Git

- toc
  - Local
  - Remote

General

- node (commit) specifier
  - HEAD
  - HEAD~2, HEAD^^^, HEAD@{2}, . . .
  - commit id (node hash)
  - branch name (indicator of last node of the branch)

Local

- git does not see `.gitignore` specified pathes
- changing levels
  - ignored
  - working tree
    * untracked
    * tracked
  - staged

Changes

- having 2 file in 2 branch in same time with easy navigation and easy changing each!
  - for this you can duplicate your git project with all git informations, one of this replicas is for navigation between different git nodes and watching them and another is for work that you currently on.
- we want to move our current change from current branch to another branch!
  - first save your changes into the stash stack:

    ```
    git stash save
    ```
  - go to the branch you want:

    ```
    git checkout wantedbranch
    ```
  - pop your changes from stash stack:

    ```
    git stash pop
    ```
  - !!!! remember to pop your stash when you back to branch :/ !!!!
- see the stashes list:

`git stash list`

- how to stash just single file

  `git stash -- filename`

    if file is untracted first do to trackted :)

- removing current changes

  `git restore path`

- removing current changes added to stage

  `git restore --staged path`

- removing untrackted files

  `git clean -fd`

  -f means files, -d means directories

- remove files from git to add them to `.gitignore`

  `git rm --cached path`

- the only place that you can see the dangling commits!!!

  `git reflog`

Branches

- when you create a branch, git make copies of the node that you are now on it.
- create new branch

  `git branch branchname`

- create new branch and switch on it

  `git checkout -b branchname`

- checkout to a commit

  `git checkout commitspecifier`

  when you checkout to a commit you are coding to a dangling branch!!!
  Detached HEAD is a dangling branch

- how to delete a branch:

  `git branch -d branchname`

  `git branch -D branchname`

    `-D` is equal to `-fd`!

Remote

always try to push fast if you add or modify a `.gitignore` file!!!

- see git current server remotes

  ```
  git remote -v
  ```

- change remote

  ```
  git remote set-url origin new.git.url/here
  ```

- github remote format (ssh format)

  ```
  git@github:username/reponame.git
  ```

- remove a git branch from remote

  ```
  git push origin -d branchname
  ```

Conflict

- solve binary file conflict

  ```
  git checkout --theirs -- path/to/conflicted-file.txt
  ```

  ```
  git checkout --ours -- path/to/conflicted-file.txt
  ```

- push conflict (**DANGEROUS**)

  ```
  git push --force-with-lease
  ```

  – you should not use this command

    ```
    git push --force
    ```