



By Nada Mostafa

Undoing things

- ▶ `rm -rf .git`
 - ▶ To remove local repo
- ▶ `Git restore new.txt`
 - To restore from stage area (ignore changes in working tree)
- ▶ `git commit --amend -m "your new msg"`
 - The above command will amend the added change to the last commit

Un staging changes

- ▶ `git restore --staged new.txt`

To unstage the changes

- ▶ `git reset HEAD new.txt`

To unstage the changes

Undoing a Commit

- ▶ `git reset HEAD~n`

To unstage the changes ...keep changes in your WT but not in the index
to show differences == `git diff ...` to restore it ===`git restore .`

- ▶ `git reset --hard HEAD~n`

To restore changes into working tree direct .. Changes won't stay in your WT
Delete the commit and the change from staging area and delete from WT

- ▶ `git reset --soft HEAD^` ===== ~1

- ▶ Delete the last commit and git back to the staging area

- ▶ `git reset --soft HEAD^^` ===== ~2

- ▶ Delete the last 2 commits and git back to the staging area

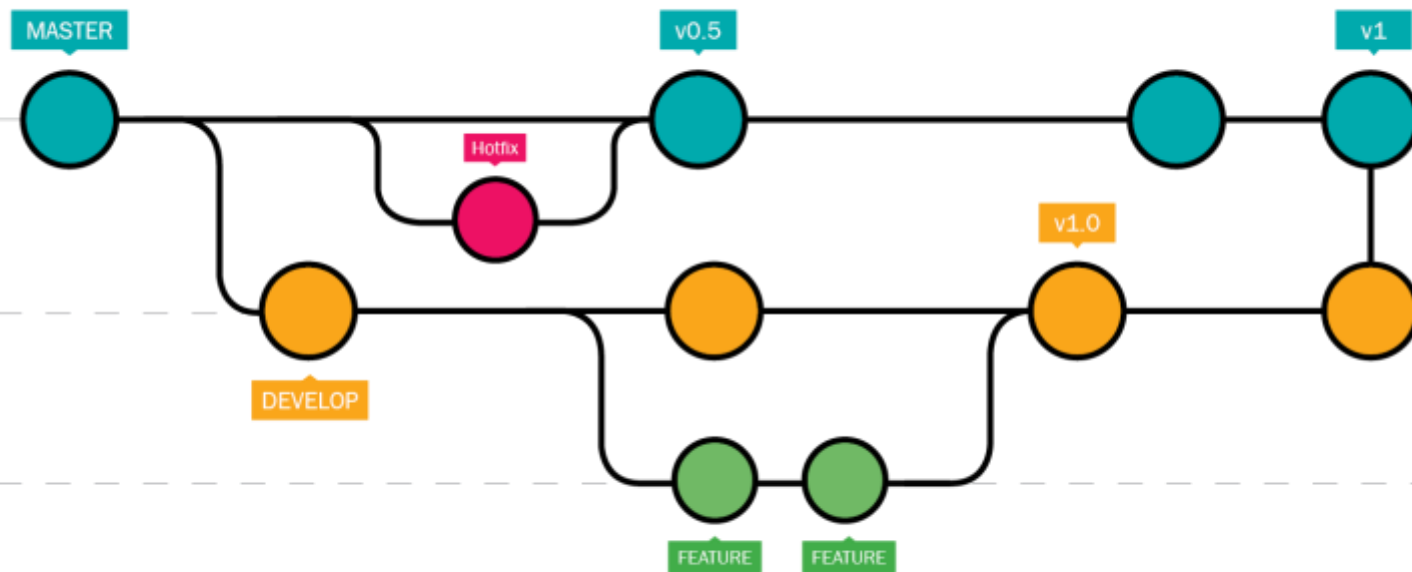
Add / Remove Remote Repo from existing project

- ▶ `git remote add origin https://github.com/Nadamostafa2/anyrepo.git`
To add a remote repository
- ▶ `git remote -v`
To list the remote repositories
- ▶ `git remote rm origin`
to remove the remote repo

Push / Pull Remote Repo

- ▶ `git push origin master`
To push the local changes to the remote repo
- ▶ `git pull origin master`
To get the changes that made by others

Branching



Branching

- ▶ `git branch new_branch_name`
 - ▶ To make a **new** branch
- ▶ `git checkout branch_name` *// switch*
 - ▶ To **switch** to a branch
- ▶ `git branch`
 - ▶ To **list** all the branches
- ▶ `git checkout -b new_branch_name`
 - ▶ To create a branch and checkout it in **one step**

Create a Remote Branch

- When you need another people to work on your branch
 - ▶ Then you have to make your branch available remotely
 - ▶ `git push origin branch_name`
- To list remote branches
 - ▶ `git branch -r`

Remove a Branch

- To delete a **remote** branch
 - ▶ `git push origin -d branch-name` / `git push origin :branch_name`
- To delete a **local** branch
 - ▶ `git branch -d branch_name` (soft delete)
 - ▶ `git branch -D branch_name` (hard delete)

Merging Branches

- ▶ After finishing your work on the branch,
 - ▶ you've to **merge** it with the Master branch.
-
- First, go to the Master branch
 - ▶ `git checkout master`
-
- Then, merge the two branches with each other
 - ▶ `git merge branch_name`

Cherry-pic

- ▶ Git cherry-pic 40.....
- ▶ To merge specific commit

Pull Request

- ▶ Pull requests let you **tell others** about changes you've pushed to a branch in a repository on GitHub.
- ▶ Once a pull request is opened, you can discuss and review the **potential changes** with collaborators and add follow-up commits before your changes are merged into the base branch.

forking



Tagging & Versioning

A tag is a reference to a commit - used mostly in release versioning.

▶ Git supports two types of tags:

- Lightweight
- Annotated.

Tags Types

- ▶ `git tag v1.0`
 - ▶ To create a **lightweight** tag
- ▶ `git tag -a v2.0 -m "version 2.0"`
 - ▶ To create an **annotated** tag

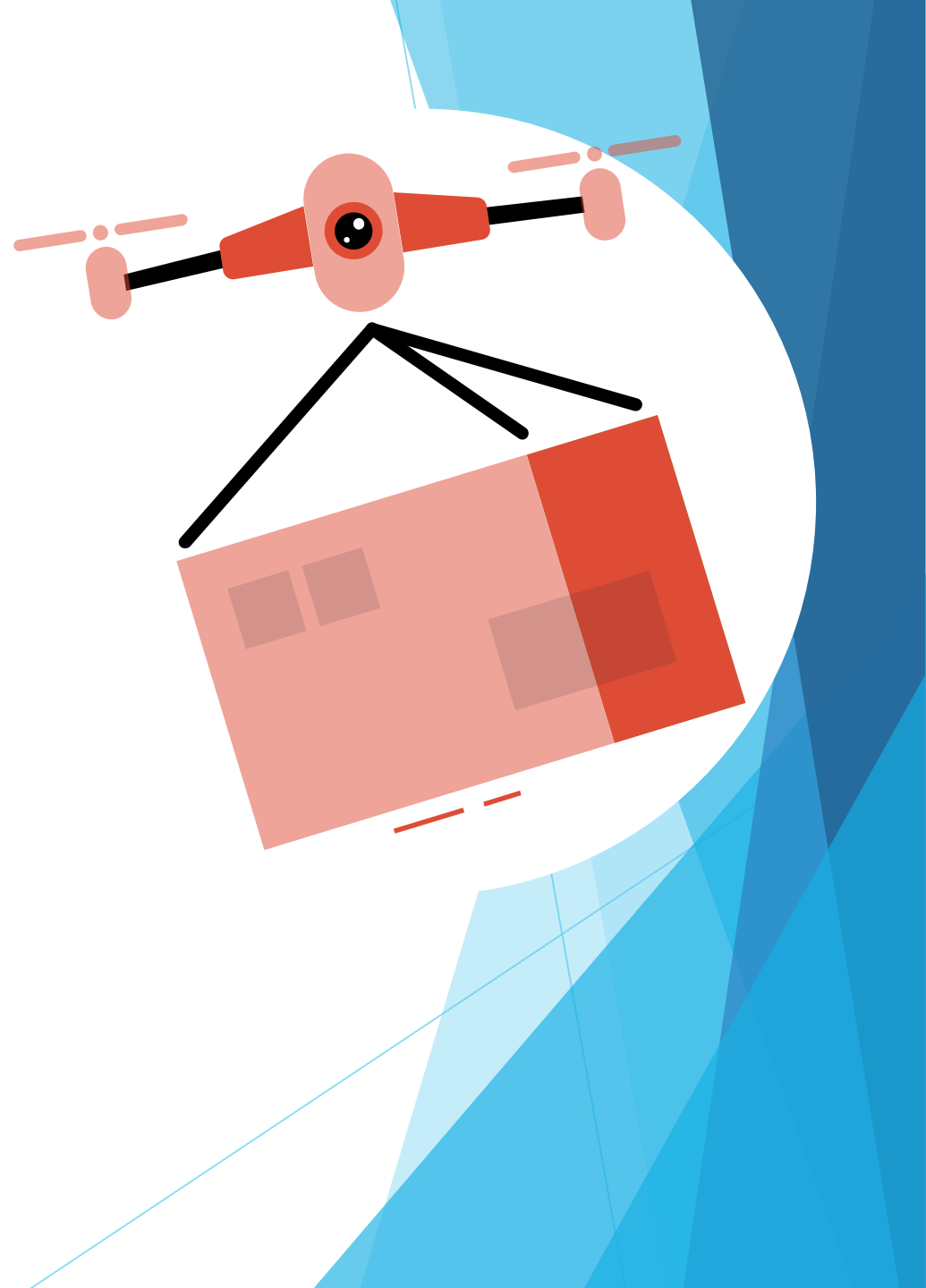
Push Tags

- To list all tags
 - ▶ `git tag`
- To push tags
 - ▶ `git push origin <tag_name>`
 - ▶ `git push --tags`

Delete Tags

- To delete remote tag
 - ▶ `git push origin --delete v1.0`
- To delete local tags
 - ▶ `git tag -d v1.0`

Ignoring Files



Ignoring Files

- Often, you will have a class of files that you don't want git to **automatically add** or even show to you as being untracked.
- In such cases you can create a file called **.gitignore** to contains all the unwanted files or directories.

→ cache/

→ logs/*.log

Lab 2

- Create a new project on your local machine, then push it your remote repo.
- Create two branches (dev & test) then create one file on each branch, and push this branches to the remote repo.
- Merge this changes on Main branch locally and then push it to your remote main branch.
- Create an annotated tag with tagname (v1.7)
- Push it to the remote repository
- Send an invitation to me “nadosatafa42127@gmail.com // nm4378586@gmail.com”

