Basics

Configuration

```
git config –global user.name “Jan Kowalski”

git config –global user.email kowalski@gmail.com

git config –global core.editor “code –wait”                    //Ustawienie domyślnego IDE

git config –global core.autocrlf true    (input – Mac, Linux)     //Ustawienie auto-commit’owania LF (koniec linii)
```

Initializing a repository

```
git init
```

Stage files

```
git add file1.js              //Stages a single file1

git add file1.js file2.js     //Stages multiple files

git add *.js                  //Stages all files with extension “js”

git add Dir1/*                //Stages all files in directory “Dir1”

git add .                     //Stages all the modifications in current directory
```

View the status of the repository

```
git status                    //Full status

git status -s                 //Short status
```

Committing the staged files

```
git commit -m “Message”       //Commits with a one-line message

git commit                    //Opens the default editor to type a long message
```

Skipping the staging area

```
git commit -am “Message”
```

Removing files

```
git rm file1.js          //Removes file from working directory and staging area (index)
git rm –cached file1.js  //Removes from staging area only
```

Viewing the staged/unstaged changes

```
git diff                 //Shows unstaged changes
git diff –staged         //Shows staged changes
git diff –cached         //Same above
```

Viewing the history

```
git log                  //Full history
git log –oneline         //Summary
git log –reverse         //Lists the commits from the oldest to the newest
```

Viewing a commit

```
git show 92a2ff          //Shows the given  commit
git show HEAD            //Shows the last commit
git show HEAD~2          //Two steps before the last commit
git show HEAD:file.js
```

Unstaging files (undoing git add)

```
git restore –staged file.js
```

Discarding local changes

```
git restore file.js
git restore file1.js file2.js
git restore .
git clean -fd
```

Restoring an earlier version of a file

```
git restore –source=head~2 file2.js
```

Viewing the history

```
git log –stat

git log –patch
```

Filtering the history

```
git log -3

git log –author=”mosh”

git log –before=”2020-01-01”

git log –after=”one week ago”

git log –grep=”gui”

git log -s”gui”

git log hash1..hash2

git log file.txt
```

Viewing a commit

```
git show head~2

git show head~2:file1.txt
```

Comparing commits

```
git diff head~3 head

git diff head~3 head file.txt
```

Checkout out a commit

```
git checkout dad47ed

git checkout master
```

Finding a bad commit

```
git bisect start

git bisect bad

git bisect good ca49180

git bisect reset
```

Finding contributors

```
Git shortlog
```

Viewing the history of a file

```
git log file.txt

git log –stat file.txt

git log –path file.txt
```

Finding the author of lines

```
git blame file.txt
```

Tagging

```
git tag v1.0

git tag v1.0 5e7a828

git tag

git tag -d v1.0
```

Managing branches

```
git branch bugfix

git checkout bugfix

git switch bugfix

bug switch -c bugfix
```

```
Git branch -d bugfix
```

Comparing branches

```
git log master..bugfix

git diff master..bugfix
```

Stashing

```
git stash push -m “changed login”

git stash list

git stash show stash@{1}

git stash show 1

git stash apply 1

git stash drop 1

git stash clear
```

Git merging

```
git merge bugfix

git merge –no-ff bugfix

git merge –squash bugfix

git merge –abort
```

Viewing the merged branches

```
git branch –merged

git branch –no-merged
```

Rebasing

```
git rebase master
```

Cloning a repository

```
git clone url
```

Syncing with remotes

```
git fetch origin master

git fetch origin

git fetch

git pull

git push origin master

git push
```

Sharing tags

```
git push origin v1.0

git push origin -delete v1.0
```

Sharing branches

```
git branch -r

git branch -vv

git push -u origin bugfix

git push -d origin bugfix
```

Managing remotes

```
git remote

git remote add upstream url

git remote rm upstream
```