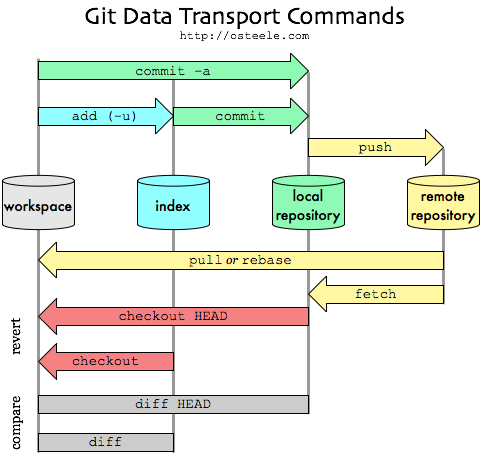


team city diff screen

|  |  |
| --- | --- |
| old version | new version |

|  |
| --- |
| **master** default name of main branch when creating a new repository  **origin** default name of remote repository  **HEAD** special pointer to local branch currently selected  FETCH\_HEAD short-lived ref  a branch is movable pointer to one commit for local branches;  is an un-movable pointer for remote branches (origin/master) on local (see fetch)  tracking branch local branche with a remote counterpart , git pull and git push are using this info when called without argument to determine remote counterpart to act upon  git branch --track  refs references, either heads, tags or repositories  fast forward merge a commit that is directly upstream (direct common ancestry), so there is no divergent work:  git checkout –b hotfix-80    …work…  git commit -a -m “fixed …”  git checkout master  git merge hotfix-80  rebasing timeline gets polluted with unnecessary micro-merges, rebasing ensures that the commits are always re-applied so that the history stays linear; |
|  |



# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_useful usage\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_git help commit\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| $ git branch list branches and active branch (HEAD) $ git branch -vlist last commit on each branch$ git diff --word-diffhighlight word changes, not entire lines$ git status -sbshort status | $ git checkout -t origin/feature **$ git push**  track a remote branch from someone else  **$ git name-rev 0c6a39d43**  find label for commit  **$ git branch --contains 0c6a39d43**  find branch for commit  **$ git show :/^.\*comm --name-only**  find commit by regexp |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_init\_\_\_\_\_\_\_\_\_\_\_\_\_repository\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# via git bash:

# $ git init

**$ git config --global user.name "al42"**

**$ git config --global core.editor sublime\_text**

**$ git status**

# --global flag instructs git to save settings in ~/.gitconfig, and not in <repository>/.git/config

To start staging files, change directory to source code folder (/WebContent and /src) and issue:

**$ git add .**

**$ git status**

Before commit to see what will be added to the repository:

**$ git diff --staged --name-only**

Commit:

**$ git commit -m 'primordial commit'**

**$ git log –-pretty=oneline –-stat** (--graph --decorate)

Get a certain version in the working directory:

**$ git log 0c6a39d --name-only**

**$ git checkout 0c6a39d --name-only**

**$ git checkout 0c6a39d index.jsp**

Working with a remote repository:

**$ git remote add origin** [**git@github.com:testuser/project.git**](mailto:git@github.com:testuser/project.git)

**$ git push origin master**

# \_\_\_\_\_\_\_init\_\_\_\_from\_\_\_\_repository\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**git clone** [**http://al042@sourcescontrol.lu.europe.intranet:9987/gitblit/r/channels/project/ing-channels-suites.git**](http://al042@sourcescontrol.lu.europe.intranet:9987/gitblit/r/channels/project/ing-channels-suites.git) **ing-channels-suites-clone**

now ing-channels-suites-clone folder is on “master”

**git checkout -b dev origin/dev**

now ing-channels-suites-clone folder is on “dev”

Change in folder structure after VSS to GIT migration:

|  |  |  |  |
| --- | --- | --- | --- |
| VSS | wks\portal2\ | portal2-webapp\ | src\lu\crediteurop\portail\credidem |
| GIT | sbx\ing-channels-suites\ing-portal-root\ing-portal-core-root\ | ing-portal-webapp\ | src\main\java\lu\crediteurop\portail\credidem |

# \_\_\_\_\_\_\_\_\_\_123\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**git checkout -b bugfix-53**

create a new branch and switch to it

1) **git** **fetch** alias of **git fetch origin**

HEAD master = HEAD origin/master, update locally state of remote repository;

**git log master origin/master --graph --shortstat**

2) …local work…

**git** **commit-m**

HEAD master advances <> HEAD origin/master (still local:/origin/master)

3) update local branch from remote

4) merge local version with pre-fetched remote version

# git merge current branch

# \_\_\_\_\_\_\_456\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**git add**

Is unique to git, in the sense that it does not add files to repository, but it creates the stage version of a file. Further using commit will add staged content to the repository, logical buckets of commit can be applied;

**git log** alias of **git log HEAD**

Local history. Nice options: --shortstat

**git log origin/master**

Local history or local mirror for remote repository local:origin/master.

**git log FETCH\_HEAD ^master**

????

**git fetch --dry-run**

Contextual command, update LOCAL with the latest REMOTE without merging /affecting the working copy. Creates a special head reference FETCH\_HEAD that will be erased by the next merge (that aligns HEAD).

Diff directly between local and remote is not possible, local mirror of remote has to be updated (git fetch) to obtain the effect.

**$ git fetch --dry-run**

remote: Counting objects: 5145, done

remote: Finding sources: 100% (3797/3797)

remote: Getting sizes: 100% (3159/3159)

remote: Compressing objects: 99% (3646562/3647048)

remote: Total 3797 (delta 1756), reused 3687 (delta 1737)

Receiving objects: 100% (3797/3797), 1.20 MiB | 0 bytes/s, done.

Resolving deltas: 100% (1803/1803), completed with 453 local objects.

**git fetch** alias of  **git fetch origin**

Contextual command, updates local mirror of remote repository, dosn’t merge with local changes. Updates origin/branchx pointer with remote state, origin/branchx branch (pointer) is not modifiable locally => to create a workable local copy:

**git checkout –b branchx origin/branchx**

**git remote update** alias of  **git fetch** for all remotes

**git pull** alias of  **git fetch** + **git merge**

**git push**

git push will by default push all branches that have the same name on the remote;

**git branch –vv**

Contextual command, lists local branches (and HEAD sha1) with remote counterpart in [].

Current branch is market with \*.

**$ git branch --no-color -vv**

**6968-clog 2ea46ee [origin/6968-clog] Bug fixing Tranche 6968 - CLOG**

**\* dev fe11e22 [origin/dev] Merge remote-tracking branch 'origin/dev' into dev**

**master 9771b51 [origin/master] Update version number to 1.0-SNAPSHOT**

**git branch –avv**

Contextual command, lists all branches local +remote (and HEAD sha1) , remote counterpart for local branches in [].

**git branch –rvv**

Contextual command, lists all branches remote (and HEAD sha1).

**git branch –d bugfix-53**

delete branch (that is ancestor of HEAD)

**git branch –D bugfix-53**

delete branch forceful and discard changes

**git checkout <branch>**

Change current branch in working dir to <branch>: HEAD pointer is now associated with the newly selected branch, working directory is updated with the snapshot of the selected branch (files are added/erased/modified as needed unless uncommitted changes detected);

# \_\_\_\_\_\_\_file\_\_\_\_\_structure\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

git sub-folders (.git/)

|  |  |
| --- | --- |
| refs/heads | refs/remotes/<remote>/ |
| local | remote-tracking |

# \_\_\_\_\_command\_\_\_\_line\_\_\_\_\_parameters\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Format**

%x09 TAB

%h abbreviated commit hash

%an author name

%ae author email

%ar author date, relative

%ai author date, ISO 8601 format

%cd committer date

%s subject

# \_\_\_\_\_configuration\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

how-to ‘setup github.com/mardare/rishiri.git’ on laptop@work

git config --global user.name "mrd@w"

git config --global user.email "n/a"

git config --global push.default simple

git config --global http.proxy http://so20pe:mass123@192.168.5.116:8080

set GIT\_SSL\_NO\_VERIFY=true

git clone http://github.com/mardare/rishiri.git

[…]

git pull

[…]

git checkout

git add . (or git commit –am “forrest/dungeons …”)

git commit -m "forrest /dungeonss"

git log --pretty=oneline –stat

git push

# \_\_\_\_conventions\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# \_\_\_\_documentation/pres\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

https://www.kernel.org/pub/software/scm/git/docs/git.html

https://sandofsky.com/images/fast\_forward.pdf

https://www.atlassian.com/git/

Scott Chacon (ProGit) Presentation <https://www.youtube.com/watch?v=ZDR433b0HJY>

<http://rogerdudler.github.io/git-guide/files/git_cheat_sheet.pdf>

<http://rogerdudler.github.io/git-guide/>

Enhanced Behavior Scripts (<https://github.com/josephdpurcell/git-extras>)

**hub** is a command-line wrapper for git that makes you *better at GitHub (*<http://hub.github.com/>**)**

manpage log https://www.kernel.org/pub/software/scm/git/docs/git-log.html