Traitement de corpus

Cours n°4 - Dictionnaires et Traitement de textes

Hee-Soo Choi

UFR Sciences Humaines et Sociales
Université de Lorraine

3 novembre 2021

Table de matières

Rappels : La syntaxe

Les deux points après les « if », les « for... », les « def »

```
1 def mention(note):
2     if note >= 16:
3         print("Très Bien")
4     elif 14 <= note < 16:
5         print("Bien")
6     elif 12 <= note < 14:
7         print("Assez Bien")
8     else:
9         print("Pas de mention")
```

L'indentation

Rappels : La notion de fonction

Sans fonction

```
1 a = 4
2 b = 3
3 c = 15
4 d = 20
5 somme1 = a + b
6 somme2 = c + d
```

Avec fonction

```
1 def somme(a, b):
2     resultat = a + b
3     return resultat
4
5 somme1 = somme(4, 3)
6 somme2 = somme(15, 20)
```

$\rightarrow$ Évite de réécrire plusieurs fois le même code pour une opération qui se répète

Les dictionnaires

Définition : dictionnaire

Structure de données permettant d'associer une **clé** à une **valeur**.

```
notes = {"Maths" : 17, "Anglais" : 10, "Philo" : 14}

voiture = {
  "marque": "Audi",
  "modèle" : "A3",
  "année": 2010,
  "couleurs": ["rouge", "blanche", "noire", "bleue"]
}
```

→ Les valeurs et les clés des éléments d'un dictionnaire peuvent être des chaînes de caractères, des entiers, des listes...

Accéder aux éléments d'un dictionnaire

```
notes = {"Maths" : 17, "Anglais" : 10, "Philo" : 14}

# Obtenir les clés du dictionnaire
print(notes.keys())

# Obtenir les valeurs
print(notes.values())
```

```
dict_keys(['Maths', 'Anglais', 'Philo'])
dict_values([17, 10, 14])
```

```
# Obtenir les paires clé-valeur
print(notes.items())
```

```
dict_items([('Maths', 17), ('Anglais', 10), ('Philo', 14)])
```

```
# Obtenir la valeur d'une clé spécifique
print(notes["Anglais"])
```

```
10
```

Ajouter et supprimer des éléments

```
notes = {"Maths" : 17, "Anglais" : 10, "Philo" : 14}

# Ajouter un nouvel élément
notes["Physique"] = 19
print(notes)
```

```
{'Maths': 17, 'Anglais': 10, 'Philo': 14, 'Physique': 19}
```

```
# Retirer un élément
notes.pop("Philo")
print(notes)
```

```
{'Maths': 17, 'Anglais': 10, 'Physique': 19}
```

Parcourir un dictionnaire

```
notes = {"Maths" : 17, "Anglais" : 10, "Philo" : 14}

# Afficher toutes les clés avec une boucle
for element in notes:
    print(element)
```

```
Maths
Anglais
Philo
```

```
# Autre méthode
for cle in notes.keys():
    print(cle)
```

```
Maths
Anglais
Philo
```

Parcourir un dictionnaire

```
# Afficher toutes les valeurs avec une boucle
for valeur in notes.values():
    print(valeur)
```

```
17
10
14
```

```
# Afficher toutes les paires clé-valeur avec une boucle
for element in notes.items():
    print(element)
```

```
('Maths', 17)
('Anglais', 10)
('Philo', 14)
```

Plus d'infos...

Pour en savoir plus sur les dictionnaires :

https://docs.python.org/fr/3/library/stdtypes.htmltypesmapping

Types de traitement

- Retirer les ponctuations
- Retirer certains mots comme les mots-vides
- Couper les phrases en mots
- Compter le nombre d'occurrences des mots
- Faire des statistiques sur les textes

Fonctions utiles

Définition : upper() et lower()

upper() permet de mettre une chaîne de caractères en majuscules.
lower() permet de la mettre en minuscule.

```
1 texte = "Bonjour tout le monde"
2 texte_maj = texte.upper()
3 texte_min = texte.lower()
4
5 print(texte_maj)
6 print(texte_min)
```

```
1 BONJOUR TOUT LE MONDE
2
3 bonjour tout le monde
```

Fonctions utiles

Définition : `split()`

`split()` permet de couper la chaîne de caractères à un séparateur spécifique. La fonction retourne une liste.

```
texte = "Bonjour tout le monde"
resultat = texte.split(" ")

print(resultat)
```

```
["Bonjour", "tout", "le", "monde"]
```

```
texte = "Bonjour tout le monde"
resultat = texte.split("o")

print(resultat)
```

```
["B", "nj", "ur t", "ut le m", "nde"]
```

Fonctions utiles

Définition : `replace()`

`replace()` permet de remplacer une chaîne de caractère par une autre.

```
1 texte = "Bonjour tout le monde"
2 nouveau_texte = texte.replace("Bonjour", "Bonsoir")
3 print(nouveau_texte)
```

```
1 Bonsoir tout le monde
```

Attention

Toutes les occurrences de la chaîne à remplacer seront changées.

Fonctions utiles

Définition : count()

count() permet de compter le nombre d'occurrences d'une chaîne de caractère.

```
1 texte = "Bonjour tout le monde"
2
3 print(texte.count("o"))
4 print(texte.count("ou"))
```

```
1 4
2 2
```

Autres fonctions

Pour explorer d'autres fonctions pour les chaînes de caractères :

https://www.w3schools.com/python/python_strings_methods.asp

Exemple d'application

« Les familles heureuses se ressemblent toutes ; les familles malheureuses sont malheureuses chacune à leur façon. »

Léon Tolstoï, *Anna Karénine*

Exemple d'application

```
texte = "Les familles heureuses se ressemblent toutes ; les familles
    malheureuses sont malheureuses chacune à leur façon."

# Suppression des ponctuations
ponctuations = ".,!?:"
for caractere in texte:
    if caractere in ponctuations:
        texte = texte.replace(caractere, "")

print(texte)
```

```
Les familles heureuses se ressemblent toutes  les familles malheureuses sont
     malheureuses chacune à leur façon
```

```
# Mettre le texte en minuscules
texte_min = texte.lower()
print(texte_min)
```

```
les familles heureuses se ressemblent toutes  les familles malheureuses sont
     malheureuses chacune à leur façon
```

Exemple d'application

```python
# Obtenir les mots du texte dans une liste
mots_texte = texte_min.split(" ")

# Retirer les mots-vides
mots_vides = ["le", "la", "les", "une", "un", "des", "à", "et", "de"]
mots_finaux = []

for mot in mots_texte:
    if mot not in mots_vides:
        mots_finaux.append(mot)

print(mots_finaux)
```

```
['familles', 'heureuses', 'se', 'ressemblent', 'toutes', '', 'familles', '
    malheureuses', 'sont', 'malheureuses', 'chacune', 'leur', 'façon']
```

Exemple d'application

```python
# Obtenir les occurrences de chaque mot dans un dictionnaire

dico = {}
for mot in mots_finaux:
    dico[mot] = mots_finaux.count(mot)

print(dico)
```

```
{'familles': 2, 'heureuses': 1, 'se': 1, 'ressemblent': 1, 'toutes': 1, '': 1, 'malheureuses': 2, 'sont': 1, 'chacune': 1, 'leur': 1, 'façon': 1}
```