

Les fonctions en Python

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

Romain Tavenard

Le but de cette séance est de réaliser vos premiers programmes en Python dans l'IDE (*Integrated Development Environment*) PyCharm. Sachez que la documentation Python est de très bonne qualité : utilisez-la (<https://docs.python.org/3/tutorial/>).

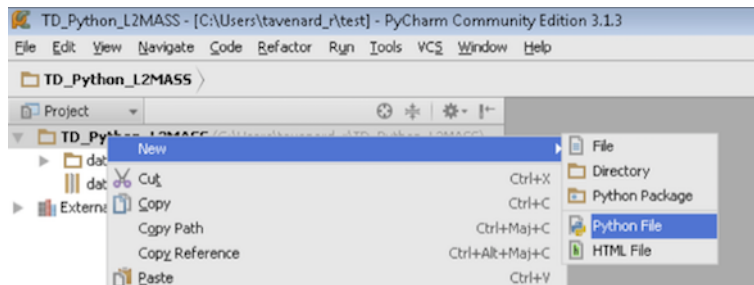
1 Avant-propos

Lors de ce TD, vous allez créer, sur votre disque **M:/**, un répertoire dans lequel vous travaillerez **tout au long du semestre**. Sauf indication contraire, vous créerez, **pour chaque séance de TD, un nouveau fichier Python** dans lequel vous écrirez votre code. Pour ce premier TD, vous serez guidés (un peu plus bas dans cet énoncé) dans la création de votre répertoire/projet Python et du fichier Python correspondant au TD1.

2 Prise en main de PyCharm

Démarrez PyCharm (qui se trouve dans le dossier Développement du menu Démarrer, sous l'item JetBrains) et créez un nouveau projet nommé **L2_Python** dans un endroit bien identifié (vous utiliserez ce projet tout au long du semestre) de votre disque **M:/**.

Ajoutez un nouveau fichier Python à votre projet, que vous nommerez **td1**.



3 Organisation de votre code

Dans ce nouveau nouveau fichier, votre code sera organisé de la manière suivante :

```
# Section 1 : les imports (peut être vide, selon les TD)

# Section 2 : les définitions de fonctions
def fibonacci(n):
    [...]

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

# Section 3 : les tests (un ou plusieurs par fonction codée)
# Vous prendrez soin d'indiquer sous l'appel de fonction
# la sortie observée.
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.

4 Fonctions et procédures en Python

1. 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.
2. 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.
3. 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
```

4. 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.
5. 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

6. Écrivez une fonction `somme` qui prenne en entrée 2, 3, ou 4 nombres et en retourne la somme.
7. 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 Algorithmique récursive

8. 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 :

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

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

7 Exercice de synthèse

10. 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.