

# Plan de cours

Dix heures au niveau master

Adrien Técher · édition du 29 septembre 2026

---

## Versionnement et intégration continue

**Niveau :** master 1. **Durée :** dix heures en cinq séances de deux heures. **Effectif :** vingt-quatre étudiants au plus, douze binômes. **Outils :** Git, un GitLab privé et, au choix du binôme, Python avec pytest ou TypeScript sur Node.js 24. Le carnet et les dépôts suffisent pour suivre le cours ; le manuel n'est pas un prérequis.

### Objectifs

Lire un historique Git, distinguer répertoire de travail, index et commit, intégrer des contributions et choisir une réparation adaptée. Construire un pipeline simple, interpréter son résultat, chercher un défaut que les tests existants manquent et vérifier qu'une archive correspond à la source demandée.

Au bout du compte : prendre une décision justifiée sur un changement, en nommant l'état concerné, l'effet attendu, la vérification choisie et les limites du résultat.

### Calendrier

Heures de Paris. Chaque séance comprend une pause de cinq minutes.

| Séance | Créneau                 | Contenu                                                                                         |
|--------|-------------------------|-------------------------------------------------------------------------------------------------|
| 1      | 29 septembre, 9 h–11 h  | Retrouver la bonne version : lire les références, préparer un commit cohérent.                  |
| 2      | 29 septembre, 11 h–13 h | Intégrer deux demandes : fusionner, tester leur interaction, annuler et retrouver.              |
| 3      | 29 septembre, 14 h–16 h | Un pipeline vert, un test rouge : construire une CI, propager les échecs, identifier l'archive. |
| 4      | 20 octobre, 11 h–13 h   | Relier l'archive au commit ; passage individuel A.                                              |
| 5      | 20 octobre, 14 h–16 h   | Décider à partir des preuves ; passage individuel B.                                            |

Les trois premières séances alternent explications courtes, manipulations, revue et retour collectif. Les deux dernières laissent un atelier autonome pendant les passages.

## Le fil conducteur

Vous maintenez le service de tarification d'un petit opérateur de recharge de véhicules électriques. Une annonce a changé sur une autre branche, des sessions avortées faussent une réduction de flotte, un test échoue derrière un pipeline vert, une archive garde un ancien tarif. Ces incidents sont construits pour être reproductibles. Le service existe en Python et en TypeScript, avec les mêmes incidents et les mêmes montants ; chaque binôme choisit sa version en séance 1.

Les rôles de pilote et de relecteur tournent toutes les vingt minutes. Avant d'agir, on écrit une prédiction ; après, on choisit une observation capable de la contredire. Les aides et les reprises sont libres.

## Évaluation

La note sur vingt provient uniquement de deux passages individuels le 20 octobre : dix points pour l'histoire Git et l'action choisie, dix pour le diagnostic de CI et la relation entre source et artefact. Chaque passage dure environ trois minutes.

Le sujet est présenté sur place. Vous disposez de papier, des cartes et de la fiche de commandes fournie, sans annotation, sans ordinateur personnel ni assistant. La note porte sur la compréhension et l'adaptation observées, pas sur l'aisance ni sur la récitation.

**Aucun dépôt, code, pipeline, rapport, historique, devoir ou résultat de binôme ne rapporte de point.** Les activités du premier jour sont formatives. Une préparation personnelle peut aider à apprendre ; aucune production préparée n'est évaluée. Les ateliers se déroulent sans assistant génératif, avec documentation et entraide.

| Passage | Cinq critères, deux points chacun                                                                        |
|---------|----------------------------------------------------------------------------------------------------------|
| A       | Lire l'état ; prédire l'effet ; choisir l'action ; justifier ; adapter à une condition nouvelle.         |
| B       | Diagnostiquer ; utiliser une preuve ; choisir la vérification ; relier l'artefact à sa source ; adapter. |

Une autocorrection raisonnée est recevable. Une absence ou une accommodation donne lieu à un passage adapté sur les mêmes compétences, jamais à un devoir produit hors observation.

## Préparation

Avant le premier jour : suivre la fiche d'installation, vérifier Git et Python ou Node.js 24, ouvrir son espace privé. Les accès sont remis individuellement avec l'archive `kit-etudiants.zip`, qui permet de travailler hors ligne.

Pendant le cours : carnet, fiche de commandes, cartes et dépôts de séance. Pour approfondir : le manuel, ses exemples commentés et les exercices corrigés, que le carnet indique séance par séance.