

Plaquette 1 – Boucles, seuils, listes et simulations techniques

Algorithmique et Python — Première STI2D

Exercices

Méthode

Quatre gestes à maîtriser : **dérouler** un programme à la main, écrire une **fonction** qui calcule une grandeur, chercher un **seuil** avec une boucle « tant que », et **simuler** une expérience aléatoire. Les contextes sont ceux de la filière : températures, tolérances, contrôles, débits.

Exercice 1 - Suivre des affectations

Que valent a et b à la fin du script ?

```
a = 4
b = a + 3
a = 2 * b
b = a - b
```

Exercice 2 - Dérouler une boucle for

Qu'affiche ce programme ?

```
t = 200
for i in range(3):
    t = t * 0.8
print(round(t, 2))
```

Exercice 3 - Fonction de calcul technique

Écrire une fonction *puissance(u, i)* qui renvoie la puissance électrique $P = U \times I$, puis donner *puissance(230, 0.5)*.

Exercice 4 - Fonction avec un test

Une pièce est conforme si sa longueur appartient à $[24,5 ; 25,5]$ mm. Compléter la fonction.

```
def conforme(longueur):
    if longueur >= 24.5 and longueur <= ...:
        return True
    else:
        return False
```

Que renvoie *conforme(25.7)* ?

Exercice 5 - Algorithme de seuil

Une pièce à 360 °C perd 15 % de sa température chaque minute. Écrire une fonction qui renvoie le nombre de minutes au bout duquel elle descend sous 100 °C.

Exercice 6 - Somme et moyenne d'une liste

Écrire une fonction *moyenne(L)* qui renvoie la moyenne des éléments d'une liste non vide, sans utiliser *sum*. Tester avec $L = [12.1, 11.8, 12.4, 12.0, 11.7]$.

Exercice 7 - Compter avec une condition

Écrire une fonction *nbHorsTolerance(L)* qui compte, dans une liste de longueurs, celles qui sortent de l'intervalle $[24,5 ; 25,5]$. Tester avec $L = [25.1, 24.3, 25.0, 25.6, 24.9]$.

Exercice 8 - Tableau de valeurs d'une fonction

Écrire un programme qui affiche la hauteur $h(x) = -0,25x^2 + 2x$ (en mètres) pour x entier de 0 à 8 (en mètres), puis indiquer où la hauteur est maximale.

Exercice 9 - Simuler un contrôle qualité

Une production comporte 8 % de pièces défectueuses. Compléter le programme qui estime la fréquence de pièces défectueuses sur n prélèvements.

```
from random import random

def frequenceDefaut(n):
    defaults = 0
    for i in range(n):
        if random() < ...:
            defaults = defaults + 1
    return defaults / ...
```

De quelle valeur le résultat doit-il s'approcher ?

Exercice 10 - Simuler une répétition d'épreuves

On prélève 4 pièces (production à 8 % de défauts) et le lot est refusé s'il contient au moins deux pièces défectueuses.

Écrire une fonction qui estime, sur N simulations, la fréquence de lots refusés, et donner la valeur théorique attendue.

Exercice 11 - Quotient et reste en Python

On range 125 pièces dans des cartons de 12.

```
n = 125
q = n // 12
r = n % 12
```

Donner q et r , puis le nombre de cartons nécessaires.

Exercice 12 - Liste des termes d'une suite

Un ressort mesure 40 mm et s'allonge de 6 mm par masse ajoutée. Écrire une fonction *longueurs(n)* qui renvoie la liste des longueurs pour 0, 1, ..., $n - 1$ masses, puis donner *longueurs(5)*.

Exercice 13 - Vider une cuve

Une cuve contient 5 000 L et perd 350 L par minute. Écrire une fonction qui renvoie le nombre de minutes au bout duquel il reste moins de 1 000 L.

Exercice 14 - Simuler des lancers de dé

Écrire une fonction *frequence_six(n)* qui simule n lancers d'un dé équilibré et renvoie la fréquence des 6. De quelle valeur doit-elle s'approcher ?

Exercice 15 - Synthèse : corriger un programme

Ce programme doit renvoyer le plus petit entier n tel que $1\,000 \times 1,05^n$ dépasse 2 000. Il comporte deux erreurs : les trouver et donner la version corrigée avec son résultat.

```
def seuil():  
    c = 1000  
    n = 0  
    while c > 2000:  
        c = c * 1.05  
    return c
```