



GIT

Agenda

- GIT
- Difference between GIT & GitHub
- Git Repository (what is, init)
- Git Commands: *stage, commit*
- *.gitignore* File



Agenda

- Git Branches (*create, checkout, delete, merge*)
- Git *remote repository*
- Sync with remote (*push, clone, fetch*)
- Git Tools
- Quiz
- Assignment

What Is GIT?

Git Install



GIT

- A Version Control System
- It records the changes made to our code overtime. (It tracks changes in source code, during software development)
- It can be used to track changes in any set of files.
- Git goals include: speed, data integrity, support for distributed, non-linear workflows
- The creator of Git is Linus Torvalds – Linux creator

GIT - Install

- <https://git-scm.com/> Source Control Management (SCMs)
- Default, commit-as-is, environment variables for git should point towards git cmd binary



Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change

<https://gitforwindows.org/>

Only show new options

Next >

Cancel



Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- Additional icons
 - On the Desktop
- Windows Explorer integration
 - Git Bash Here
 - Git GUI Here
- Git LFS (Large File Support)
- Associate .git* configuration files with the default text editor
- Associate .sh files to be run with Bash
- Use a TrueType font in all console windows
- Check daily for Git for Windows updates

Current selection requires at least 259.1 MB of disk space.

<https://gitforwindows.org/>

Only show new options

< Back

Next >

Cancel



Choosing the default editor used by Git

Which editor would you like Git to use?

Use Vim (the ubiquitous text editor) as Git's default editor



The [Vim editor](#), while powerful, [can be hard to use](#). Its user interface is unintuitive and its key bindings are awkward.

Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.

Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.

<https://gitforwindows.org/>

Only show new options

< Back

Next >

Cancel



Adjusting the name of the initial branch in new repositories

What would you like Git to name the initial branch after "git init"?

Let Git decide

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project [intends](#) to change this default to a more inclusive name in the near future.

Override the default branch name for new repositories

NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

This setting does not affect existing repositories.

<https://gitforwindows.org/> —

Only show new options

< Back

Next >

Cancel



Adjusting your PATH environment

How would you like to use Git from the command line?

Use Git from Git Bash only

This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

Git from the command line and also from 3rd-party software

(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools.

You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

Use Git and optional Unix tools from the Command Prompt

Both Git and the optional Unix tools will be added to your PATH.

Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

<https://gitforwindows.org/>

Only show new options

< Back

Next >

Cancel



Choosing HTTPS transport backend

Which SSL/TLS library would you like Git to use for HTTPS connections?



Use the OpenSSL library

Server certificates will be validated using the ca-bundle.crt file.

Use the native Windows Secure Channel library

Server certificates will be validated using Windows Certificate Stores.

This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

<https://gitforwindows.org/>

Only show new options

< Back

Next >

Cancel



Configuring the line ending conversions

How should Git treat line endings in text files?

Checkout Windows-style, commit Unix-style line endings

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

Checkout as-is, commit Unix-style line endings

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

Checkout as-is, commit as-is

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

<https://gitforwindows.org/>

Only show new options

< Back

Next >

Cancel



Choose the default behavior of `git pull`

What should `git pull` do by default?

Default (fast-forward or merge)

This is the standard behavior of `git pull`: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.

Rebase

Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

Only ever fast-forward

Fast-forward to the fetched branch. Fail if that is not possible.

<https://gitforwindows.org/>

Only show new options

< Back

Next >

Cancel



Choose a credential helper

Which credential helper should be configured?

Git Credential Manager Core

(NEW!) Use the new, [cross-platform version of the Git Credential Manager](#).
See more information about the future of Git Credential Manager [here](#).

Git Credential Manager

(DEPRECATED) The [Git Credential Manager for Windows](#) handles credentials e.g.
for Azure DevOps and GitHub (requires .NET framework v4.5.1 or later).

None

Do not use a credential helper.

<https://gitforwindows.org/>

Only show new options

< Back

Next >

Cancel



Let's check out if we installed git correctly

```
git --version
```

```
git config --global user.name "Your Name"
```

```
git config --global user.email "you@example.com"
```

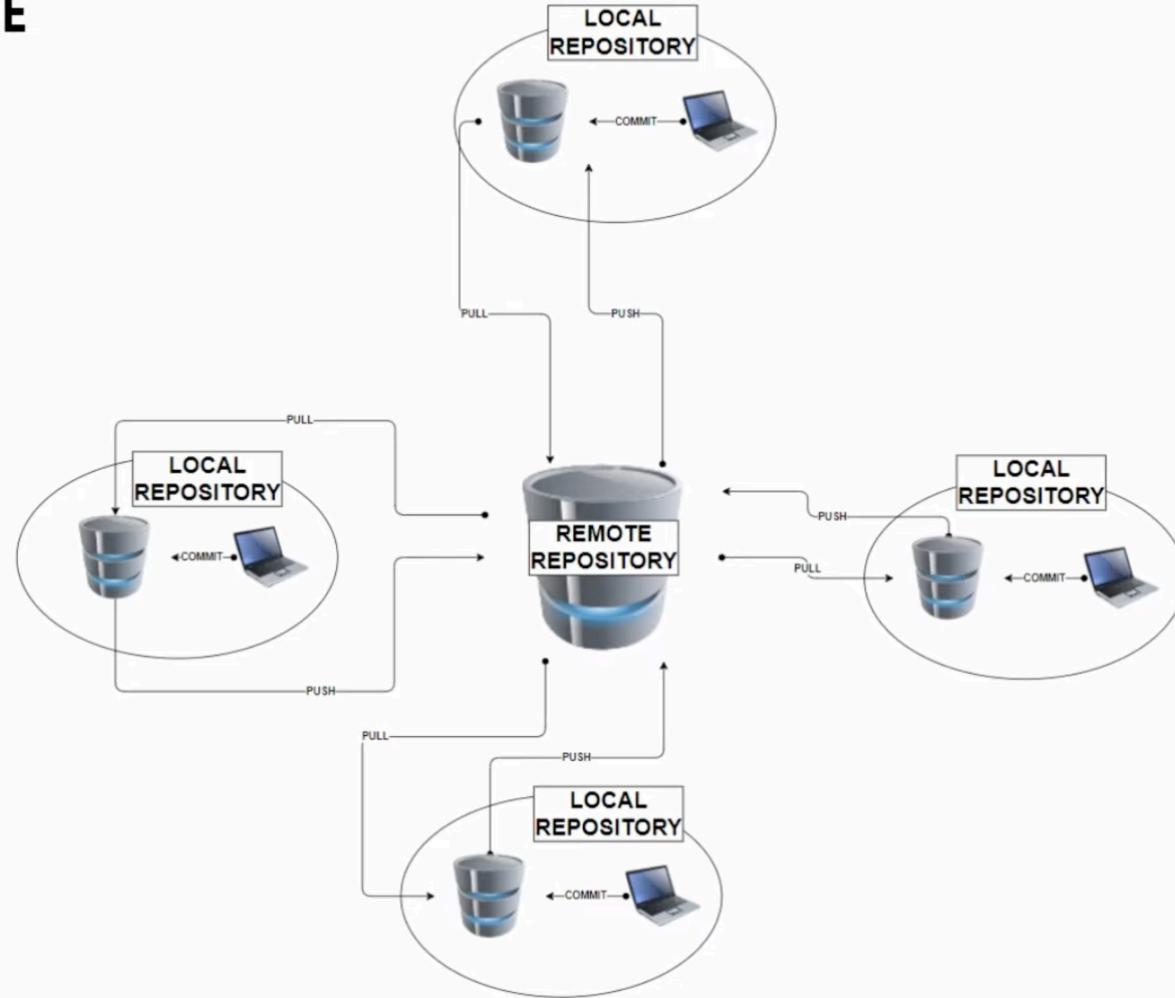
GIT vs GitHub distinction

GIT vs GitHub distinction

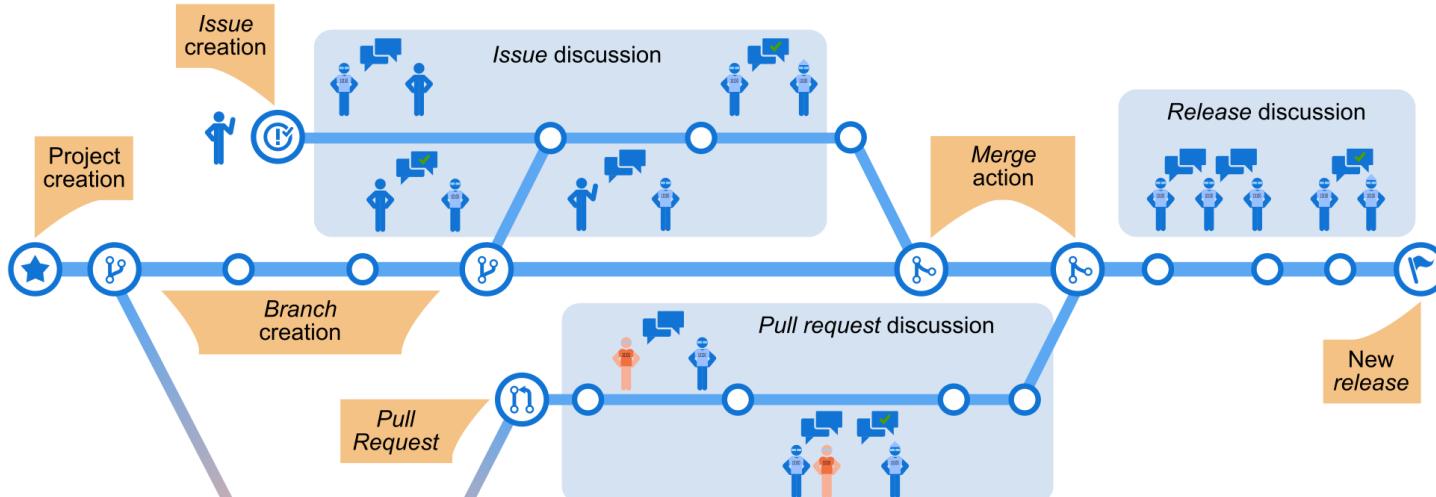
- They are completely different things
- Git is a technology, a distributed system version whereas Github is a platform that uses Git Technology
- Github is owned by Microsoft since 2018 and it's the most popular platform where you can store code

Git Repositories

LOCAL VS REMOTE

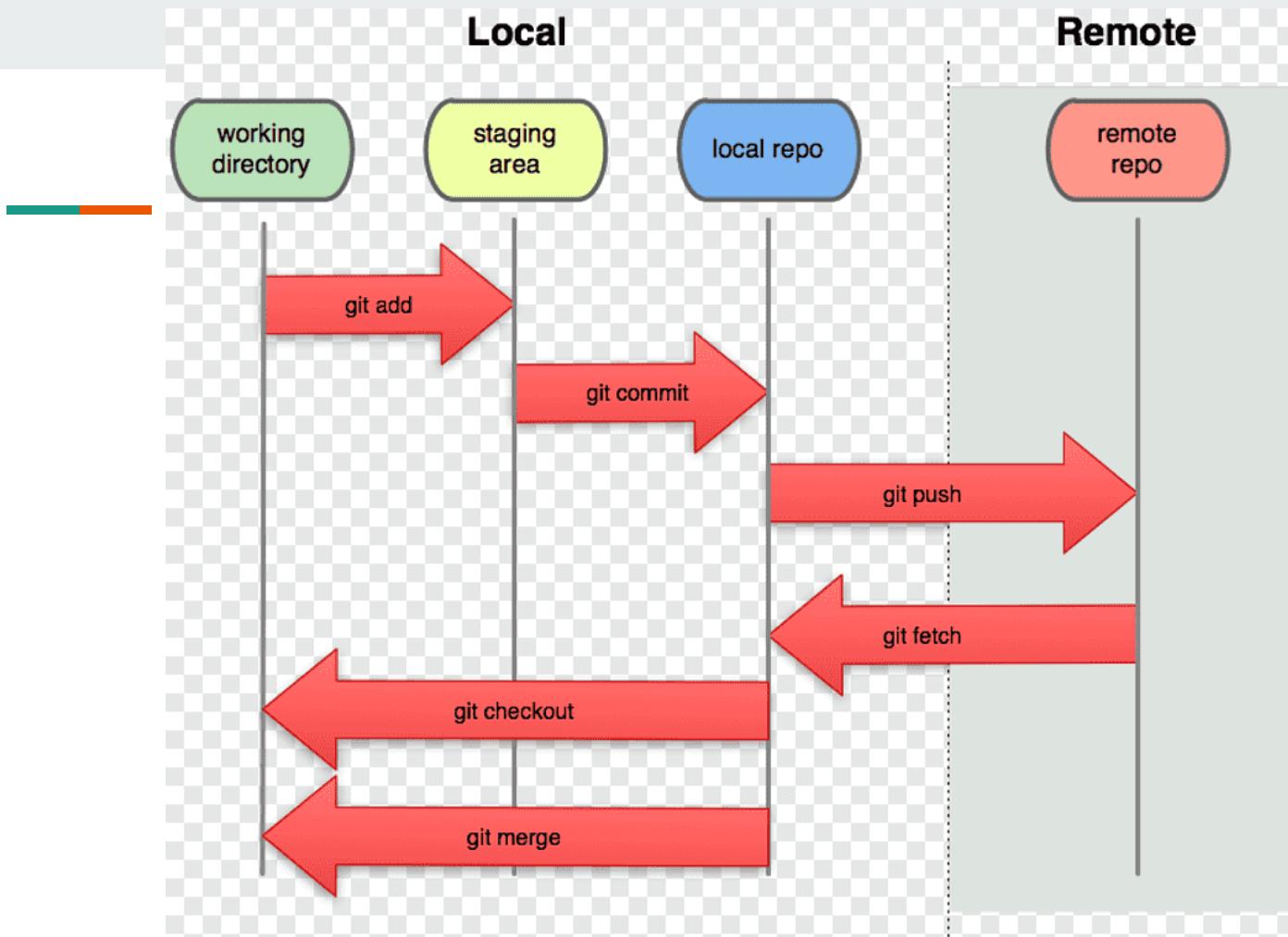


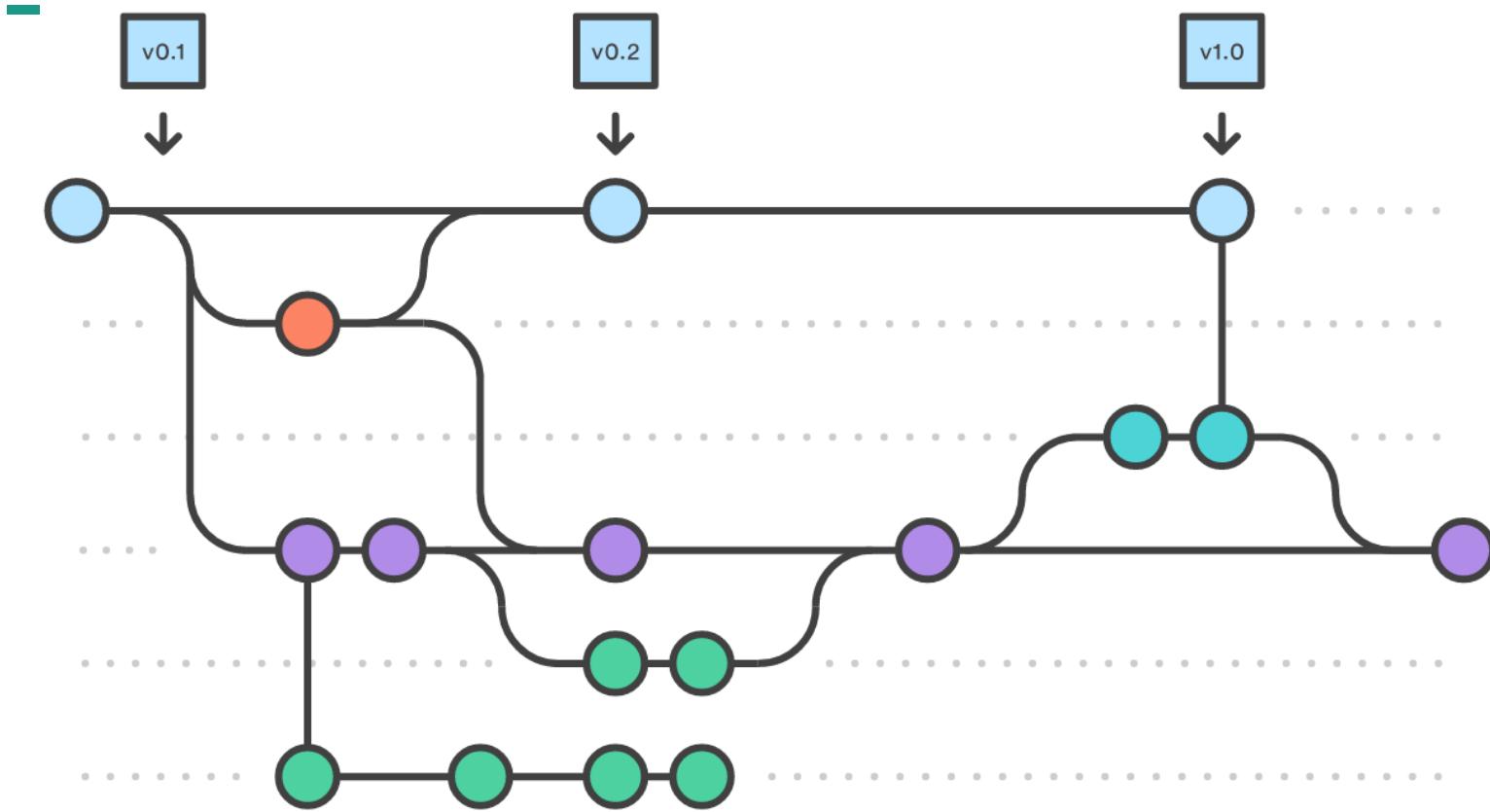
How GitHub projects are developed? Where are the main discussion points?



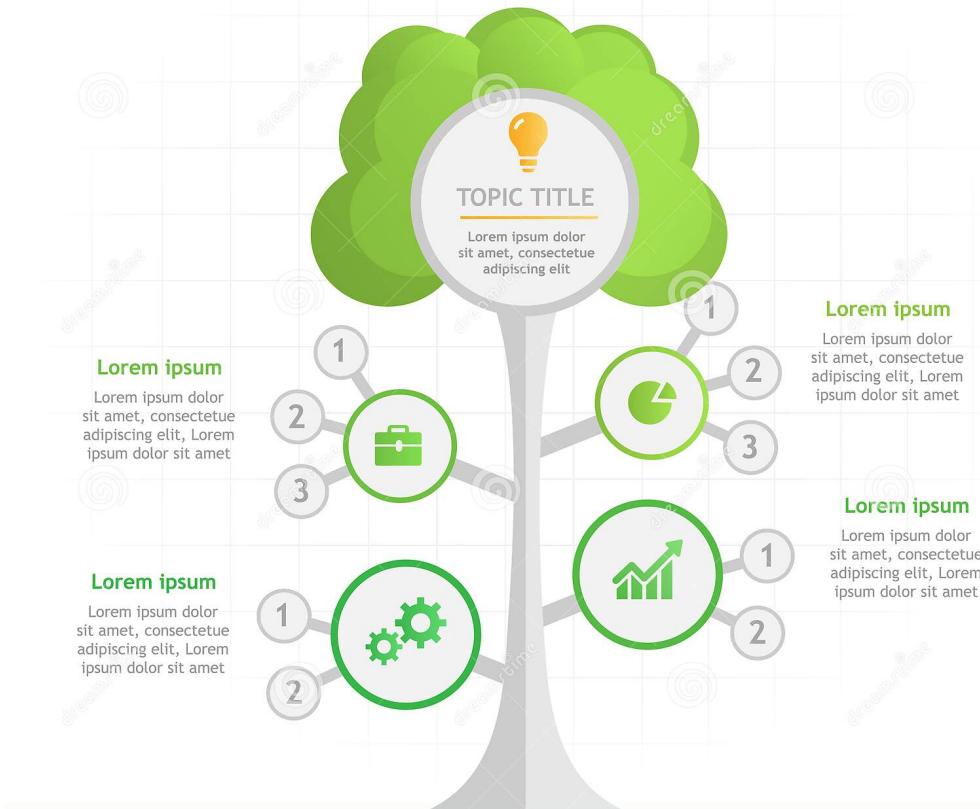
- | | | |
|---------------------|---------------------|--|
| Project | (★) | Contains the source code of a software application |
| Issue | (💬) | Change request (bug or improvement) for an application |
| Commit | (○) | Set of changes of the source code of an application |
| Branch | (—) | Set of commits usually addressing an application's feature |
| Fork | (🍴) | Creation of a new branch |
| Pull Request | (Pull Request icon) | Request to join two branches |
| Merge | (Merge icon) | Action of joining two branches |
| Release | (Flag icon) | Publication of a new version of an application |







MINDMAP INFOGRAPHIC



Git Commands – Hands-on

- git init
- git status
- git add (what is git staging)

Git Commands – Stage & Commit

- git commit
- git log

Git Commands – gitignore File

- Files can be ignored individually
- By extensions
- By folders

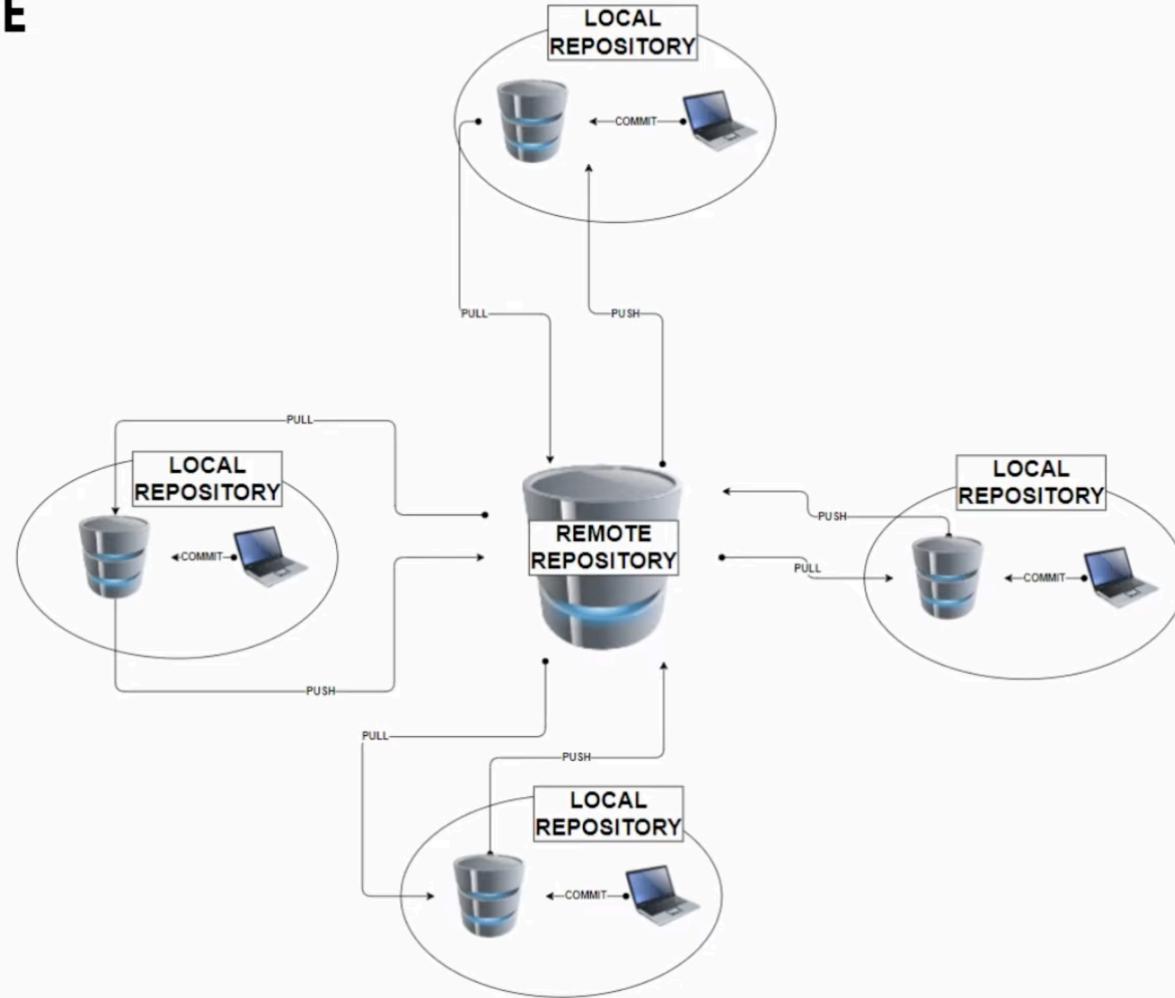
Git Commands – Branches

- `git branch`
- `git checkout branchname`
- `git checkout -b branchname`
- `git branch -d branchname | git branch -D branchname`
- We work individually, independently on the branches we make
- All the modifications that were made up until we created the new branch are kept in the old branch
- All the modifications are copied **from** the branch we are starting the new one

Git Commands – Branches

- `git merge branchname`

LOCAL VS REMOTE



Git Commands – Remote Repositories

- git remote add origin <https://github.com/git-demo.git>
- git remote -v
- git push -u origin master (-u stands for sync with master – only used first time)
- git push
- git pull
- git fetch

Git Commands – Remote Repositories

- git fetch is the command that tells your local git to retrieve the latest meta-data info from the original (yet doesn't do any file transferring. It's more like just checking to see if there are any changes available).
- git pull on the other hand does that AND brings (copy) those changes from the remote repository.

Git Commands – Remote Repositories

- Fixing merge conflicts

GitHUB account

- Let's create a GitHub account - <https://github.com/signup>

Git Windows SSH

- **What is SSH?**
- SSH is a way to authenticate with a server like GitHub, without exposing your username and password.
- SSH stands for Secure Shell and is a cryptographic protocol based on the concept of public-private keys.
- We are using SSH with Git because it is much easier than typing your username and password all the time, and it is also more secure.
- You have probably noticed that GitHub is complaining that you don't have an SSH key.

Git Windows SSH Key

- For Github authorization or other platforms
- Open Git Bash that you just installed (Start->All Programs->Git->Git Bash)
- Type in the following: `ssh-keygen -t rsa` (when prompted, enter password, key name can stay the same)
- Open file `your_home_directory/.ssh/id_rsa.pub` with your favorite text editor, and copy contents to your Git repository's keys field (`GitHub`, `Gitlab` or other provider), under your account.
- Be sure that you **don't copy** any whitespace while copying public key's content (`id_rsa.pub`)

Git Windows SSH Key

- Note: `your_home_directory` is either `C:\Users\your_username` (on Windows Vista / 7 / 8 / 10), or `C:\Documents and Settings\your_username` (on Windows XP)
- When you create private/public SSH keys on your machine (that's what you did in the above steps), it's not enough. You need to give your public key to the repository in order to pair the Git server with your local machine (that'd be steps 4. and 5. above).

Git OSX SSH

- Enter the following command in the Terminal window.
- ssh-keygen -t rsa -this command generated the ssh rsa key (press enter on each step)
- pbcopy < ~/.ssh/id_rsa.pub -this command copies the newly generated key

Signed in as **alexghi** [Edit status](#)[Your profile](#)[Your repositories](#)[Your codespaces](#)[Your projects](#)[Your stars](#)[Your gists](#)[Upgrade](#)[Feature preview](#) [Help](#)[Settings](#)[Sign out](#)

All activity

sindresorhus released v1.8.1 of **sindresorhus/Actions** 16 hours ago **sindresorhus / Actions**[Sponsor](#)**v1.8.1**

- Fix description for the "Parse CSV" action.

[v1.8.0...v1.8.1](#)

6

AndreiJoldos and **BoraElena** started following you 22 hours ago **AndreiJoldos**

1 repository 1 follower

[Unfollow](#) **craftzdog** created a repository **craftzdog/chat-bubbles-for-yt** yesterday **craftzdog/chat-bubbles-for-yt**

Real-time recording for chat bubbles

Updated Mar 5

[Star](#) **sindresorhus** released v1.8.0 of **sindresorhus/Actions** 3 days ago**Our response to the war in Ukraine**

GitHub is united with the people of Ukraine and the international community.

[Read more](#)**Explore repositories****edgar-zigis/CocoaTextField**

Apple UITextField created according to the Material Design guidelines of 2019. Featured at Medium.

Swift 257

chainPaneth/react-native-jw-media-play

React-Native Android/iOS bridge for JWPlayer (<https://www.jwplayer.com/>)

Objective-C 104

siriwatknp/mui-treasury

A collection of ready-to-use components based on Material-UI

JavaScript 1.9k

[Explore more →](#)



Search or jump to...



Pull requests Issues Marketplace Explore



+



Alex Ghiurau
Your personal account

[Go to your personal profile](#)

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

SSH and GPG keys

Organizations

Moderation

Code, planning, and automation

Repositories

Packages

Pages

Saved replies

Security

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



Alex MacbookPRO

SSH

Last used within the last week — Read/write

[Delete](#)

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

[New GPG key](#)

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

Vigilant mode

[Flag unsigned commits as unverified](#)

This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.

[Learn about vigilant mode.](#)



Search or jump to...



Pull requests Issues Marketplace Explore



Alex Ghiurau
Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

SSH and GPG keys

Organizations

Moderation

Code, planning, and automation

Repositories

Packages

Pages

Saved replies

Security

Code security and analysis

Integrations

Applications

Scheduled reminders

SSH keys / Add new

Title

some name

Key

here you should paste the public key you generated with [keygen](#)

Add SSH key

Git Reference

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

Git Quiz



Git Quiz

- https://www.w3schools.com/git/git_quiz.asp?remote=github
- <https://play.kahoot.it/v2/?quizId=85f8952a-2854-4fc4-9e12-ef770b4adc0d>

Git Assignment

Git Assignment - 1

- Create a local / remote repository
- Create a branch other than master called development
- Create 3 commits on master that should have three files (
 - index.html first commit,
 - style.css second commit,
 - folder assets/image.png third commit)
- Push your local commits to origin
- Checkout on development branch
- Populate index.html, style.css in one commit

Git Assignment - 1

- Commit the populated files but don't push them to remote just yet
- Checkout on master branch (notice that index.html and style.css are still empty on master yet)
- Merge branch development into master
- When you merged it successfully push your latest commits to origin

Git Assignment - 2

- Ask a colleague to clone your repository
- After this your colleague should edit in his local repo some of the files by adding his own code
BUT in another branch that he created
- After this your colleague should push his modifications on origin (after he committed them, of course)

Git Assignment – 3

- clone this repo from here <git@github.com:alexghi/fasttrack-web-course.git>
- You can check your resources from all the courses there

Git Assignment – 4

- Create another local / remote repo (distinct from assignment no1) and add the files that you worked on in C3 in ONE commit
- After this push it to remote (this way I can see your homework from C3 there)