

## Série d'exercices 2 : variables et opérations

### Exercice 1 – carré de nombres

Écrivez un programme python qui demande à l'utilisateur de donner un nombre entier en affichant '*Donner un nombre entier :* ', calcule le carré de ce nombre (puissance 2) et l'affiche sous cette forme par exemple dans le cas où l'utilisateur entre le nombre 4 : '*Le carré du nombre 4 est : 16.*'

Pour vérifier votre solution nommez votre script python `exercice1.py` et suivez les instructions à la fin de ce document.

### Exercice 2 – calculateur de moyennes

Écrivez un programme qui demande à l'utilisateur d'entrer 3 notes en affichant successivement '*Entrez votre première note:* ', '*Entrez votre deuxième note:* ' et '*Entrez votre troisième note:* '.

Le programme devra ensuite calculer la moyenne des 3 notes et l'afficher sous la forme : '*Votre moyenne est: 5*' par exemple dans le cas où l'utilisateur entre les notes 4, 5 et 6.

Pour vérifier votre solution nommez votre script python `exercice2.py` et suivez les instructions à la fin de ce document.

### Exercice 3 – puissances de nombres

Écrivez un programme qui demande à l'utilisateur d'entrer un nombre à élever à une puissance en affichant '*Donner un nombre entier:* ' puis une puissance à laquelle élever ce nombre en affichant '*Donner une puissance entière:* '.

Si l'utilisateur entre x comme premier nombre et y comme puissance, le programme devra calculer  $x^y$  et l'afficher sous la forme : '*Le nombre 3 à la puissance 4 est : 81.*' par exemple.

Pour vérifier votre solution nommez votre script python `exercice3.py` et suivez les instructions à la fin de ce document.

## Exercice 4 – capacité de x bits

Comme vu en théorie de représentation de l'information, on doit souvent savoir combien de valeurs différentes on peut représenter en utilisant un nombre fixe de 0 et 1 (bits). Par exemple avec 7 bits on peut représenter  $2^7 = 128$  valeurs différentes, ce qui est utilisé pour l'encodage de texte avec ASCII.

Écrivez un programme qui demande à l'utilisateur un nombre de bits pour représenter différentes valeurs et affiche le nombre total de valeurs.

Par exemple, le programme demandera :

*'Sur combien de bits d'information voulez-vous stocker vos valeurs ?'*

Si l'utilisateur répond en écrivant le nombre '5', le programme affichera :

*'On peut stocker 32 valeurs différentes sur 5 bits.'*

Pour vérifier votre solution nommez votre script python `exercice4.py` et suivez les instructions à la fin de ce document.

## Exercice 5 – conversion entre bytes et KB

Comme vu en théorie de représentation de l'information, un byte est une séquence de 8 bits et un kilobyte (KB) représente 1000 bytes. Un fichier sur un ordinateur prend un certain nombre de bytes comme place sur le disque dur, mais si il fait 3200 bytes, on préfère voir apparaître que le fichier fait 3.2 KB ou même la valeur arrondie plus visible 3 KB.

Écrivez un programme qui demande à l'utilisateur un nombre de bytes et affiche ce nombre converti en KB à la fois en valeur décimale exacte et en valeur arrondie vers le bas à l'entier le plus proche. Par exemple si l'utilisateur entre 2903 bytes, le programme affichera successivement *'2903 bytes = 2.903 KB (valeur exacte)'* puis *"2903 bytes = 2 KB (valeur arrondie)'*.

Pour vérifier votre solution nommez votre script python `exercice5.py` et suivez les instructions à la fin de ce document.

## Vérification

Vérifiez vos fichiers en les plaçant sur le **Bureau** et en téléchargeant le fichier `python_checker_serie2.sh` depuis Moodle et en le copiant sur le **Bureau**. Ouvrez ensuite le **terminal** (touches espace + commande et tapez « ter » pour voir apparaître l'application terminal), entrez les 2 commandes suivantes :

```
cd ~/Desktop
```

```
sh python_checker_serie2.sh
```

Pour chaque exercice, le terminal devrait afficher « **exercice est correct !** ». Toute erreur sera affichée avec un message en **rouge**.