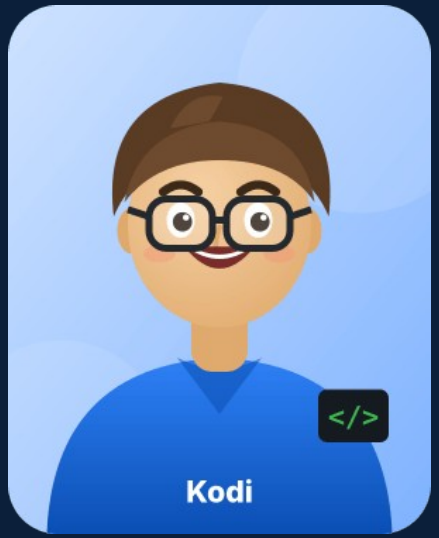
HEF-010 · MISSION 01 / 10

Modéliser un problème

Un algorithme, c'est une recette précise. Avant de cuisiner, il faut savoir ce qu'on veut dans l'assiette.



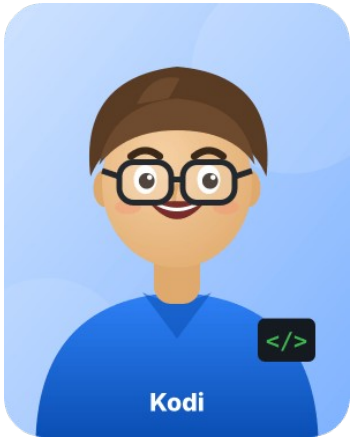


Badge : Architecte de recettes

65 min · débutant

L'histoire

Maya arrive avec une demande : « **Trouve-moi le mentor le plus disponible pour chaque nouveau membre.** »
Kodi ouvre son éditeur. Sam l'arrête gentiment.



> *« Avant d'écrire du code : c'est quoi l'entrée ? C'est quoi la sortie ? Et c'est quoi, "le plus disponible" ? »*

Kodi réalise qu'il ne peut pas coder une phrase floue. Il faut d'abord la **transformer en problème précis**.

Ta mission

À la fin, je serai capable de…

1. définir un algorithme avec mes mots
2. identifier l'entrée, la sortie et les règles d'un problème
3. écrire un pseudocode simple
4. exécuter un algorithme à la main sur un exemple
5. repérer ce qui rend un énoncé ambigu

Défi d'échauffement

Défi d'échauffement

Écris les étapes pour **trouver le plus grand nombre dans cette liste**, comme si tu donnais les instructions à une personne qui ne sait pas lire les nombres d'un coup d'œil, seulement un par un :

`7, 3, 12, 9, 12, 4`

Écris tes étapes en français, numérotées. Garde ta feuille.

Le concept

Un algorithme, c'est quoi ?

Un **algorithme** est une suite d'étapes **précises** qui transforme une **entrée** en **sortie**, et qui se termine.
Analogie : une recette. Les ingrédients sont l'entrée, le plat est la sortie, les étapes sont l'algorithme. Une bonne recette est claire pour quelqu'un qui ne connaît pas le plat.

Les trois questions d'un problème

Question	Pour « le mentor le plus disponible »
Entrée ?	La liste des mentors avec leurs heures libres
Sortie ?	Un seul mentor
Règle ?	Celui qui a le plus d'heures libres. En cas d'égalité ? À décider !

Le concept

Les trois questions d'un problème (suite)

Une règle floue donne un code qui fait n'importe quoi. Les cas d'égalité, de liste vide, de valeur manquante font partie du problème.

Le pseudocode

Le **pseudocode** est une écriture entre le français et le code. Il n'a pas de syntaxe officielle. Il sert à réfléchir sans se battre avec un langage.

```
fonction plus_grand(liste):
    si la liste est vide: signaler une erreur
    meilleur ← premier élément
    pour chaque élément de la liste:
        si élément > meilleur:
            meilleur ← élément
    renvoyer meilleur
```

Le concept

La trace

Pour vérifier un algorithme, on le **déroule à la main** en notant l'état à chaque étape : c'est une **trace**.

🧠 ***Astuce*** *: si tu ne sais pas faire la trace sur un exemple, tu ne comprends pas encore ton algorithme.*

Exemple pas à pas

Exemple pas à pas

Problème : trouver le plus grand nombre de `[7, 3, 12, 9, 12, 4]`.

Trace de l'algorithme `plus_grand` :

Élément lu	`meilleur` avant	Comparaison	`meilleur` après
(départ)	:	:	7
3	7	3 > 7 ? non	7
12	7	12 > 7 ? oui	12
9	12	non	12
12	12	12 > 12 ? non	12
4	12	non	12

Sortie : 12.

Exemple pas à pas (suite)

Ce qu'on remarque : l'égalité (12 deux fois) est gérée par la règle `>` : le premier 12 gagne. Si l'énoncé demandait « le dernier », il faudrait `>=`. **Les détails de la règle changent le résultat.**
En Python :

```
def plus_grand(liste):
    if not liste:
        raise ValueError("liste vide")
    meilleur = liste[0]
    for element in liste:
        if element > meilleur:
            meilleur = element
    return meilleur

print(plus_grand([7, 3, 12, 9, 12, 4]))
```

À toi, avec aide

Problème : *« Compter combien de membres ont terminé au moins 3 missions. »* Les données sont une liste de nombres : le nombre de missions terminées par chaque membre, par exemple `[0, 5, 3, 2, 3, 1]`.

1. Écris l'entrée, la sortie, la règle.
2. Écris le pseudocode.
3. Fais la trace sur l'exemple.

À toi, avec aide · réponse

Indices progressifs

Indice 1

La sortie est un seul nombre : combien de membres répondent à la règle.

Indice 2

Il te faut un compteur qui démarre à 0 et qu'on augmente quand la condition est vraie.

Indice 3

La règle est « au moins 3 », donc 3 compte : utilise `>= 3`.

À toi, avec aide · réponse

Réponse

- Entrée : liste de nombres. Sortie : un nombre. Règle : compter les valeurs `>= 3`.

```
compteur ← 0
pour chaque valeur de la liste:
    si valeur >= 3:
        compteur ← compteur + 1
renvoyer compteur
```

Trace sur `[0, 5, 3, 2, 3, 1]` : 0 non, 5 oui (1), 3 oui (2), 2 non, 3 oui (3), 1 non.
Résultat : 3.

À toi, en solo

Problème : *« Trouver le mentor le plus disponible. »* Chaque mentor a un nom et un nombre d'heures libres : `[("Aïcha", 4), ("Bruno", 7), ("Chloé", 7)]`.

1. Liste **les ambiguïtés** de l'énoncé (au moins 3).
2. Choisis une règle pour chacune.
3. Écris l'algorithme en pseudocode puis fais la trace.

À toi, en solo · réponse

Pistes de correction

Ambiguïtés possibles : égalité entre Bruno et Chloé ; liste vide ; heures négatives ou manquantes ; « disponible » cette semaine ou ce mois ?
Exemple de choix : en cas d'égalité, le premier de la liste ; liste vide → erreur claire ; heures négatives refusées. Le pseudocode ressemble à `plus_grand`, mais compare `heures` et garde le **mentor entier**. Trace : Aïcha (4) → Bruno (7) remplace → Chloé (7) ne remplace pas (pas strictement plus grand). **Résultat : Bruno.**

Pièges classiques

Piège	Pourquoi c'est un problème	Comment l'éviter
Énoncé flou	On code une règle qu'on a devinée	Poser les questions avant de coder
Oublier les cas limites	Plantage sur liste vide ou égalité	Les lister dans le problème
Étape imprécise	« Trouve le meilleur » : comment ?	Chaque étape doit être exécutable à la lettre
Pas de trace	On croit que ça marche	Dérouler à la main sur un petit exemple

Mini-boss

Mini-boss

Maya demande : *« Dis-moi quels membres sont inactifs. »*

Produis une **fiche de problème** complète :

1. entrée, sortie ;
2. définition précise de « inactif » (à toi de choisir, mais justifie) ;
3. 4 cas limites ;
4. pseudocode ;
5. une trace sur un exemple de 5 membres.

Mini-boss · réponse

Pistes de correction

Exemple de définition : « inactif = aucune mission terminée depuis 30 jours ». Cas limites : nouveau membre inscrit hier ; date manquante ; liste vide ; date dans le futur. Une bonne fiche donne un pseudocode qui parcourt la liste, compare la date avec « aujourd'hui − 30 jours » et ajoute le membre à une liste résultat. La trace montre chaque membre et la décision.

Quiz éclair

1. Un algorithme est :

A. une suite d'étapes précises qui se termine
B. un programme dans un langage précis
C. une idée générale

2. Quelle est la sortie d'un problème « trouver le plus grand nombre » ?

A. La liste de départ
B. Un nombre
C. Un tri de la liste

3. Le pseudocode sert à :

A. remplacer le code définitivement

Quiz éclair

Quiz éclair (suite)

B. effacer les bugs
C. réfléchir à l'algorithme sans la syntaxe d'un langage

4. 🔁 *Révision HEF-000, mission 01* : « Le site est lent » est :

A. un symptôme
B. une solution
C. une contrainte

5. 🔁 *Révision HEF-000, mission 01* : Une hypothèse est :

A. un fait
B. une explication possible non vérifiée
C. une solution

Quiz éclair · réponse

Réponses

1. **A**.
2. **B**.
3. **C**.
4. **A**.
5. **B**.

Je réfléchis

1. Compare ta recette d'échauffement avec le pseudocode du concept. Qu'est-ce qui était imprécis chez toi ?
2. Quel problème de ta vie quotidienne pourrais-tu décrire avec entrée, sortie, règle ?

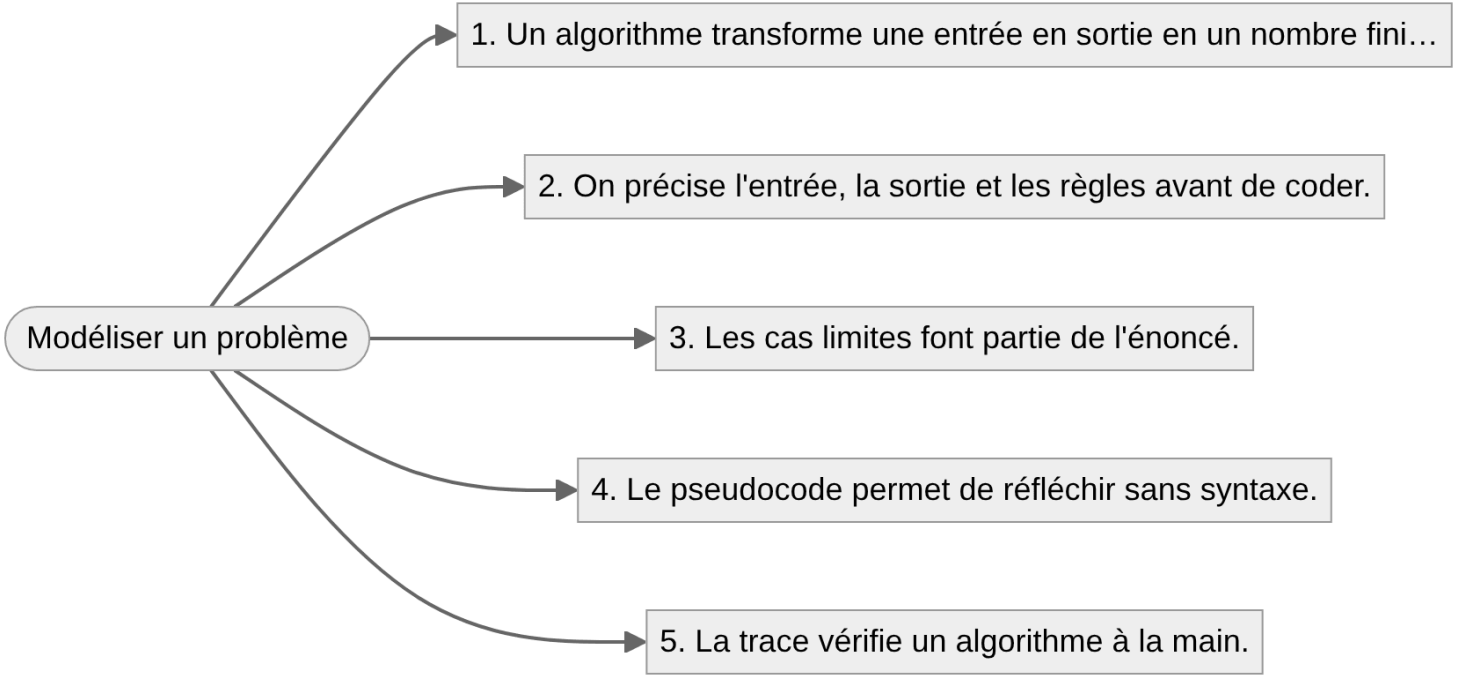
À retenir

À retenir

1. Un algorithme transforme une entrée en sortie en un nombre fini d'étapes précises.
2. On précise l'entrée, la sortie et les règles avant de coder.
3. Les cas limites font partie de l'énoncé.
4. Le pseudocode permet de réfléchir sans syntaxe.
5. La trace vérifie un algorithme à la main.

À retenir

Le schéma de la mission



MISSION ACCOMPLIE

+100 XP

Badge débloqué : Architecte de recettes

Prochaine mission : Variables, conditions et boucles

