

# Info101 - TD4

L'objectif du troisième TD est d'introduire les tableaux, tout en continuant d'utiliser boucles for et structures conditionnelles. Les concepts vus dans ce TD sont :

- les fonctions (toujours !),
- les structures conditionnelles (if ... elif ... else ...),
- les boucles (for i in range(0, n, 1) : ...),
- les tableaux.

## OPÉRATIONS DE BASE SUR LES TABLEAUX

**Exercice 1.** Quelle est la valeur des expressions *Python* suivantes ?

```
[1, 2, 3] + [4]           len(["Hello, I'm Bob!"])
[0] * 10                 ["bla"] * 4
len([1, 1, 2, 3, 5, 8])  len([42, "Blabla", True]) + 4
[1, 2, 3, 4][2]         (["a", "b"] + ["c"])[0]
```

*Note* : on dispose pour les tableaux d'opérations similaires à celles des chaînes de caractères : la concaténation (notée +), la répétition (notée \*), la longueur (notée len(t)) et l'accès à un élément (notée t[i]).

**Exercice 2.** On entre dans l'interpréteur python les lignes suivantes :

```
>>> t1 = [0] * 5           >>> t2 = [1, 2, 3, 4]
>>> t1[2] = 42            >>> t3 = t1 + t2
```

Quelle est la valeur des expressions *Python* suivantes ?

```
t1[0]                     len(t3) + 12
t1[2]                     len(t1 + t2 + t3)
t1 * 2                    t3[len(t2)]
t2 + t1                   t3 + [1] * 2 + [len(t2)]
```

**Exercice 3.** Après avoir entré les lignes suivantes, quelle est la valeur de t1 et t2 ? Inversez la deuxième et la troisième ligne puis répondez à nouveau.

```
>>> t1 = [0, 1, 2, 3, 4]
>>> t1[3] = t1[2] + 41
>>> t2 = t1 + [0] * 4
```

## TABLEAUX ET NOMBRES

**Exercice 4.** Définissez une fonction `sum` prenant en argument un tableau de nombres `t` et qui en calcule la somme. Définissez ensuite une fonction `prod` qui calcule le produit des éléments d'un tableau de nombres `t`, puis une fonction `moyenne` qui calcule la moyenne des éléments du tableau.

**Exercice 5.** Définissez une fonction `max` qui prend en argument un tableau de nombres `t` non vide, et qui retourne leur maximum. Utilisez ensuite cette fonction pour redéfinir les fonction `max2` et `max3` du premier TD.

**Exercice 6.** Définissez une fonction `table` qui prend en argument un entier `n`, et qui retourne un tableau contenant les 20 premiers multiples de `n`.

**Exercice 7.** Définissez une fonction `scal_mul` qui prend en argument un nombre `n` ainsi qu'un tableau de nombres `t` et qui en retourne une copie où chaque nombre a été multiplié par `n`.

**Exercice 8.** Définissez une fonction `zip_max` qui prend en argument deux tableaux `t1` et `t2` et qui retourne le tableau des nombre maximums de ces deux tableaux deux à deux.

```
>>> zip_max([1, 2, 3], [4, 0, 2])
[4, 2, 3]
```

**Exercice 9.** Définissez une fonction `pair` qui prend en argument un tableau d'entiers `t` et qui retourne le même tableau auquel on a retiré les nombres impairs.

## TABLEAUX ET CHAÎNES

**Exercice 10.** Définissez une fonction `concat` qui prend en argument un tableau de chaînes de caractères `t` et qui retourne la concaténation des éléments de `t`.

**Exercice 11.** Définissez une fonction `chars` qui prend en argument une chaîne de caractères et qui retourne le tableau de ses caractères. On aura par exemple :

```
>>> chars("Hello")
["H", "e", "l", "l", "o"]
```

**Exercice 12.** Définissez une fonction `mots` qui prend en argument une chaîne de caractères `s` et qui retourne le tableau des mots de `s`.

```
>>> mots("Bonjour je m'appelle Bob!")
["Bonjour", "je", "m'appelle", "Bob!"]
```