



# **SUPPORT TP INFORMATIQUE S1**

## **L1 MINES**

Année Universitaire 2020/2021



## Contenu de TP

- 01 Notions de Bases**
  - 1.1 Ordinateur et Informatique**
  - 1.2 Les Types de Données**
- 02 Introduction à l'Environnement de Programmation**
  - 2.1 Comparaison MatLab vs Octave**
  - 2.2 Manipulations dans l'Environnement**
  - 2.3 Vecteurs et Matrices**
  - 2.4 Les Entrées et les Sorties**
- 03 Références**



## 1. Notions de Bases

### 1.1. Ordinateur et Informatique

Un ordinateur est une machine qui permet la manipulation des informations telles que *nombres, mots, images, sons*. Ces informations s'appellent données.

Les ordinateurs traitent avec une très grande vitesse des quantités énormes de données, les stocker et les afficher. La plupart de ces opérations de stockage, de transfert et de communication se font avec l'électronique numérique (en anglais Digital Electronics). L'électronique numérique utilise *le système binaire (ON =1/OFF=0)*. Un signal avec une série d'impulsions *ON / OFF* est égal à un nombre binaire.

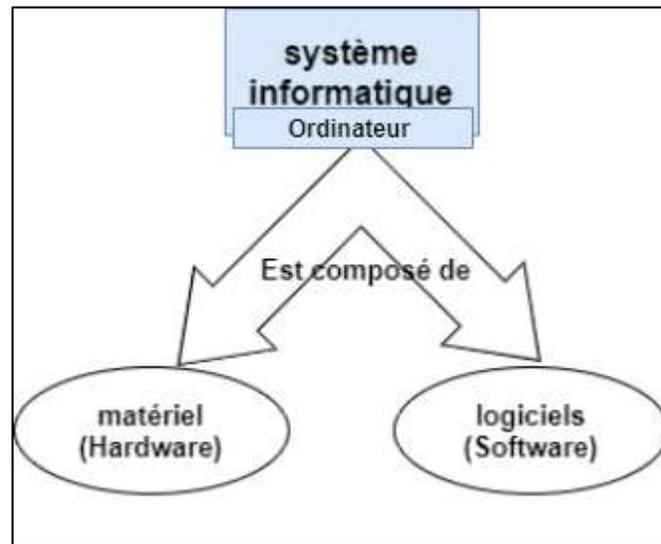


Figure 1 : Composants d'un system Informatique

Selon Figure 1, un system informatique est composé d'une infrastructure technique qui représente le matériel (Hardware) et un ensemble des logiciels (Software).

- **Matériel** : c'est l'ensemble des éléments physiques et les circuits électroniques de la machine. Tel que : L'unité centrale (UC) et ses composantes électronique (processeur, carte mère...), et les périphériques (Clavier, souris, Scanner, Webcam dites périphériques entrées et Imprimante, écran périphériques de sorties), comme dans la Figure 2.

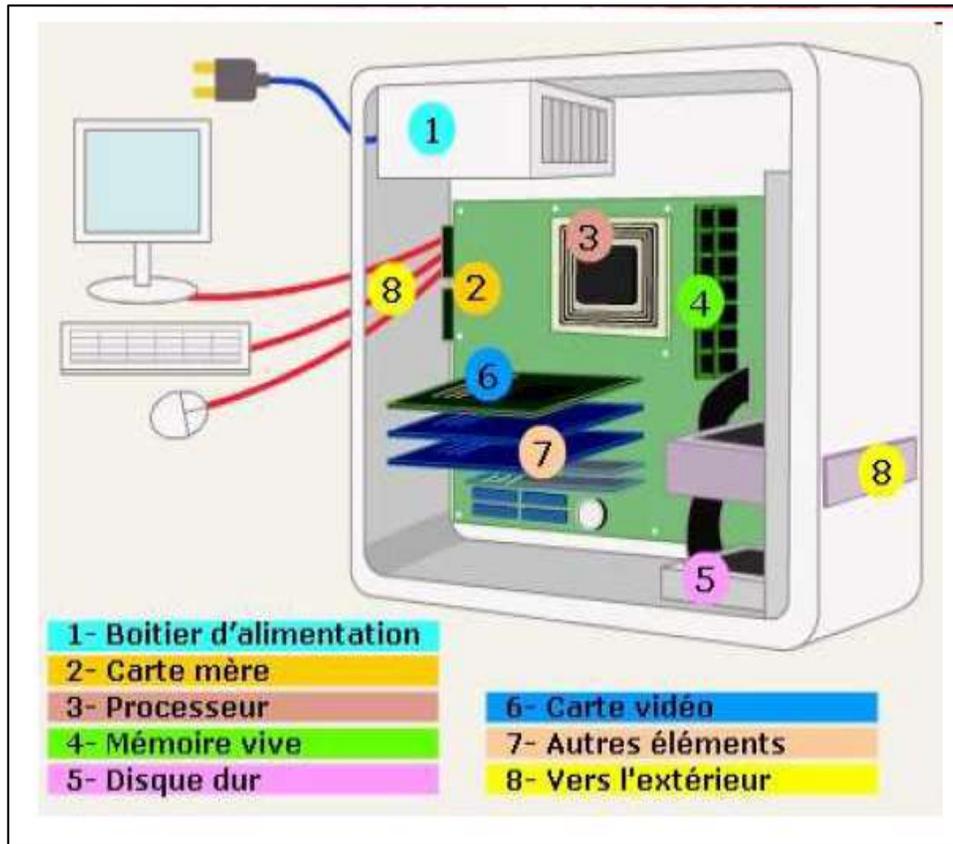


Figure 2 : Matériel et schéma d'UC

1- **Boitier d'alimentation** (*Bloc d'alimentation*, ou *alimentation*), d'un ordinateur est le matériel informatique l'alimentant.

2- **Carte Mère** (En anglais *Motherboard*) est une carte électronique qui relie et supporte la plupart des composants et des connecteurs dans un ordinateur. Elle est composée de circuits et de port de connexion. Elle est comme système nerveux pour l'ordinateur.

3- **Processeur** (*Unité Centrale de Traitement*, UCT ou en anglais *Central Processing Unit*, CPU), c'est le cerveau de l'ordinateur. Il permet de manipuler des informations numériques (*les informations codées sous forme binaire*), et d'exécuter les instructions stockées en mémoire. En général, les Ses fonctionnalité s'englobe dans les étapes suivantes :

- Fetch*, recherche et récupération de l'instruction.
- Decode*, interprétation de l'instruction (opération et opérandes).
- Execute*, exécution de l'instruction.
- Writeback*, écriture du résultat.

Un processeur est composé d'un ensemble de parties à fin de bien maintenir ses fonctionnalités. Ces composants sont : Unité de Contrôle, Registres, Horloge et Bus d'Informations.



4- **Mémoire Vive** (*Mémoire Volatile* aussi nommée **RAM** de l'anglais *Random Access Memory* et en français *Mémoire à Accès Aléatoire* ou *Mémoire à Accès Direct*). C'est une mémoire informatique dans laquelle l'ordinateur place les données dedans lors de leur traitement pour un stockage provisoire. Ses caractéristiques sont les suivantes :

- a- La rapidité d'accès par rapport au disque dur pour apporter rapidement les données au processeur
- b- La volatilité qui implique que les données sont perdues dès que l'ordinateur cesse d'être alimenté en électricité.

#### **Quelle est la différence entre la mémoire vive et la mémoire morte ?**

La *mémoire morte* (En anglais *Read Only Memory, ROM*, ou *Mémoire en Lecture Seule*) s'oppose à la *mémoire vive*, car son contenu est fixé lors de sa programmation. Elle ne s'efface pas lors de la mise hors tension du système. Exemple : le *BIOS* notamment fait appel à la mémoire morte pour pouvoir conserver un certain nombre de paramètres.

**5- Disque Dur** (*DD*, en anglais *Hard Disk Driver, HD* ou *HDD*) est un disque tournant magnétique. Il conserve les données sous forme binaires d'une manière permanente. Le disque dur contient le système Windows, vos programmes installés ainsi que vos données personnelles.

- **Logiciels** : sont les programmes, ou instructions, qui indiquent au matériel ce qu'il faut faire. Il y a deux types de logiciels : les *systèmes d'exploitation* qui contrôlent le fonctionnement des différents éléments matériels et les *logiciels d'application*, qui donnent à l'ordinateur des instructions pour qu'il réalise des tâches spécifiques. par exemple traitement de texte *Word*, *Google Chrome*.

Les ordinateurs modernes sont multiformes, Ils remplissent des tâches très variées. Ce qui implique un changement sur les deux niveaux Matériel et Logiciel. Avec l'introduction de l'Intelligence Artificielle (En anglais *Artificial Intelligence AI*) et l'*Apprentissage Automatique* (En anglais *Machine learning ML*), les programmes s'apprennent et s'adaptent facilement avec les nouvelles architectures des machines et les situations. Quelques Exemples : Supercalculateur (Supercomputer), Casque de Réalité Virtuelle (ou *Casque VR* , en anglais *Virtual Reality Headset*), Apple Watchs et Smart Phones Figure 3.



Figure 3 : Nouvelles Technologies

## 1.2. Les Types de Données

### 1.2.1. Données

Le but principal de l'informatique c'est : *collecter et traiter* des informations d'une façon automatique et bien sur les *stockées*. Ces informations sont appelées des *données*. D'autre terme une donnée est la représentation d'une information dans un programme c.-à-d. : dans un *code source* ou dans la *mémoire durant l'exécution*. Les informations (données) peuvent être sous plusieurs formes tel que : Document texte, Image, Audio, Software, ou d'autres types. Les données dans un programme contiennent des *valeurs (valeurs de données peuvent être fixes (Constantes) ou variables)*, qui peuvent nous aider pour comprendre :

- Ce qu'est fait ;
- Ce qui est stocké ;
- Ce qui est sorti pour l'utilisateur (Visio, Audio).

Les valeurs de données peuvent être saisies par un utilisateur ou obtenues à partir d'une gamme de sources différentes tel que :

- À partir d'un dispositif d'entrée (Clavier, Camera...);
- À partir d'un fichier, d'une base de données ou d'un autre logiciel... ;
- *Même les mouvements de la souris et les clics de bouton sont des données d'entrée, et leur emplacement et leur moment peuvent affecter ce qui se passe.*



### a. Variables

Les variables sont noms qu'on donne à un emplacement mémoire, pour stocker des valeurs dedans, dans un programme.

*Q : Pourquoi utiliser des variables ?*

*R : Les variables sont un élément clé de la programmation. Ils sont utilisés pour les calculs, pour stocker des valeurs pour une utilisation ultérieure, dans les décisions et dans l'itération.*

**Dans ce fait un/une variable doit avoir ces trois éléments :**

- **Nom.** Chaque variable est nommée de manière à indiquer clairement quelle variable est utilisée à tout moment. Voici quelques règles pour nommer une variable : Cohérence, pas d'espacement, pas de Chiffres au début

(var1= 10, 1var=10).

- **Type.** Les variables sont de toutes formes et tailles. Certains sont utilisés pour stocker des nombres, certains sont utilisés pour stocker du texte et certains sont utilisés pour des types de données beaucoup plus complexes. D'autre terme un type de donnée définit la nature des valeurs que peut prendre une donnée.

- **Valeur.** Les variables représentent toute valeur que nous choisissons de leur attribuer.

b. **Types de données** Les types de données à connaître sont :

- i. **String / Chaîne de caractère /Str/Text.** Est utilisée pour une combinaison de tous les caractères qui apparaissent sur un clavier, tels que des lettres, des chiffres et des symboles.

- ii. **Character/ Caractère/Char.** Est utilisé pour les lettres simples.

- iii. **Integer/ Entier/Int.** Utilisé pour les nombres entiers.

- iv. **Float/ Real/ Flottant/Réel.** Utilisé pour les nombres contenant des points décimaux ou pour les fractions.

- v. **Boolean/ Bool/ Booléen.** Utilisé lorsque les données sont limitées aux options *Vrai / Faux* ou *Oui / Non (Logique)*.

- Dans de nombreux langages de programmation, *les variables doivent être déclarées avant de pouvoir être utilisées*, comme *Java*.

- Dans certains langages, tels que *MatLab/ Octave* ou *Python*, vous pouvez *simplement commencer à utiliser la variable sans la déclarer*.



### 1.2.2. Algorithme et Programme

**a. Algorithme.** Le mot algorithme vient du nom d'un mathématicien perse du IX<sup>e</sup> (9) siècle, *Al-Khwârizmî* (en arabe: الخوارزمي).

Un algorithme est un plan, un processus logique étape par étape pour résoudre un problème. Une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème. Les algorithmes sont normalement écrits sous forme d'organigramme ou de pseudocode.

Lors de la conception d'un algorithme, considérez s'il existe plusieurs façons de résoudre le problème.

**b. Programme.** La programmation consiste à écrire du code informatique pour créer un programme, pour résoudre un problème. Les programmes sont créés pour implémenter des algorithmes. La programmation est la traduction de ceux-ci en un programme informatique.

## 2. Introduction à l'Environnement de Programmation

### 2.1. Comparaison MatLab vs Octave

N°	Les Points de comparaisons	<i>MatLab</i>	<i>Octave</i>
01	<b>Définition</b>	C'est un laboratoire matriciel (Matrix Laboratory). Langage utilisé pour le calcul technique.	C'est un langage de programmation utilisé pour le calcul numérique.
02	<b>Programmé</b>	Il a été écrit en langage de programmation <i>C</i> , <i>C++</i> et <i>Java</i> .	Il a été écrit en langage <i>C</i> , <i>C++</i> et <i>Fortran</i> .
03	<b>Interface</b>	Il a une meilleure interface.	Son interface n'est pas aussi bonne que celle de <i>MATLAB</i> .
04	<b>RAM</b>	Il consomme plus de <i>RAM</i> par rapport à <i>Octave</i> .	Il consomme moins de <i>RAM</i> que <i>MatLab</i> .
05	<b>Licence</b>	Ce n'est pas gratuit	C'est gratuit
06	<b>Site officiel</b>	<a href="https://www.mathworks.com/">https://www.mathworks.com/</a>	<a href="https://www.gnu.org/software/octave/">https://www.gnu.org/software/octave/</a>
07	<b>Vitesse d'exécution</b>	La vitesse d'exécution est plus rapide qu' <i>Octave</i> .	La vitesse d'exécution est plus lente que <i>MatLab</i> .

#### Remarques

- *MatLab* et *Octave* sont principalement utilisés dans le même but.
- La principale différence est la syntaxe et d'autres fonctionnalités.
- *MatLab* se compose de boîtes à outils (toolboxes) spécialisées qui ne font pas partie d'*Octave*.
- Ils ne sont pas totalement compatibles (le code).

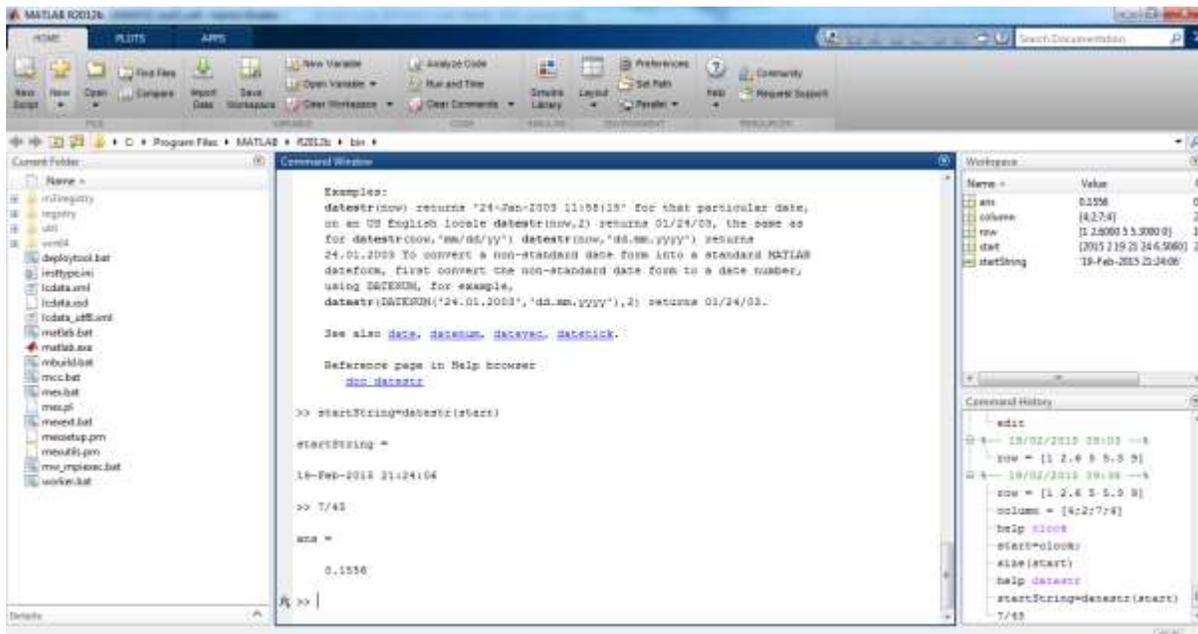


- Le principal avantage de *MatLab* est le grand nombre d'outils prêts à l'emploi.

## 2.2. Manipulations dans l'Environnement

*MatLab* (*Matrix Laboratory*) est un système interactif de calcul numérique, qui dispose d'un grand nombre de fonctions, d'un langage de programmation et d'outils de visualisation graphique. *Matlab* est un langage interprété, c'est à dire qu'il exécute directement (sans compilation) les commandes, que vous entrez dans la fenêtre de commandes. La structure de base sur laquelle travaille *MatLab* est la matrice.

### a. Interface *MatLab*



Au démarrage *MatLab* affiche plusieurs fenêtres sur l'écran, les fenêtres les plus importantes sont :

- ✓ *Command Window*: permet d'entrer les commandes *MatLab* (*Mode interactif*). L'utilisateur peut introduire des commandes des équations, après les chevrons '>>', sur le côté gauche de la fenêtre. Avec un simple 'Entrée', l'exécution s'affiche.
- ✓ *Command History*: Historique de commandes, contient tous les commandes, qu'ont été exécutés.
- ✓ *Current Folder*: Permet la visualisation de répertoire du travail courant.
- ✓ *Workspace*: Permet la visualisation des variables qu'ont été initialisées jusqu'à présent.
- ✓ *Help browser*: Ce navigateur nous permet de parcourir l'aide *MatLab* (*Interface graphique de l'aide*).



- ✓ *Matlab* fonctionne suivant deux modes, le *mode interactif* et le *mode programmation (script)*.
- ✓ *Matlab* fait la distinction *entre minuscules et majuscules*.
- ✓ Commencer toute nouvelle page d'instructions *Matlab* par *clear all (clear var)*. Cette instruction détruit toutes les variables personnelles créées auparavant.
- ✓ *exit* ou *quit* : permet de quitter l'interface *MatLab*.
- ✓ Dans *MatLab* tout ce qui suit un % est considéré comme un commentaire (Dans *Script/Interactif*), pour *Octave* c'est % ou bien #.
- ✓ Deux méthodes sont possibles pour l'aide sur *MatLab*, soit en *mode texte*, soit via l'*interface graphique*. En mode texte, la commande *help* vous donne un aperçu des commandes disponibles.

**b. L'interface de Commandes (Command Window)**

>> 10 ans = 10	<b>ans</b> : Most recent <b>answer</b> , La réponse calculé la plus récente qui n'a pas été explicitement affecté à une <i>variable</i> .
>> a = 12 a = 12	12 est affecté à une variable nommée <i>a</i> , '=' est le signe d'affectation sous <i>MatLab/ Octave</i>
>> A = 1 A = 1 >> c = A + a c = 13	<b>MatLab et Octave</b> font la différence entre variable en Majuscule ou Minuscule, c.-à-d. : <i>a</i> est différent de <b>A</b> . <b>La même chose pour les mots clés <i>sqrt(4)</i> ; <i>SQRT(4)</i></b>
>> 4 + 2    >> 5 ^ 3 >> 8 - 1 >> 7 * 3 >> 20 / 2	Les Operations arithmétiques : Addition (+), Soustraction (-), Multiplication (*), Division (/), Puissance (^)
>> a = 12; >> b = 15; >> c = a - b c = -3	a = 12, b = 15, c = a - b a = 12 b = 15 c = -3 <i>MatLab/Octave</i> supprime la sortie des commandes terminées par un point-virgule et affiche la sortie des commandes terminées par une virgule.
>> a1 = 5 a1 = 5 >> 1a = 9 <b>error: parse error:</b> <b>syntax error</b> >>> 1a = 9 ^	La façon correcte de nom d'une variable (écrite en Vert)
who : liste des variables existantes	Whos : Informations détaillées
>> c = pi	pi est une valeur prédéfini.



C= 3.1416	
>>s= 'Hello' s = Hello	s est une chaine de caractères, l'utilisation des " est obligatoire.
clear s clear all	Efface la valeur de s seulement Efface les valeurs de tous les variables existants dans le workspace
s = 'hello'; >> s (2) ans = e test s(6)	s(2) faire sortir le 2em élément de la chaine s
<b>Exemple 1 :</b> Calculer le surface d'un cercle : $R^2 * \pi / R=5$	<b>Exemple 2 :</b> Calculer le surface d'un cercle: $R^2 * \pi / R= 8$ . Avec : - Commentaires pour expliquer chaque ligne ; (Nom Code, les autres lignes) - Sans l'affichage de R= 8 ; - Ecrire variable « Surface » pour afficher le résultat final.

### 2.3. Vecteurs et Matrices

**a. Vecteur.** Cette notion mathématique est exploitée dans différentes disciplines : Physique (Champs, Electrique...), Informatique (Représentation des données...). Les points suivants résument l'utilité des Vecteurs dans les Maths et l'Informatique.

✓ Un vecteur est une notion utilisé dans une vaste partie des Maths : Algèbre, Géométrie et Analyse.

✓ L'Algèbre utilise les vecteurs dans les systèmes d'équations linéaires.  
Exemple : Trois équations à trois inconnues → Recherche des vecteurs de dimension trois.

✓ En Géométrie, un vecteur est un segment orienté (Point départ et point d'arrivée), employé dans des notions tell que : La Projection et le Parallélisme.

✓ Pour l'Analyse, dans la représentation du graph d'une fonction dans le plan euclidien, on trouve l'espace vectoriel  $\mathbb{R}^2$  pour par exemple la détermination des droites relative à la courbe.

✓ L'Informatique utilise le terme vecteur pour des raisons géométriques et algébrique. Le codage des images sur un écran peut utiliser une technique vectorielle où l'image est composée d'objets géométriques.

✓ La représentation des données en Informatique, utilise aussi le concept de vecteur. Cette représentation dans la mémoire se pose sur des tableaux d'Octets, ces tableaux appartiennent à la famille de composantes vectorielles. Un vecteur est un tableau d'une dimension 1X1.



La manipulation des vecteurs dans *Matlab/ Octave*, se fait comme suit :

<pre>&gt;&gt; x = [1 2 3 ] OU x = [1, 2, 3 ] x = 1 2 3</pre>	<p><b>Un vecteur lignes</b> : les éléments sont séparés par les blancs ou des virgules.</p>
<pre>&gt;&gt; x = [1:9] x = 1 2 3 4 5 6 7 8 9</pre>	<p>C'est une autre forme de vecteur de lignes, les ':' signifiés afficher tous les éléments entre le point de début et le point de fin successivement.</p>
<pre>&gt;&gt;X= ['A': 'G'] OU X= 'A': 'G' X = ABCDEFG</pre>	<p>Vecteur ligne avec le type Char</p>
<pre>&gt;&gt; X (5) ans = E</pre>	<p>Faire sortir le 5<sup>em</sup> élément de ligne</p>
<pre>&gt;&gt; V = [5; 7; 2] V = 5 7 2</pre>	<p>Vecteur colonnes</p>
<pre>&gt;&gt; V(2) ans = 7</pre>	<p>Faire sortir le 2<sup>em</sup> élément de colonne</p>
<pre>&gt;&gt;length(X) ans = 7</pre>	<p>Longueur d'un vecteur (Vecteur Ligne)</p>
<p><b>Exemple 1</b> : Afficher le vecteur ligne <b>VL</b> des nombres <b>25</b> jusqu'à <b>35</b>, et calculer ça longueur. Affiche le vecteur colonne <b>VC</b> du même ensemble.</p>	<p><b>Exemple 2</b> : Affiche le vecteur qui contient les lettres de <b>N</b> jusqu'à <b>Z</b>, dans la variable Lettre.</p>

### b. Matrice

- ✓ En maths, les matrices sont des tableaux bidimensionnels des éléments (nombres, caractères).
- ✓ Une matrice à  $m$  lignes et  $n$  colonnes :  $m \times n$ . Il y a  $m$  lignes, et dans chaque ligne  $n$  éléments.
- ✓ En informatique la deuxième technique de représentation des images c'est l'image matricielle, qui est une image constituée d'une matrice de points colorés. Ces points sont les pixels.
- ✓ Un vecteur est un cas spécial de Matrice.



La manipulation des Matrices dans *Matlab/ Octave*, se fait comme suit :

<pre>&gt;&gt; M = [1 2 3; 4 5 6; 7 8 9 ] OU M = [1, 2, 3; 4, 5, 6; 7, 8, 9 ] M =     1  2  3     4  5  6     7  8  9</pre>	<p><b>Une matrice</b> : les éléments de ligne sont séparés par les blancs ou des virgules. Et les colonnes sont séparées par des points virgules</p>
<pre>&gt;&gt; M (2,1) ans = 4</pre>	<p>Pour faire sortir un élément selon sa position Ligne/Colonne : dans l'exemple M(2,1) c'est l'élément de la <b>deuxième Ligne</b> et la <b>1ere colonne</b>.</p>
<pre>&gt;&gt; M(1,:) ans =     1  2  3</pre>	<p>Affiche tous les éléments de la <b>1ere ligne</b>.</p>
<pre>&gt;&gt;M( :,1) ans =     1     4     7</pre>	<p>Affiche tous les éléments de la <b>1ere Colonne</b>.</p>
<pre>&gt;&gt;size (M) ans =     3  3 &gt;&gt; s = size (M) s =     3  3 &gt;&gt; [L,C]= size (M) L = 3 C = 3</pre>	<p>La taille ou la dimension de la matrice.</p>
<pre>&gt;&gt; det(M) ans = 6.6613e-16</pre>	<p>Le déterminant de la matrice carrée</p>

**Exemple 1** : Créer une matrice carrée de nombre M 4X4.  
 -Calculer le déterminant **D** et la taille en Ligne et Colonne,  
 - Sortir le 3em élément de la 4eme ligne.  
 -Afficher la 2eme ligne dans **L2** et la 1ere colonne dans **C1**.

**2.4. Les Entrées et les Sorties.** La communication avec l'utilisateur se fait par le moyen d'entrées/sorties de données, dans notre cas on s'intéresse par la communication via le clavier.

**a. Les Entrées.** Pour demander de programme d'attendre la saisie d'une valeur pour une variable, on utilise l'instruction suivante :

```
>> X = input ('ecrire la valeur de X: ') % cas numerique
ecrire la valeur de X: 8
X = 8
```



```
>> S = input ('Ecrire un message ', 'E') % Cas String  
Ecrire un message Hello  
S = Hello
```

- b. **Les Sorties.** Pour afficher les données, il y a d'autres instructions pour le faire. La commande simple d'affichage de MatLab/ Octave c'est **disp**.

```
>> disp('La valeur de X: '), disp(X)  
La valeur de X:  
8
```

*Le même principe est utilisé pour l'affichage de String.*

```
14 1 0  
Exemple : soit la matrice M -89 58 25  
10 2 99
```

-Créer la Matrice  $M$ , en utilisant l'instruction `input`

- Afficher le déterminant de  $M$  en utilisant `disp`

### 3. Références

- 01 Tutorials Point originated, "Computer Programming Tutorial", <https://bit.ly/2KHKABM>
- 02 BBC, Bitesize, "Computer Science", <https://bbc.in/2KYyrbw>
- 03 Difference Between MATLAB and Octave, <https://bit.ly/2Lsfjn5>
- 04 Matlab vs Octave, <https://bit.ly/35N2FWq>
- 05 Lien de téléchargement gratuit d'Octave, <https://bit.ly/3qp090t>
- 06 Lien pour exécution en ligne MatLab/Octave, <https://bit.ly/3oSnlnD>
- 07 Notes sur Octave, <https://bit.ly/2Kff73f>
- 08 Octave Forge, Extra packages for GNU Octave, <https://bit.ly/3976NTq>
- 09 Programmation : interaction, structures de contrôle, scripts, fonctions, entrées-sorties fichier, debugging, <https://bit.ly/3p5kasN>