

# Chapitre : Introduction aux Algorithmes

EcoleDigitale Promotion : P12 – UNCHK

## Objectifs du chapitre

- Comprendre ce qu'est un algorithme.
- Savoir écrire un algorithme simple.
- Apprendre à représenter un algorithme avec des instructions claires.
- Distinguer les étapes : entrée, traitement, sortie.

### 1. Définition d'un algorithme

Un **algorithme** est une **suite d'instructions logiques et ordonnées** permettant de résoudre un problème ou d'accomplir une tâche donnée.

→ C'est le **cœur de toute programmation** : un programme informatique n'est rien d'autre qu'un algorithme traduit dans un langage de programmation.

### 2. Exemples de situations algorithmiques

- Faire bouillir de l'eau :
  1. Prendre une casserole.
  2. Ajouter de l'eau.
  3. Mettre sur le feu.
  4. Attendre jusqu'à ébullition.
- Calculer la moyenne de deux nombres :
  1. Lire les deux notes.
  2. Les additionner.
  3. Diviser par 2.
  4. Afficher la moyenne.

### 3. Structure d'un algorithme

Un algorithme est souvent organisé selon trois grandes étapes :

Étape	Rôle	Exemple
<b>Entrée</b>	Collecter les données nécessaires	Lire deux nombres $a$ et $b$
<b>Traitement</b>	Appliquer des opérations sur les données	Calculer $\text{somme} = a + b$
<b>Sortie</b>	Afficher ou transmettre le résultat	Afficher $\text{somme}$

## 4. Représentations possibles d'un algorithme

Un algorithme peut être représenté de différentes façons :

◆ a. En langage naturel

Lire deux nombres → Les additionner → Afficher la somme.

◆ b. En pseudo-code

Début

Lire a

Lire b

somme  $\leftarrow a + b$

Afficher somme

Fin

c. En organigramme (diagramme de flux)

(à représenter graphiquement)

- Un bloc "DÉBUT"

- Une flèche vers "Lire a et b"
  - Puis "Calculer somme = a + b"
  - Ensuite "Afficher somme"
  - Et enfin "FIN"
- 

## 5. Les instructions de base d'un algorithme

Un algorithme repose sur **trois types d'instructions** :

● *a. Instructions séquentielles*

→ Exécutées l'une après l'autre, dans l'ordre.

● *b. Instructions conditionnelles (si...alors...)*

→ Permettent d'exécuter des actions **en fonction d'une condition**.

Exemple :

Si  $\text{âge} \geq 18$  alors

Afficher "Majeur"

Sinon

Afficher "Mineur"

FinSi

*c. Instructions répétitives (boucles)*

→ Utilisées pour **répéter** une ou plusieurs instructions.

Exemple (tant que) :

Tant que  $\text{compteur} \leq 5$

Afficher compteur

compteur  $\leftarrow$  compteur + 1

FinTantQue

## 6. Qualités d'un bon algorithme

- **Finit** : il doit toujours s'arrêter après un nombre limité d'étapes.
- **Correct** : il doit donner le bon résultat.
- **Clair** : les étapes doivent être faciles à comprendre.
- **Efficace** : rapide, et utilisant peu de ressources.

## 7. Exercice simple en pseudo-code

Écrire un algorithme qui affiche le carré d'un nombre.

Début

Lire x

carré  $\leftarrow$  x \* x

Afficher carré

Fin

## Récapitulation

Un algorithme est **le plan de construction** de tout programme. Savoir l'écrire, le lire, et le représenter, c'est déjà penser comme un informaticien. Ce n'est qu'après avoir bien compris les algorithmes qu'on peut les transcrire dans un langage de programmation (comme Python, C, Java, etc.).

Pour tester vos connaissances, vous allez maintenant répondre à une série de **QCM (Questions à Choix Multiples)** portant sur les notions vues dans ce cours.

Prenez le temps de bien relire vos notes, puis lancez-vous dans les QCM pour valider votre compréhension.

EcoleDigitale