

GIT GUIDE

СОЗДАНИЕ НОВОГО РЕПОЗИТОРИЯ

Для того чтобы создать новый репозиторий git необходимо открыть папку где вы хотите его разместить и выполнить команду

```
git init
```

Проверка состояния репозитория

```
git status
```

ПОДГОТОВКА И КОММИТ

Чтобы подготовить изменения (добавить их в Индекс) используйте

```
git add <имя_файла>  
git add *
```

Это первый шаг в основном рабочем процессе. Сделать коммит подготовленных изменений можно командой

```
git commit -m "Описание коммита"
```

Теперь изменения закреплены в локальном репозитории, но еще не в удаленном репозитории.

ОТПРАВКА ИЗМЕНЕНИЙ

Чтобы отправить эти изменения в ваш удаленный репозиторий, выполните

```
git push origin master
```

Можно изменить master на любую другую ветвь чтобы отправить изменения на неё.

Если вы еще не клонировали существующий репозиторий и хотите подключить ваш к удаленному, вам нужно на GitHub создать новый репозиторий, для этого переходим по ссылке <https://github.com/new> и потом добавить его, выполнив

```
git remote add origin <адрес_сервера>
```

Теперь вы можете отправлять изменения на удаленный репозиторий. Для этого нужно будет ввести логин и пароль от вашего GitHub

ПОЛУЧЕНИЕ РЕПОЗИТОРИЯ

Создать локальную рабочую копию репозитория можно командой

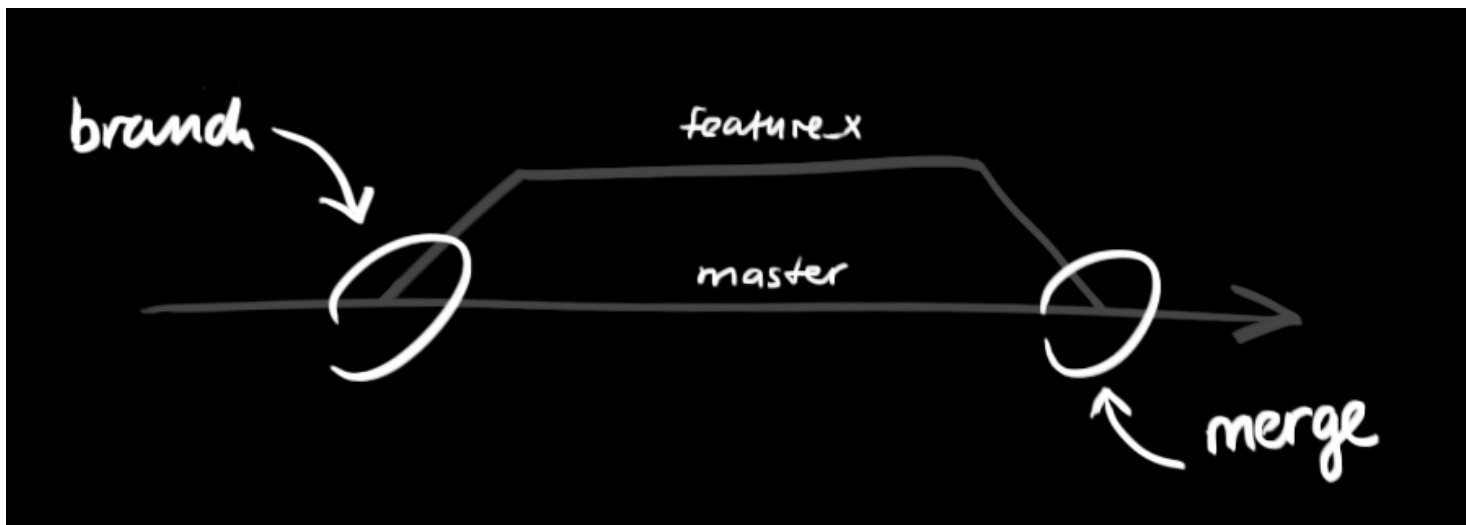
```
git clone /путь/к/репозиторию
```

когда используется удаленный сервер, команда будет

```
git clone юзер@хост:/путь/к/репозиторию
```

ВЕТВЛЕНИЕ

Ветки используются для разработки функционала изолированного от остального. Ветка **master** используется по умолчанию когда вы создаете репозиторий. Используйте другие ветки для разработки и слияние в **master** когда разработка завершена.



Создать новую ветку с названием "feature_x" и переключиться на неё можно командой

```
git checkout -b feature_x
```

переключиться обратно на *master*

```
git checkout master
```

удалить ветку

```
git branch -d feature_x
```

ветка не будет доступна тем, кто пользуется с вами удаленным репозиторием пока вы не отправите её туда

```
git push origin <имя_ветки>
```

BЕТКА GITHUB PAGES

Чтобы мы могли "смотреть" наши репозитории по ссылке типа <http://alexcss.github.io/css-diner-shopping-list/> (где *css-diner-shopping-list* это имя репозитория, а *alexss* ваш логин) нужно создать ветку **gh-pages**

```
git branch gh-pages
```

сменить ветку на gh-pages

```
git checkout gh-pages
```

синхронизировать gh-pages с master (git merge master)

```
git rebase master
```

отправить ветку на сервер.

```
git push origin gh-pages
```

вернуться в master

```
git checkout master
```

Эти команды можно объединить в одну строку

```
git checkout gh-pages && git merge master && git checkout master && git push --all
```

После нужно будет ввести логин и пароль от вашего GitHub

Чтобы быстро добавить все изменения и залить на GitHub используйте

```
git add . && git commit -m "Some descriptive commit message"
```

```
git checkout gh-pages && git merge master && git checkout master && git push --all
```

ОБНОВЛЕНИЕ И СЛИЯНИЕ

Обновить ваш локальный репозиторий, можно командой

```
git pull
```

которая заберет изменения из удаленного репозитория и проведет слияние с активной веткой. Для того чтобы слить другую ветку с активной (например master), используйте команду

```
git merge <имя_ветки>
```

В обоих случаях git пытается автоматически слить изменения. К сожалению, это не всегда возможно и результатом станет *конфликт*. Вы ответственны за разрешение возникших *конфликтов*, путем ручного редактирования файлов указанных git. После изменений вам надо пометить их как слитые

```
git add <имя_файла>
```

перед слиянием вы можете предварительно посмотреть на изменения

```
git diff <имя_ветки> <имя_другой_ветки>
```

РАБОТА С УДАЛЕННЫМ РЕПОЗИТОРИЕМ

Добавление удаленного репозитория.

```
git remote add origin <адрес_репозитория>
```

Получить список удаленных репозитория

```
git remote
```

Заливка файлов в удаленный репозиторий. (Попросит ввести логин и пароль)

```
git push origin master
```

Скачивание файлов с удаленного репозитория

```
git pull origin master
```

Переименование удаленного репозитория

```
git remote rename староеИмя новоеИмя
```

Удаление

```
git remote rm имя
```

ПОЛЕЗНЫЕ КОМАНДЫ

Помощь по команде

```
git --help [имя_команды]
```

использовать цветной вывод в терминале

```
git config color.ui true
```