

# TABLEAUX ET INDEX

AJOUT D'ÉLÉMENTS, AFFICHAGES D'ÉLÉMENTS, TAILLE...

UNE VARIABLE DE TYPE NOMBRE  
NE PEUT CONTENIR  
**QU'UN SEUL NOMBRE À LA FOIS**

IL EXISTE DES VARIABLES  
CAPABLES DE CONTENIR  
**PLUSIEURS VALEURS**  
LES UNES À LA SUITE DES AUTRES

IL S'AGIT DES **TABLEAUX**

# QU'EST-CE QU'UN TABLEAU ?

- UN TABLEAU EST UNE VARIABLE QUI PEUT CONTENIR UN ENSEMBLE DE VALEURS
- CHAQUE VALEUR CONTENUE DANS LE TABLEAU POSSÈDE UNE POSITION DANS LE TABLEAU : ON APPELLE CE NUMÉRO DE POSITION UN INDICE (OU INDEX)
- DANS ALGOBOX, LES TABLEAUX SONT IMPROPREMENT APPELÉS « LISTE ». CES LISTES CONTIENNENT FORCÉMENT DES NOMBRES

EN ANGLAIS ON DIT « ARRAY »

LES TABLEAUX  
PEUVENT ÊTRES PARCOURUS  
DANS DES BOUCLES

CE SONT DE BONS OUTILS  
POUR FAIRE DES  
TRAITEMENTS À LA CHAINE

# EXEMPLE : LOGICIEL D'INVENTAIRE DES FACTURES

- IMAGINONS QUE DANS UN PROGRAMME, NOUS AYONS BESOIN SIMULTANÉMENT DE 12 MONTANTS DE FACTURE (PAR EXEMPLE POUR CALCULER UNE MOYENNE OU POUR TROUVER LA PLUS CHÈRE).
- C'EST UNE MAUVAISE IDÉE DE CRÉER 12 VARIABLES (FACT1, FACT2, FACT3, ETC.) : PEU ÉVOLUTIF, PEU PERFORMANT, PEU ÉLÉGANT...

# EXEMPLE : LOGICIEL D'INVENTAIRE DES FACTURES

- D'AILLEURS, SI ON VEUT CALCULER LA MOYENNE, CECI DONNERA LIEU À UNE OPÉRATION UN PEU TROP LOURDE, UN PEU TROP VERBEUSE :
- **AFFECTER** montantMoyen = (Fact1 + Fact2 + Fact3 + Fact4 + Fact5 + Fact6 + Fact7 + Fact8 + Fact9 + Fact10 + Fact11 + Fact12) / 12

# EXEMPLE : LOGICIEL D'INVENTAIRE DES FACTURES

- DE PLUS, SI L'ON VEUT SAISIR N MONTANTS DE FACTURE (UN NOMBRE DE FACTURE QUI NE SERAIT PAS CONNU À L'AVANCE) : C'EST IMPOSSIBLE !
- **UN TABLEAU SERA DONC LA SOLUTION IDÉALE**, CAR IL PERMETTRA DE RASSEMBLER TOUTES CES VARIABLES EN UNE SEULE, AU SEIN DE LAQUELLE CHAQUE MONTANT DE FACTURE SERA DÉSIGNÉE PAR UN NUMÉRO (L'INDEX).
- C'EST BEAUCOUP PLUS PRATIQUE, PROPRE, ÉVOLUTIF, ÉLÉGANT ET PERFORMANT

DANS ALGOBOX,  
LES TABLEAUX SONT  
**IMPROPREMENT** APPELÉS  
**« LISTE »**

MAIS CE SONT BEL ET BIEN  
**DES TABLEAUX**

# COMMENT REmplir un TABLEAU AVEC DES DONNÉES EN DUR

## VARIABLES

facture **EST\_DU\_TYPE** Liste

## DEBUT\_ALGORITHME

**AFFECTER** facture[0] = 72.15

**AFFECTER** facture[1] = 63.80

**AFFECTER** facture[2] = 19.44

## FIN\_ALGORITHME



**À SAVOIR : EN PROGRAMMATION, ON COMMENCE TOUJOURS PAR REMPLIR UN TABLEAU À LA POSITION 0 !**

# COMMENT REmplir un TABLEAU AVEC DES DONNÉES DYNAMIQUES

## VARIABLES

facture **EST\_DU\_TYPE** Liste

## **DEBUT\_ALGORITHME**

**LIRE** facture[0]

**LIRE** facture[1]

**LIRE** facture[2] **ASTUCE : AU MOYEN  
D'UNE BOUCLE**

## **FIN\_ALGORITHME**

**« POUR », IL SERA  
FACILEMENT POSSIBLE DE  
GÉNÉRER UN GRAND  
TABLEAU**



# COMMENT AFFICHER LA DEUXIÈME VALEUR DU TABLEAU

## VARIABLES

```
facture EST_DU_TYPE Liste  
DEBUT_ALGORITHME  
    AFFICHER facture[1]  
FIN_ALGORITHME
```



EXPLICATION : LE PREMIER ÉLÉMENT DU TABLEAU EST EN POSITION 0,  
DONC LE DEUXIÈME EST EN POSITION 1

COMMENT  
AFFICHER LA  
VALEUR SITUÉE  
À LA POSITION  
CHOISIE PAR  
L'UTILISATEUR  
DANS UN  
TABLEAU

## VARIABLES

facture **EST\_DU\_TYPE** Liste

index **EST\_DU\_TYPE** Nombre

## DEBUT\_ALGORITHME

**AFFICHER** "Lire quelle position ?"

**LIRE** index

**AFFICHER** facture[index]

## FIN\_ALGORITHME

# COMMENT CONNAITRE LA TAILLE D'UN TABLEAU ?

LA « TAILLE » D'UN TABLEAU CORRESPOND AU NOMBRE D'ÉLÉMENTS DANS UN TABLEAU

AFFICHER "Vous venez de saisir "  
AFFICHERCALCUL facture.length  
AFFICHER " factures."



IMPORTANT : L'INDEX N°0 DU TABLEAU DOIT ÊTRE UTILISÉ POUR OBTENIR UNE TAILLE QUI SOIT JUSTE

# COMMENT AFFICHER LA DERNIÈRE VALEUR DU TABLEAU

## VARIABLES

facture EST\_DU\_TYPE Liste

**DEBUT\_ALGORITHME**

AFFICHER facture[facture.length - 1]

**FIN\_ALGORITHME**



CECI FONCTIONNE  
PEUT IMPORTE  
LA TAILLE DU TABLEAU

```
▼ VARIABLES
  |-facture EST_DU_TYPE LISTE
  |-index EST_DU_TYPE NOMBRE
  |-i EST_DU_TYPE NOMBRE

▼ DEBUT_ALGORITHME
  |-AFFICHER "Bienvenue dans le logiciel de suivi des factures"
  |-AFFICHER "Saisissez le montant de vos trois dernières factures"
  |-LIRE facture[0]
  |-LIRE facture[1]
  |-LIRE facture[2]
  |-AFFICHER "Vous venez de saisir "
  |-AFFICHERCALCUL facture.length
  |-AFFICHER " factures."
  |-AFFICHER "Quel n° de facture voulez-vous retrouver ? 0, 1 ou 2 ?"
  |-LIRE index
  |-AFFICHER "La facture était était de "
  |-AFFICHER facture[index]
  |-AFFICHER "€"
  |-AFFICHER "Liste des factures :"

▼ POUR i ALLANT_DE 0 A 2
  |-DEBUT_POUR
  |-AFFICHER "Position n°"
  |-AFFICHER i
  |-AFFICHER ":"
  |-AFFICHER facture[i]
  |-FIN_POUR

FIN_ALGORITHME
```

# DÉMONSTRATION AVEC ALGOBOX

# DÉFI TECHNIQUE « 07a. Factures et calcul de moyenne.alg »

```
VARIABLES
facture EST_DU_TYPE LISTE
index EST_DU_TYPE NOMBRE
i EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME
AFFICHER "Bienvenue dans le logiciel de suivi des factures"
AFFICHER "Saisissez le montant de vos trois dernières factures"
LIRE facture[0]
LIRE facture[1]
LIRE facture[2]
AFFICHER "Vous venez de saisir "
AFFICHERCALCUL facture.length
AFFICHER " factures."
AFFICHER "Quel n° de facture voulez-vous retrouver ? 0, 1 ou 2 ?"
LIRE index
AFFICHER "La facture était était de "
AFFICHER facture[index]
AFFICHER "€"
AFFICHER "Liste des factures :"

POUR i ALLANT_DE 0 A 2
DEBUT_POUR
AFFICHER "Position n°"
AFFICHER i
AFFICHER ":"
AFFICHER facture[i]
FIN_POUR

FIN_ALGORITHME
```

**AMÉLIOREZ CET ALGO :**

- A. AFFICHEZ LA SOMME TOTALE DES FACTURES À LA FIN**
- B. AFFICHEZ LA MOYENNE DES FACTURES À LA FIN**
- C. PERMETTEZ À L'UTILISATEUR D'INDIQUER LE NOMBRE DE FACTURE DE SON CHOIX (À DEMANDER AU DÉBUT)**

**(ADAPTEZ L'ALGO EN CONSÉQUENCE)**

## DÉFI TECHNIQUE « 07b. Ajouter deux tableaux.alg »

**CONCEVEZ UN ALGORITHME  
QUI DEMANDE LA SAISIE DE  
DEUX TABLEAUX  
AYANT 6 VALEURS CHACUN  
PUIS CONSTRUISEZ ET AFFICHEZ  
UN 3<sup>E</sup> TABLEAU À 6 VALEURS  
QUI AFFICHERA LA SOMME  
DES DEUX AUTRES TABLEAUX**

4	7	2	0	3	3
---	---	---	---	---	---

2	3	25	9	5	3
---	---	----	---	---	---

6	10	27	9	8	6
---	----	----	---	---	---

VEILLIEZ À BIEN SAUVEGARDER VOTRE  
ALGORITHME UNE FOIS TERMINÉ

## DÉFI TECHNIQUE « 07c. Multiplier deux tableaux.alg »

**CONCEVEZ UN ALGORITHME  
QUI DEMANDE LA SAISIE DE  
DEUX TABLEAUX  
UN DE 4 VALEURS  
ET UN DE 6 VALEURS**

PUIS AFFICHEZ LE RÉSULTAT  
DE LA MULTIPLICATION DE TOUS  
LES NOMBRES DU PREMIER  
TABLEAUX AVEC CHAQUE  
NOMBRE DU DEUXIÈME TABLEAU

$$2 \times 4 + 2 \times 2 + 2 \times 6 + 2 \times 8 + 2 \times 3 + 2 \times 5 + \\ 5 \times 4 + 5 \times 2 + 5 \times 6 + 5 \times 8 + 5 \times 3 \dots \text{ ETC.}$$



2	5	3	4
---	---	---	---

4	2	6	8	3	5
---	---	---	---	---	---

392

VEILLIEZ À BIEN SAUVEGARDER VOTRE  
ALGORITHME UNE FOIS TERMINÉ

# DÉFI TECHNIQUE « 07d. Rendu de monnaire.alg »

## CONCEVEZ UN ALGORITHME DE RENDU DE MONNAIE

- DÉCLAREZ CE TABLEAU DANS L'ALGORITHME :

500	200	100	50	20	10	5	2	1	0,50	0,20	0,10	0,05	0,02	0,01
-----	-----	-----	----	----	----	---	---	---	------	------	------	------	------	------

IL CORRESPOND AUX BILLETS ET AUX PIÈCES

- L'ALGORITHME DEMANDE LE MONTANT DE LA NOTE À PAYER
- L'ALGORITHME INDIQUE COMBIEN LE CLIENT A PAYÉ (EN ESPÈCES)
- L'ALGORITHME INDIQUE EN DÉTAIL

CE QUE LE CAISSIER DOIT LUI RENDRE

IMPORTANT : L'ALGORITHME DOIT TOUJOURS

FAIRE EN SORTE DE RENDRE LE MOINS DE BILLETS ET DE  
PIÈCES POSSIBLE (COMME DANS LA RÉALITÉ)

ON NE REND PAS LA MONNAIE EN PIÈCES DE 1 CENTIMES

BIENVENUE DANS CE LOGICIEL DE  
RENDU DE MONNAIE !

ENTRER MONTANTNOTE : 53.77

ENTRER MONTANTPAYÉ : 200

IL FAUT RENDRE 146,23 €, SOIT :

1x 100 € 2x 20 € 1x 5 € 1x 1 €  
1x 0,2 € 1x 0,02 € 1x 0,01 €

AU REVOIR

VEILLIEZ À BIEN SAUVEGARDER VOTRE ALGORITHME UNE FOIS TERMINÉ

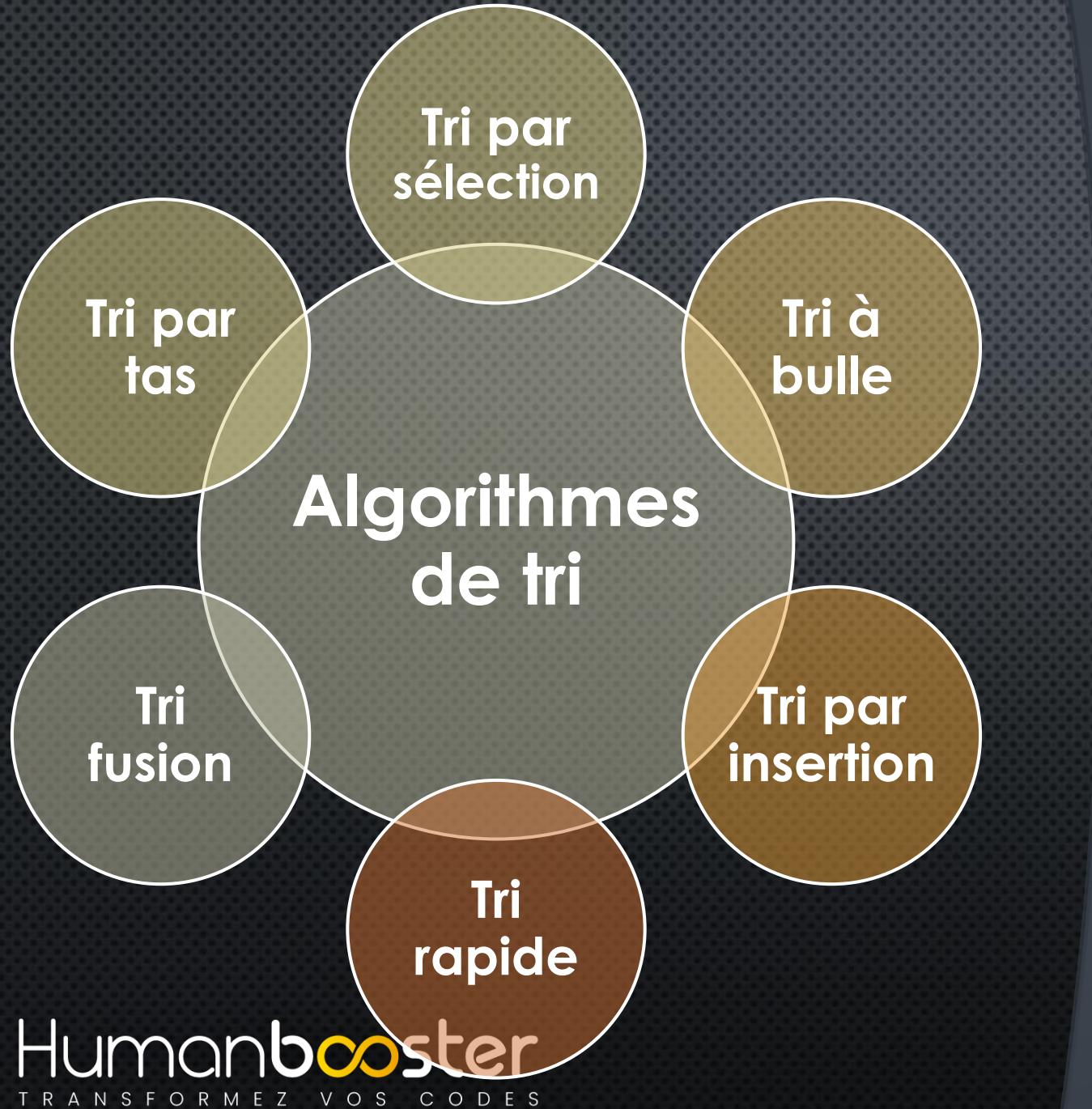
**DÉFI TECHNIQUE** « 07e. Manipulation de tableaux et tri.alg »

**CONCEVEZ UN ALGORITHME QUI PERMET À L'UTILISATEUR DE SAISIR **UN TABLEAU** DE LA **TAILLE DE SON CHOIX PUIS** QUI RÉPOND À CES **COMMANDES** :**

- « **Q** » : QUITTE LE LOGICIEL
  - « **L** » : DONNE LA TAILLE DU TABLEAU
  - « **A** » : AFFICHE LE TABLEAU ENTIER
  - « **R** » : DEMANDE UNE VALEUR NUMÉRIQUE À L'UTILISATEUR, PUIS CHERCHE DANS LE TABLEAU ET INDIQUE SI LA VALEUR EXISTE DANS LE TABLEAU OU PAS. SI ELLE EXISTE : L'INDEX EST AFFICHÉ
  - « **M** » : RETOURNE LA PLUS HAUTE VALEUR DU TABLEAU
  - « **T** » : TRIE LE TABLEAU DU PLUS PETIT AU PLUS GRAND
- OPTION (BONUS) POUR « **R** » :  
INDIQUE LE NOMBRE DE DOUBLONS
- 
- 
- 

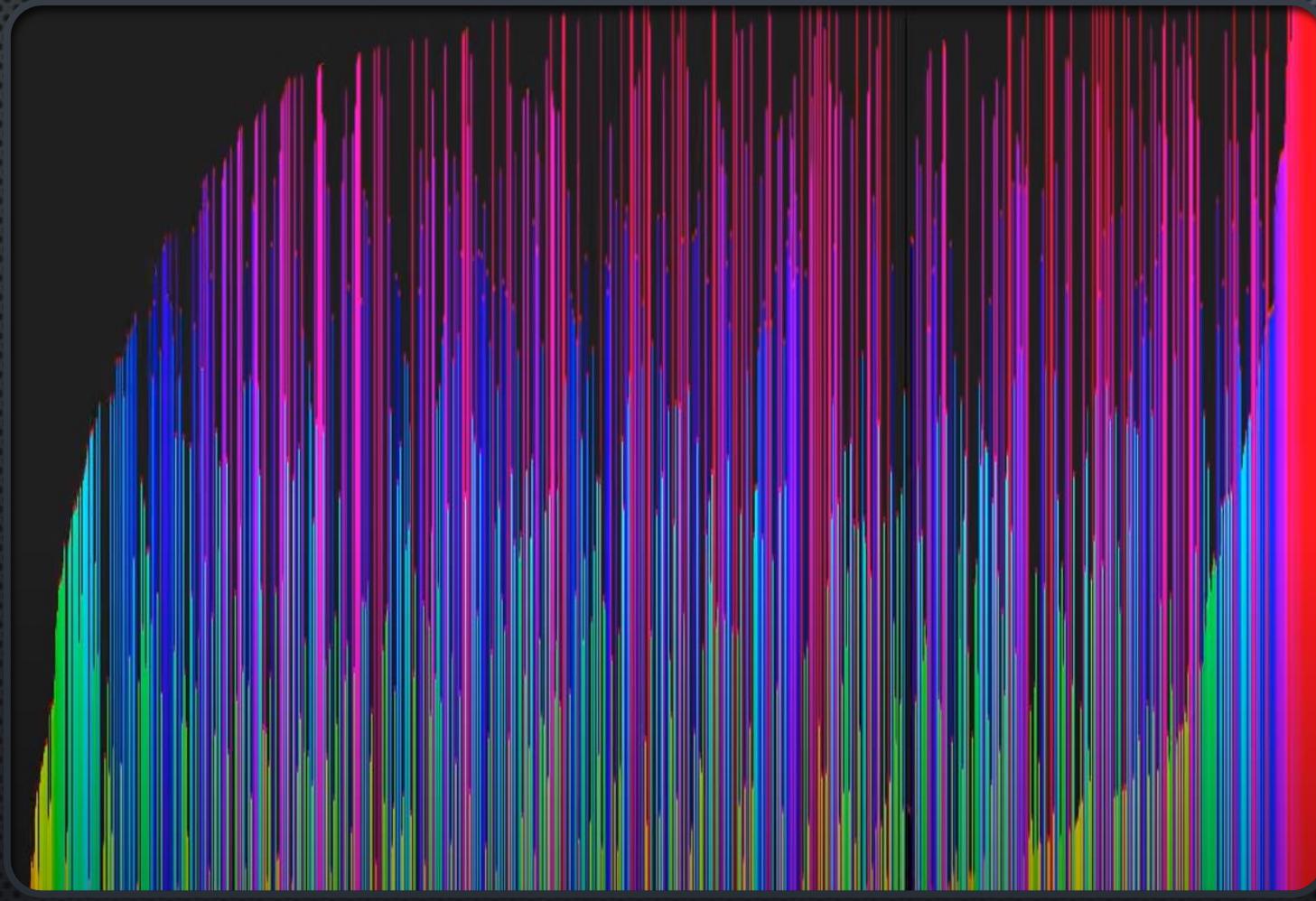
OPTION (BONUS) :

- « **D** » : SUPPRIME TOUS LES DOUBLONS DU TABLEAU AVEC « `tableau.splice(index)` »



- TRIER UN TABLEAU EST UN PROBLÈME BIEN CONNU EN ALGORITHMIQUE : IL EXISTE DE NOMBREUSES APPROCHES
- LES DIFFÉRENTS ALGORITHMES N'ONT PAS LA MÊME COMPLEXITÉ ; LEUR PERFORMANCE N'EST PAS LA MÊME SELON LA SITUATION (TABLEAU EN PARTIE TRIÉ, GRAND TABLEAU, ETC.)
- UN TRI PEUT SE FAIRE SUR DES NOMBRES ENTIERS, DES NOMBRES RÉELS, DES CHAINES, ETC.
- VOUS TROUVEREZ DE NOMBREUX EXEMPLES SUR INTERNET ET DÉMONSTRATION SUR YOUTUBE

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=RA79TDfotno](https://www.youtube.com/watch?v=ra79TDfotno)



**VIDÉO  
ILLUSTRENT LES  
ALGORITHMES  
DE TRI EN  
SON & IMAGE**

[HTTPS://WWW.YOUTUBE.COM  
/WATCH?v=QMOTL6PPCl0](https://www.youtube.com/watch?v=QMOTL6PPCl0)



IL EST POSSIBLE DE FAIRE  
DES TABLEAUX  
**DANS DES TABLEAUX**  
ON PARLE ALORS DE  
**TABLEAUX**  
**À DEUX DIMENSIONS**

EXEMPLE : SUDOKU

IL EST AUSSI POSSIBLE DE FAIRE  
DES TABLEAUX DE TABLEAUX  
DE TABLEAUX !

ON PARLE ALORS DE  
**TABLEAUX**  
**À TROIS DIMENSIONS**  
ET AINSI DE SUITE...

EN PROGRAMMATION ORIENTÉE OBJET  
(PLUS TARD)  
ON UTILISERA ÉGALEMENT  
**LES COLLECTIONS**  
DE MANIÈRE TRÈS COMPARABLE  
**AUX TABLEAUX**