



# Algorithmie

04 – Les tableaux

**DWWM PE6 LYON**

Barthélémy DELUY – CC-BY-NC-SA 4.0

# Introduction

La plupart des langages proposent des types de variables appelés « types primitifs » : nombres entiers, nombres décimaux, chaînes de caractères, booléens.

Ils proposent également au moins un type dit « complexe » : les tableaux (ou listes).

# Présentation des tableaux

En informatique, un tableau est une liste non ordonnée de valeurs, stockées dans la même variable.

Pour déclarer un tableau, on ajoute « [ ] » au nom de la variable :

Déclarer `mon_tableau[ ]`

Pour ajouter la valeur 4 à la fin du tableau, on utilise :

`mon_tableau[] ← 4`

Attention : il n'existe pas de fonction universelle permettant d'insérer un élément à un emplacement spécifique du tableau, par défaut on ajoute toujours à la fin !

# Présentation des tableaux

Pour accéder à un élément du tableau (récupérer la valeur située à la n-ième position dans la liste), on utilise :

Déclarer `ma_variable`  $\leftarrow$  `mon_tableau[5]`

ATTENTION : les indices du tableau (= les numéros des positions) commencent à 0 : le premier élément du tableau est situé dans `mon_tableau[0]`, le deuxième dans `mon_tableau[1]`, etc.

Pour connaître la taille du tableau, on utilisera `taille(mon_tableau)` en algo.

# Présentation des tableaux

Les éléments du tableau sont donc indexés (positionnés) de 0 à (taille du tableau – 1) : les valeurs contenues dans un tableau de 10 éléments seront indexées de 0 à 9.

Comme pour toutes les variables, on peut remplacer un élément dans un tableau :

`mon_tableau[2] ← 5`

`mon_tableau[2] ← 7`

# Présentation des tableaux

ATTENTION : on ne peut pas affecter une valeur « en-dehors » du tableau.

Si j'ai un tableau de 10 éléments, je ne peux pas faire :

`mon_tableau[20] ← « raté ! »`

Certains langages vous autoriseront à le faire, d'autres crasheront, d'autres placeront la valeur « raté ! » à la fin du tableau (donc en position 11 au lieu de 20).

Considérez que ça ne fonctionne pas quoiqu'il arrive.

# Rappels sur la boucle TANTQUE

Déclarer compteur  $\leftarrow 0$

TANT QUE compteur  $<$  VALEUR FAIRE

[instructions à répéter]

compteur  $\leftarrow$  compteur  $+ 1$

FIN TANT QUE

# Parcourir un tableau

Déclarer compteur  $\leftarrow 0$

TANT QUE compteur  $<$  taille(mon\_tableau) FAIRE

[instructions à répéter]

compteur  $\leftarrow$  compteur + 1

FIN TANT QUE



# Conclusion

