

Modèles de Linguistique Computationnelle

CM 5 : formatage / compréhension de listes / expressions régulières et groupes

M1 Langue et Informatique

Crédits: Gaël Lejeune, Karĕn Fort, Iana Atanassova, Djame Seddah, Eleni Kogkitsidou, Olga Seminck

Alice Millour prenom.nom@sorbonne-universite.fr

Yoann Dupont prenom.nom@sorbonne-universite.fr

2020-2021

Sorbonne-Université

Séance d'aujourd'hui

- (une) correction du TD3
- CM 5 :formatage des chaînes de caractères / compréhension de listes / expressions régulières et groupes
- TD 4 : Implémentation d'un agent conversationnel

Plan

Formatage

Compréhensions de listes

Expressions régulières et groupes

la syntaxe `{ }.format(value) ...` pourquoi ?

- lisibilité
- économie de code
- flexibilité

Chaînes de caractères et formatage

On peut utiliser plusieurs formats pour le paramètre à afficher :

- s – strings
- d – decimal integers (base-10)
- f – floating point display
- c – character
- b – binary
- o – octal
- x – hexadecimal with lowercase letters after 9
- X – hexadecimal with uppercase letters after 9
- e – exponent notation

Chaînes de caractères et formatage

Cette syntaxe permet de passer des *paramètres* formatés à une chaîne de caractères

```
# Formattage
valeur_s = "mot"
print("{}".format(valeur_s))

valeur_d = 3
print("{}".format(valeur_d))

valeur_f = 3.5
print("{}".format(valeur_f))

valeur_c = 37
# conversion au format caractère
print("{:c}".format(valeur_c))
print("{} est le code ASCII en base 10 du caractère {:c}".format(valeur_c, valeur_c))
```

```
mot
3
3.5
%
37 est le code ASCII en base 10 du caractère %
```

Chaînes de caractères et formatage

S'il n'y a pas d'ambiguïté, on n'est pas obligé de spécifier les formats :

```
# Mélanger les formats
```

```
name = input("nom : ")
date = input("aujourd'hui, nous sommes le : ")
duree_du_cours = input("la durée du cours est de (en heures) : ")

print('Le {} octobre, {} aura passé {}h dans la salle 219.'.format(date,name, duree_du_cours))
```

```
nom : Alice
aujourd'hui, nous sommes le : 14
la durée du cours est de (en heures) : 2,5
Le 14 octobre, Alice aura passé 2,5h dans la salle 219.
```

Chaînes de caractères et formatage

Le formatage permet d'améliorer la **lisibilité** des expressions :

```
# comment écrire : lundi précède mardi, mercredi, jeudi et vendredi ?  
print(lundi + " précède " + mardi + ", " + mercredi + ", " + jeudi + " et " + vendredi + ".")  
print('{} précède {}, {}, {} et {}'.format(lundi, mardi, mercredi, jeudi, vendredi))
```

```
lundi précède mardi, mercredi, jeudi et vendredi.  
lundi précède mardi, mercredi, jeudi et vendredi.
```


Stocker les paramètres dans une liste

Le formatage permet d'améliorer la **compacité** des expressions :

La liste de valeurs est parcourue grâce à l'opérateur *

```
jours = "Lundi mardi mercredi jeudi vendredi samedi dimanche"
liste_jours = jours.split()
print(liste_jours)

print('{} précède {}, {}, {} et {}'.format(*liste_jours))
```

```
['Lundi', 'mardi', 'mercredi', 'jeudi', 'vendredi', 'samedi', 'dimanche']
Lundi précède mardi, mercredi, jeudi et vendredi.
```

Plan

Formatage

Compréhensions de listes

Expressions régulières et groupes

Compréhension de liste

```
new_list = [function(item) for item in list if condition(item)]
```

```
a = [1,4,2,7,1,9,0,3,4,6,6,6,8,3]
```

```
##### CODE 1 #####
```

```
b = []
```

```
for x in a:
```

```
    if x > 5:
```

```
        b.append(x)
```

```
print(b)
```

```
##### CODE 2 #####
```

```
b = [x for x in a if x > 5]
```

```
print(b)
```

```
[7, 9, 6, 6, 6, 8]
```

```
[7, 9, 6, 6, 6, 8]
```

Les codes 1 et 2 sont équivalents

Formatage

Compréhensions de listes

Expressions régulières et groupes

Expressions régulières et groupes

Les groupes sont des portions d'expressions régulières délimitées par des **parenthèses**

```
# les parenthèses permettent de définir des groupes
jours = "Lundi mardi mercredi jeudi vendredi samedi dimanche"
prefixe_di = [re.search(r'(\w+)di\b', jour) for jour in jours.split()]

print([prefixe_di[i].group(1) for i in range(len(prefixe_di)) if prefixe_di[i] != None])

['Lun', 'mar', 'mercre', 'jeu', 'vendre', 'same']
```

Voir `code_cours.ipynb`