



POLITECNICO
MILANO 1863



ITECNICO
LANO 1863

Computing Systems

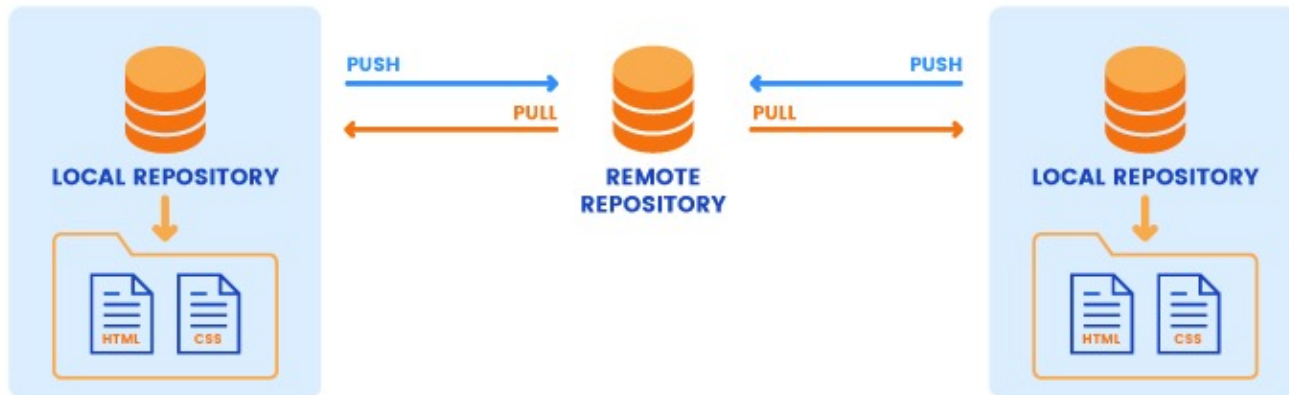
Git

Andrea Masciadri, PhD
andrea.masciadri@polimi.it

What is GIT



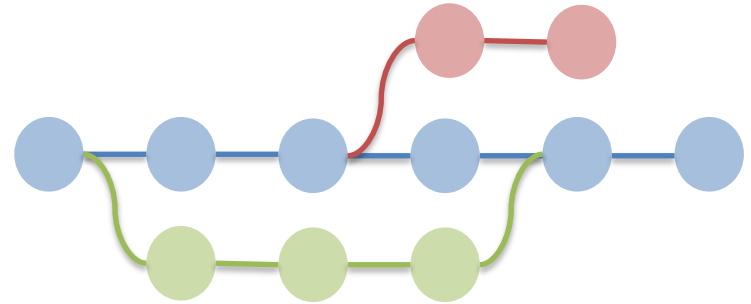
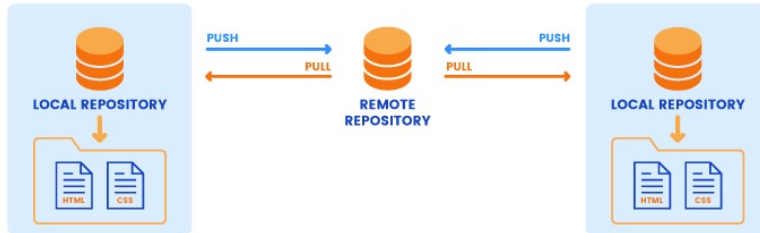
Git is a free and open source
distributed version control system.



<https://rubygarage.org/blog/most-basic-git-commands-with-examples>



Why GIT?



Having a centrally located place where you can upload your changes and download changes from others, enable you to **collaborate** more easily with other developers: Git can **automatically merge** the changes, so two people can even work on different parts of the same file and later merge those changes without losing each other's work.

Version control and track changes!



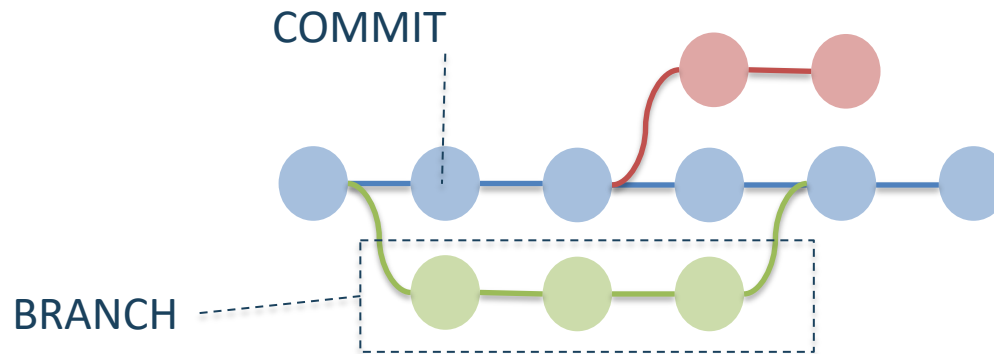
Definitions

BRANCH A "branch" is an active line of development. The most recent commit on a branch is referred to as the tip of that branch. The tip of the branch is referenced by a branch head, which moves forward as additional development is done on the branch. A single Git repository can track an arbitrary number of branches, but your working tree is associated with just one of them (the "current" or "checked out" branch), and HEAD points to that branch.

COMMIT

As a noun: A single point in the Git history; the entire history of a project is represented as a set of interrelated commits. The word "commit" is often used by Git in the same places other revision control systems use the words "revision" or "version". Also used as a short hand for commit object.

As a verb: The action of storing a new snapshot of the project's state in the Git history, by creating a new commit representing the current state of the index and advancing HEAD to point at the new commit.

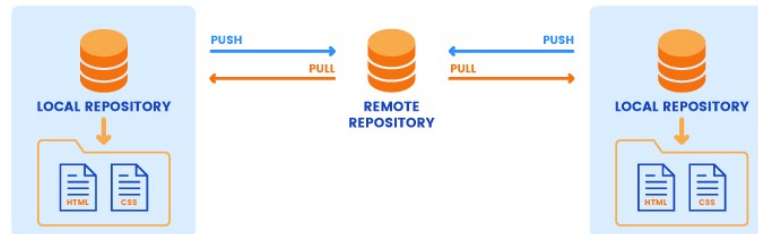


Definitions

PULL Pulling a branch means to fetch it and merge it.

- **FETCH** Fetching a branch means to get the branch's head ref from a remote repository, to find out which objects are missing from the local object database, and to get them, too.
- **MERGE** To bring the contents of another branch into the current branch. Merging is performed by an automatic process that identifies changes made since the branches diverged, and then applies all those changes together. In cases where changes conflict, manual intervention may be required to complete the merge.

PUSH Pushing a branch means to get the branch's head ref from a remote repository, find out if it is an ancestor to the branch's local head ref, and in that case, putting all objects, which are reachable from the local head ref, and which are missing from the remote repository, into the remote object database, and updating the remote head ref. If the remote head is not an ancestor to the local head, the push fails.



Install git

You can use GitHub with the student pack

- **Benefit:** Free while you are a student.

GitHub for Windows <https://windows.github.com>

GitHub for Mac <https://mac.github.com>

Git for All Platforms <http://git-scm.com>

GitHub is a platform for hosting and collaborating on Git repositories



Commands: Create a repository

`$ git init`

Turn an existing directory into a git repository

`$ git clone [url]`

Clone (download) a repository that already exists on GitHub, including all of the files, branches, and commits



Commands: Branches

\$ git branch [branch-name]

Creates a new branch

\$ git checkout [branch-name]

Switches to the specified branch and updates the working directory

\$ git merge [branch]

Combines the specified branch's history into the current branch. This is usually done in pull requests, but is an important Git operation

\$ git branch -d [branch-name]

Deletes the specified branch



Commands: Synchronization

\$ git fetch

Downloads all history from the remote tracking branches

\$ git merge

Combines remote tracking branch into current local branch

\$ git push

Uploads all local branch commits to GitHub

\$ git pull

Updates your current local working branch with all new commits from the corresponding remote branch on GitHub.
git pull is a combination of git fetch and git merge



Commands

For more commands you can access the Git documentation here:

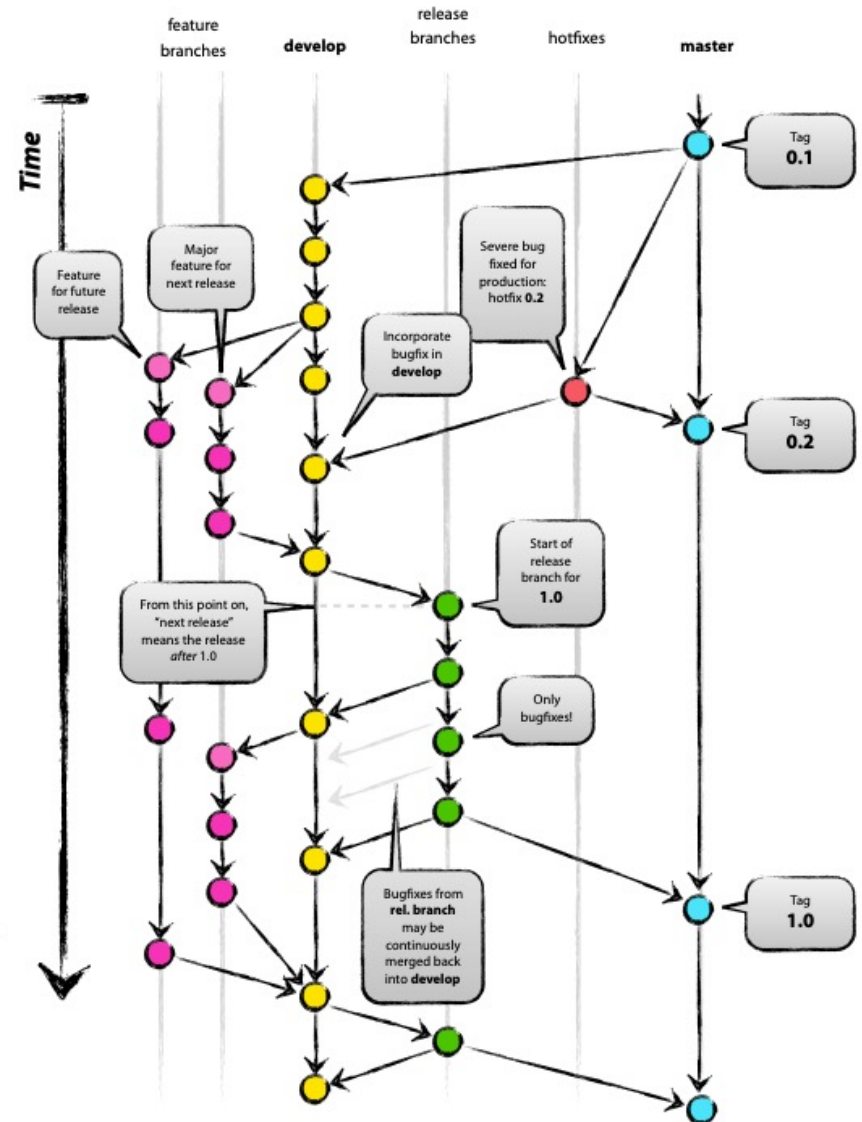
https://git-scm.com/docs/git#_git_commands

A complete tutorial to learn how to import a new project into Git, make changes to it, and share changes with other developers is available here:

<https://git-scm.com/docs/gittutorial>



How to use git? A successful Git branching model



Author: Vincent Driessen

Original blog post: <http://nvie.com/posts/a-successful-git-branching-model>

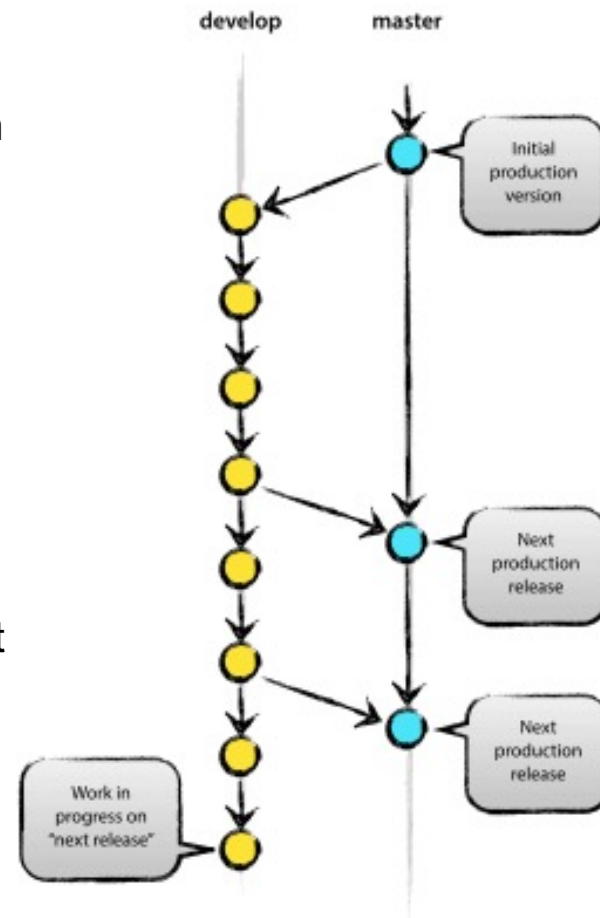
License: Creative Commons BY-SA



Main branches

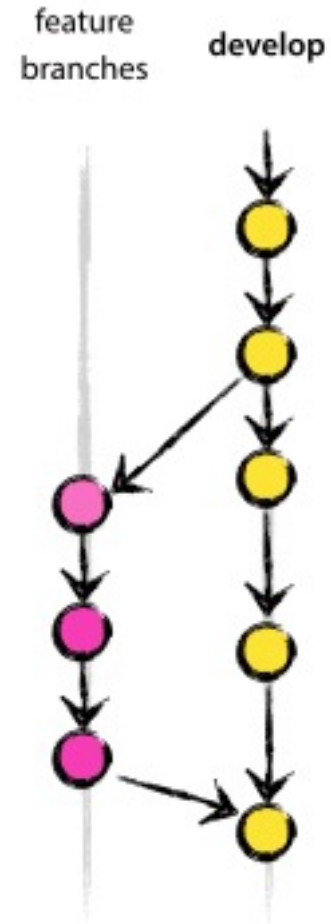
The central repo holds two main branches with an infinite lifetime:

1. We consider **origin/master** to be the main branch where the source code of HEAD always reflects a *production-ready* state.
2. We consider **origin/develop** to be the main branch where the source code of HEAD always reflects a state with the latest delivered development changes for the next release.



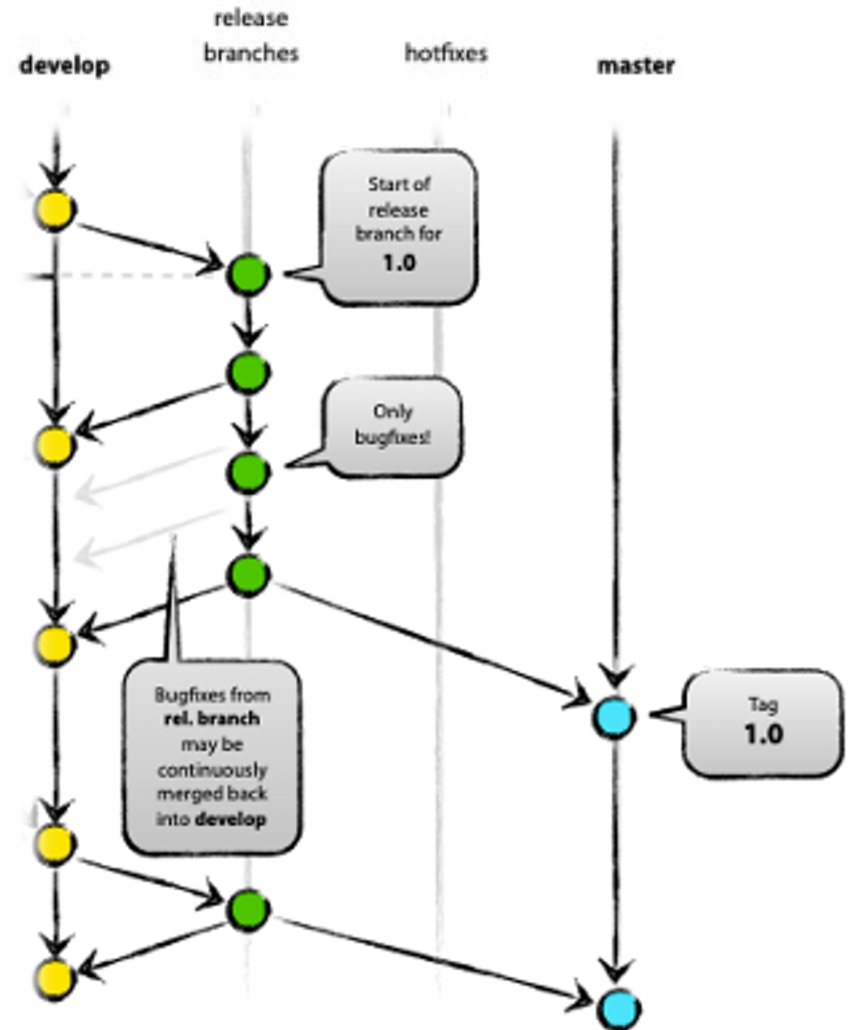
Feature branches

Feature branches (or sometimes called topic branches) are used to develop new features for the upcoming or a distant future release.



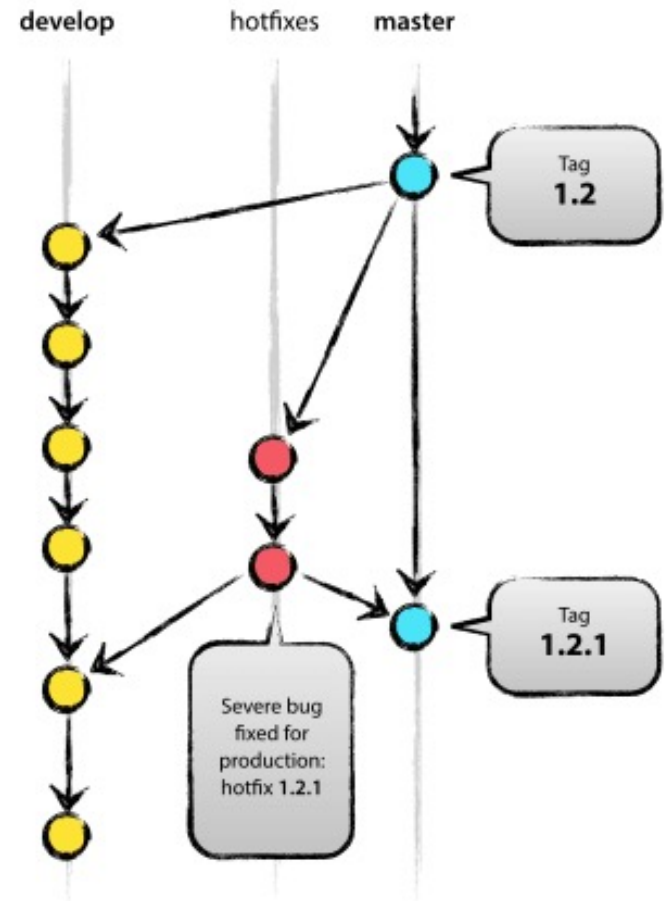
Release branches

The key moment to branch off a new release branch from `develop` is when `develop` (almost) reflects the desired state of the new release. At least all features that are targeted for the release-to-be-built must be merged in to `develop` at this point in time. All features targeted at future releases may not—they must wait until after the release branch is branched off.



Hotfix branches

When a critical bug in a production version must be resolved immediately, a hotfix branch may be branched off from the corresponding tag on the master branch that marks the production version.



References

- Git documentation <https://git-scm.com/>
- GitHub Education <https://education.github.com>
- A successful Git Branching Model <https://nvie.com/posts/a-successful-git-branching-model/>





POLITECNICO
MILANO 1863



ITCNICO
LANO 1863

Questions?

andrea.masciadri@polimi.it

