Git – basic commands

Installation

- Linux

$ apt-get install git

- Windows

➢Git for Windows https://gitforwindows.org/

➢GitHub desktop https://desktop.github.com/

Starting configuration

- Name and email

$ git config --global user.name "Your Name "

$ git config --global user.email your_email@whatever.com

Create local repository

1. Create folder:

```
$ mkdir hello
$ cd hello
```

2. Create repository in this folder:

```
$ git init
Initialized empty Git repository in /hello/.git/
```

3. Add testing file:

```
$ git add hello.html
$ git commit -m "First Commit"
[master (root-commit) 911e8c9] First Commit
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 hello.html
```

Check status / local changes

Without any changes

$ git status

On branch master

nothing to commit (working directory clean)

File hello.html changed

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

#

modified: hello.html

#

no changes added to commit (use "git add" and/or "git commit -a")

Commit changes

Index changes

$ git add hello.html

$ git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

#

modified: hello.html

#

Or add full catalogue

$ git add .

Commit changes

$ git commit

|

Please enter the commit message for your changes. Lines starting

with '#' will be ignored, and an empty message aborts the commit.

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

#

modified: hello.html

#

Project history and versions hashes

History:

$ git log

History with hashes:

$ git config --global alias.hist "log --pretty=format:'%h %ad | %s%d [%an]' --graph --date=short"

$ git hist

* fa3c141 2011-03-09 | Added HTML header (HEAD, master) [Alexander Shvets]

* 8c32287 2011-03-09 | Added standard HTML page tags [Alexander Shvets]

* 43628f7 2011-03-09 | Added h1 tag [Alexander Shvets]

* 911e8c9 2011-03-09 | First Commit [Alexander Shvets]

Return to previous version

$ git checkout 911e8c9

Note: checking out '911e8c9'.

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -b with the checkout command again. Example:

```
git checkout -b new_branch_name
```

HEAD is now at 911e8c9... First Commit

$ cat hello.html

Hello, World

Cancelling changes

- Without commit:

$ git add hello.html

$ git reset HEAD hello.html
$ git checkout hello.html

- After commit:

$ git commit -m "Oops, we didn't want this commit“
$ git revert HEAD --no-edit

Moving files

```
$ mkdir lib
$ git mv hello.html lib
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   renamed:    hello.html -> lib/hello.html
#


$ git commit -m "Moved hello.html to lib"
```

Branch creation and navigation

- Use branches for large modifications of code without changing original version.

Creation of branch “style”

1) $ git checkout -b style = 2) $ git branch style
$ git checkout style

Навигация и переключение

$ git checkout master

Состояние всех веток

$ git hist –all

Слияние (если нет конфликтов)

git checkout style

git merge master

Несколько репозиториев

1. Клонирование

$ git clone hello cloned_hello

2. Удаленные репозитории – просмотр веток

$ git remote show origin

* remote origin

  Fetch URL: /Users/alex/Documents/Presentations/githowto/auto/hello

  Push URL: /Users/alex/Documents/Presentations/githowto/auto/hello

  HEAD branch (remote HEAD is ambiguous, may be one of the following):

    style

    master

  Remote branches:

    style tracked

    master tracked

  Local branch configured for 'git pull': master merges with remote master

  Local ref configured for 'git push': master pushes to master (up to date)

Синхронизация с удаленным репозиторием

Извлечение новых коммитов из оригинального репозитория

$ git fetch

$ git merge origin/master

Updating 6e6c76a..2faa4ea

Fast-forward

README | 1 +

1 files changed, 1 insertions(+), 0 deletions(-)

$ git pull

Добавление изменений в удаленный репозиторий

1) Коммит локального изменения

$ git checkout master

$ git add README

$ git commit -m "Added shared comment to readme«

2) Отправка изменений

$ git push shared master

Tutorials about git and github desktop

- GitHowTo https://githowto.com/ru

- An Intro to Git and GitHub for Beginners

https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners

- https://help.github.com/en/desktop/getting-started-with-github-desktop