ALGORITHMIQUE · PARCOURS · FONDATIONS

Poser le problème et décrire les données

Découvrez les entrées, sorties, variables, opérateurs, bases de numération et collections avant d'écrire un algorithme en pseudo-code, Python ou MATLAB.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Définir les données reçues et le résultat attendu avant le calcul.
- Distinguer valeurs, types, opérateurs et représentations des nombres.
- Convertir une entrée et accéder à un élément avant de le parcourir.
- Écrire puis tracer une méthode en pseudo-code anglais.

Prérequis. Aucun prérequis de programmation.

Comprendre les données avant les instructions

Les fondations de l'algorithmique commencent par une question précise : quelles données reçoit-on et quel résultat doit-on produire ? Convertir 125 secondes en minutes paraît immédiat, mais il faut choisir une unité, annoncer les valeurs autorisées et vérifier que le résultat représente la même durée. Le premier chapitre transforme cette demande en un contrat explicite.

Les chapitres suivants donnent les outils pour suivre le calcul : une valeur, un type, une variable, une constante et une expression. Vous apprendrez à choisir les opérateurs, à déterminer leur ordre, à convertir les bases 2, 8, 10 et 16, puis à distinguer les valeurs mathématiques de leurs représentations en machine. Ces mots sont définis avant leurs utilisations dans les exemples. Une trace note ensuite ce qui change après une instruction. Les entrées textuelles, les conversions et les premiers accès à une collection complètent ces fondations. Ce travail prépare le pseudo-code sans supposer que vous savez déjà lire une boucle ou une fonction.

Les chapitres de cette partie

- 1. Passer d'un énoncé à un problème précis
- 2. Valeurs, variables et affectation
- 3. Choisir une opération arithmétique
- 4. Déterminer l’ordre des opérations
- 5. Expressions, comparaisons et logique
- 6. Écrire un nombre en bases 2, 8, 10 et 16
- 7. Comprendre les limites des nombres en machine
- 8. Passer du texte aux données du calcul
- 9. Organiser du texte et des suites de valeurs
- 10. Lire et écrire du pseudo-code

Vérifier une étape avant de poursuivre

Après la conversion, expliquez la différence entre la durée de départ et son reste. Après les variables, refaites l'échange de 4 et 9 sans perdre une valeur et distinguez une donnée variable de la constante 60 secondes par minute. Après les opérateurs, reconstituez 17 avec son quotient et son reste dans la division par 5. Convertissez ensuite 26 en binaire et en hexadécimal, distinguez conversion et validation d’une quantité, puis retrouvez le dernier élément d’une collection. Après la logique, justifiez la réponse pour une température exactement égale à une borne. Vous pourrez alors écrire les étapes d'un calcul de commande en pseudo-code.

Les blocs sont en anglais et le cours explique leurs mots-clés en français. Choisissez d'abord un langage pour exécuter un exemple, puis comparez l'autre onglet. Chercher immédiatement une correspondance caractère par caractère entre Python et MATLAB ferait oublier le résultat commun à vérifier.

À retenir - fiche mémo

Données

Définissez les entrées, les sorties et leurs unités avant les instructions.

État

Une trace consigne les valeurs après chaque instruction.

Notation

Le pseudo-code du parcours emploie des mots anglais et annonce ses conventions.

Pseudo-code

```
// example.pseudo
// assignment_and_comparison
value <- 4
PRINT value == 4
```

Point de vigilance. Affecter une valeur ne teste pas une égalité.

Consulter la cheatsheet : Fondations · **Télécharger la fiche du thème en PDF**

Version web : https://www.clearcode.fr/courses/programming/algorithmic/foundations