

Basic Commands

coursera.org/learn/git-distributed-development/supplement/ZSuPA/basic-commands

You can see the version of git you have installed with:

```
$ git --version  
git version 1.5.5.6
```

Detailed help information in the form of a man page can be obtained about any subcommand by doing:

```
$ git help [subcommand]
```

For example, the two following statements produce the same result:

```
$ git help status  
$ man git-status
```

which furthermore points out that there are two ways to invoke particular commands, e.g. these two commands are equivalent:

```
$ git status  
$ git-status
```

The second hyphenated form is historical and is now disfavored, so use the **git subcommand** form instead.

You can get a basic list of git commands by just typing **git**, which will give you the list showed in the following screenshot:

```

File Edit View Search Terminal Help
c7:/tmp>git
usage: git [--version] [--help] [-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>]

The most commonly used git commands are:
  add           Add file contents to the index
  bisect        Find by binary search the change that introduced a bug
  branch        List, create, or delete branches
  checkout      Checkout a branch or paths to the working tree
  clone         Clone a repository into a new directory
  commit        Record changes to the repository
  diff          Show changes between commits, commit and working tree, etc
  fetch         Download objects and refs from another repository
  grep          Print lines matching a pattern
  init          Create an empty Git repository or reinitialize an existing one
  log           Show commit logs
  merge         Join two or more development histories together
  mv            Move or rename a file, a directory, or a symlink
  pull          Fetch from and merge with another repository or a local branch
  push          Update remote refs along with associated objects
  rebase        Forward-port local commits to the updated upstream head
  reset         Reset current HEAD to the specified state
  rm            Remove files from the working tree and from the index
  show          Show various types of objects
  status        Show the working tree status
  tag           Create, list, delete or verify a tag object signed with GPG

'git help -a' and 'git help -g' lists available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
c7:/tmp>

```

There are only a few global options that apply, those prefixed with -- in the above listing. Many of the subcommands have their own options, which are included in **[ARGS]** in the above.

If you cannot resist seeing the more complete set of commands, do:

```
$ git help --all
```

```

File Edit View Search Terminal Help
c7:/tmp>git help --all
usage: git [--version] [--help] [-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 '/usr/libexec/git-core'

add                  cvsexportcommit      index-pack          patch-id            send-pack
add--interactive     cvsimport             init                peek-remote         sh-i18n--envsubst
am                  cvsserver             init-db            prune               shell
annotate            daemon               instaweb           prune-packed        shortlog
apply               describe             log                pull                show
archive             diff                 lost-found         push                show-branch
bisect              diff-files           ls-files           quiltimport         show-index
bisect--helper      diff-index           ls-remote          read-tree           show-ref
blame               diff-tree            ls-tree            rebase              stage
branch              difftool             mailinfo           receive-pack        stash
bundle              difftool--helper     mailsplit          reflog              status
cat-file            fast-export          merge              relink              stripspace
check-attr          fast-import          merge-base         remote              submodule
check-ignore        fetch                merge-file         remote-ext          subtree
check-ref-format    fetch-pack           merge-index        remote-fd           svn
checkout             filter-branch        merge-octopus      remote-ftp          symbolic-ref
checkout-index       fmt-merge-msg        merge-one-file     remote-ftps         tag
cherry              for-each-ref         merge-ours         remote-http         tar-tree
cherry-pick          format-patch         merge-recursive    remote-https        unpack-file
citool              fsck                 merge-resolve      remote-testpy       unpack-objects
clean               fsck-objects         merge-subtree      remote-testsvn      update-index
clone               gc                   merge-tree         repack              update-ref
column              get-tar-commit-id    mergetool          replace             update-server-info
commit              grep                 mktag              repo-config         upload-archive
commit-tree         gui                  mktree             request-pull        upload-pack
config              gui--askpass         mv                 rerere              var
count-objects       hash-object          name-rev           reset               verify-pack
credential          help                 notes              rev-list            verify-tag
credential-cache     http-backend         p4                 rev-parse           web--browse
credential-cache--daemon http-fetch           pack-objects       revert              whatchanged
credential-gnome-keyring http-push            pack-redundant     rm                  write-tree
credential-store     imap-send           pack-refs          send-email

```

'git help -a' and 'git help -g' lists available subcommands and some concept guides. See 'git help <command>' or 'git help <concept>' to read about a specific subcommand or concept.

The long list in the screenshot above may seem rather intimidating, but some of them are really for expert usage and rarely used, or are more efficiently invoked through shorthand combinatorial commands.

Furthermore, there are several graphical interfaces to git which avoid having to be able to name all the plumbing fixtures.