

# TD: programmation fonctionnelle:

**Exercice 1: composition de fonctions:** On s'intéresse au programme suivant:

```
0 def addition(a,b):  
    return a+b  
2  
4 def soustraction(a,b):  
    return a-b  
6  
8 def multiplication(a,b):  
    return a*b  
10  
def division(a,b):  
    return a/b
```

fonctionnelle1.py

1. Expliquer ce que font chacune des fonctions de ce programme.
2. Le paradigme fonctionnel est-il respecté?
3. Sans utiliser  $+$ ,  $-$ ,  $*$ ,  $/$ :
  - 3.1 Créer une fonction `operation1` permettant de réaliser l'opération  $(a + b).c$
  - 3.2 Créer une fonction `operation2` permettant de réaliser l'opération  $(a - b)/(c + d)$

**Exercice 2: fonction lambda:** On s'intéresse au programme suivant:

```
0 majorite = lambda x : print("mineur") if x < 18 else print("majeur")
```

fonctionnelle2.py

1. Expliquer ce que fait cette fonction.
2. La tester.

**Exercice 3: fonction map:** La fonction `map` s'utilise comme suit:

```
In [11]: objet = map(lambda x: 2*x, range(10))  
In [12]: print(list(objet))  
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

Figure 1: Fonction `map`

Utiliser la fonction `map` pour créer une fonction qui prend comme argument un tableau et qui retourne chaque élément augmenté de 1. Vérifier que le paradigme fonctionnel est bien respecté.