Git Bash Repository/Local Repository

A free open source development environment for creating Microsoft Windows applications:

- **All “git” commands start with “git” unless indicated**
- **The $ indicates the line you are on, it appears automatically**
- **User@Computer is the name of the computer; yours will be different than the example.**
- **MINGW32 or MINGW64, determines if you are using a 32 or 64 bit computer.**
- **Master is the branch of GitHub you are on.**
- **Our resource is the YouTube video: How to Get Started With GitHub – Beginner Tutorial by MicrowaveSam**

GIT COMMANDS

```
$ git help -a
usage: git [--version] [--help] [-C <path>] [-c name=value]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--
info-path]
           [-p | --paginate | --no-pager] [--no-replace-objects]
[--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--
namespace=<name>]
           <command> [<args>]

available git commands in 'C:\Program
Files\Git\mingw32/libexec/git-core'

  add                 fsck-objects        rebase
  add--interactive    gc                  receive-pack
  am                  get-tar-commit-id   reflog
  annotate            grep                relink
  apply               gui                 remote
  archimport          gui--askpass        remote-ext
  archive             gui--askyesno       remote-fd
  bisect              gui.tcl             remote-ftp
  bisect--helper      hash-object         remote-ftps
  blame               help                remote-http
  branch              http-backend        remote-https
  bundle              http-fetch          repack
  cat-file            http-push           replace
  check-attr          imap-send           request-pull
  check-ignore        index-pack          rerere
  check-mailmap       init                reset
  check-ref-format    init-db             rev-list
  checkout            instaweb            rev-parse
  checkout-index      interpret-trailers  revert
  cherry              log                 rm
  cherry-pick         ls-files            send-email
```

```
citool              ls-remote           send-pack
clean               ls-tree             sh-i18n--envsubst
clone               mailinfo            shortlog
column              mailsplit           show
commit              merge               show-branch
commit-tree         merge-base          show-index
config              merge-file          show-ref
count-objects       merge-index         stage
credential          merge-octopus       stash
credential-manager  merge-one-file      status
credential-store    merge-ours          stripspace
credential-wincred  merge-recursive     submodule
cvsexportcommit     merge-resolve       submodule--helper
cvsimport           merge-subtree       subtree
cvsserver           merge-tree          svn
daemon              mergetool           symbolic-ref
describe            mktag               tag
diff                mktree              unpack-file
diff-files          mv                  unpack-objects
diff-index          name-rev            update-index
diff-tree           notes               update-ref
difftool            p4                  update-server-info
difftool--helper    pack-objects        upload-archive
fast-export         pack-redundant      upload-pack
fast-import         pack-refs           var
fetch               patch-id            verify-commit
fetch-pack          prune               verify-pack
filter-branch       prune-packed        verify-tag
fmt-merge-msg       pull                web--browse
for-each-ref        push                whatchanged
format-patch        quiltimport         worktree
fsck                read-tree           write-tree
```

Major GIT Commands With Examples

Here you will find a list with the major commands, their short descriptions and exemplary usage. For a detailed description of all the GIT commands please visit

http://www.kernel.org/pub/software/scm/git/docs/

Major GIT commands:

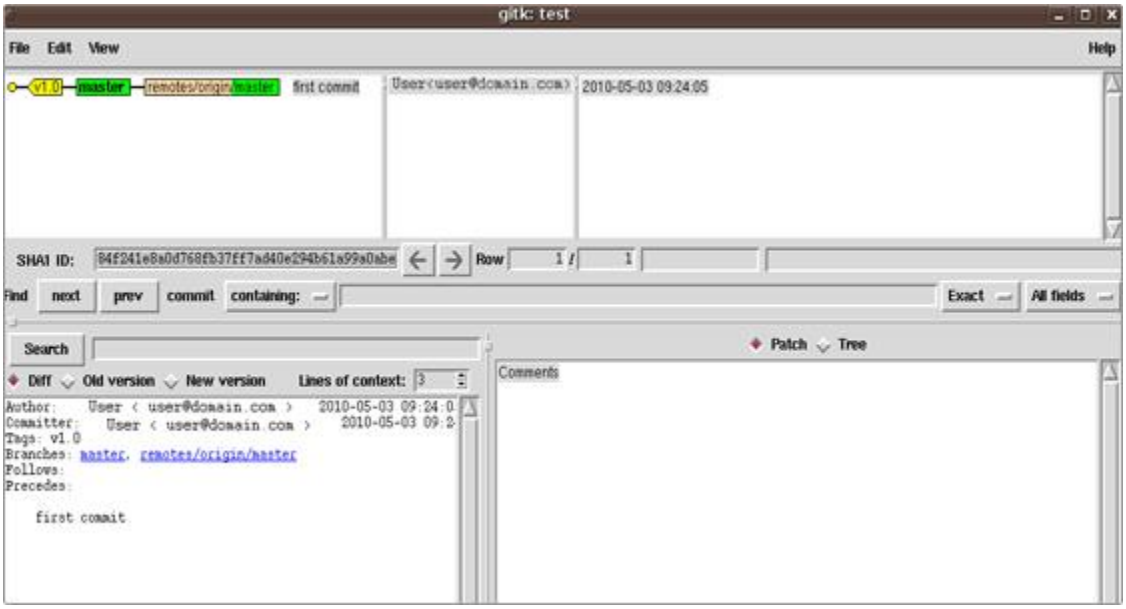
- git config
  Sets configuration values for your user name, email, gpg key, preferred diff algorithm, file formats and more. Example: git config --global user.name "My Name" git config --global user.email "user@domain.com" cat ~/.gitconfig [user] name = My Name email = user@domain.com

- git init
  Initializes a git repository – creates the initial '.git' directory in a new or in an existing project. Example: cd /home/user/my_new_git_folder/ git init
- git clone
  Makes a Git repository copy from a remote source. Also adds the original location as a remote so you can fetch from it again and push to it if you have permissions. Example: git clone git@github.com:user/test.git
- git add
  Adds files changes in your working directory to your index. Example: git add .
- git rm
  Removes files from your index and your working directory so they will not be tracked. Example: git rm filename
- git commit
  Takes all of the changes written in the index, creates a new commit object pointing to it and sets the branch to point to that new commit. Examples: git commit -m 'committing added changes' git commit -a -m 'committing all changes, equals to git add and git commit'
- git status
  Shows you the status of files in the index versus the working directory. It will list out files that are untracked (only in your working directory), modified (tracked but not yet updated in your index), and staged (added to your index and ready for committing). Example: git status # On branch master # # Initial commit # # Untracked files: # (use "git add <file>..." to include in what will be committed) # # README nothing added to commit but untracked files present (use "git add" to track)
- git branch
  Lists existing branches, including remote branches if '-a' is provided. Creates a new branch if a branch name is provided. Example: git branch -a * master remotes/origin/master
- git checkout
  Checks out a different branch – switches branches by updating the index, working tree, and HEAD to reflect the chosen branch. Example: git checkout newbranch
- git merge
  Merges one or more branches into your current branch and automatically creates a new commit if there are no conflicts. Example: git merge newbranchversion
- git reset

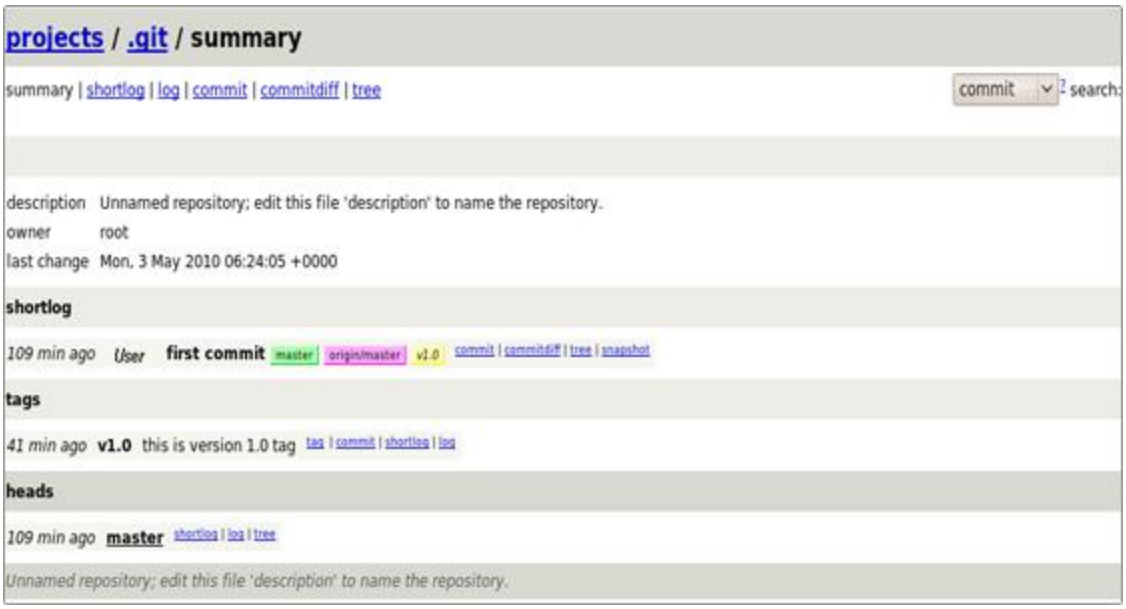
Resets your index and working directory to the state of your last commit. Example: git reset --hard HEAD

- git stash
  Temporarily saves changes that you don’t want to commit immediately. You can apply the changes later. Example: git stash Saved working directory and index state "WIP on master: 84f241e first commit" HEAD is now at 84f241e first commit (To restore them type "git stash apply")
- git tag
  Tags a specific commit with a simple, human readable handle that never moves. Example: git tag -a v1.0 -m 'this is version 1.0 tag'
- git fetch
  Fetches all the objects from the remote repository that are not present in the local one. Example: git fetch origin
- git pull
  Fetches the files from the remote repository and merges it with your local one. This command is equal to the git fetch and the git merge sequence. Example: git pull origin
- git push
  Pushes all the modified local objects to the remote repository and advances its branches. Example: git push origin master
- git remote
  Shows all the remote versions of your repository. Example: git remote origin
- git log
  Shows a listing of commits on a branch including the corresponding details. Example: git log commit 84f241e8a0d768fb37ff7ad40e294b61a99a0abe Author: User <user@domain.com> Date: Mon May 3 09:24:05 2010 +0300 first commit
- git show
  Shows information about a git object. Example: git show commit 84f241e8a0d768fb37ff7ad40e294b61a99a0abe Author: User <user@domain.com> Date: Mon May 3 09:24:05 2010 +0300 first commit diff --git a/README b/README new file mode 100644 index 0000000..e69de29
- git ls-tree
  Shows a tree object, including the mode and the name of each item and the SHA-1 value of the blob or the tree that it points to. Example: git ls-tree master^{tree} 100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391 README

- git cat-file
  Used to view the type of an object through the SHA-1 value. Example: git cat-file -t e69de29bb2d1d6434b8b29ae775ad8c2e48c5391 blob
- git grep
  Lets you search through your trees of content for words and phrases. Example: git grep "www.siteground.com" -- *.php
- git diff
  Generates patch files or statistics of differences between paths or files in your git repository, or your index or your working directory. Example: git diff
- gitk
  Graphical Tcl/Tk based interface to a local Git repository. Example: gitk



- git instaweb
  Runs a web server with an interface into your local repository and automatically directs a web browser to it. Example: git instaweb --httpd=webrick git instaweb --stop



- git archive
  Creates a tar or zip file including the contents of a single tree from your repository. Example: git archive --format=zip master^ README >file.zip
- git gc
  Garbage collector for your repository. Optimizes your repository. Should be run occasionally. Example: git gc Counting objects: 7, done. Delta compression using up to 2 threads. Compressing objects: 100% (5/5), done. Writing objects: 100% (7/7), done. Total 7 (delta 1), reused 0 (delta 0)
- git fsck
  Does an integrity check of the Git file system, identifying corrupted objects. Example: git fsck
- git prune
  Removes objects that are no longer pointed to by any object in any reachable branch. Example: git prune

Repository Commands
$ git config --global user.name "*Your GitHub User Name***"**

User@Computer MINGW64 ~
$ git config --global user.email "*Your GitHub Email Address***"**

Create a new repository in GitHub

- Add the following components
    - Repository name
    - Description
    - Public or private (costs money)
    - Initialize with a README.md (a type of about page)
    - Click create repository

You must specify a repository to clone, copy from the clipboard or the GitHub repository URL address bar. (*The green Clone or Download button has a dropdown icon in it. Click the dropdown icon and a link and a clip board will appear. Click the clip board; it will state that the link is copied. Return to the Git Bash window and continue to follow the instructions below.)*

User@Computer MINGW64 ~
$ cd Desktop

User@Computer MINGW64 ~/Desktop (Sample)
$ ls
about.html contact.html fonts/ index.html LICENSE
blog.html css/ img/ js/ README.md

User@Computer MINGW64 ~
clear
User@Computer MINGW64 ~/Desktop
$ git clone https://github.com/Nkcurlett/whats-up.git
(*Paste the copied project name with a space next to the git clone command. The example is in bold lettering with an underline.*)

cd whats-up/
(whats-up is our example repository name, ending with the forward slash (/)

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ ls

README.me

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ vim README.md
(Here you can go to the folder that's usually on the desktop and open the document with your favorite text editor)

The vim editor can only be accessed this way:

- The word insert will display at the bottom of the screen
- Tap "I" or the "ins" key to add text
- Movement is confined to the arrow keys "left, right, up and down"
- Add your text
- Press the esc or escape button to exit the insert area
- :q quits the process of the vim but does not save your entry
- :x saves and exits from the vim entry
- press enter to exit
- You can go back to the folder to see the vim entry on your text editor

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git add README.md
(Git add is the staging area that prepares the local machine to upload "push" to the repository (single file)

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git commit -m "your message for your commit**"**
(Documenting or updates your changes)

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git log
(tracking of your contributions to the GitHub site)

Example:
commit f787d72a7c8ca479d9ad60791046c2e2cbf3cda0
Author: Nkcurlett <nkcurlett@gmail.com>
Date: Mon Sep 19 10:20:21 2016 -0500

updated readme file to indicate changes
User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
(use "git branch --unset-upstream" to fixup)
nothing to commit, working tree clean

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git remote add origin (*https://github.com/your project name*.git)

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git remote add origin (*https://github.com/your project name*.git)
fatal: remote origin already exists.

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git push -u origin master

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git push -u origin master
Branch master set up to track remote branch master from orgin.
Everything up-to-date

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ touch testfile.html
(touch adds new file)

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ ls
about.html blog.html contact.html css/ fonts/ img/ index.html js/ LICENSE README.md testfile.html

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ vim testfile.html

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git status
On branch master
Your branch is up-to-date with 'orgin/master'.
Untracked files:

```
  (use "git add <file>..." to include in what will be committed)
        testfile.html
nothing added to commit but untracked files present (use "git add" to track)

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git add .
warning: LF will be replaced by CRLF in testfile.html.
The file will have its original line endings in your working directory.

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git commit -m "testfile html for sample of adding a file"
[master cce7b0a] testfile html for sample of adding a file
 1 file changed, 1 insertion(+)
 create mode 100644 testfile.html

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git log
commit cce7b0a941ca651631435f7a58ceb6be7fa8a965
Author: Nkcurlett <nkcurlett@gmail.com>
Date:   Mon Sep 19 10:51:14 2016 -0500
    testfile html for sample of adding a file
commit f787d72a7c8ca479d9ad60791046c2e2cbf3cda0
Author: Nkcurlett <nkcurlett@gmail.com>
Date:   Mon Sep 19 10:20:21 2016 -0500
    updated readme file to indicate changes

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git status
On branch master
Your branch is ahead of 'orgin/master' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working tree clean

User@Computer MINGW64 ~/Desktop/whats-up (master)
$ git push
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 309 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
```

remote: Resolving deltas: 100% (1/1), completed with 1 local objects.
To https://github.com/Nkcurlett/personal-website.github.io.git
  f787d72..cce7b0a master -> master

HOW TO CREATE A DOMAIN

1. Get your domain account from a web domain purveyor with your preferred domain name (Aabasco, Godaddy etc.)
2. Go to your repository and add a CNAME file in the gh-pages branch
    a. Create a folder named CNAME without an ending tag
    b. Place the file name in the CNAME folder
        i. nkcurlett.net
        ii. commit changes
3. Refresh the page
4. The domain name must be configured in your DNS in the A record
    a. The GitHub resolve IP addresses are:
        i. 192.30.252.153
        ii. 192.30.252.154
    b. TTL is custom\
    c. Seconds do it 600
5. It may take some time refresh the page and this will display your new website name.