1. Git, c'est quoi ?

Git = l'outil qui garde toutes les versions de ta plateforme e-commerce.

- Chaque modification sur le site (nouveau produit, design de page, fonctionalité ajoutée) peut être **enregistrée**. =>nouvelle version après chaque commit (créer une version)
- Tu peux **revenir à une version précédente** si ça casse quelque chose.
- Tu peux travailler avec d'autres développeurs sans se mélanger.

Métaphore :

Ton projet = **ton site e-commerce**
Git = **le carnet qui enregistre chaque version du site**

2. Les mots essentiels de Git (version e-commerce)

Terme Git	Explication simple	Métaphore e-commerce
Repository (repo)	Le projet avec son historique	Tout le site e-commerce avec toutes ses pages et fonctionnalités
Commit	Sauvegarde de ton travail	Enregistrer le site tel qu'il est maintenant
Branch	Nouvelle version pour tester	Créer une copie du site pour tester une nouvelle page, un nouveau design, ou une promo
Merge	Fusionner deux versions	Appliquer les changements testés dans le site principal
Clone	Copier un projet existant	Copier le site d'un autre développeur pour l'essayer chez toi
Push	Envoyer tes changements	Mettre en ligne tes changements sur le serveur ou GitHub
Pull	Récupérer les changements des autres	Importer les changements faits par d'autres développeurs
Staging area	Zone d'attente avant commit	Préparer les fichiers/modifications avant de les appliquer sur le site

3. Les commandes Git principales (version e-commerce)

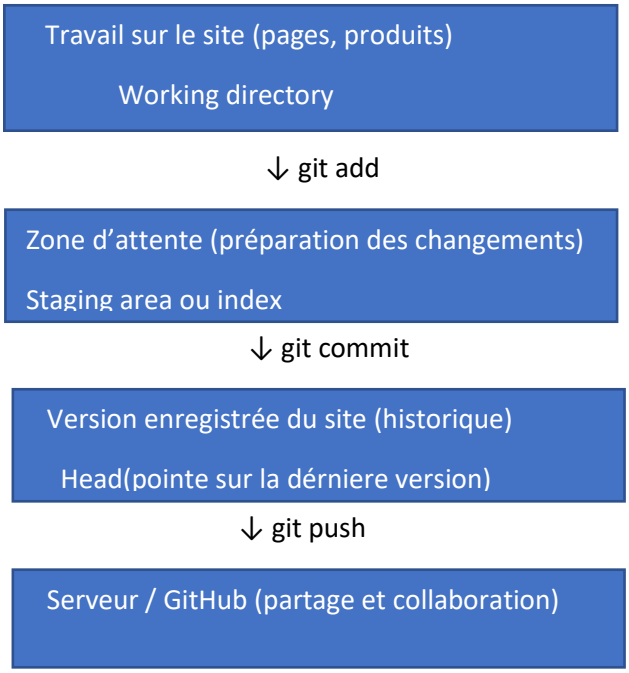
Commande	Ce que ça fait	Métaphore e-commerce
git init	Démarrer un projet Git	Créer ton site e-commerce version Git
git status	Voir ce qui a changé	Vérifier les pages ou produits modifiés
git add .	Préparer les fichiers pour sauvegarde	Mettre toutes les modifications dans la zone de préparation
git commit -m "message"	Sauvegarder le projet	Enregistrer le site tel qu'il est maintenant
git log	Voir toutes les sauvegardes	Feuilleter l'historique des versions du site
git branch nom_branche	Créer une nouvelle version	Créer une copie du site pour tester un nouveau design ou fonctionnalité
git checkout nom_branche	Passer à une branche	Travailler sur la version test du site

git merge nom_branche	Fusionner une branche	Intégrer la version test réussie dans le site principal
git clone url	Copier un projet existant	Copier le site d'un autre développeur pour l'essayer
git push	Envoyer les changements	Mettre en ligne tes changements sur GitHub ou serveur
git pull	Récupérer les changements	Importer les changements des autres développeurs

Exemple : application e-commerce

Commande	Description
git init	Tu démarres ton projet e-commerce
git add .	Tu modifies la page ''Produits'' → tu prépares les fichiers
git commit -m "Ajout page produits"	Tu sauvegardes ton site
git branch promo git checkout promo	Tu veux tester une nouvelle promotion → créer une branche
git add . git commit -m "Ajout promotion Ramadan"	Tu modifies le site pour la promo → tu sauvegardes
git checkout main git merge promo	Tout est OK → fusionner dans le site principal
git push origin main	Partager les changements sur GitHub :

Résumé visuel – e-commerce



Bonnes pratiques

- Faire des commits souvent = enregistrer chaque étape de ton site
- Créer une branche par fonctionnalité = tester une nouvelle promo ou une page sans toucher au site principal

- Toujours pull avant push = récupérer les mises à jour des autres développeurs
- Ne jamais travailler directement sur main = ne jamais modifier le site principal sans test

Exercice d'application :

Tu dois gérer une mini plateforme e-commerce locale avec Git.

1. Installer et configurer git
2. C'est quoi l'objectif de la configuration
3. Initialise un dépôt Git. (git init)
4. Ajouter les composants
5. Sauvegarde toutes les modifications (commit). (git add . | git commit -m' … ')

   On a utiliser l unique branche (main branch)

6. Crée une branche promotion pour tester une page promo. (git branch promo) => git checkout promo
7. Modifie produits.html en ajoutant une promotion.
8. Ajoute et commit les changements sur la branche promotion. Git add . git commit
9. Reviens sur la branche main et fusionne la branche promotion. Git checkout main
10. Supprime les modifications non désirées si tu as fait une erreur sur produits.html.

    git merge promo

11. Vérifie l'historique des commits et la différence entre Index et WD(Working directory)

    Git log