The ultimate
Git Cheatsheet

Cloud is for everyone.

CCE

DOWNLOADING & INITIALIZATION

Initialize an existing directory as Git repository

```
$ git init
```

Clone a repository that already exists on Github, including all of the files, branches and commits

```
$ git clone [repo_url]
```

After using the git init command, link the local repository to an emty GitHub repository

```
$ git remote add origin [url]
```

Cloud is for everyone.

CCE

SETUP CREDENTIALS

Set a username

```
$ git config --global user.name
“[firstname lastname]“
```

Set an email address

```
$ git config --global user.email
“[email address]“
```

BRANCHES

List your branches & check active branch

```
$ git branch
```

Create a new branch

```
$ git branch [branch-name]
```

Create a new branch & switch to that branch (one command only)

```
$ git checkout -b [branch-name]
```

Delete specific branch

```
$ git branch -d [branch-name]
```

Cloud is for everyone.

STAGE & SNAPSHOT

Show modified files in working directory, staged for your next commit

```
$ git status
```

Add a file as it looks now to you next commit (stage)

```
$ git add [file-name]
```

Add all changed files to staging area

```
$ git add .
```

Cloud is for everyone.

Commit your staged content as new commit snapshot

```
$ git commit -m “[message]“
```

Fetch and merge any commits from the tracking remote branch

```
$ git pull origin <branch-name>
```

Transmit local branch commits to the remote repository branch

```
$ git push origin <branch-name>
```

Cloud is for everyone.

Synchronize your local repository with the remote repository

```
$ git fetch
```

MERGING

Merge a remote branch into your current branch to bring it up to date

```
$ git merge <branch-name>
```

REDO COMMITS

Reset staging area to match most recent commit but leave the working directory unchanged

```
$ git reset
```

Reset staging area and working directory to match most recent commit and overwirte all changes in the working directory

```
$ git reset --hard
```

Create a new commit, reverting changes from the specified commit.

```
$ git revert <commit>
```

Cloud is for everyone.

TEMPORARY COMMITS

Put current changes in your working directory into stash for later use

```
$ git stash
```

Apply stored stash content into working directory, and clear stash

```
$ git stash pop
```

Delete a specific stash from all your previous stashes

```
$ git stash drop
```

Cloud is for everyone.

CCE

FOLLOW FOR MORE

linkedin.com/in/tobias-schuemann/

instagram.com/tobias.schuemann/

Cloud is for everyone.