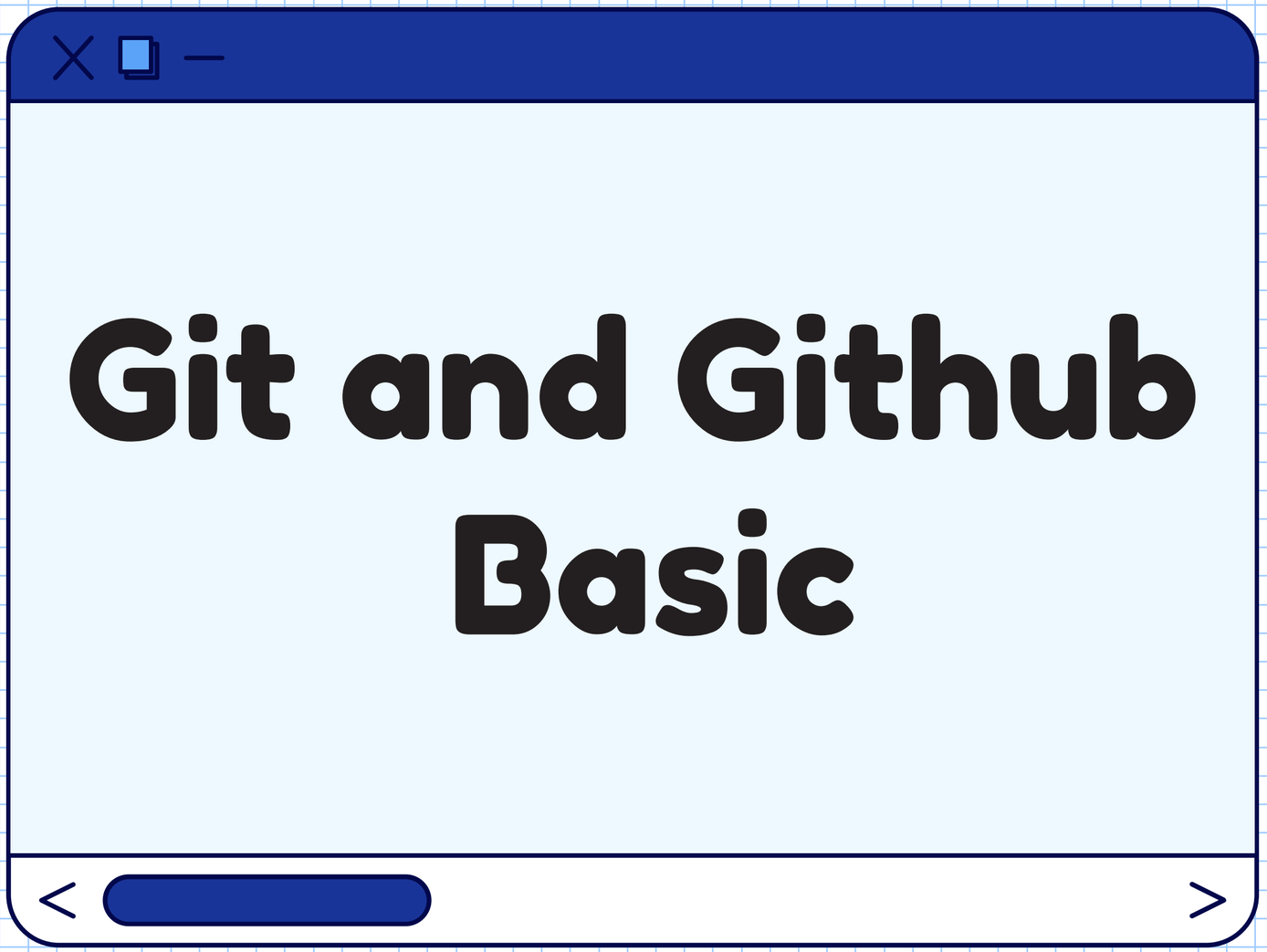


Git and Github Basic

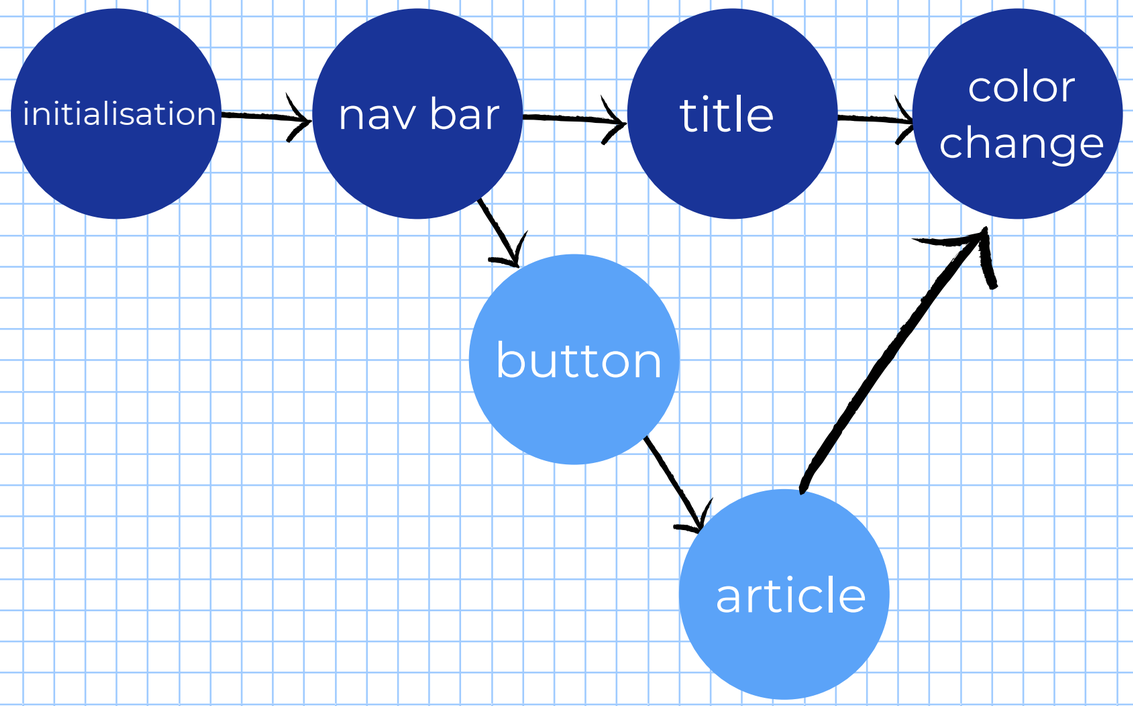
Git & Github why do we need them?

- Git & Github are not the same thing
  - Git - version control system
  - Github - cloud based service

- Git - tracking and manages different versions of files/ repositories over time
- Github - stores git repositories in cloud (you can share your code with anyone!)
  - hosts websites for free

Visualisation of Git

infpals.github.io
*go there and try to imagine how was it done



General workflow

WARNING:
Do Not initialise git in folders which is already being tracked by git.

WARNING:
Do not initialise git at the root folders of your computer

If your git is not initialized

1. **Open Terminal**
2. **Navigate to folder where you want to track your work**
3. **git status (Checking if there is not already a git initialised)**
4. **Initialise git at your folder with git init**

OPTIONAL: git status

What Commands are we going to learn?

git status
git init
git add
git commit

git log
git diff
git clone
git push

We hope you are having fun! ;)

Let's open the terminal!

Check your version of git

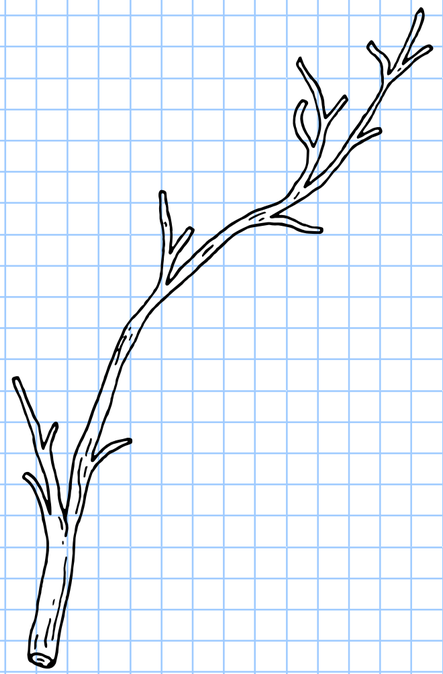
git --version

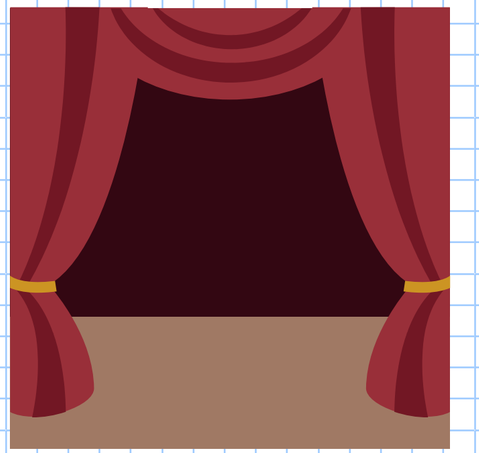
Link to download Git:

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

git status

Prints the current status of repository





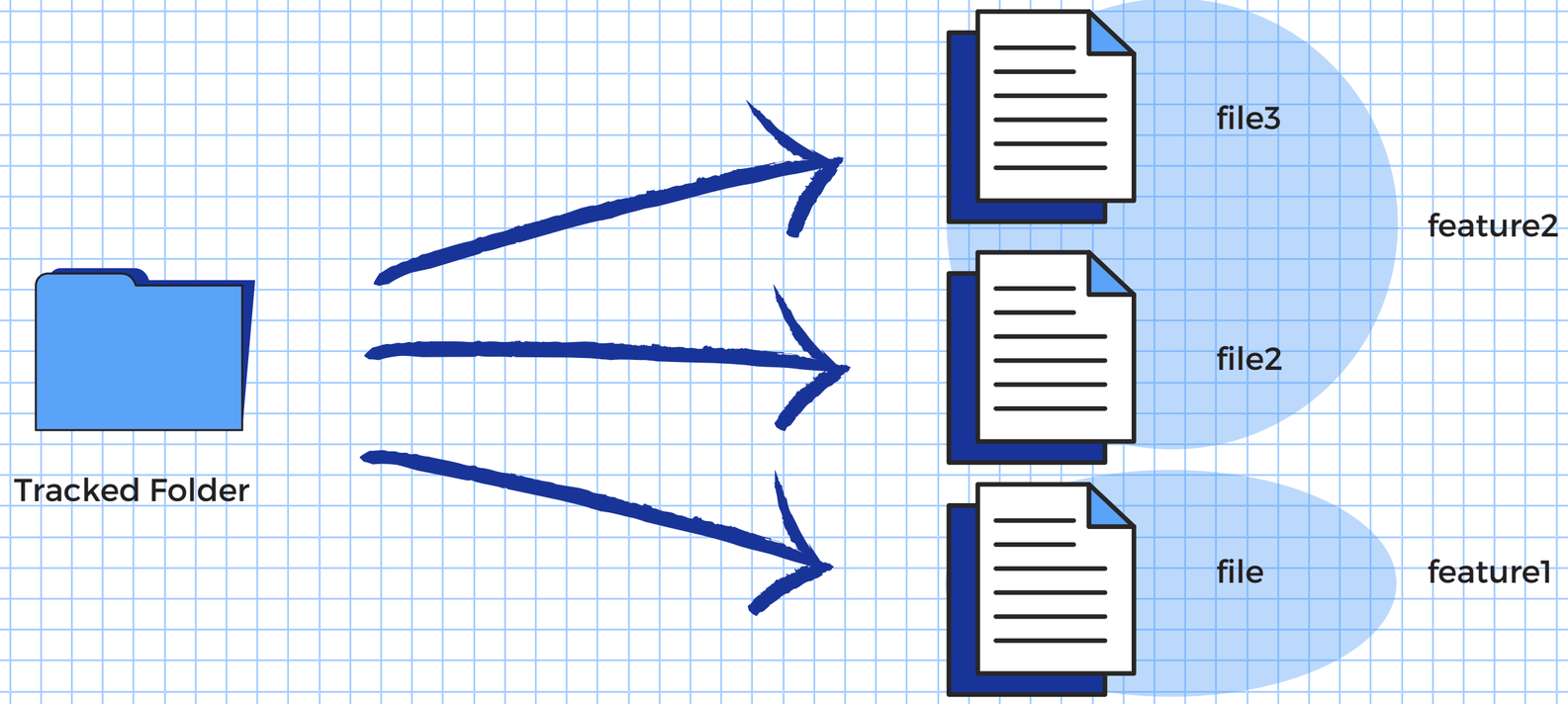
Files to be commited

git init

Initialises git (tracker of your changes) at a given folder

git add [[file] or .]

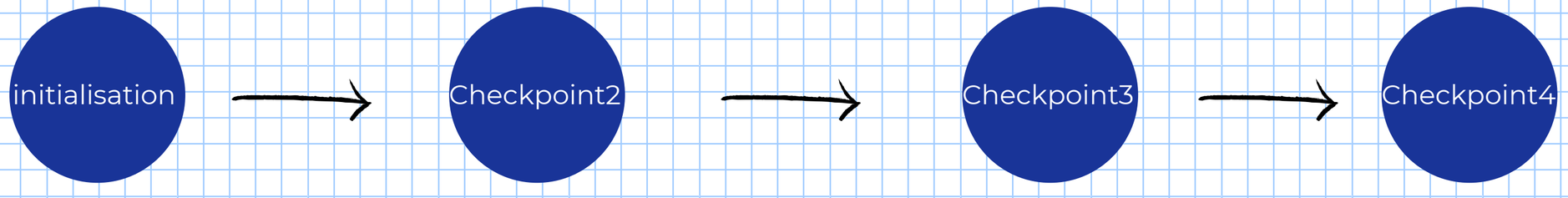
Stages the files with changes / newly added



Optional: git reset <file>

git commit -m '[message]'

Commits/saves staged changes



Note: If you do not use -m (do not add message) you will be directed to Vim. Here is a link to article which explains how to get out of it: https://www.cyberciti.biz/faq/linux-unix-exit-vim-editor/

git log

Prints ordered commits with given message / author / date / hash

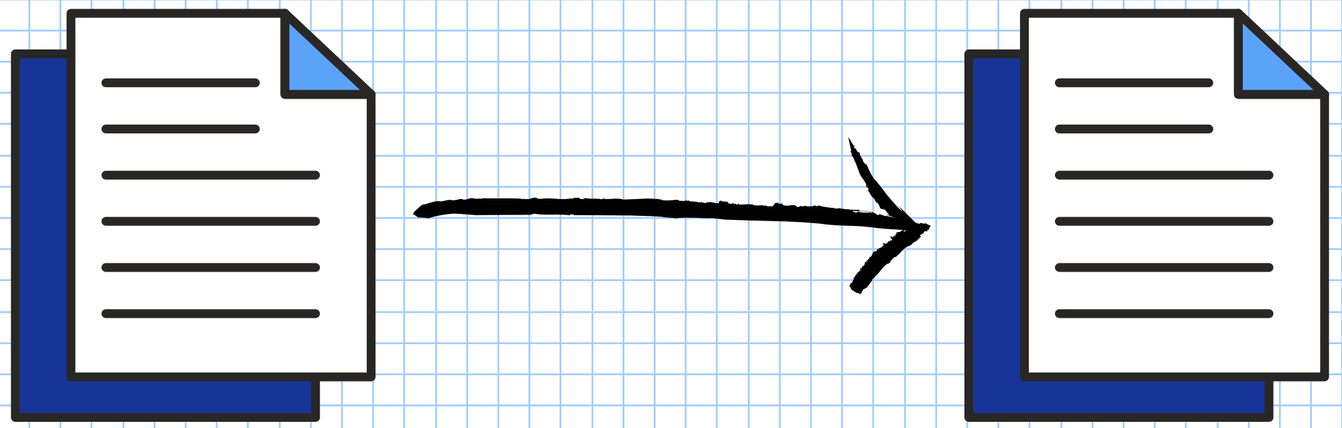
hash

(don't know how to visualise)

Optional: git log --oneline
Note: Short hash can be used for git diff

git diff [HEAD or commit1..commit2 or branch1..branch or nothing]

Prints differences between commits, branches, files ...



version 1.1

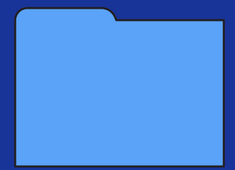
version 3.2

We hope you are having fun! ;)

Let's open github!

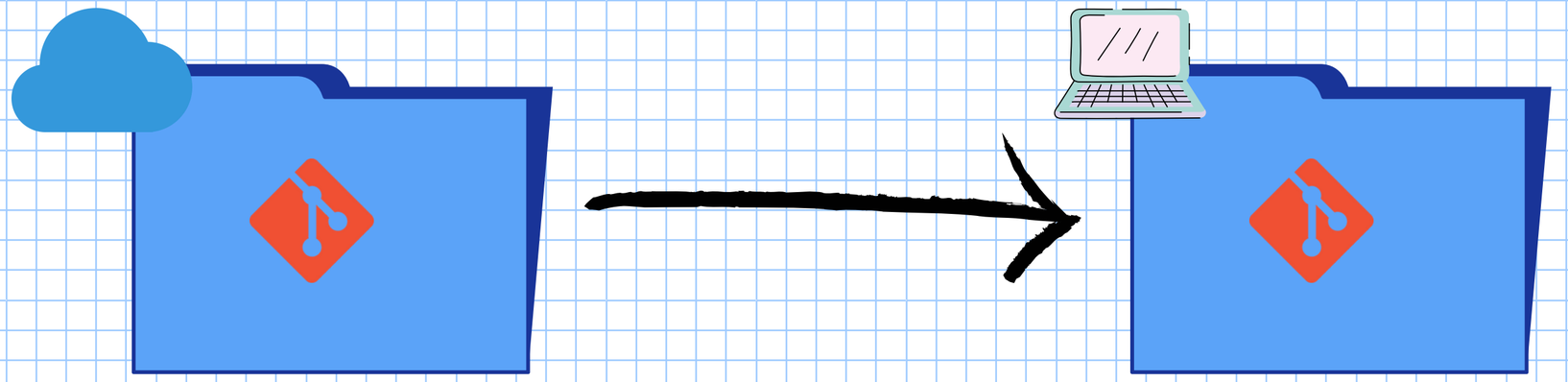
https://github.com/

If you do not have account, now its the time to set it up :)



git clone [HTTPS or SSH]

Clones a git repository (folder) into your local computer with all the historical git data

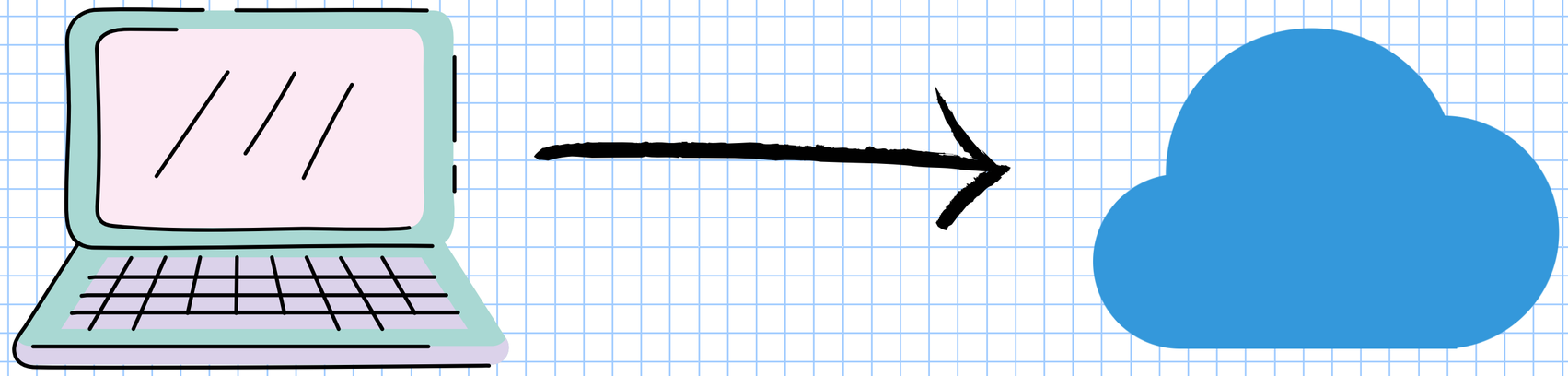


Cloud (Github)

Local (your) computer

git push

Pushes all historical git data into the cloud (github)

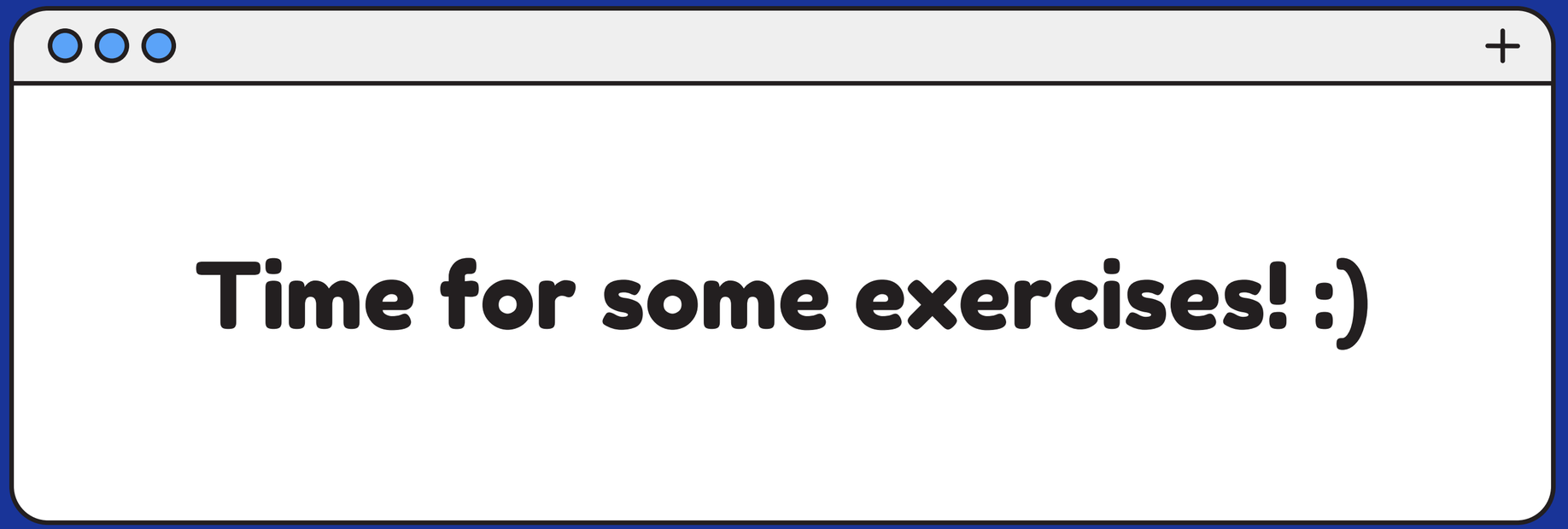


Local (your) computer

Cloud (Github)

Amazing visualisation tool

https://learngitbranching.js.org/


Time for some exercises! :)

Try to do things listed in the window!

If you have any problems don't hesitate to raise your hand and the Leader will come and help you. :)

1. Open Terminal and navigate yourself to your Desktop
2. Create Folder PracticeGit
3. Check the Status of Git in PracticeGit
4. Initialise Git in PracticeGit
5. Create a file Me.txt and do initial commit with message 'initial commit'
6. Write 'Hello! :) My name is [your name].' into the Me.txt and do another commit with message 'add my name'
7. Write 'I study [name of your degree]' into the Me.txt, create Colors.txt and do another commit with message 'add my degree' where you only commit changes in Me.txt
8. Do commit for Colors.txt with message 'add Colors.txt file'
9. Print all commits/author/dates/hashes
10. Print difference between initial commit and last commit
11. Create private repository on Gihub Hello-World
12. Establish connections between PraticeGit and Hello-World
13. Write your 3 favourite colors and commit changes with message 'add my 3 favourite colors' and push changes into Github repository
14. Clone Materials from : https://github.com/infpals/ip2022-git-and-github-basic into your local computer (somewhere where you do not have already initialised git)
15. Print commits done in ip2022-git-and-github-basic