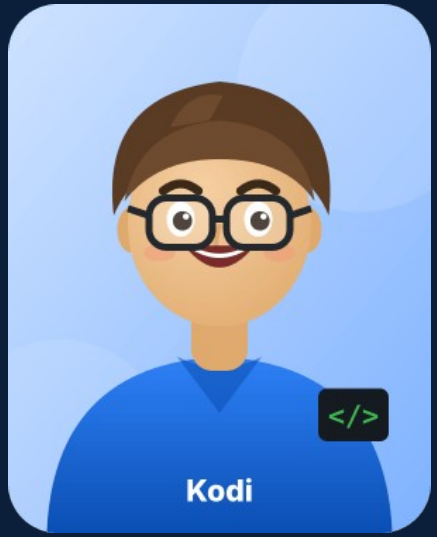
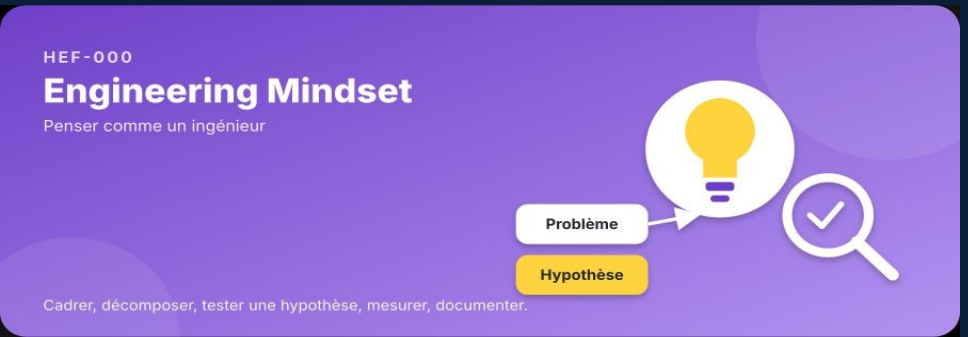
HEF-000 · MISSION 05 / 9

# Documenter et expliquer

*Ce qui reste dans ta tête n'aide pas ton équipe.*




**Badge : Passeur·se de savoir**

70 min · débutant

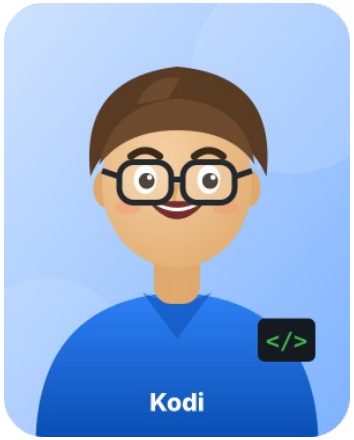
# L'histoire

Sam part en vacances deux semaines. Le jour même, le serveur plante. Kodi ouvre le dépôt : aucun guide, aucun schéma, pas une ligne qui explique comment relancer le service.

> *« Sam est le seul à savoir… » soupire Maya.*

Kodi se promet une chose : **plus jamais ça.**

**Dans cette mission, tu apprends à écrire ce qu'il faut, là où il faut, pour que quelqu'un d'autre (ou toi dans six mois) puisse comprendre et agir.**



Ta mission

# À la fin, je serai capable de…

1. expliquer pourquoi la connaissance « dans la tête » est un risque ;
2. choisir le bon type de document : README, schéma d'architecture, ADR, runbook ;
3. séparer faits, hypothèses et décisions dans un texte ;
4. écrire pour un lecteur précis ;
5. relire un document avec une liste de contrôle.

Défi d'échauffement

# Défi d'échauffement

Explique **par écrit, en 6 lignes maximum**, comment préparer ton plat préféré à quelqu'un qui n'a jamais cuisiné.

Ensuite, donne ta recette à une personne (ou imagine-la) : où se bloquerait-elle ? Note les trous.

Le concept

# Documenter, ce n'est pas tout écrire

On documente pour **répondre à une question que quelqu'un se posera**.

> ***Quelle connaissance dois-je rendre accessible, et à qui ?***

## Quatre documents utiles

| Document | Il répond à la question | Exemple |
|---|---|---|
| **README** | « C'est quoi, et comment je commence ? » | Description du projet, installation, premier lancement |
| **Architecture** | « Comment c'est organisé ? » | Schéma des composants et de leurs liens |
| **ADR** (décision) | « Pourquoi a-t-on choisi ça ? » | « Nous utilisons PostgreSQL, car… » |

Le concept

# Quatre documents utiles (suite)

| Document | Il répond à la question | Exemple |
|---|---|---|
| **Runbook** | « Que faire quand ça casse ? » | Étapes pour relancer le service |

Un **ADR** (*Architecture Decision Record*) est une petite note datée qui garde une décision : le contexte, le choix, les alternatives, les conséquences. Six mois plus tard, on sait pourquoi.

## Fait, hypothèse, décision

Mélanger ces trois types rend un document trompeur.

- **Fait** : « Le serveur répond sur le port 443. »
- **Hypothèse** : « Le problème vient peut-être de la saturation des connexions. »
- **Décision** : « Nous utilisons PostgreSQL comme base principale. »

Le concept

# Fait, hypothèse, décision (suite)

Écris-les de manière à ce qu'on les reconnaisse.

## Documentation proche du code

Une doc dans un fichier oublié se périme. Une doc **dans le dépôt**, relue comme le code, reste vivante. C'est la *documentation as code*.

## Écrire pour le futur lecteur

Avant d'écrire, demande-toi : **qui lira ?** Un nouveau, une personne en urgence à 3 h du matin, un client ? Le niveau de détail change.

> 🧠 ***Astuce** : un bon runbook se lit en situation de stress. Phrases courtes, étapes numérotées, commandes à copier.*

# Exemple pas à pas

**Besoin** : documenter comment relancer le service de notifications.

**1. Le lecteur** : un·e membre de l'équipe de garde, nuit, pressé·e.

**2. Le type** : un runbook.

**3. Le contenu** :

```
# Relancer le service de notifications

Quand l'utiliser : les e-mails ne partent plus.

1. Vérifier l'état : `systemctl status notifications`
2. Lire les 20 dernières lignes du journal : `journalctl -u notifications -n 20`
3. Relancer : `sudo systemctl restart notifications`
4. Vérifier : envoyer un e-mail de test depuis l'interface admin.
```

Exemple pas à pas

# Exemple pas à pas (suite)

```
Si ça échoue encore : appeler l'astreinte (numéro dans le tableau d'équipe).
Dernière vérification de ce document : 2 novembre.
```

**4. Pourquoi c'est bon** : on sait *quand* l'utiliser, chaque étape est une action, il y a une vérification, un plan B, et une date.

# À toi, avec aide

Rédige un **mini-ADR** pour la décision : *« Nous stockons les certificats au format PDF dans un stockage de fichiers, et non dans la base de données. »*
Complète :

- Contexte : _____
- Décision : _____
- Alternatives envisagées : _____
- Conséquences : _____

À toi, avec aide · réponse

# Indices progressifs

## Indice 1

Contexte : pourquoi se pose-t-on la question ? (taille des fichiers, nombre d'utilisateurs…)

## Indice 2

Une conséquence peut être positive (base plus légère) **et** négative (deux endroits à sauvegarder).

# Exemple

- **Contexte** : les certificats sont des fichiers de plusieurs centaines de ko, et il y en aura des milliers.
- **Décision** : les stocker dans un service de fichiers, et garder en base seulement leur adresse.
- **Alternatives** : les mettre en base (alourdit les sauvegardes) ; les générer à chaque demande (plus lent).
- **Conséquences** : base plus légère ; il faut sauvegarder aussi le stockage de fichiers.

# À toi, en solo

Choisis un projet (le tien, un projet scolaire, ou imaginaire) et écris un **README** d'une page avec :

1. ce que fait le projet, en 2 phrases ;
2. ce qu'il faut pour l'installer ;
3. comment le lancer ;
4. comment vérifier que ça marche ;
5. où demander de l'aide.

Demande à quelqu'un de suivre ton README **sans te poser de question**. Note où il se bloque.

À toi, en solo · réponse

# Critères de réussite

Une personne qui ne connaît pas le projet peut le lancer seule. Chaque étape est une action. Les mots techniques sont expliqués ou liés.

# Pièges classiques

| Piège | Pourquoi c'est un problème | Comment l'éviter |
|---|---|---|
| **Documenter à la toute fin** | On oublie, ou on ne sait plus pourquoi | Écrire au fil de l'eau |
| **Tout documenter** | Personne ne lit, tout se périme | Documenter ce qui répond à une vraie question |
| **Copier sans expliquer** | Une commande sans contexte est dangereuse | Dire à quoi elle sert |
| **Ne jamais mettre à jour** | Le document ment | Mettre une date et relire |
| **Écrire pour soi seul** | Le lecteur ne comprend pas | Choisir un lecteur précis |

Mini-boss

# Mini-boss

*Un·e nouveau·elle membre rejoint une équipe. La personne responsable de l'infrastructure est absente.*

Prépare un **kit d'accueil** de 3 documents courts :

1. un README (ce qu'est le projet, comment démarrer) ;
2. un schéma d'architecture simple (en Mermaid ou à la main) avec sa description en texte ;
3. un runbook pour un incident courant de ton choix.

Chaque document doit préciser **pour qui** il est écrit.

Mini-boss · réponse

# Exemple de kit

README du site HashBox ; schéma : navigateur → API → base + stockage ; runbook : « Que faire si la page de paiement est lente ? » avec 4 étapes et un plan B.

Quiz éclair

# Quiz éclair

**1.** Un ADR sert à :

A. lister les bugs
B. garder une décision et ses raisons
C. remplacer le code

**2.** « Le serveur pourrait être saturé » est :

A. un fait
B. une hypothèse
C. une décision

**3.** Un bon runbook :

A. est long et détaillé

Quiz éclair

# Quiz éclair (suite)

B. a des étapes courtes et une vérification
C. est écrit pour soi

**4.** 🔁 *Révision mission 04* : quand on corrige un bug, il faut :

A. s'arrêter dès que ça semble marcher
B. vérifier avec le scénario exact
C. changer plusieurs choses à la fois

**5.** 🔁 *Révision mission 01* : un problème bien écrit contient :

A. qui, situation, impact, contexte
B. une solution
C. un coupable

Quiz éclair · réponse

# Réponses

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

# Je réfléchis

1. Qu'est-ce que tu sais, que personne d'autre ne sait, et qui pourrait poser problème si tu étais absent·e ?
2. Qu'écrirais-tu en premier pour y remédier ?

À retenir

# À retenir

1. Ce qui n'est pas écrit n'existe pas pour l'équipe.
2. On documente pour répondre à des questions réelles.
3. README, architecture, ADR, runbook : chacun a son rôle.
4. Faits, hypothèses et décisions se séparent.
5. Un document a un lecteur, une date, et une vérification.

MISSION ACCOMPLIE

# +100 XP

## Badge débloqué : Passeur·se de savoir

Prochaine mission : Tout assembler : la méthode C-D-R-P-D-V-A

