

## TP N02(Matlab)

---

**Définition:** Un **fichier script** permet regrouper des séries de commandes Matlab. Cela évite d'avoir à saisir plusieurs fois de longues suites d'instructions. A son lancement, les instructions qu'il contient s'exécutent séquentiellement comme si elles étaient lancées depuis l'invite de commande. Un script stocke ses variables dans le workspace, lequel est partagé par tous les scripts. Ainsi, toutes les variables créées dans les scripts sont visibles depuis la command window et vice versa. Lorsque Matlab détecte une erreur, le programme s'arrête et un message d'erreur s'affiche à l'écran (avec le numéro de la ligne où l'erreur est détectée).

```
>>x = 2;  
>>y = 4;  
>>resultat(1) = x-y ;  
>>resultat(2) = x/y ;  
>>resultat(3) = sqrt(x^2+y^2);  
>>resultat
```

Notre script peut ensuite être exécuté, soit en tapant son nom (sans l'extension) à l'invite de commande, soit en cliquant sur le bouton « run » de l'éditeur (icône avec un triangle vert)

```
>>Nom_du_script
```

### **Partie 1 : Les nombres en matlab**

voici quelques format possibles pour l'affichage d'un nombre donné :

<b>format rat</b>	<b>forme rationnelle (a/b)</b>
<b>format long</b>	<b>forme décimale avec 15 chiffres avec notation en virgule flottante.</b>
<b>format long e</b>	<b>forme exponentielle longue.</b>
<b>format hex</b>	<b>forme hexadécimale telle que représentée en mémoire</b>
<b>format short</b>	<b>forme par défaut à 5 chiffres avec notation en virgule flottante.</b>

format short e	forme exponentielle courte.
format bank	Affiche les nombres avec deux chiffres apres la virgule

- Completez l'exemple suivant :

```
>>8/3
//////////
//////////
>>format long
//////////
>>8/3
//////////
//////////
>>format bank
>>8/3
//////////
>>format rat
>>7.2*3.6
//////////
//////////
```

- Quel est l'affichage de **pi** avec les formats donnés dans le tableau précédant ?

```
>>x=1.2631548
>>round(x) %arrondi un nombre vers l'entier le plus proche
//////////
>>floor(x) %arrondi un nombre vers l'entier le plus petit # max{n|n≤x, n
entier}
//////////
>>ceil(x) %arrondi un nombre vers l'entier le plus grand # min{n|n≥x, n
entier}
//////////
```

## Partie 2 : Les chaines de caractères

Les chaînes de caractères sont définies entre des 'quotes' simples : c = 'Ceci est un texte' définira la variable c comme une chaîne de caractères qui pourra être utilisée pour afficher le message entre les quotes.

- Voici quelques fonctions qui s'applique sur les chaines de caractères

La longueur d'une chaîne	<b>length(c)</b>
Les dimensions d'une chaîne	<b>size(c)</b>
La comparaison entre 2 chaînes	<b>strcmp(c1,c2)</b>
La majuscule d'une chaîne	<b>upper(c)</b>
La minuscule d'une chaîne	<b>lower(c)</b>
Le code ASCII d'une chaîne	<b>abs(c)</b>
Conversion d'un code ASCII en caractères	<b>setstr(code)</b>
La concaténation horizontale	<b>strcat(c1,c2)</b>
La concaténation verticale	<b>strvcat(c1,c2)</b>
La recherche d'une sous chaîne dans autre chaîne	<b>findstr(c,'n')</b>
La vérification de type d'une variable	<b>isstr(c1)</b>
Chaîne vers un nombre	<b>ch='2018'</b> <b>str2num(c)</b>
Numéro vers une chaîne	<b>X=27</b> <b>num2str(x)</b>

- Crée un script Fonction\_chainer.m et Répondez soigneusement aux questions suivantes :
  - Calculer la taille de la chaine ch='bonne continuation'
  - quelle est la dimension de ch
  - comparer entre la chaine ch1='Bonjour' et ch2='Bonsoir'
  - mettez la chaine ch1 et ch2 en majuscule
  - mettez la chaine ch2 en minuscule
  - quel est le code ASCII de ch
  - convertir le code ASCII affiché précédemment en une chaine de caractères
  - chercher la présence du caractères 'n' dans la chaine ch
  - convertir le nombre 127 en une chaine et faites une concaténation avec la chaine 'etudiant' de telle sort à avoir le résultat comme suit Z='ETUDIANT n 127'

### Exercice 01 :

<b>switch (choix )</b>  <b>case val1</b> action 1 <b>case val2</b> action 2  ...	<b>case valn</b> action n <b>otherwise</b> autre  <b>end</b>
---	---

- Écrivez un programme qui lit une valeur entière au clavier comprise entre 1 et 7 et qui affiche le nom du jour correspondant (Dimanche, Lundi, Mardi, . . . , Samedi) en utilisant un switch. . . case.
- Prévoir un message d'erreur si l'entier entré n'est pas compris entre 1 et 7
- enregistrer dans un script jour.m
- Écrivez un programme qui lit une valeur entière au clavier comprise entre 1 et 12 et qui affiche le nom du mois correspondant en utilisant un switch. . . case.
- Prévoir un message d'erreur si l'entier entré n'est pas compris entre 1 et 12
- enregistrer dans un script mois.m

### Exercice 02 :

1. Calculez le P.G.C.D. de deux entiers naturels a et b entrés au clavier en utilisant l'algorithme d'EUCLIDE.
2. Vérifiez vos résultats en utilisant la fonction prédéfinie Matlab gcd(a,b).

### Exercice 03 :

<b>while (condition )</b>  <b>Instruction 1;</b> ... <b>Instruction n;</b> <b>end</b>	<b>for indice=valeur initiale:pas:valeur finale</b>  <b>Instruction 1;</b> ... <b>Instruction n;</b> <b>end</b>
--	--

Écrivez un programme qui lit N nombres entiers au clavier et qui affiche leurs somme, leurs produit et leurs moyenne.

1. en utilisant la boucle while,
2. en utilisant la boucle for.
3. sauvegarder votre script sous le nom nombre\_entier.m

**Exercice 04 :**

Dans un script nommé factorielle1.m, écrivez un programme qui permet de calculer la factorielle d'un nombre entier naturel N saisi au clavier.

**Exercice 05 :**

Un nombre parfait est un nombre présentant la particularité d'être égal à la somme de tous ses diviseurs, excepté lui-même.

Exemple :  $6 = 3 + 2 + 1$  est un nombre parfait

- Ecrivez un scripte qui verifie si un nombre introduit par l'utilisateur est parfait ou pas

**Exercice 06(supplémentaire) :**

- Ecrivez un script qui vérifie si un nombre introduit par l'utilisateur est premier ou non?
- Afficher N nombre successive de nombre premier en comptant à partir de zéro.