1.  
Git - це розподілена система керування версіями, яка використовується для зберігання та відстеження змін в програмному коді та інших текстових документах. Вона дозволяє командам розробників спільно працювати над проектами, відстежувати історію змін, вирішувати конфлікти та багато іншого. Основні дії та команди, які виконуються в Git, включають:

1. **Клонування репозиторію**: **git clone <URL>** - створює копію віддаленого репозиторію на локальному комп'ютері.
2. **Ініціалізація нового репозиторію**: **git init** - створює новий репозиторій у поточній директорії.
3. **Додавання змін**: **git add <файл>** - додає зміни у вказаний файл до стадії готовності для коміту.
4. **Зроблення коміту**: **git commit -m "Повідомлення про коміт"** - фіксує зміни, додані до стадії готовності, і додає їх до історії репозиторію.
5. **Відстеження гілок**: **git branch** - відображає список усіх гілок в репозиторії, а **git checkout <гілка>** - перемикається на іншу гілку.
6. **Стягування змін**: **git pull** - оновлює локальний репозиторій з віддаленого, а **git fetch** - завантажує зміни, але не об'єднує їх з локальною гілкою.
7. **Відправлення змін**: **git push** - відправляє локальні зміни на віддалений репозиторій.
8. **Подивитися історію комітів**: **git log** - виводить історію комітів у поточній гілці.
9. **Вирішення конфліктів**: При конфліктах під час злиття гілок чи злиття змін, використовують команди **git mergetool** та ручні засоби для вирішення конфліктів.
10. **Видалення та переміщення файлів**: **git rm <файл>** - видаляє файл із репозиторію, **git mv <старий\_файл> <новий\_файл>** - переміщує або перейменовує файл.
11. **Створення нової гілки**: **git branch <ім'я\_гілки>** - створює нову гілку, а **git checkout -b <ім'я\_гілки>** - створює і переходить на нову гілку.
12. **Об'єднання гілок**: **git merge <гілка>** - об'єднує зміни з іншої гілки в поточну.
13. **Синхронізація гілок**: **git rebase <гілка>** - перебудовує історію комітів для гармонізації змін з іншою гілкою.

Git надає потужні можливості для роботи з версіями та спільної роботи над проектами і є незамінним інструментом для розробників у сфері програмування.

2.

"Коміт" (або "commit" в англійській мові) в системі керування версіями, такій як Git, є операцією, яка фіксує (зберігає) зміни в файловій системі та в історії проекту. Кожен коміт представляє собою фіксацію (знімок) стану всіх файлів у репозиторії в певний момент часу. Він дозволяє відслідковувати, які зміни були внесені в проект, коли вони були внесені, і хто вніс ці зміни.

Основні функції та значення комітів включають:

***Збереження історії***: Кожен коміт зберігає інформацію про зміни, які були внесені в файлову систему з моменту попереднього коміту. Ця інформація включає змінені файли, видалені файли, нові файли і повідомлення про коміт, яке пояснює, що саме було зроблено.

* ***Відслідковування змін***: Кожен коміт дозволяє вам відслідковувати, які конкретні зміни були внесені у файлі або наборі файлів. Це допомагає розробникам розуміти, що було змінено у коді, і виявляти помилки чи невідповідності.
* ***Спільна робота***: Коміти дозволяють різним розробникам працювати над одним проектом одночасно. Вони можуть робити свої зміни, фіксувати їх у комітах і спільно працювати над великими проектами без конфліктів.
* ***Повернення до попередніх станів***: Ви можете повертатися до попередніх комітів, якщо виникнуть проблеми або якщо ви захочете повернутися до старого коду. Це дозволяє вам легко виправляти помилки та відновлювати попередні версії проекту.

В загальному, коміти в системі керування версіями є ключовим інструментом для збереження, відстеження та управління змінами в проекті. Вони допомагають розробникам зберігати структуровану історію проекту та ефективно спільно працювати над ним.