GIT CHEAT SHEET

SET UP & CONFIGURE

Show config
```
$ git config --list
```

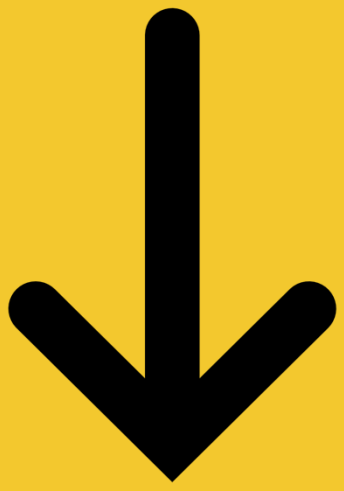
Set your git username
```
$ git config --global user.name "Anders"
```

Set your git email
```
$ git config --global user.email "anders@domain.com"
```

Set a global .gitignore
```
$ git config --global core.excludesfile "C:/path/.gitignore_global"
```

Set a editor for commit messages
```
$ git config --global core.editor "c:/windows/system32/notepad.exe"
```

MORE COMMANDS ON NEXT PAGE



GIT CHEAT SHEET

OVERVIEW

```
$ git init - Create a new local git repo
$ git status - View the changes to your project code
$ gid add - Add files to staging area
$ git commit - Creates a new commit with files from staging area
$ git log - View recent commits
```

CREATE

Clone an existing repository
```
$ git clone ssh://user@domain.com/repo.git
```

Crete a new local repository
```
$ git init
```

SET UP REMOTE REPOSITORY

See registered remotes – Shows two remotes - **push** and **pull**
```
$ git remote -v
```

Set remote repository and push to remote
```
$ git remote set-url origin https://0.0.0.0/ymi/myProject.git
$ git push -u origin master
```

SHOW LOG – LAST 10 COMMITS

```
$ git log -10 --oneline
```

BRANCHES

Show all branches and active branch
```
$ git branch -av
```

Switch branch
```
$ git checkout myBranchName
```

Create new local branch - and switch to that branch
```
$ git checkout -b myNewBranchName
```

Fetch remote branch - and switch to that branch
```
$ git fetch origin
$ git checkout --track origin/remoteBranchName
```

UPDATE & PUBLISH

Download and merge changes to active branch
```
$ git pull
```

Publish changes to Remote repository (GitLab / GitHub / BitBucket, etc)
```
$ git push
```

MERGE

Merge branch into active branch (!!! First checkout the branch to be merged **INTO** !!!)
```
$ git merge myOtherBranch
```

UNDO

Undo changes to a single file
```
$ git checkout -- nameOfFileToUndo.js
```

Add a file to a commit directly after running "git commit"
```
$ git add forgotten_file.js
$ git commit --amend
```
(Ctrl+c, wq, Enter)

```
$ git reset --<soft/hard> [id]
```