

# Faites vos premiers pas en Python – Exercices

## 1 Compétences ciblées

Savoir :

1. déclarer et initialiser une variable
2. utiliser les structures conditionnelles.
3. utiliser les boucles
4. gérer les exceptions

## 2 Déclarer et initialiser une variable

1. Variables réelles
  - (a) Affectez les variables temps et distance par les valeurs 6.892 et 19.7
  - (b) Affichez la valeur de la vitesse
  - (c) Améliorer la vitesse en imposant un chiffre après la point décimal
2. Interactions avec l'utilisateur
  - Saisissez un nom et un age en utilisant l'instruction `input()`. Puis les afficher.

## 3 Contrôler le flux d'instructions

1. Saisissez un flottant. S'il est positif ou nul, affichez sa racine, sinon affichez un message d'erreur :
  - (a) d'abord en utilisant une boucle de contrôle `if / else`
  - (b) puis en utilisant les mots clés `try` et `except`
2. L'ordre *lexicographique* est celui du dictionnaire.
  - (a) Saisissez deux mots, comparez-les pour trouver le *plus petit* et affichez le résultat.
  - (b) Refaites l'exercice en utilisant l'instruction ternaire :

`<res> = <a> if <condition> else <b>.`

3. On désire sécuriser une enceinte pressurisée. On se fixe une pression seuil et un volume seuil :  $\text{pression\_seuil} = 2.3$ ,  $\text{volume\_seuil} = 7.41$ . On demande de saisir la pression et le volume courant de l'enceinte et d'écrire un script qui simule le comportement suivant :
  - (a) si le volume *et* la pression sont supérieurs aux seuils : arrêt immédiat ;
  - (b) si seule la pression est supérieure à la pression seuil : demander d'augmenter le volume de l'enceinte ;
  - (c) si seul le volume est supérieur au volume seuil : demander de diminuer le volume de l'enceinte ;
  - (d) sinon déclarer que *tout va bien*.
 Ce comportement sera implémenté par une alternative multiple.
4. Initialisez deux entiers :  $a = 0$  et  $b = 10$ .
  - (a) Écrire une boucle affichant et incrémentant la valeur de  $a$  tant qu'elle reste inférieure à celle de  $b$ .
  - (b) Écrire une autre boucle décrémentant la valeur de  $b$  et affichant sa valeur si elle est impaire. Bouclez tant que  $b$  n'est pas nul.
5. Écrire une *saisie filtrée* d'un entier dans l'intervalle 1 à 10, bornes comprises. Affichez la saisie.
6. Affichez chaque caractère d'une chaîne en utilisant une boucle `for`. Affichez chaque élément d'une liste en utilisant une boucle `for`.
7. Affichez les entiers de 0 à 15 non compris, de trois en trois, en utilisant une boucle `for` et l'instruction `range()`.
8. Utilisez l'instruction `break` pour interrompre une boucle `for` d'affichage des entiers de 1 à 10 compris, lorsque la variable de boucle vaut 5.
9. Utilisez l'instruction `continue` pour modifier une boucle `for` d'affichage de tous entiers de 1 à 10 compris, sauf lorsque la variable de boucle vaut 5.
10. Utilisez une exception pour calculer, dans une boucle évoluant de  $-3$  à 3 compris, la valeur de  $\sin(x)/x$ .
11. La clause `else` des boucles.
  - (a) Dans cet exercice, effectuez les saisies avec des `integerbox` et les affichages avec des `msgbox`, tous deux appartenant au module `easygui`.
  - (b) Utilisez une liste avec 5 entiers de votre choix puis saisissez un entier.
  - (c) Dans une boucle `for`, parcourez la liste. Si l'entier saisi appartient à la liste, sauvez-le et interrompez la boucle (puisque vous l'avez trouvé). Si la boucle s'est bien terminée, utilisez une clause `else` pour affichez un message l'annonçant.
  - (d) Entrez maintenant une autre entier, cette fois-ci positif.
  - (e) Ecrivez une boucle `while` pour déterminer si cet entier est premier. S'il ne l'est pas, la boucle devra afficher le premier diviseur trouvé et s'interrompre. S'il est premier, l'afficher dans une clause `else`.