

Конфигурация:

1. Имя пользователя

```
git config --global user.name "Your_user_name_on_github"
```

2. Почта пользователя

```
git config --global user.email "Your_email_on_github"
```

3. Выбор редактора

```
git config --global core.editor vim
```

4. Утилита сравнения

```
git config --global merge.tool vimdiff
```

5. По файловое отображение изменений

```
git config --global status.showUntrackedFiles all
```

6. Создание псевдонима для команд

```
git config alias.<alias> <command>
```

7. Проверка настроек

```
git config --list
```

Работа с репозиторием:

1. Инициализация репозитория

```
git init
```

2. Добавление файлов под контроль

```
git add <file>
```

3. Добавление всех файлов под контроль

```
git add .
```

4. Фиксация изменений

```
git commit
```

5. Игнорирование индексации

```
git commit -a -m "comment"
```

6. Изменение последнего коммита

```
git commit --amend
```

7. Клонирование удаленного репозитория

```
git clone <remote>
```

8. Клонирование удаленного репозитория в указанный каталог (директорию)

```
git clone <remote> <dir>
```

9. Определение состояния файлов

```
git status
```

10. Просмотр измененных, но не проиндексированных файлов, которые войдут в следующий коммит

```
git diff
```

11. Просмотр проиндексированных файлов, которые войдут в следующий коммит

```
git diff --staged
```

12. Удаление файла с индексацией

```
git rm <file>
```

13. Удаление файла из индекса (при этом файл остается в рабочем каталоге)

```
git rm --cached <file>
```

14. Перемещение файлов

```
git mv <old file> <new file>
```

15. Просмотр истории коммитов

```
git log
```

16. Отмена индексации файла

```
git reset HEAD <file>
```

17. Отмена изменений файла

```
git checkout -- <file>
```

Ветвления:

1. Показ существующих веток

```
git branch
```

2. Создание новой ветки

```
git branch <branch>
```

3. Удаление ветки

```
git branch -d <branch>
```

4. Просмотр последнего коммита на каждой из веток

```
git branch -v
```

5. Просмотр веток, для которых выполнены слияния

```
git branch --merged
```

6. Просмотр веток, для которых не выполнено слияние

```
git branch --no-merged
```

7. Принудительное удаление ветки

```
git branch -D <branch>
```

8. Создание новой ветки из ветки на удаленном сервере

```
git branch --track <branch> <remote branch>
```

9. Переход на ветку

```
git checkout <branch>
```

10. Создание ветки с переходом на неё

```
git checkout -b <branch>
```

11. Слияние веток

```
git merge <branch>
```

12. Запуск графического инструмента для разрешения конфликтов

```
git mergetool
```

13. Отправка ветки на удаленный сервер

```
git push <remote> <branch>
```

14. Удаление ветки на удаленном сервере

git push <remote> :<branch>

Работа с метками:

1. Просмотр меток

git tag

2. Создание аннотированных меток

git tag -a <tag> -m "comment"

3. Создание легковесной метки

git tag <tag>

4. Выставление меток позже

git tag -a <tag> <hash>

5. Просмотр данных метки вместе с коммитом

git show <tag>

6. Обмен метками

git push <remote> <tag>

7. Отправка всех меток за один раз

git push <remote> --tags

Работа с удаленным репозиторием:

1. Отображение удаленных репозиториев

git remote

2. Добавление удаленных репозиториев

git remote add <alias> <remote>

3. Информация об удаленном репозитории

git remote show <remote>

4. Переименование удаленных репозиториев

git remote rename <old name> <new name>

5. Удаление удаленных репозиториев

git remote rm <remote>

6. Получение данных из удаленных репозиториев

git fetch <remote>

7. Получение данных из удаленного репозитория и слияние с локальным

git pull <remote> <branch>

8. Отправление локальных изменений на удаленный сервер

git push <remote> <branch>

Работа с файлом '.gitignore':

1. # — комментарий

2. **/file.a** — Игнорировать файл **'file.a'** в текущем каталоге
3. ***~** — Игнорировать все файлы, заканчивающиеся на **~**
4. ***.[ab]** — Игнорировать файлы, заканчивающиеся на **.a** или **.b**
5. ***.a** — Игнорировать все файлы, заканчивающиеся на **.a**
6. **dir/** — игнорировать все файлы в каталоге **dir/**
7. **!file.a** — Но не игнорировать файл **file.a**
8. **/*.a** — Игнорировать все файлы, заканчивающиеся на **.a** в текущем каталоге
9. **/dir/** — Игнорировать каталог **dir** и всё его содержимое