

Les fonctions en Python

Romain Tavenard

1 TODO TD fonctions

- Bouger Question 2 plus bas (ils ne sont pas à l'aise avec la récursion, ne la mettre qu'après les durées)
- Expliciter 1 fichier, et son organisation

Le but de cette séance est de manipuler vos premières fonctions en Python. Comme d'habitude, un rappel : la documentation Python est de très bonne qualité : utilisez-la (<https://docs.python.org/3/tutorial/>).

2 Organisation de votre code

Pour ce TD, vous créerez un nouveau fichier `td2.py` dans le répertoire que vous avez créé à la première séance. Dans ce fichier, votre code sera organisé de la manière suivante :

```
def fibonacci(n):  
    [...]  
  
def mon_autre_fonction(truc, machin):  
    [...]  
  
# Tests  
print(fibonacci(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.

3 Échauffement

1. Écrivez, sur papier et en pseudo-code, un algorithme **récuratif** (c'est-à-dire dans lequel une fonction **f** contient un ou plusieurs appels à la même fonction **f** avec des paramètres différents) permettant de calculer le terme de rang n de la suite de Fibonacci définie par :

$$\begin{aligned}f_0 &= f_1 = 1 \\f_n &= f_{n-1} + f_{n-2}, \forall n \geq 2\end{aligned}$$

2. Implémentez en Python l'algorithme précédent. On écrira une fonction **fibonacci** qui prend en argument le rang **n** et renvoie la valeur du terme.

4 Fonctions et procédures en Python

3. Écrivez une fonction **duree_secondes** qui prend en argument une durée en heures, minutes, secondes, exprimée sous la forme de trois variables, et qui renvoie la durée équivalente en nombre de secondes.
4. Écrivez une fonction **delta_secondes** qui retourne la différence (en secondes) entre deux durées exprimées en heures, minutes, secondes. On considère que la première durée est toujours supérieure à la seconde.
5. Écrivez une fonction **table** qui prenne pour argument une base et affiche la table de multiplication de cette base. Par exemple **table(5)** affichera la table de 5, soit une sortie de la forme :

```
5*1=5
5*2=10
[...]
5*10=50
```

6. Proposez une implémentation récursive du calcul de la somme des **n** premiers entiers. On écrira une fonction **somme_entier** qui prend en argument la valeur de **n** et renvoie la somme des entiers.
7. Écrivez une fonction **aire_perimetre_rectangle** qui retourne l'aire et le périmètre d'un rectangle dont les longueurs des côtés sont passés en argument.
8. Analysez la structure des paroles de la chanson *La jument de Michao* et écrivez une fonction qui affiche ces paroles dans le terminal pour un nombre d'années **n** initial passé en argument. Voici la structure de la chanson pour **n=4** :

C'est dans 4 ans je m'en irai. J'entends le loup et le renard chanter.
{2x}

J'entends le loup, le renard et la belette. J'entends le loup et le renard chanter. {2x}

C'est dans 3 ans je m'en irai. La jument de Michao a passé dans le pré.

La jument de Michao et son petit poulain. A passé dans le pré et mangé tout le foin. {2x}

L'hiver viendra les gars, l'hiver viendra. La jument de Michao, elle s'en repentira. {2x}

C'est dans 2 ans je m'en irai. J'entends le loup et le renard chanter. {2x}

J'entends le loup, le renard et la belette. J'entends le loup et le renard chanter. {2x}

C'est dans 1 ans je m'en irai. La jument de Michao a passé dans le pré.

La jument de Michao et son petit poulain. A passé dans le pré et mangé tout le foin. {2x}

L'hiver viendra les gars, l'hiver viendra. La jument de Michao, elle s'en repentira. {2x}

5 Arguments facultatifs et valeurs par défaut

9. Écrivez une fonction `somme` qui prenne en entrée 2, 3, ou 4 nombres et en retourne la somme.
10. Écrivez une version améliorée de la fonction `table`, que vous appellerez `table_multiplication`, qui prenne en entrée une base, un multiplicateur de début et un autre de fin et affiche la table de multiplication de cette base entre les deux bornes fournies. Par exemple `table_multiplication(5, 4, 7)` affichera une sortie de la forme :

```
5*4=20
5*5=25
5*6=30
5*7=35
```

Notez qu'un appel à `table_multiplication(5)` devra avoir le même effet qu'un appel à `table(5)`

6 Exercice de synthèse

11. Écrivez une fonction qui calcule l'[aire d'une ellipse](#). Si l'argument correspondant au demi-petit axe n'est pas spécifié, on supposera qu'il s'agit d'un cercle et que donc le demi-petit axe est égal au demi-grand axe.