

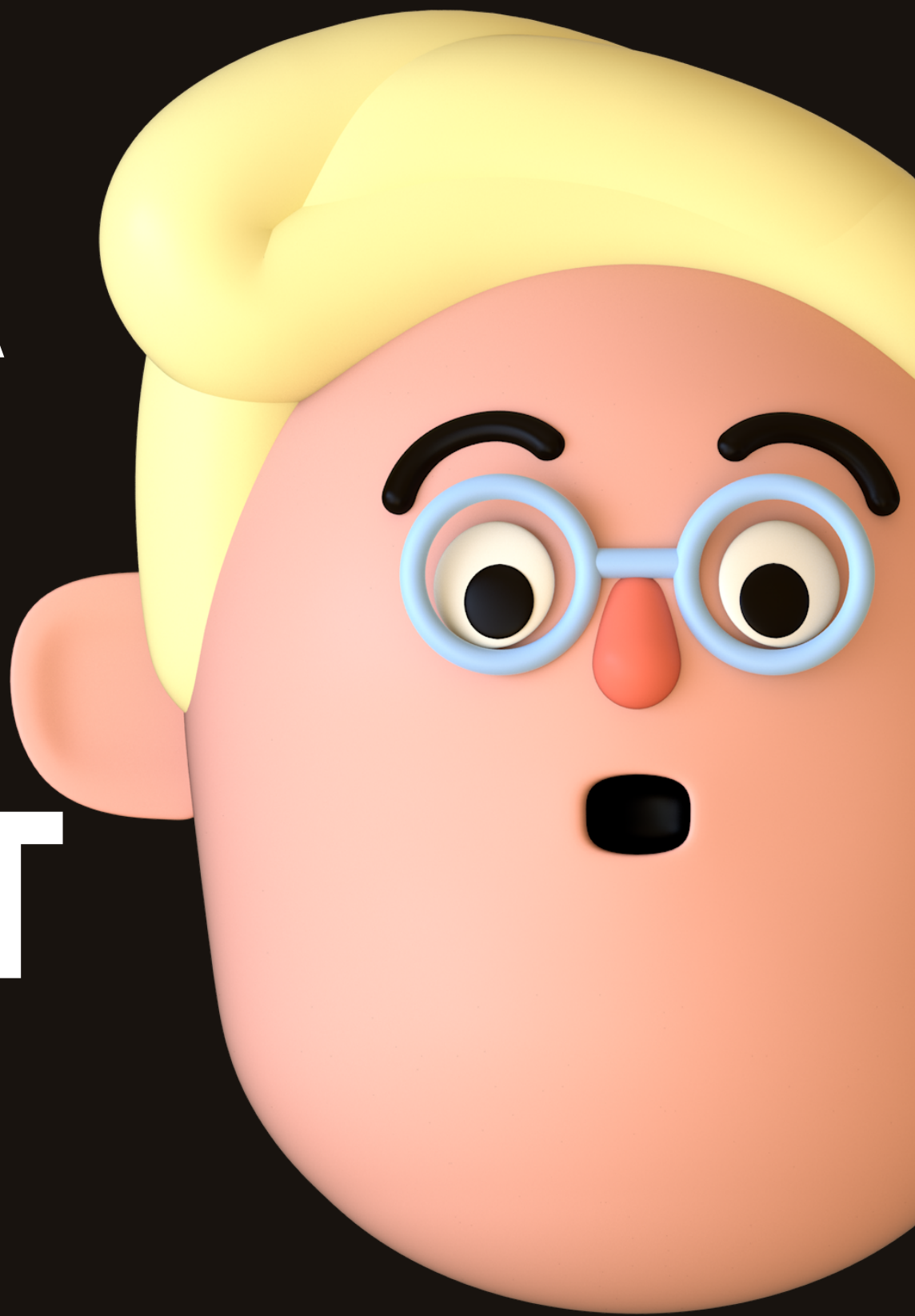
MEGA



git

CHEAT

SHEET



1.SETUP & INIT

**git config-global username [Firstname
Lastname]**

>> set name that is identifiable for credit when reviewing version history

**git config-global useremail
[vallid email]**

>> set an email address that will be associated with each history maker

git config-global color UI auto

>> set automatic command line for git for easy writing

git init

>> initialize an existing repository as a git repository



git clone [URL]

>> retrieve an entire repository from a hosted location via URL

2. STAGES & SNAPSHOTS

git branch

>> list your branch. a will appear next to the currently active branch

git branch [branch-name]

>> create a new branch at current commit

git checkout

>> switch new another branch and check it out

git merge

>> merge the specified branches history into the current one

git diff-staged

>> show modified files in working directory staged for your next committed

git commit [descriptive message]

>> commit your staged content as a new commit snapshot

3. BRANCH & MERGE

git branch

>> show modified files in working directory staged for your next comment

git add [file]

>> add the file as it looks now to your next commit

git reset [file]

>> unstage a file while remaining the changes in the working directory

git diff

>> diff of what is staged but not yet committed

4. INSPECT & COMPARE

git log

>> show all commits in the current branch history

git log branch_B ... branch_A

>> show all commits that are in branch A that are not on branch B

git log --stat-M

>> show all commit logs with an indication of any path that move



5. REWRITE HISTORY

`git rebase [branch]`

>> apply any commits to the current branch ahead of specified one

`git reset-hard [commit]`

>> show all commits that are in branch A that are not on branch B

6. IGNORING PATTERNS

`logs/ notes/ pattern`

>> save file with desired patterns as git ignore with either direct string matches

`git config-global core excludes file [file]`

>> system-wide ignore pattern for all local repositories



7. TEMPORARY COMMITS

git stash

>> save modified and staged changes git

git stash list

>> list stack order of stashed file changes

git stash pop

>> write working from top of stash stack

git log --follow [life]

>> show the commits that changed file even across renames

git show [SHA]

>> show any object in git in human readable format

8. SHARE & UPDATE

git remote add [alias] [URL]

>> add git as an alias

git fetch [alias]

>> fetch down all the branches from that git remote

git merge [alias]/[branch]

>> merge a remote branch into your current branch to bring it up to date

git push [alias] [branch]

>> transmit local branch commit to the remote repository branch

git pull

>> fetch and merge any commit from the tracking remote



9. TRACKING PATH CHANGE

`git rm [file]`

>> delete the file from project and stage removal for commit

`git mv [existing path] [new path]`

>> change the existing file path and stage the move



Turn on post notifications

FOLLOW



@NamasteDevOfficial

on  Instagram

for more exciting updates!

don't feel shy spread
knowledge to everyone

