1. создать локальный репозиторий в текущей папке

cd (путь)

git init

2. посмотреть статус текущего репозитория

git status

3. что такое ветка и какая веткая является обычно основной?

Ветка - последовательность коммитов, основная ветка - master.

4. добавить файл в контекст который будет коммититься

git add (имя файла)

5. создать коммит на основе текущего контекста для комита и указать для него комментарий

git commit -m "(комментарий)"

6. создать коммит, включающий изменения всех наблюдаемых файлов, и указать для него комментарий

git commit -a -m "blablabla"

7. посмотреть протокол коммитов

git log (чтобы выйти - нажмите 'q')

8. просмотреть информацию о текущих настройках

git config -l

9. убрать файл из контекста

git reset (имя файла)

10. посмотреть изменения в файле по сравнению с последним коммитом

git diff (имя файла)

11. убрать изменения относительно последнего коммита из файла

Если файл уже в конексте, то git reset (имя файла) и git checkout (имя файла)

Если же не в конексте, то просто git checkout (имя файла)

12. просмотреть в графическом интерфейсе историю коммитов

gitk

13. добавить в контекст коммита все измененные и созданые файлы

git add --all

14. изменить глобальные/локальные настройки

git config --global user.name LOH

15. переписать имя пользователя

git config --global user.name LOH

16. просмотреть существующие ветки

git branch

17. создать новую ветку

git branch (имя ветки)

18. переключиться на другую ветку

git checkout (имя ветки) или git checkout - ("-" - переключают между 2мя ветками)

19. создать новую ветку и сразу же переключиться на нее

git checkout -b (имя файла)

20. удалить ветку/ удалить ветку, даже если она не примержена

git branch -d (имя ветки); если не примержена (merge), то git branch -D (имя ветки)

21. просмотреть в графическом интерфейсе историю коммитов по всем веткам

gitk

22. примержить изменения из указанной ветки в текущую

git merge (имя ветки)

23 - 25. в каком случае могуть появиться конфликты? сделайте так!

Конфликты появляются, когда в одной и той же строке разный текст(код) в обеих ветках;

когда в одной ветке файл удалили, а в другой изменили.

ПРИМЕР, господа:

git branch master

touch 1.txt

git add 1.txt

git commit -m "asd"

git branch help

echo 1 > 1.txt

git add 1.txt

git commit -m "ads"

git checkout master

echo 2 > 1.txt

git add 1.txt

git commit -m "das"

git merge help ; После редактор скажет, что случился конфликт и надо его пофиксить вводим

cat 1.txt Выведет что-то похожее на это:

1

2

и выбираем то, что требуется (1 или 2)

echo 1 > 1.txt

git add 1.txt

git commit -m "sad"

ГОВОРИТЕ, если не работает, не получается и прочее.

24. как посмотреть в каких файлах конфликты

25. как устранить конфликты

26. как переключиться на указанный коммит

git log

git checkout de4837ce... (рандомный набор символов рядом со словом commit)

27. сделать ребазирование(rebase) текущей ветки

git rebase help (http://habrahabr.ru/post/161009/ - хорошая статья про rebase)

28. устранение конфликтов во время ребазирования

Тоже что и в мерже

29. отменить ребазирование во время конфликтов

git rebase --abort

30 пропустить текущний конфликтный коммит и перейти к следующему

git rebase --skip

31. отправить изменения из локального репозитория для указанной ветки в удаленный(дистанционный)

git push origin master

32. Забрать изменения из репозитория, для которого были созданы удаленные ветки по умолчанию

git fetch

git checkout -b (имя удалённой ветки)

33. забрать изменения удаленной ветки из репозитария по умолчанию, основной ветки

git pull origin (имя ветки)

34. создание копии репозитория

git clone (url)

35 удаление и переименовывание файла

git rm ; git mv

36. создание, удаление, отправка файла в удаленный репозиторий, удаление файла в

удаленном репозитории

На битбакете найдёте кнопки для этого.

37. отправка ветки в удаленный репозиторий, удаление ветки в удаленном репозитории

git push origin --delete (имя ветки)