



Instructions : for, while, if, ...

**MATLAB**



**MATrix LABoratory**

**1- Commandes particulières**

**2- Boucle FOR et WHILE**

**3- Instruction IF et SWITCH**

**4- Scripts et fonctions**

### ● **input**

**Variable = input('message')** ➡ Valeur numérique

**Chaîne = input('message','s')** ➡ Chaîne de caractère

### ● **disp**

**disp('message')** ➡ Affiche sur la fenêtre de commande le message

**disp(['phrase1','phrase2'])**

● **pause** ➡ Fait une pause dans un programme.  
Pour poursuivre le calcul, il faut appuyer sur une touche du clavier.

- **num2str** Commande qui convertit une donnée numérique en chaîne de caractère

**Exemple :** Soit  $x = 2$ , le résultat d'un calcul que l'on désire afficher.

```
» x=2;  
» disp([ 'la valeur de x est :', x ])
la valeur de x est :  
» disp(['la valeur de x est :', num2str(x) ])
la valeur de x est :2  
»
```

- **str2num** Commande qui convertit une chaîne de caractère en donnée numérique

### Opérateurs relationnels

<	Inférieur à
>	Supérieur à
<=	Inférieur ou égal à
>=	Supérieur ou égal à
==	Égal à
~=	Différent de

### Opérateurs logiques

<b>&amp;</b>	<b>et</b>
<b> </b>	<b>ou</b>
<b>~</b>	<b>non</b>

## Syntaxe

```
for compteur = début : pas : fin  
    instructions  
end
```

## Exemple :

```
% Tableau des carrés des n premiers entiers naturels  
N=10 ;  
X=[];  
for ii = 1:N  
    X=[X  ii^2];  
end  
X
```




```
» X =  
1 4 9 16 25 36 49 64 81 100
```

## Syntaxe

```
while conditions  
    instructions  
end
```

## Exemple :

```
% Valeur de n, le plus petit entier tel que  $n^2 < X$   
n=0 ;  
X=15;  
while  $n^2 < X$   
    n = n+1 ;  
end
```



```
» n =  
    4
```

## Syntaxe

```
if conditions  
    instructions  
else  
    instructions  
end
```

```
if conditions  
    instructions  
elseif conditions  
    instructions  
else  
    instructions  
end
```



## Exemple :

```
n = input('Entrer un nombre entier :')  
  
if rem(n,2) == 0  
    disp('Nombre pair')  
else  
    disp('Nombre impair')  
end
```

```
» Entrer un nombre entier :10  
n =  
    10  
Nombre pair
```

```
» Entrer un nombre entier :41  
n =  
    41  
Nombre impair
```

## Syntaxe



Avec une valeur numérique

```
switch variable
    case valeur1
        instructions
    case valeur2
        Instructions
    case {valeur3, valeur4}
        instructions
    otherwise
        instructions
end
```

## Syntaxe



Avec une chaîne de caractère

```
switch chaîne
    case 'expression 1'
        instructions
    case 'expression 2'
        instructions
    case {'expression 3', 'expression 4'}
        instructions
    otherwise
        instructions
end
```

**Script** (syntaxe : **fichier.m**)

Contient des suites d'instructions qui seront exécutées une à une

**Exemple :**

```
% Ceci est un exemple de script .m  
close all; clear all; clc;  
X= 0 : 10;  
Y=randn(1,10);  
Z=2*Y;  
plot(X,Y)  
figure  
plot(X,Z)
```

➡ Le programme est exécuté et les deux courbes sont tracées

**Fonction** Sous-programme contenant des variables locales

La première ligne commence toujours par :

<b>function</b>	<b>[s1,s2] =</b>	<b>nomfonction(e1,e2,e3)</b>
		
<b><u>Paramètres de sortie</u></b>		<b><u>Paramètres d'entrée</u></b>



Si un calcul est répété souvent au cours d'un programme, on crée une fonction qui a, pour paramètre de sortie, le résultat escompté.



Les fichiers de fonctions sont stockés sous **un nom identique à celui de la fonction.**

**Exemple :** (nom du fichier : **eff.m**)



```
function y = eff (x)
```

```
% Pour un vecteur, eff(x) donne la valeur efficace  
% Pour un matrice, eff(x) donne un vecteur contenant  
% la valeur efficace de chaque colonne
```

```
[m,n] = size(x)
```

```
if m ==1
```

```
    m = n ;
```

```
end
```

```
y=sqrt(sum(x.*x)/m) ;
```



Si on appelle cette fonction dans un autre programme (ou sous la console) :

```
Vecteur1 = 5*randn(1,10);  
Vecteur2= 2*randn(1,10);  
% Calcul de la valeur efficace  
valeff1= eff (vecteur1) ;  
valeff2 = eff( vecteur2) ;
```



Si des variables intermédiaires sont utilisées dans la fonction, elles seront inconnues en dehors de la fonction.



**Ce sont des variables locales.**



A noter que les commandes MATLAB peuvent être éditables par l'utilisateur voir modifiable par vous ! **Alors attention !**