

## TD 1 - Accompagnement du premier chapitre

- 1 – Section 1 : Essayer la mise en place des espaces de travail et mettre en pratique les points 1 à 8.
- 2 – Section 1.2 : Essayer des commandes dans une session PYTHON .
- 3 – Sections 1.3.1-2 : Utiliser les différents types présentés ainsi que les opérateurs correspondants.
- 4 – Section 1.3.3 :
  1. Ecrire une fonction `f` qui à un `float`  $x$  renvoie  $2x + 3$ . Utiliser la syntaxe avec `def` et celle avec `lambda`.
  2. Ecrire une fonction `positif` qui à un `float`  $x$  renvoie `True` si  $x \geq 0$  et `False` si  $x < 0$ .
- 5 – Section 1.4.1 :
  1. Ecrire une fonction `heavyside` qui à un nombre  $x$  renvoie 1 si  $x \geq 0$  et 0 si  $x < 0$ .
  2. La fonction `critere` suivante prend en entrée trois nombres et renvoie un booléen.

```
def critere(a,b,c):  
    return a**2+b**2==c**2
```

Ecrire en utilisant cette fonction une fonction `pythagore` telle que pour trois nombres `a,b,c`, `pythagore(a,b,c)` renvoie `True` si un triangle de côtés de longueurs `a,b,c` est rectangle, `False` sinon.
  3. Ecrire une fonction `eq1` telle que `eq1(a,b)` renvoie l'ensemble des solutions de l'équation en  $x$  :  $ax + b = 0$ . Attention à bien traiter tous les cas.
  4. Ecrire une fonction `eq2` telle que `eq2(a,b,c)` renvoie l'ensemble des solutions *réelles* de l'équation en  $x$  :  $ax^2 + bx + c = 0$ . Attention à bien traiter tous les cas, et utiliser la fonction `eq1` quand elle s'avère utile.
- 6 – Section 1.4.2 :
  1. Ecrire une fonction `somme_entiers` telle que pour  $n$  entier, `somme_entiers(n)` renvoie la somme des  $n$  premiers nombres entiers positifs.
  2. Ecrire une fonction `somme_impairs` telle que pour  $n$  entier, `somme_impairs(n)` renvoie la somme des  $n$  premiers nombres impairs positifs.
  3. Ecrire une fonction `fact` telle que pour  $n$  entier, `fact(n)` renvoie la factorielle de  $n$  :
$$n! = 1 \times 2 \times \cdots \times n.$$
  4. Ecrire une fonction `somme` telle que pour une liste  $l$  d'entiers, `somme(l)` renvoie la somme des éléments de  $l$ .
  5. Ecrire une fonction `produit` telle que pour une liste  $l$  d'entiers, `produit(l)` renvoie le produit des éléments de  $l$ .