Git Cheatsheet

```
git pull
```

Retrieves up-to-date files from the **remote** repository. Should be the first command launched every time you start working on anything.

```
git add <file1> [<file2> ...]
```

Adds the specified files to the **local** repository.

***Note*: it is possible to specify also folders, not necessarily single files**

```
git rm <file1> [<file2> ...]
```

Removes the specified files from the **local** repository.

***Note*: as for the *add* command, folders may be specified instead of files**

```
git commit -m "Message for the commit"
```

Commits all the modified files to the **local** repository.

```
git push <remote_name> <branch_name>
```

Pushes all the pending commits from the **local** repository to the **remote** repository.

Usually, <remote_name> is *origin* and <branch_name> is *master*.

```
git status
```

Prints the status of the **local** repository, telling which files are modified, added or removed. Also tells which of these modifications are already committed.

```
git branch
```

Prints the list of all the existing branches in the **local** repository.

```
git branch <branch_name>
```

Creates a new branch with the specified name in the **local** repository.

```
git branch -d <branch_name>
```

Deletes the specified branch from the **local** repository.

```
git checkout <branch_name>
```

Switches the context into the specified branch in the **local** repository.

```
git merge <branch_name>
```

Merges the modifications from the specified branch into the current branch.

```
git remote add <remote_name> <remote_repository_url>
```

Links the **local** repository with the **remote** repository located at the specified url.

```
git stash save
```

Saves the modified files locally, and cleans the **local** repository to have no pending modifications.

```
git stash pop
```

Retrieves modifications from the last stash saved.

```
git stash list
```

Prints the list of all the existing stashes saved.

```
git stash drop <stash_ID>
```

Deletes the specified stash.

```
git revert <commit>
```

Reverts the **local** repository to the specified commit. To revert only the last commit, use *HEAD* as < commit>. In order to revert the **remote** repository, simply push the modifications.

***Note*: this command deletes the modifications of all the files**

```
git reset <commit>
```

Similar to *revert* command, but it takes the modifications to the files.

```
git diff
```

Shows the differences between unstaged files and the last version available.

```
git diff -staged
```

Shows the differences between staged files and the last version available.

```
git diff <branch1> <branch2>
```

Shows differences between files belonging to the specified branches.

Git configuration commands

```
git config --global user.email <e-mail_address>
```

Saves the e-mail address used for commits.

```
git config --global user.name <username>
```

Saves the username used for commits.

```
git config --global credential.helper store
```

After the first operation on the **remote** repository, credentials are cached and never asked again.