**<https://docs.github.com/en>**

**Git** - distributed, de/centralized source system

**Git** – це розподілена система контролю версій, яка дозволяє відстежувати історію розробки ПЗ і спільно працювати над складними проектами з будь-якої точки світу. Головна перевага Git – в тому, що він дуже швидкий і прозорий.

Git — це суто система керування версіями.

**Repository** contains files, history, config managed by Git

**Commit** - saved changes to Git repo, impacts history

**Master branch** = **main** branch, default branch

**Head** - pointer to the last commit on the branch

**Remote** - related repo but no local (GitHub, GitLab)

**How to change branch?**

Go to GitHub acc > Settings > Repositories > replace > Update

<https://globallogic.udemy.com/course/git-going-fast/learn/lecture/4816034#overview>

cmd:

git version

git help

**Git Configuration - name, email - під\*єднати ноут до гіта**

pwd to go to a home directory

git config --global user.name "Sofiya Firman"

git config --global user.email "..."

git config --global --list

cat ~/.gitconfig

To create a folder:

cd ~

mkdir projects

cd projects

git init git-demo The git init command transforms the current directory into a Git repository

ls -a - shows all files + hidden ones (dots)

subfolder is created

git add README.rtf - to add to staging

git status - to check the status

git commit -m "initial commit" - the file goes to Git Repository

git commit -am "some text" - does both: adds to stg + commits

git add . - adds to stg all files were modified.Make sure to **commit** them too

When files are on **STG**:

git reset HEAD README.rtf - **unstages** the file

git checkout -- README.rtf - to discard the changes in working directory

git log - history

git help log

git rm debug.log

git commit -m "removed logs" - to remove a file

mkdir web - to create a new directory (folder)

git mv index.html web - to move a file to the new dir

git commit -m "moving to web"

.gitignore files - Git exludes files we don’t want to have

mate .gitignore

\*.log - this should be entered in the opened window. \* means all that ends with .log

git add .gitignore

git commit -m "ignored log"

Going remote

1.

cd ~

mkdir .ssh

cd .ssh

ssh-keygen -t rsa -C ["hrytsyna.sph@gmail.com"](mailto:\"hrytsyna.sph@gmail.com\") - to generate SSH key

**<enter passphrase> use your pass**

enter,enter,enter

mate id\_rsa.pub - open the key file (located in .ssh)

Copy the key > go to <https://github.com> > log in > go to ‘Settings’ > ‘SSH keys’ > Add SSH key.

To access GitHub via ssh, do this:

ssh -T [git@github.com](mailto:git@github.com)

enter your passphrase: latenight

To upload the repo to GitHub

On the GitHub page, go to ‘Create a repository’ > enter the name (e.g. git-demo) > set public/private > UNcheck the initialize option > Create.

On the opened page, select SSH > go down > **copy first command** from ‘…or push an existing repository from the command line’

Go to the needed cd > git status > pass the command

git remote -v - to check that origin is associated with the remote repo

3.

To push the repo to Github

-u - establishes an upstrim link btw our remote repo and our local repo

git push -u origin main - the third line in ‘…or push an existing repository from the command line’

Enter the passcode.

For future pushes, just use git push origin main

Locally, we needed to edit a file. Once edited, we added and commited the changes.

Then we need to pull not push them:

git pull origin main

Then:

git push origin main

Pull is used to fetch any changes that were made and merge them into the GitHub repo before pushing them to GitHub.

**Why pull before push?**

- your push might fail if others have made changes to the remote repo since your last pull

- it allows you to integrate and test other’s changes with your changes

- it’s a good habit to a dev

- it’s **not required** if you’re the **sole** user of the rempote repox

To download a remote copy to PC

<https://docs.github.com/en/get-started/using-git/getting-changes-from-a-remote-repository>

Use git clone

+ link from Code > SSH (on Git Hub repo)

git clone git@github.com:ahjop/git-demo.git

UPDATES OF GIT ON **MAC** - how to resolve issues

which git

git version

sudo xcode-select --install

password + install