

# code\_cours\_CM5

November 5, 2020

## 1 CM5 (code cours)

Contact : Alice Millour, [alice.millour@sorbonne-universite.fr](mailto:alice.millour@sorbonne-universite.fr)

### 1.1 Formatage de chaînes de caractères

```
[ ]: # Formatage
valeur_s = "mot"
print("{}".format(valeur_s))

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

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

valeur_c = 47
# 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))
```

```
[ ]: # 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))
```

```
[ ]: # une chaîne de caractères peut contenir des éléments inconnus

lundi = "lundi"
mardi = "mardi"
mercredi = "mercredi"
```

```
jeudi = "jeudi"
vendredi = "vendredi"
samedi = "samedi"
dimanche = "dimanche"

# 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))
```

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

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

## 1.2 Compréhension de listes

```
[ ]: 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 #####
def fonction(b) :
    b = [x for x in a if x = 4]
    print(b)
```

## 1.3 Groupes et expressions régulières

```
[ ]: # Les groupes avec match/search (1/2)

import re

jours = "Lundi mardi mercredi jeudi vendredi samedi dimanche"

jour_di = re.search(r'\w+di\b', jours)
#print(jour_di)
```

```

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

#print(prefixe_di)

#print(prefixe_di[0].group(1))
#print(prefixe_di)

liste=[]
for prefixe_jour in prefixe_di:
    if prefixe_jour != None :
        liste.append(prefixe_jour.group(1))
print(liste)

#print(liste)

#print([prefixe_jour.group(1) for prefixe_jour in prefixe_di if prefixe_jour !=
↳None ])

```

```

[ ]: # La fonction group (2/2)

import re

phrase = "Pendant ce temps, Marie cuisine des aubergines"

qui_quoi = re.search(r'(\b\w+\b) cuisine des (\b\w+\b)', phrase)
print(qui_quoi)

# group(0)
print("le motif complet qui a été détecté est \"" + qui_quoi.group(0) + "\"")
# group(1)
print("la personne qui cuisine est \"" + qui_quoi.group(1) + "\"")
# group(2)
print("la personne cuisine des \"" + qui_quoi.group(2) + "\"")

```

```

[ ]: # Les groupes avec findall

phrase = "Marie cuisine des aubergines, Jean cuisine des courgettes, Sébastien
↳cuisine des carottes."

qui_quoi = re.findall(r'(\b\w*\b) cuisine des (\b\w+\b)', phrase)

print("### findall renvoie une liste de tuples correspondant aux groupes qui ont
↳été capturés\n")
print(qui_quoi)

```

```

print("La liste des personnes qui cuisinent est : " + str([partie[0] for partie_
↳in qui_quoi]) )
print("La liste des légumes cuisinés est : " + str([partie[1] for partie in_
↳qui_quoi]))

print()
print("### Bilan du découpage : \n")

print(phrase)
print(str([partie[0] for partie in qui_quoi]) + " cuisinent des " +_
↳str([partie[1] for partie in qui_quoi]))

```

```

[ ]: phrase_et_parentheses = "J'aime bien l'algo (mais pas trop)."  

# on doit "échapper" les parenthèses  

parentheses = re.search(r'(\([\\w\\s]*\\))', phrase_et_parentheses)  

print(parentheses.group(1))

```