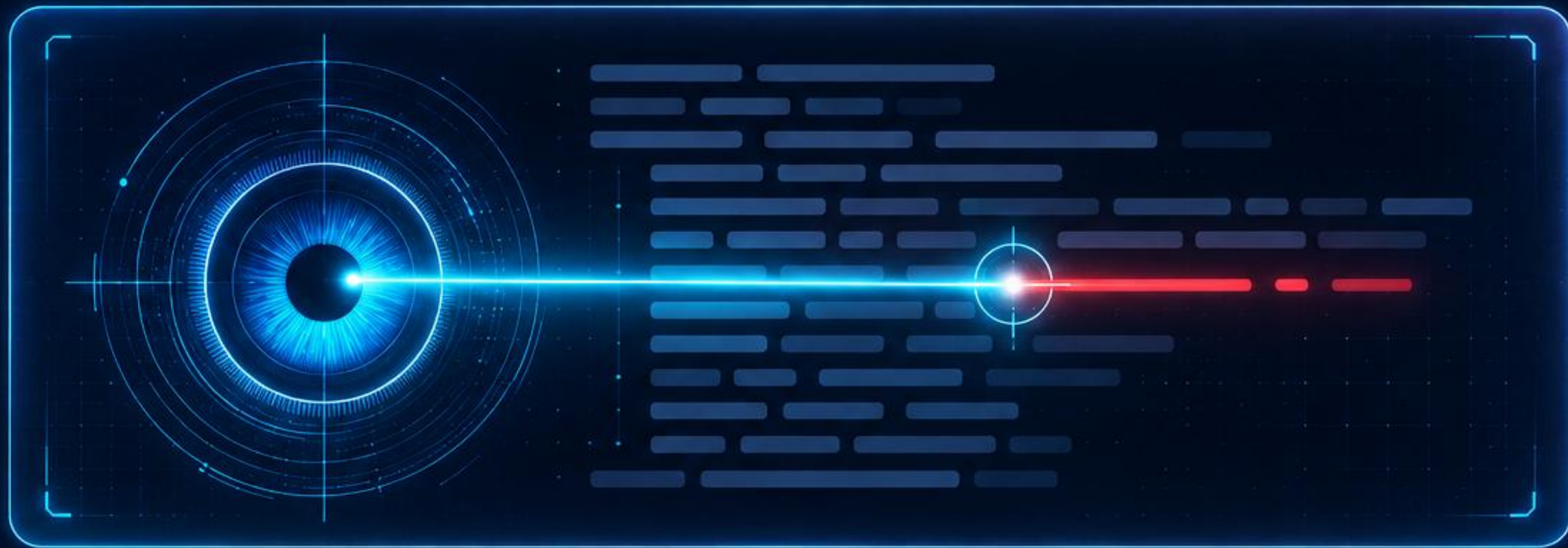


• Point de théorie • qualité du code •

Faire relire votre code par une machine

Formatage • Lint • Analyse statique



Mardi, jeudi



- **Mardi** : une PR de 20 lignes, GitHub en affiche 300. L'éditeur du binôme a tout réindenté.
- **Jeudi** : un identifiant inexistant dans l'URL → erreur 500. Personne n'avait pensé au `null`.



Même remède : des outils qui lisent le code à votre place.



Le soulignement rouge de VS Code

C'est de l'analyse statique.

Ce chapitre l'étend à **tout le projet** et à **toute l'équipe**.



Trois questions, trois outils

Formatage • « Est-ce écrit de manière uniforme ? » • réécrit sans changer le comportement

Lint • « Y a-t-il des motifs suspects ? » • signale, corrige parfois

Analyse statique • « Les types collent-ils ? » • prouve une erreur avant toute exécution

Le même code, trois verdicts

Analyse statique :

Cannot access property \$name
on Conversation|null

C'est l'erreur 500 de jeudi, attrapée avant le commit.

Trois stacks, une seule idée



PHPStan • Cannot access
property \$name on Conversation|null



mypy • Item "None"
has no attribute "name"



tsc • Object is possibly
'undefined'



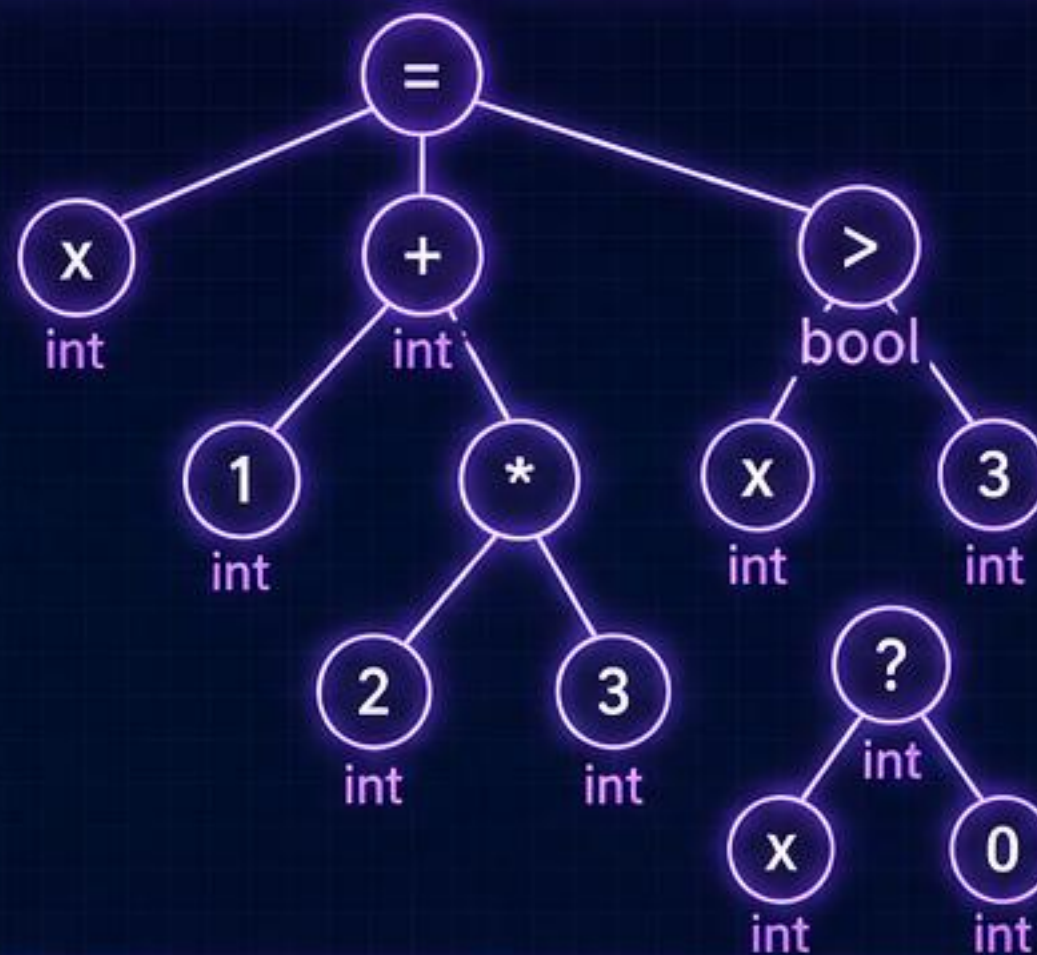
Cette valeur peut être vide, et vous faites comme si elle ne l'était jamais.

Lire · Déduire · Vérifier

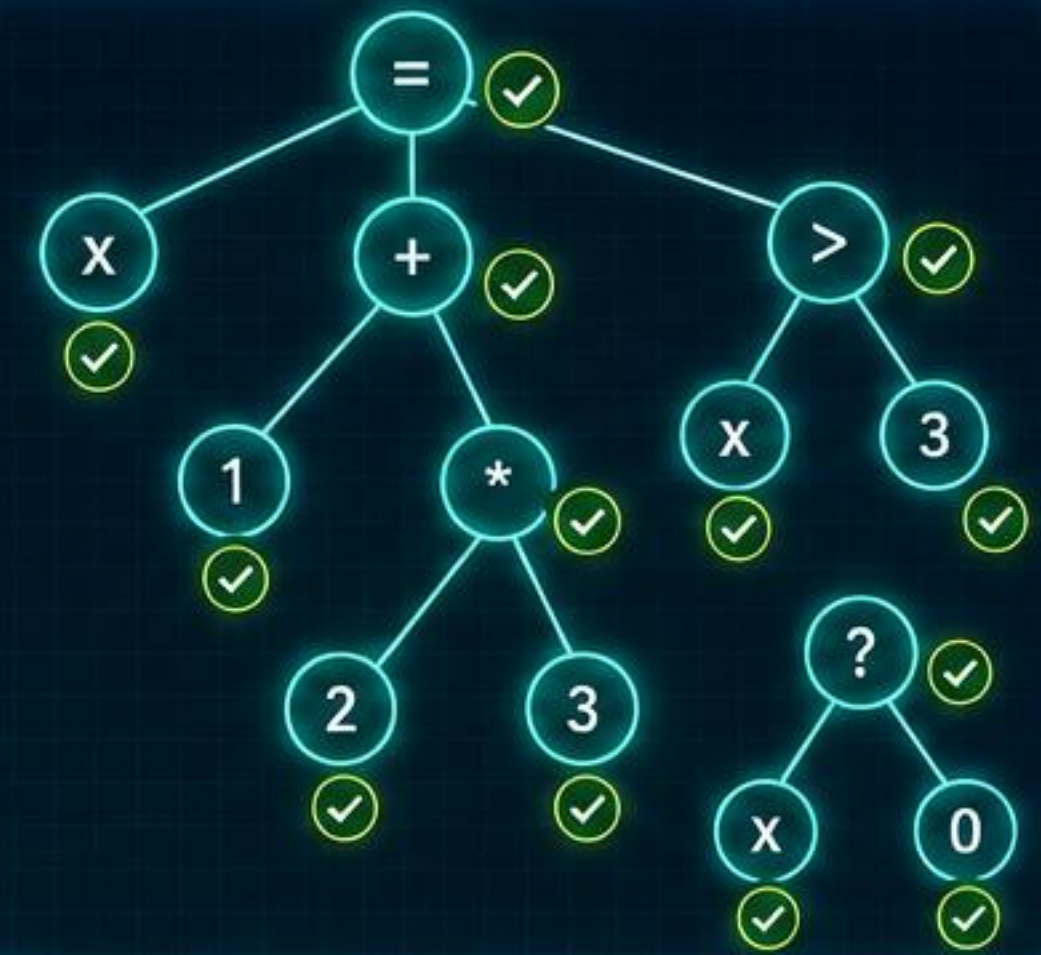
1. Lire le code et construire l'arbre syntaxique

```
let x = 1 + 2 * 3;  
let y = x > 3 ? x : 0;  
function f(n) {  
  if (n > 0) {  
    return n * f(n - 1);  
  } else {  
    return 1;  
  }  
}  
let z = f(y);
```

2. Déduire le type de chaque expression, de proche en proche



3. Vérifier que chaque usage est compatible



Sans exécuter le programme. Quelques secondes, tout le projet.



Fichier · ligne · code de règle · message

```
chat/views.py:3:8: F401 [*] `json` imported but unused
```


Votre trousse



Laravel

Pint • PHPStan + Larastan



Django

ruff • mypy + django-stubs



Node / TS / Vue

Prettier • ESLint • tsc / vue-tsc



On commence bas, on monte



PHPStan : niveaux 0 → 10

10



5

0



mypy : défaut → `--strict`



TypeScript : `"strict": true`



Faux positif ? On ignore une ligne, avec un commentaire.
Jamais le réglage global.

Éditeur · Commit · CI



Auditer les dépendances



composer audit • pip-audit • npm audit



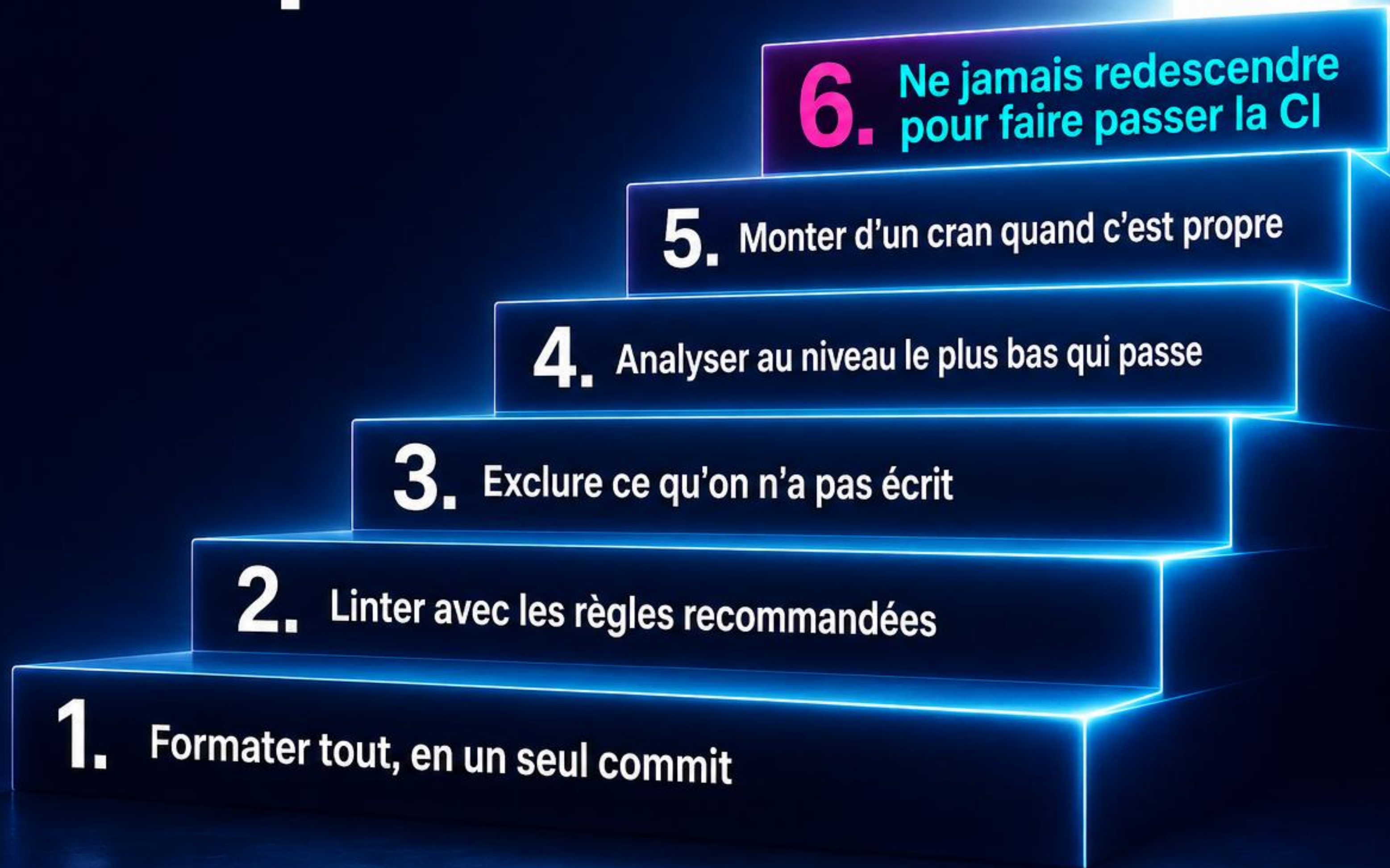
+ Dependabot dans les réglages du repo



Une faille dans une bibliothèque
est une faille chez vous.



Marche par marche



TP • 30 minutes

- ✓ Formatez, committez « Formatage » seul
- ✓ Lintez, corrigez avec `--fix`
- ✓ Analysez, montez jusqu'à la première erreur
- ✓ Introduisez un bug de type : l'analyseur le voit, les tests non
- ✓ Branchez le job `lint` et le check obligatoire sur `main`



Repo d'exemple : github.com/opmvpc/cicd26

✓ lint passing

