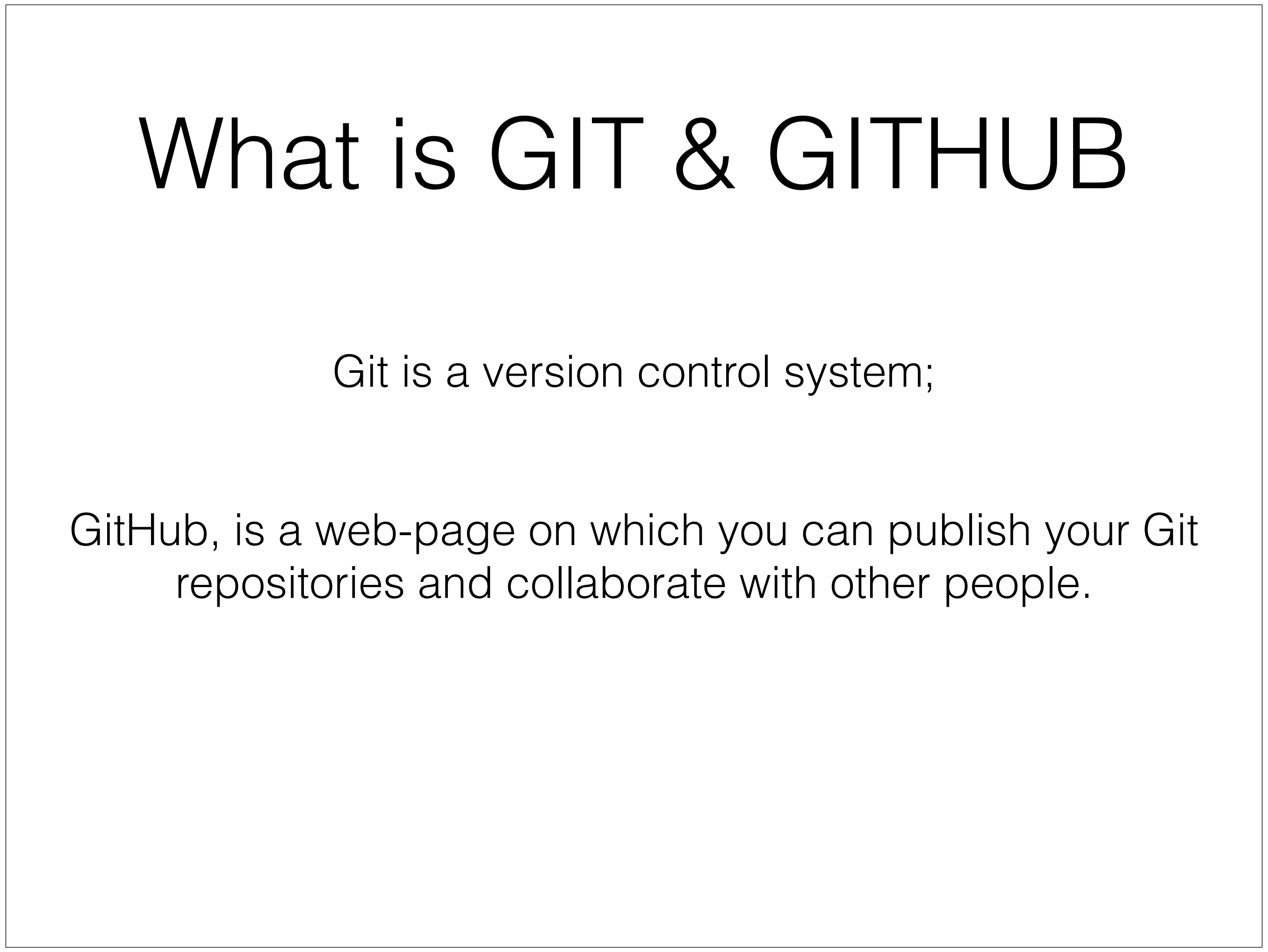
GIT & GITHUB

The basics
03/12/2014

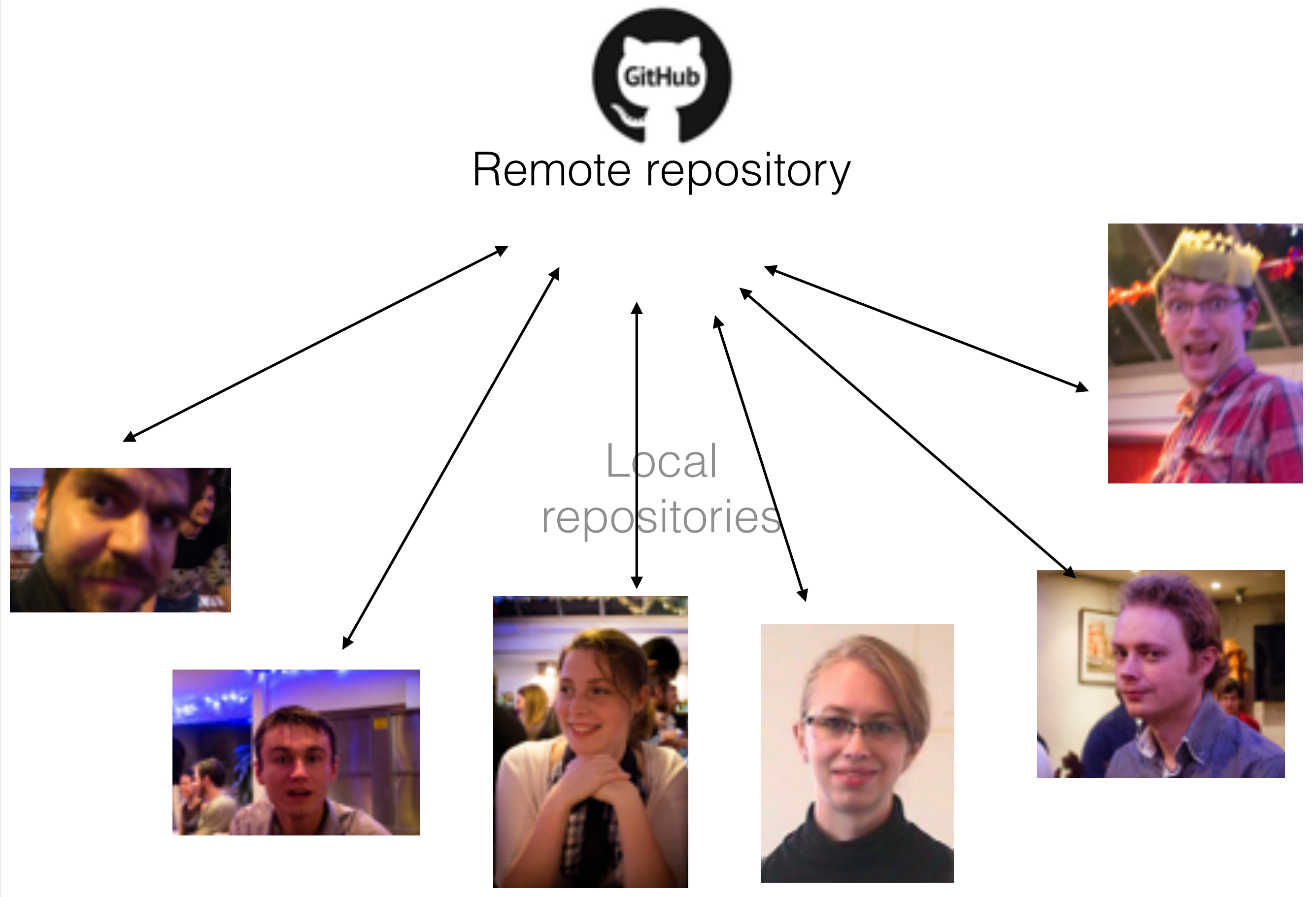


What is GIT & GITHUB

Git is a version control system;

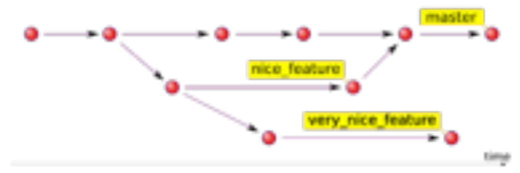
GitHub, is a web-page on which you can publish your Git repositories and collaborate with other people.

GIT: series of snapshots (commits) of your code. You see a path of these snapshots, in which order they where created. You can make branches to experiment and come back to snapshots you took.


GitHub
Remote repository
Local repositories

Terminology

Branch: A branch is a parallel version of a repository.



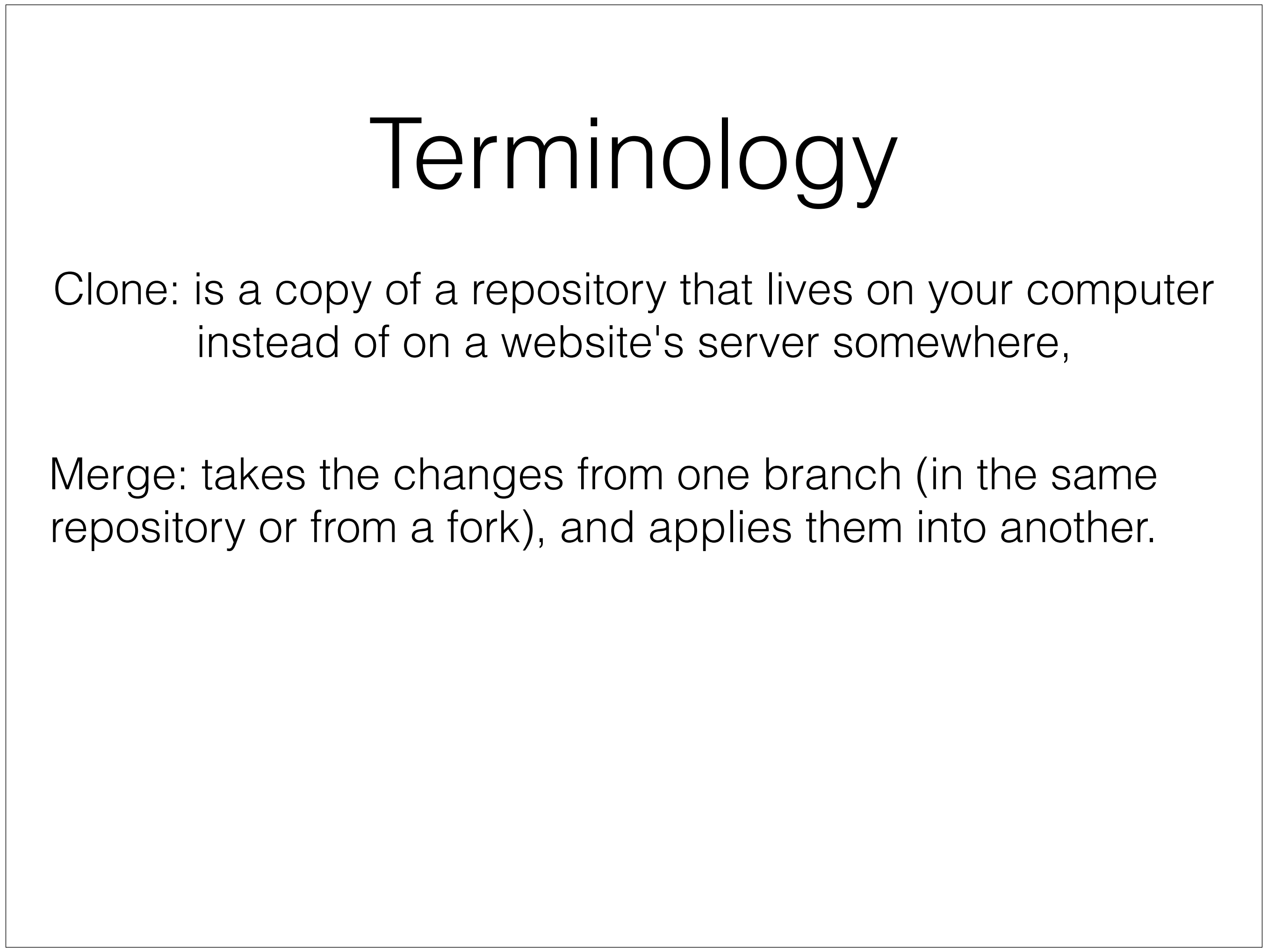
Commit: or "revision", is an individual change to a file (or set of files).

Pull: refers to when you are fetching in changes and merging them.

Branch: contained within repository, does not affect the primary or master branch allowing you to work without disrupting the "live" version. You can merge your branch back into the master branch to publish your changes.

Commit: like when you save a file, except with Git, every time you save it creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of what changes were made when and by who. Commits usually contain a commit message which is a brief description of what changes were made.

PULL: For instance, if someone has edited the remote file you're both working on, you'll want to pull in those changes to your local copy so that it's up to date.



Terminology

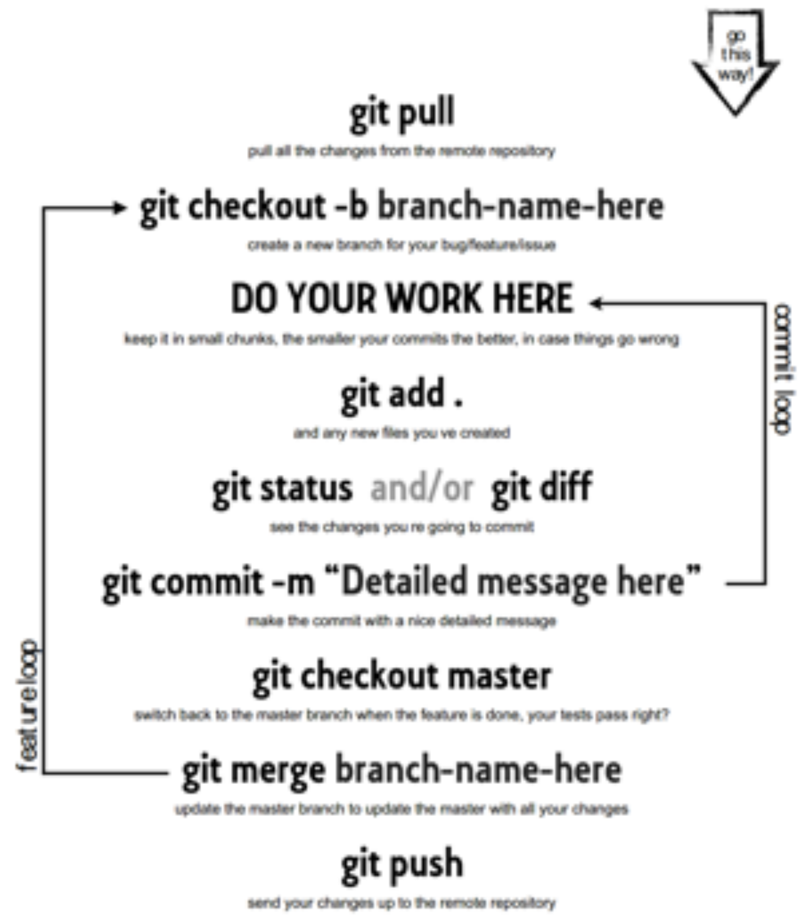
Clone: is a copy of a repository that lives on your computer instead of on a website's server somewhere,

Merge: takes the changes from one branch (in the same repository or from a fork), and applies them into another.

Clone: edit the files in your preferred editor, use Git to keep track of your changes without having to be online. It is, however, connected to the remote version so that changes can be synced between the two. You can push your local changes to the remote to keep them synced when you're online.

Merge: This often happens as a Pull Request (which can be thought of as a request to merge), or via the command line. A merge can be done automatically via a Pull Request via the GitHub.com web interface if there are no conflicting changes, or can always be done via the command line.

simple daily git workflow



git pull

pull all the changes from the remote repository

git checkout -b branch-name-here

create a new branch for your bug/feature/issue

DO YOUR WORK HERE

keep it in small chunks, the smaller your commits the better, in case things go wrong

git add .

and any new files you ve created

git status and/or **git diff**

see the changes you re going to commit

git commit -m "Detailed message here"

make the commit with a nice detailed message

git checkout master

switch back to the master branch when the feature is done, your tests pass right?

git merge branch-name-here

update the master branch to update the master with all your changes

git push

send your changes up to the remote repository

Other cool things to note:

- You can go back to old versions of code.
- You can see the differences between different versions of cod
- Issues: Way to flag issues or enhancements. Can be referred to in commits.
- Markdown: Any comments on GITHUB are in Markdown language
- Displays read me files

markdown: plain text formatting syntax[5] designed so that it can be converted to HTML