# Навигация

pwd — Показать текущий каталог  
ls - Показать файлы в данной папке, кроме скрытых  
ls -f — Показать файлы в данной папке, включая и скрытые  
cd c:/ — Перейти в конкретный каталог  
cd - — Вернуться назад  
cd .. — Выйти на 1 уровень вверх  
cd ../.. — Выйти на 2 уровня вверх

# Основная конфигурация

**git init** – создаст гит-репозиторий (появиться папочка .git в корне проекта)

**git config --global user.name "John Doe"** – уккзать свое имя

**git config --global user.email** [**johndoe@example.com**](mailto:johndoe@example.com) **–** указать свою почту

**cat <.git/config>** - показать содержимое файла <config>

Настройки бывают:

* **--system** – общесистемные, для всех пользователей, хранится в **C:\Program Files\Git\etc\gitconfig**
* **--global** – на уровне конкретного пользователя (относится ко всем проектам пользователя) пишутся в **C:\Users\Ilya\.gitconfig**
* **--local** – на уровне проекта – они пишутся в файл .**git/config**

Git ищет эти параметры снизу вверх

**git config --list (--global)** – список параметров конфигурации (только глобальные)

**git config --unset** (something) – что-то отменить в настройках

**git config –h** – посмотреть все команды конфига

**git help** **<command>** - помощь

**/q** – выход из листалки less

# Создание репозитория, первый коммит

**git init** – создаст гит-репозиторий (появиться папочка .git в корне проекта)



В index пишутся отслеживаемые на данный момент файлы, в Repository – вся история коммитов

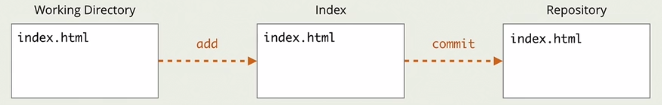
**git status** – взгляд со стороны git на файлы нашего проекта.

* Untracked files – те, которые не добавлены в index

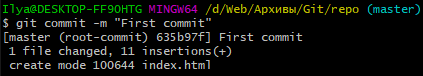
**git add <file>** – добавит файл в index

**git commit -m "comment"**  – добавить файл в прямоугольник Repository

*Когда git status ничего не выводит, значит картинка такая:*



Состояние рабочей области индекса и репозитория равны

 - в третьей строке – id коммита

**git reset HEAD <file>** – удаляет из index лишний <file>

# Игнорирование: .gitignore

<https://learn.javascript.ru/screencast/git#config-gitignore>

# Git show, автор и коммиттер

**git show <первые 4 символа id коммита>** - детали коммита, если без id – то текущего коммита

# Подключение к удаленному серверу и git репозиторию

**git remote add origin** [**https://github.com/Qexly/<repos**](https://github.com/Qexly/%3crepos)**> -** Подключение к github

**git push -u origin master –** залить из локального репо на гитхаб

**git clone** [**https://github.com/Qexly/test2.git**](https://github.com/Qexly/test2.git) - получить репо из гитхаба на ПК

**git pull** – обновит наш локальный репо с облака github

# Отмена изменений и возврат к другому коммиту

<https://ru.stackoverflow.com/questions/431520/%D0%9A%D0%B0%D0%BA-%D0%B2%D0%B5%D1%80%D0%BD%D1%83%D1%82%D1%8C%D1%81%D1%8F-%D0%BE%D1%82%D0%BA%D0%B0%D1%82%D0%B8%D1%82%D1%8C%D1%81%D1%8F-%D0%BA-%D0%B1%D0%BE%D0%BB%D0%B5%D0%B5-%D1%80%D0%B0%D0%BD%D0%BD%D0%B5%D0%BC%D1%83-%D0%BA%D0%BE%D0%BC%D0%BC%D0%B8%D1%82%D1%83>

**git log** – просмотреть коммиты

**git checkout -- <file> -** Отменить изменения, которые пока не внесены в индекс

**git reset <file>** - удалить последние изменения из индекса, а потом можно и **checkout**

**git reset --hard HEAD^1** – удалить ласт коммит и удалить его изменения из файла и индекса

**git reset --soft HEAD^1 –** удалить ласт коммит, но изменения из индекса не убирать и из файла тоже

# Ветки

**git branch** – посмотреть все существующие ветки (\* - текущая)

**git branch –v –** показывает ветки плюс последний коммит

**git branch <name> -** создать ветку

**git checkout <branchName>** - перейти в ветку. Когда переходим в другую ветку, но открытые файлы принимают состояние последнего (HEAD) комита в этой ветке

**git branch –m <name>** - переименовать текущую ветку

**git push --set-upstream origin <dev> -** добавить ветку в облако github чтобы делать push