|  |  |
| --- | --- |
| pwd | Savoir dans quel dossier je suis |
| mkdir <directory\_name> | Créer un nouveau dossier |
| touch <file\_name> | Créer un fichier |
| ls | Liste le dossier courant |
| Ls -la | Lister le contenu du répertoire courant |
| cd dossier1 | Aller dans le dossier ‘dossier1’ |
| cd .. | Remonter d’un dossier |
| git config --global user.name " " git config --global user.email " " | Commandes de configuration après installation de gitbash |
| git status | Afficher l’état du fichier |
| git diff | Afficher les modifs avant commit |
| Git init | Initialiser le dépôt (se mettre sur le bon dossier) |
| Git add . | Ajouter toutes les modifications |
| git commit -m "message du commit" | Créer un nouveau commit |
| Git log | Lister les commits |
| git log -n 2 | Afficher les deux derniers commits |
| git show <hash\_du\_commit> | Voir un commit spécifique |
| git checkout <hash\_du\_commit> | Remettre la version du sha-1 |
| git checkout master | Remettre la version la plus récente |
| Git pull | Récupérer travail depuis dépôt distant |
| git push | Pousser les modifications vers le serveur |
| git push --force | Pousser de force des modifications vers le serveurs distant |
| git checkout -b B1 | Créer une nouvelle branche B1 et y basculer |
| git checkout main | Retourner à la branche principale |
| git branch -d B1 | Supprimer la branche B1 |
| git push origin B1 | Envoyer une branche B1 vers le dépôt distant |
| git fetch | Mettre à jour le dépôt local |
| git merge B1 | Fusionner une branche B1 avec la branche active |
| Git checkout -- <nomFichier> | Annuler les modifications d'un fichier |
| Git fetch origin | Récupérer les modifications de l'origine |
| Git reset –hard origin/master | Réinitialiser à l'origine/master (réinitialisation forcée) |
| Git remoterename ‘B1’ ‘B2’ |  |
| Git remote -v | Afficher les dépôts distants |
| Git remote add origin url | Ajouter une URL de dépôt distant |
| Merge request /pull request | permet de proposer, de discuter et de réviser des modifications de code avant de les fusionner dans la branche principale d'un projet. |
| git checkout main git merge B1 | Fusionner une branche spécifique avec la branche actuelle |
| git branch -a | Lister toutes les branches |

Exercice 2 :

Pour la réalisation d’un projet, le découpage réalisé a donné le tableau d’ordonnancement suivant : Unité temps = jour

|  |  |  |
| --- | --- | --- |
| Tâche | P(X) | Durée |
| A | ------- | 6 |
| B | ------- | 5 |

|  |  |  |
| --- | --- | --- |
| C | A | 4 |
| D | B | 6 |
| E | C | 5 |
| F | A,D | 6 |
| G | E,F | 4 |

1. Réaliser le diagramme de gant
2. Calculer les marges
3. Déterminer le chemin critique
4. Réaliser le diagramme de Pert
5. Calculer les marges
6. Déterminer le chemin critique

Solution EX 2 :

| Tâche | P(X) | Suivant(s) | Durée | Date début au plutôt | Date fin au plutard | Marge libre | Marge totale |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | - | C,F | 6 | 0 | 6 | 0 | 0 |
| B | - | D | 5 | 0 | 5 | 0 | 0 |
| C | A | E | 4 | 6 | 10 | 0 | 0 |
| D | B | F | 6 | 5 | 11 | 0 | 0 |
| E | C | G | 5 | 10 | 17 | 2 | 2 |
| F | A,D | G | 6 | 11 | 17 | 0 | 0 |
| G | E,F | - | 4 | 17 | 21 | 0 | 0 |

* Le chemin critique : B → D → F → G



