

## Les matrices : les bases

Les matrices, comme les listes vues lundi dernier, permettent de stocker des séries de données. Par contre, ce sont des tableaux à 2 dimensions. Pour visualiser vos matrices, ouvrez l'application RUN-MAT puis l'onglet (Mat) [F1] ou [F3]. Sur certaines calculatrices, les matrices sont directement accessibles depuis l'application MAT.

► Notez que les Graph 20/25 (+ et Pro) ne possèdent pas les matrices. Pour suivre ce cours, [téléchargez l'émulateur officiel et gratuit de la Graph 85 SD \(v1.02.000\)](#). Cet émulateur est aussi en téléchargement sur le site de [Casio Educational online service](#).

?→Mat A[a,b]

Comme c'est un tableau à deux dimensions, il faut donc entrer deux coordonnées **a (position horizontale)** et **b (position verticale)** de la case. Vous pouvez utiliser jusqu'à 26 matrices, de A à Z (représentées par des lettres, contrairement aux listes qu'on les désigne par des numéros).

Les dimensions des matrices sont limitées à 255\*255 ou à 999\*999. Vous ne pourrez jamais créer des matrices aussi grandes car les calculatrices n'ont pas assez de mémoire (entre 24 à 144 Ko). On parle bien en Ko alors que nos ordinateurs ou même nos téléphones portables, ce sont des Go. Mais ne soyez pas rebutés par cette petite quantité de mémoire, c'est suffisant pour beaucoup de jeux.

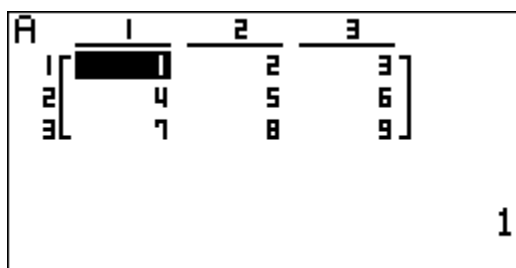
## Créer des matrices

Voici quelques techniques pour créer des matrices :

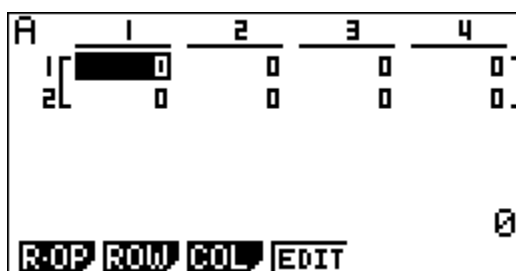
1) [[1,2,3][4,5,6][7,8,9]]→Mat A

2) {2,4}→Dim Mat A

- Pour créer une matrice, on utilise les crochets [ ]. On les obtient avec la combinaison [SHIFT] + [+] ou [-]. N'oubliez pas les deux crochets ouverts au début : le premier crochet délimite la matrice, le second crochet, la ligne. En effet, chaque ligne de la matrice est délimitée des crochets. Le code ci-dessus génère la matrice de l'image ci-contre. Bien évidemment, toutes les lignes doivent avoir les mêmes longueurs, sinon une erreur se produira.



- La seconde technique qui doit vous rappeler celle des listes avec la fonction Dim. Les deux nombres entre les accolades { } sont des dimensions de la matrice. Elle s'utilise de cette façon : {<nombre de lignes>,<nombre de colonnes>}→Dim Mat <lettre>. Toutes les cases de la matrice initialisée avec Dim prennent la valeur 0.



## Le transfert de listes à matrice et inversement

Il est possible de créer une matrice à partir de plusieurs listes. Dans ce cas, les listes concernées devront toutes avoir la même dimension. On utilise la fonction List→Mat ([OPTN]+[F1]+[F2] (L→M)). Supposons : on veut convertir les List 1, 2 et 3 en une Matrice C. La syntaxe est la suivante : List→Mat(<listes à convertir, séparées par des virgules>).

List→Mat(1,2,3)→Mat C

	List 1	List 2	List 3	List 4
1	1	1	12	
2	2	2	2	
3				
4				
5				

LIST CPLY CALC HYP PROB ▸

	1	2	3
1	1	1	12
2	2	2	2

1

On peut très bien ne convertir qu'une seule liste, dans ce cas, on obtiendra une matrice avec une seule colonne. Le maximum étant le nombre maximum de listes (de 6 à 26 selon votre modèle) et la mémoire qu'il vous reste.

L'opération inverse, d'une matrice à une liste, est également possible. Plus exactement, d'une colonne de la matrice vers une liste. On utilise Mat→List(<lettre de la matrice>,<numéro de la colonne>).

Mat→List(A,1)→List 2

A	1	2	3
1	1	2	3
2	4	5	6
3	7	8	9

R-OP ROW COL EDIT 1

	List 1	List 2	List 3	List 4
1		1		
2		4		
3		7		
4				
5				

GRAPH CALC TEST DISTR DIST ▸

## Fusionner des matrices

La fonction Augment( permet la fusion de deux matrices à condition qu'elles aient le même nombre de lignes. Via la touche [OPTN], Augment( se trouve dans l'onglet (Mat) puis [F5] (Aug).

Augment(Mat A,Mat B)→Mat C

Rappelez-vous d'une chose avec cette fonction. Certes, elle peut se révéler très pratique, mais elle consomme énormément de mémoire. Elle doit d'abord créer une matrice temporaire (Ans) pour la fusion. Avec des matrices très lourdes, vous aurez un message avec « Erreur Mémoire ».

*Je n'ai pas pointé ce problème, les listes n'ont qu'une dimension, dans la majorité des cas, vous ne rencontrerez pas ce problème de mémoire.*

## **L'utilité des matrices ?**

Dans les jeux, les matrices ont une infinité d'applications possibles, comme les bases de données, statistiques des joueurs, informations sur les cartes, l'univers du jeu, etc... .En général, on utilise plutôt les listes pour les calculs temporaires et elles sont souvent nettoyées en début et fin de programme pour optimiser la mémoire. Quant aux matrices, le programmeur conserve généralement une ou plusieurs matrices qui font office de sauvegarde.

Vendredi prochain, on abordera avec un exemple simple, à travers le jeu du morpion, l'utilité des listes et des matrices.