**Cheat Sheet – Commandes essentielles Git/GitHub & Visual Studio Code  
(débutant → avancé)**

*Version imprimable • Généré le 24/08/2025 à 00:35*

**I. Git (pour GitHub) – commandes ligne de commande**

Triées des plus courantes aux plus avancées. La dernière colonne indique l’équivalent dans GitHub Desktop lorsqu’il existe.

**Très courantes**

|  |  |  |  |
| --- | --- | --- | --- |
| Commande | Description | Exemple | Équivalent GitHub Desktop |
| git status | Affiche l’état de l’arborescence (modifiés, suivis, non suivis). |  | Onglet « Changes » / Historique |
| git add . | Ajoute toutes les modifications suivies à l’index (staging). |  | Stage all changes |
| git commit -m "message" | Valide l’instantané des fichiers indexés avec un message. | Ex. git commit -m "Initialisation" | Commit to <branch> |
| git push origin main | Envoie vos commits locaux vers la branche distante. |  | Push origin |
| git pull | Récupère et fusionne les modifications distantes dans la branche courante. |  | Pull origin |
| git clone <URL> | Copie un dépôt distant en local. | Ex. git clone https://github.com/USER/REPO.git | Clone repository |

**Fréquentes**

|  |  |  |  |
| --- | --- | --- | --- |
| Commande | Description | Exemple | Équivalent GitHub Desktop |
| git switch -c <branche> | Crée et bascule sur une nouvelle branche. | Ex. git switch -c feature/x | New branch |
| git switch <branche> | Bascule sur une branche existante. | Ex. git switch main | Switch branch |
| git merge <branche> | Fusionne <branche> dans la branche courante. | Ex. git merge feature/x | Merge into current branch |
| git stash | Range provisoirement les changements non commités. |  | Stash changes |
| git log --oneline | Affiche l’historique des commits en résumé. |  | View → History |
| git diff | Montre les différences non indexées / entre commits. |  | Onglet « Changes » (diff) |
| git remote add origin <URL> | Associe le dépôt local à un dépôt distant nommé origin. |  | Publish repository (premier push) |
| git fetch origin | Récupère les mises à jour distantes sans fusion. |  | Fetch origin |

**Occasionnelles**

|  |  |  |  |
| --- | --- | --- | --- |
| Commande | Description | Exemple | Équivalent GitHub Desktop |
| git revert <sha> | Crée un commit qui annule un commit donné (sécurisé). |  | Revert commit |
| git tag -a v1.0 -m "…" | Crée un tag annoté (versionnage). |  | — (via Git CLI / GitHub.com) |
| git rm <fichier> | Supprime un fichier suivi et enregistre la suppression. |  | Delete (puis commit) |
| git mv <src> <dst> | Renomme/Déplace un fichier suivi. |  | Rename (puis commit) |
| git blame <fichier> | Montre auteur/date par ligne (pour traçabilité). |  | — (Historique fichier) |
| git restore --staged <fichier> | Retire un fichier de l’index (unstage). |  | Unstage changes |
| git restore <fichier> | Restaure depuis le dernier commit (annule modifs locales). |  | Discard changes (fichier) |

**Avancées (prudence)**

|  |  |  |  |
| --- | --- | --- | --- |
| Commande | Description | Exemple | Équivalent GitHub Desktop |
| git reset --hard HEAD | Réinitialise l’arborescence et l’index au dernier commit (perte de modifs non commités). |  | Discard all changes |
| git rebase origin/main | Rejoue vos commits au-dessus de origin/main (historique réécrit). |  | — (Git CLI) |
| git cherry-pick <sha> | Applique un commit précis sur la branche courante. |  | — (Git CLI) |
| git clean -fd | Supprime fichiers/dossiers non suivis. |  | — (Git CLI) |
| git config --global user.name "Nom" | Configure l’identité de l’auteur (global). |  | — (Git CLI) |
| git config --global user.email "email@ex.com" | Configure l’e-mail de l’auteur (global). |  | — (Git CLI) |

**Optionnel : GitHub CLI (gh)**

|  |  |  |
| --- | --- | --- |
| Commande (gh) | Description | Exemple |
| gh auth login | Connexion à GitHub depuis la ligne de commande. |  |
| gh repo clone <owner>/<repo> | Cloner un dépôt par son nom court. | Ex. gh repo clone user/projet |
| gh repo create | Créer un dépôt (local & distant) rapidement. |  |
| gh pr create | Créer une Pull Request depuis votre branche. |  |
| gh pr status | Voir le statut des PR liées au repo/branche. |  |
| gh issue list | Lister les issues ouvertes. |  |

**II. Visual Studio Code – commandes & raccourcis (Windows)**

Triées des plus courantes aux plus avancées. Utilisez la palette (Ctrl+Shift+P) pour lancer une commande par son nom.

**Très courantes**

|  |  |  |
| --- | --- | --- |
| Commande / Action | Raccourci | Ce que ça fait |
| Ouvrir la palette de commandes | Ctrl+Shift+P | Accéder à toutes les commandes par nom. |
| Ouvrir rapidement un fichier | Ctrl+P | Recherche et ouverture par nom de fichier. |
| Enregistrer | Ctrl+S | Sauvegarder le fichier courant. |
| Enregistrer tout | Ctrl+K puis S | Sauvegarder tous les fichiers modifiés. |
| Terminal : basculer/afficher | Ctrl+` | Afficher/Masquer le terminal intégré. |
| Nouveau terminal | Ctrl+Shift+` | Créer un nouveau terminal intégré. |
| Basculer le panneau (terminal, problèmes…) | Ctrl+J | Afficher/Masquer le panneau inférieur. |
| Basculer la barre latérale | Ctrl+B | Afficher/Masquer l’explorateur latéral. |
| Rechercher dans le fichier | Ctrl+F | Chercher du texte dans l’éditeur actif. |
| Rechercher dans le projet | Ctrl+Shift+F | Recherche globale dans le dossier. |

**Fréquentes**

|  |  |  |
| --- | --- | --- |
| Commande / Action | Raccourci | Ce que ça fait |
| Source Control (Git) | Ctrl+Shift+G | Accéder au suivi de code (staging/commit/push). |
| Git : Stage All Changes |  | Indexe tous les changements (via palette). |
| Git : Commit |  | Valide les changements (via palette / SCM). |
| Git : Push |  | Pousse vers le remote (via palette / SCM). |
| Formater le document | Shift+Alt+F | Appliquer le formateur (Prettier, etc.). |
| Renommer symbole | F2 | Renommer variable/fonction partout. |
| Aller à la définition | F12 | Ouvrir la définition d’un symbole. |
| Peek Definition | Alt+F12 | Afficher la définition en aperçu inline. |
| Paramètres | Ctrl+, | Ouvrir les réglages utilisateur/espace de travail. |
| Extensions | Ctrl+Shift+X | Gérer les extensions VS Code. |

**Occasionnelles**

|  |  |  |
| --- | --- | --- |
| Commande / Action | Raccourci | Ce que ça fait |
| Scinder l’éditeur | Ctrl+\ | Ouvrir deux vues côte à côte. |
| Basculer retour à la ligne | Alt+Z | Activer/Désactiver le word wrap. |
| Commenter la ligne | Ctrl+/ | Ajouter/retirer // ou # selon le langage. |
| Aller à la ligne… | Ctrl+G | Se rendre à une ligne donnée. |
| Changer de fichier actif | Ctrl+Tab | Basculer entre fichiers ouverts. |
| Raccourcis clavier | Ctrl+K puis Ctrl+S | Personnaliser/consulter les raccourcis. |

**Avancées (utile selon besoins)**

|  |  |  |
| --- | --- | --- |
| Commande / Action | Raccourci | Ce que ça fait |
| Ouvrir l’espace de travail |  | Fichiers .code-workspace pour projets multi-dossiers. |
| Tasks: Run Task |  | Lancer des tâches déclarées (build/tests). |
| Terminal: Select Default Profile |  | Choisir CMD/PowerShell/Bash par défaut. |
| Developer: Toggle Developer Tools |  | Console DevTools pour diagnostics. |

Astuce : Dans VS Code, vous pouvez effectuer la plupart des actions Git sans quitter l’éditeur (onglet « Source Control »). Pour des opérations avancées (rebase, cherry-pick…), préférez la ligne de commande Git.