

Série d'exercices de base

Ex1 – sol_base_1.m

Écrivez un script qui affiche “Bonjour tout le monde” à l'écran.

Ex2 – sol_base_2.m

Écrivez un script qui affiche à l'écran:

Horaire:

7h00 - déjeuner

12h00 - diner

17h00 - souper

Ex3 – sol_base_3.m

Écrivez un script qui demande à l'utilisateur d'entrer un entier, le programme affiche ensuite deux fois ce nombre ex:
10 donne 20.

Ex:

Nombre entré: 10

Nombre affiché: 20

Série d'exercices de base

Ex4 – sol_base_4.m

Écrivez un script qui demande à l'utilisateur d'entrer un nombre, le programme affiche ensuite le nombre au carré.

Ex5 – sol_base_5.m

Écrivez un script qui calcule le prix à payer en plus des taxes. Il demande à l'utilisateur le nombre d'articles achetés, le prix d'un article, puis il affiche le prix total en comptant 15% de taxe de vente.

Ex6 – sol_base_6.m

Écrivez un script qui affiche le nombre pi avec différente résolution. Pour obtenir la valeur de pi, écrivez simplement pi dans Matlab, il s'agit d'une constante défini dans le système:

- a) 3.14
- b) 3.141593
- c) 3.1415926536

Série d'exercices sur le IF

Ex1 – sol_if_1.m

Écrivez un script qui demande à l'utilisateur d'entrer un nombre. Le programme doit afficher si le nombre est positif, négatif ou nulle (égal à zéro)

Ex2 – sol_if_2.m

Écrivez un script qui demande à un utilisateur d'entrer son poids (en kg) et sa taille (en m). Le programme doit en ensuite calculer l'indice de masse corporelle de l'utilisateur en utilisant la fonction suivante:

$$\text{IMC} = \text{poids}/\text{taille}^2$$

Le programme doit ensuite indiquer à l'utilisateur dans quel catégorie il se trouve:

Si plus petit que 18.5, l'utilisateur est maigre

Si plus grand ou égal à 18.5 et plus petite que 25, l'utilisateur est de corpulence normale

Si plus grand ou égal à 25, l'utilisateur est obèse

Série d'exercices sur le IF

Ex3 – sol_if_3.m

Écrivez un script qui calcule le salaire hebdomadaire d'un employé. Celui-ci est payé 23\$/heure, mais si le nombre d'heures travaillé dépasse 40 heures, chaque heures supplémentaire est payé au taux de 1.5 fois.

Le programme demande à l'employé combien d'heures il a travaillé, puis affiche son salaire.

Ex4 – sol_if_4.m

Écrivez un script qui demande à l'utilisateur d'entrer trois nombres. Le programme doit trouver quel nombre est le plus petit et l'afficher. S'il y a égalité sur le plus petit nombre, le cas doit être détecté et indiqué à l'utilisateur.

Série d'exercices sur le WHILE

Ex1 – sol_while_1.m

Écrivez un script qui crée une variable “compteur” avec une valeur égal à 0. Le script ajoute 1 à chaque itération et s'arrête quand la variable a atteint 10. Le script doit afficher la valeur finale de la variable.

Ex2 – sol_while_2.m

Écrivez un script qui demande à l'utilisateur d'entrer un nombre. Le programme initialise une variable “compteur” à 0, y additionne 2 à chaque itération et affiche le résultat. Le programme ne doit s'arrêter au moment où le nombre entré par l'utilisateur est dépassé. Par exemple:

Nombre entré : 11

Nombres affichés: 0 2 4 6 8 10

Série d'exercices sur le WHILE

Ex3 – sol_while_3.m

Écrivez un script qui initialise une variable “compteur” à 1 et y additionne 1 à chaque itération. Le programme s’arrête quand la variable atteint 10. Bien que la variable prend toutes les valeurs entre 1 et 10, le programme ne doit afficher que les valeurs impairs. Pour savoir si un nombre est pair ou impair, utilisez l’expression suivante: `mod(nombre, 2)`. Cette expression retourne 0 si le nombre est pair et 1 si le nombre est impair. L’opération utilisé est celle du modulo, que nous aborderons plus tard dans le cours.

Ex4 – sol_while_4.m

Écrivez un script qui demande à un utilisateur d’entrer un nombre minimum, un nombre maximum et un interval. Le programme doit afficher tous les nombres entre le nombre minimum et maximum inclusivement en sautant de l’interval donné.

ex:

```
nombre_minimum = 10  
nombre_maximum = 20  
interval = 2
```

résultats à afficher: 10, 12, 14, 16, 18, 20

Série d'exercices avancé

Ex1 – sol_avance_1.m

Écrivez un script qui lance un dé aléatoire à 6 faces à l'aide de la fonction randi, tant que le 6 n'a pas sorti 20 fois. Il doit compter le nombre de lancer qui a été nécessaire pour y arriver.

Ex2 – sol_avance_2.m

Écrivez un script qui lance deux dés aléatoire à 3 faces à l'aide de la fonction randi. Il fait ensuite la somme des deux, puis compte combien d'occurrence de chacune des valeurs possible: 2 à 6. Les dé sont lancé jusqu'à ce que l'une des valeurs est été lancé 100 fois. Le programme affiche ensuite l'identité de la valeur qui a été obtenue le plus souvent.

Ex3 – sol_avance_3.m

Écrivez un script qui demande à l'utilisateur un nombre minimum et un nombre maximum. Le programme doit compter tous les nombres pairs entre ces deux nombres. Pour savoir si un nombre est pair ou impair, utilisez l'expression suivante: $\text{mod}(\text{nombre}, 2)$. Cette expression retourne 0 si le nombre est pair et 1 si le nombre est impair. L'opération utilisé est celle du modulo, que nous aborderons plus tard dans le cours.