):
    [...]

def mon_autre_fonction(truc, machin):
    [...]

# Section 3 : les tests
print(ma_premiere_fonction_qui_a_un_meilleur_nom_que_celui_la(3))
# [Sortie] 3
```

Notamment, vous définirez vos fonctions en début de fichier et les appels seront listés en fin de fichier. De cette manière, vous pourrez, d'une question à l'autre, réutiliser les fonctions déjà codées au besoin.

## 2 Manipulations de dates

Le type `date` (plus précisément le type `datetime` qui permet de représenter conjointement une date et une heure) est défini dans le module `datetime`. Commencez donc par ajouter l'instruction d'importation de ce module **en début de votre script Python**.

1. Supposons que soit stocké, dans une chaîne de caractères `s`, le contenu suivant : `"24-08, 2021, 16:53"`. Chargez cette date dans une variable `d1` de type `datetime` et affichez le contenu de cette variable.

2. Écrivez une fonction qui prenne en entrée une date et retourne le nombre de secondes écoulées depuis cette date. Combien de secondes se sont écoulées depuis `d1` ?
3. Écrivez une fonction qui prenne en entrée une date `d0` et retourne la date située une semaine après `d0`. À quelle date sera-t-on rendu dans une semaine ?

### 3 Manipulations de base des dictionnaires

4. Écrivez une fonction qui prend en entrée un dictionnaire et retourne la liste de ses éléments sous forme de paires comme retournée par la méthode `items`, triés par clé.
5. Écrivez une fonction qui prend en entrée une chaîne de caractères et retourne un dictionnaire indiquant, pour chaque mot, sa fréquence dans la chaîne fournie.

### 4 Exercices de synthèse

6. On dispose d'un dictionnaire associant à des noms de commerciaux d'une société le nombre de ventes qu'ils ont réalisées. Par exemple :

```
ventes={"Dupont":14, "Hervy":19, "Geoffroy":15, "Layec":21}
```

- a. Écrivez une fonction qui prend en entrée un tel dictionnaire et renvoie le nombre total de ventes dans la société.
- b. Écrivez une fonction qui prend en entrée un tel dictionnaire et renvoie le nom du vendeur ayant réalisé le plus de ventes. Si plusieurs vendeurs sont *ex-aequo* sur ce critère, la fonction devra retourner le nom de l'un d'entre eux.
7. Écrivez une fonction qui prend en entrée une chaîne de caractères comprenant, sur chaque ligne, trois champs séparés par des caractères `;` (un numéro d'étudiant, un nom et un prénom) et retourne un dictionnaire dont les clés sont les numéros d'étudiants lus et les valeurs sont, pour chaque numéro d'étudiant, une chaîne correspondant à la concaténation des prénom et nom de la personne. On pourra tester la fonction avec la chaîne suivante :

```
chaine_etudiants = """213615200;BESNIER;JEAN
213565488;DUPOND;MARC
214665555;DURAND;JULIE"""
```

# Devoir

**\*\*Cet exercice est à rendre sur CURSUS avant la séance de TD de la semaine prochaine. Le rendu se fera sous la forme d'un unique fichier Python**

(pas de version ``.txt`` ou ``.pdf``) structuré  
comme demandé plus haut et contenant le code relatif à cette partie.\*\*

8. Écrivez une fonction qui prend en entrée 1, 2 ou 3 dictionnaires et retourne un dictionnaire fusionnant les informations de ces dictionnaires de la manière suivante : pour chaque clef présente dans au moins un dictionnaire, la valeur associée sera la plus petite des valeurs associées à cette clef dans les dictionnaires passés en argument à la fonction.

Par exemple, pour les dictionnaires suivants :

```
```python
dict1 = {"a": 1, "d": 4, "g": 7}
dict2 = {"a": 1, "d": 2, "h": 8}
dict3 = {"a": 2, "c": 3, "h": 9}
```

La fonction devra retourner :

```
{'a': 1, 'd': 2, 'g': 7, 'h': 8, 'c': 3}
```