Version Control

git

ITI - Day 2

Content

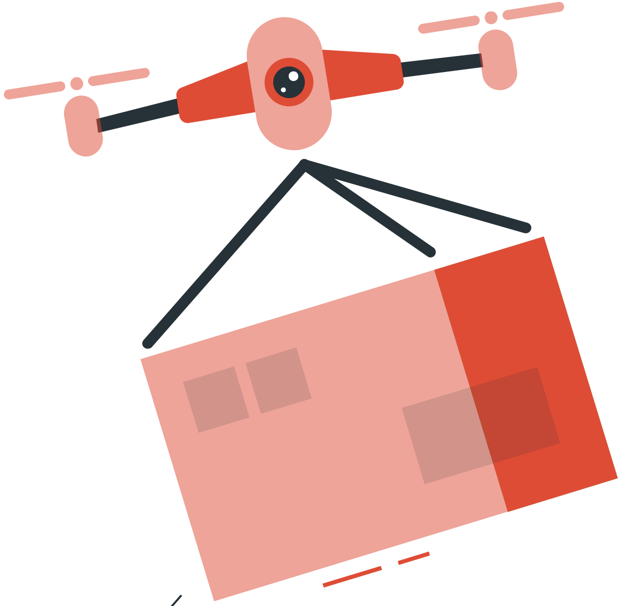
1 Branching & Rebasing

2 Pull Request

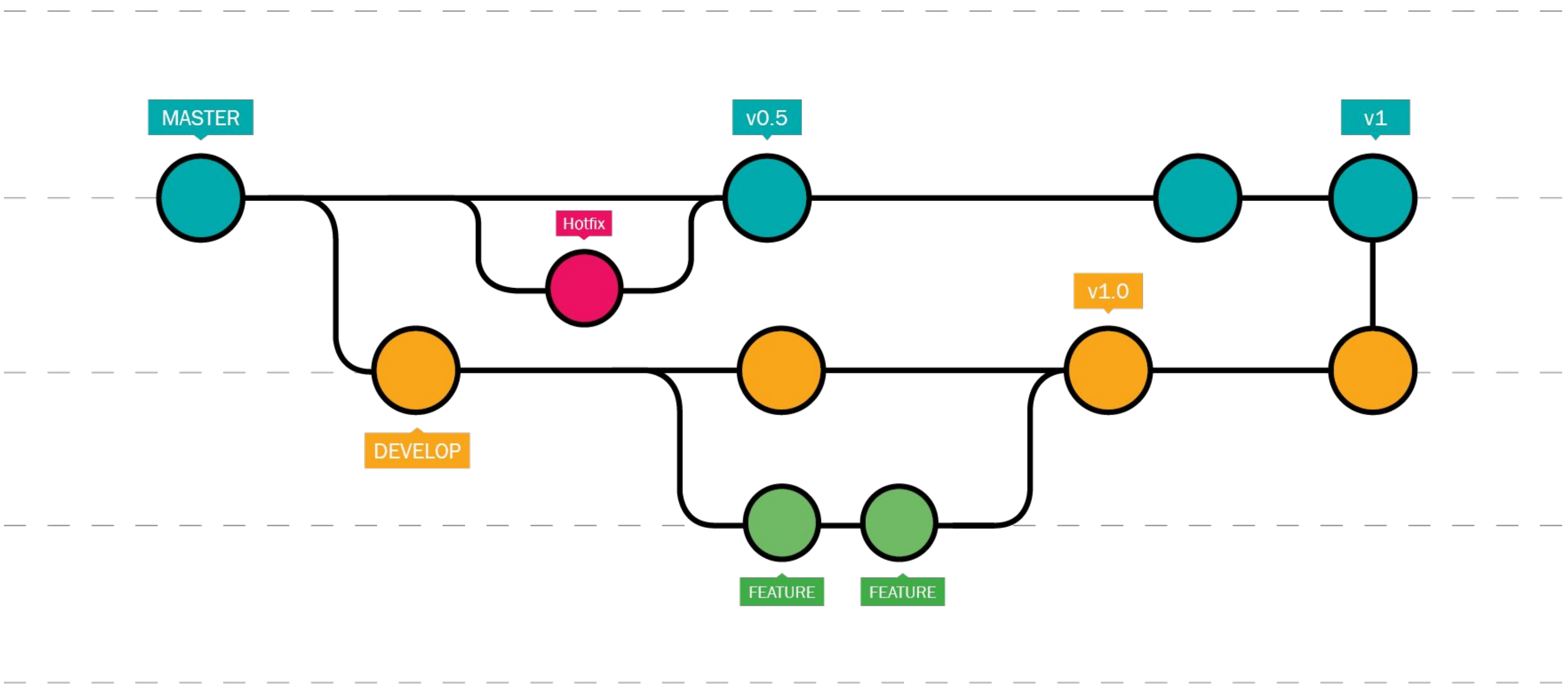
3 Tagging & Versioning

4 Ignoring Files

Branching & Rebasing



Branching Out



Branching Out

- To make a **new** branch.

```
git branch new_branch_name
```

- To **list** all the branches

```
git branch
```

- To **switch** to a branch

```
git checkout branch_name
```

- To create a branch and checkout it in **one step**

```
git checkout -b new_branch_name
```

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 :branch_name
```

- To delete a **local** branch

```
git branch -d branch_name
```

Merging Branches

After finishing your work on the branch,
you've to **merge** it with the Master branch.

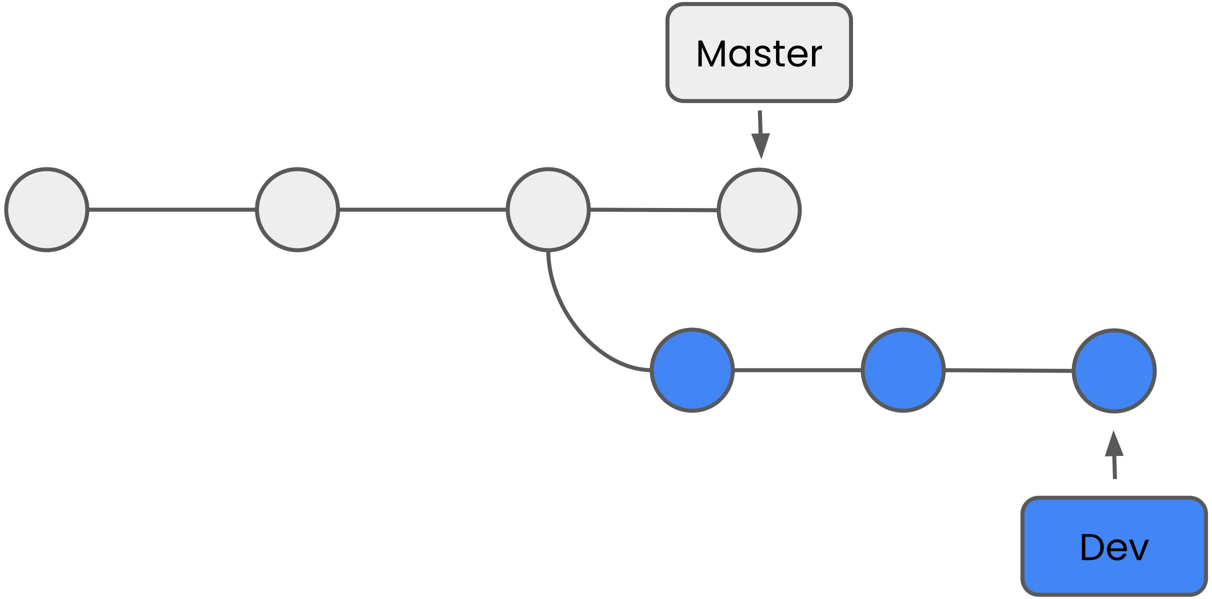
- First, go to the Master branch

  ```
  git checkout master
  ```

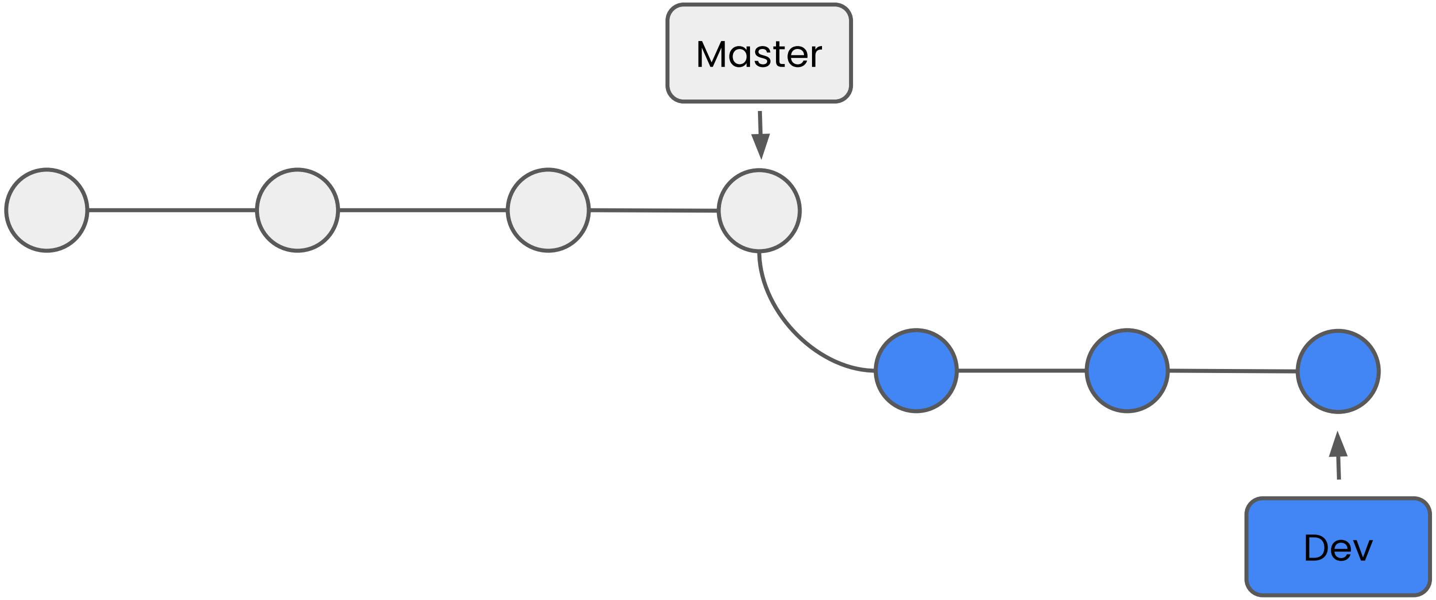
- Then, merge the two branches with each other

  ```
  git merge branch_name
  ```

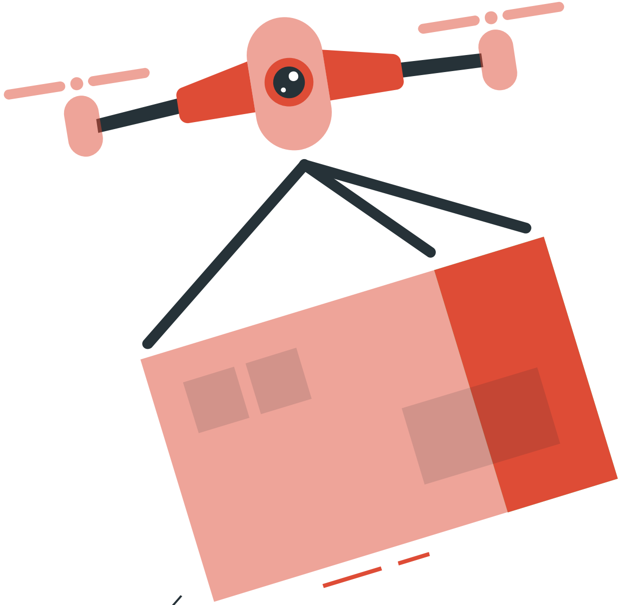
Git Rebase



Git Rebase



Pull Request



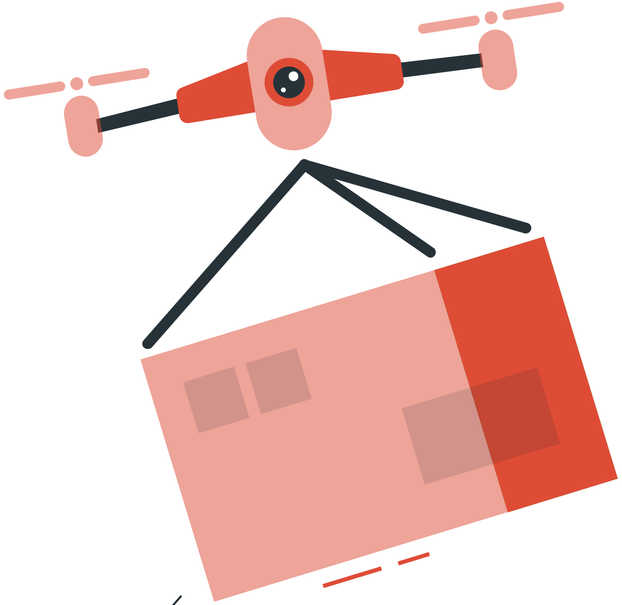
Pull Request

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.

Demo on GitHub

Tagging & Versioning



Tagging

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

Git supports two types of tags:

- Lightweight
- Annotated.

Tags Types

- To create a **lightweight** tag

```
git tag v1.0
```

- To create an **annotated** tag

```
git tag -a v2.0 -m “version 2.0”
```

Push Tags

- To list all tags

```
git tag
```

- To push tags

```
git push origin <tag_name>
git push --tags
```

Delete Tags

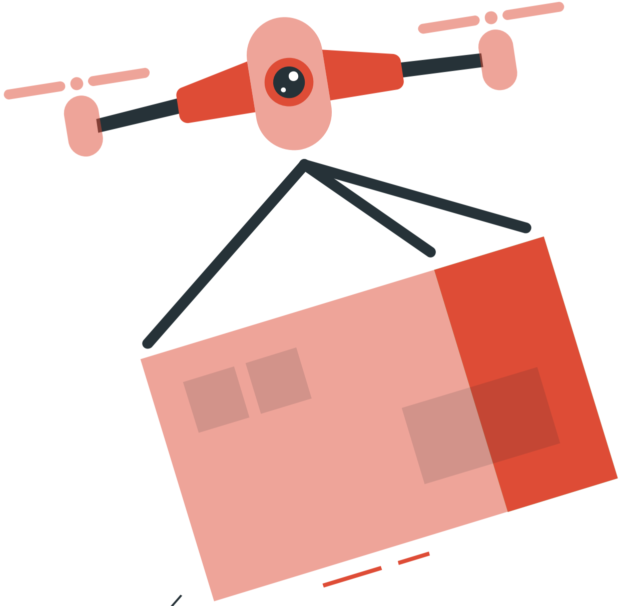
- To delete remote tag

```
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 changes to the remote repo.

- Merge this changes on Main branch and then push it to your remote main branch.

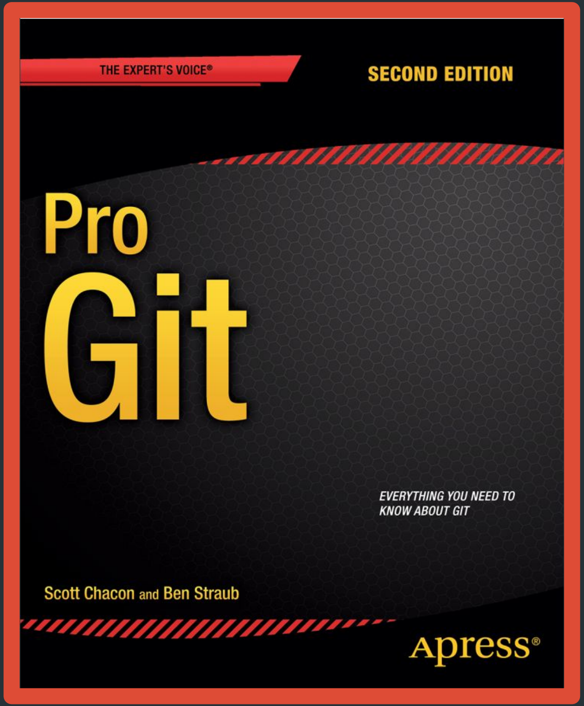
- Tell me how to remove them locally and remotely.

- Tell me how to checkout another branch without commit changes

Lab 2

- Create an annotated tag with tagname (v1.7) .
- Push it to the remote repository.
- Tell me how to list tags.
- Tell me how to delete tag locally and remotely.
- Add an image in the README.md file.

RESOURCES



THE EXPERT'S VOICE®

SECOND EDITION

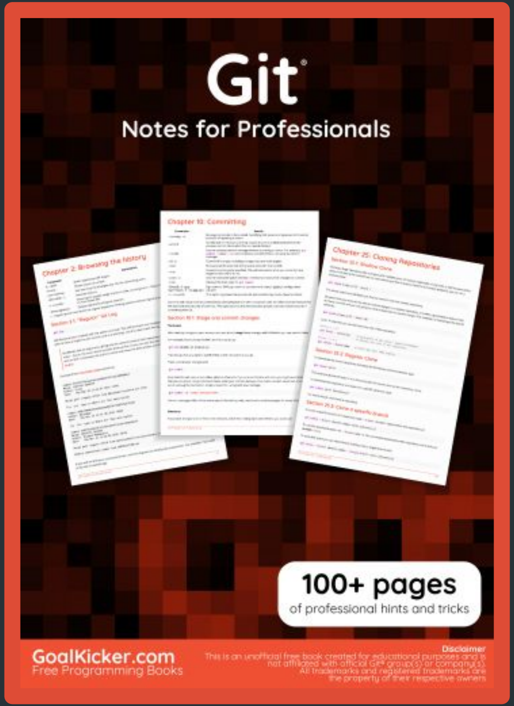
Pro Git

EVERYTHING YOU NEED TO KNOW ABOUT GIT

Scott Chacon and Ben Straub

Apress®

Pro Git - Second Edition



Git®

Notes for Professionals

100+ pages

of professional hints and tricks

GoalKicker.com

Free Programming Books

Disclaimer

This is an unofficial free book created for educational purposes and is not affiliated with official Git® group(s) or company(s). All trademarks and registered trademarks are the property of their respective owners

Git - Notes for Professionals