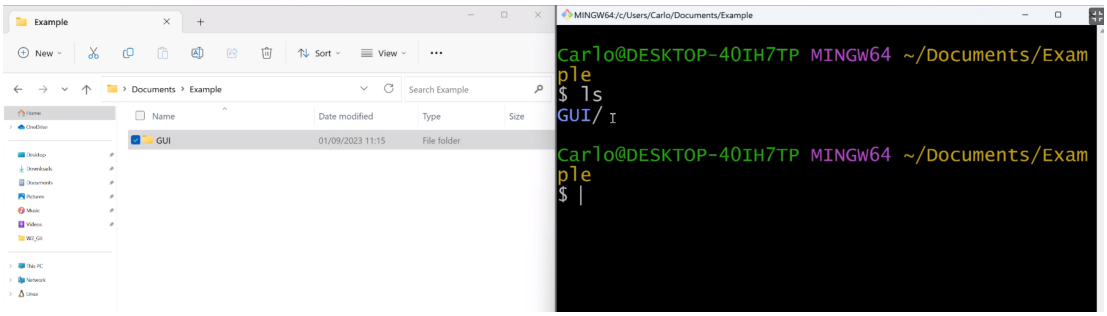
Lecture Week 2

CLI, Git, & GitHub

- basic commands in CLI
- what is Git
- what is GitHub
- learn how to submit your excerise in GitHub classes.

2.1 What is a GUI and CLI?

- GUI: Graphical User Interface
- CLI: Command-Line Interface



GUI: Graphical User Interface	CLI: Command-Line Interface
Visual interface	Text-based interface
Suitable for beginners	Preferred by advanced users
Simple and intuitive	Requires command knowledge
	Precise control and flexibility

What advantages does it have to use the CLI?

- **Speed**:
  - The console allows for quick execution of commands
  - making it efficient for performing tasks rapidly.
- **Scripting and automation**:
  - Console commands can be combined into scripts,
  - automating repetitive tasks and saving time.
- **Remote management**:
  - The console enables remote access and management of systems,
  - which can be particularly useful for servers or headless machines (without GUI).
- **Reproducibility**:
  - Console commands can be easily documented and shared,

- ensuring consistent results across different systems or users.

Disadvantages?

- Steeper learning curve
- Lack of visual representation
- Command memorization

Commands example

You can find the example video here (12:11min).

Command	Description
`ls`	List files and directories in the current directory
`cd <directory>`	Change directory to the specified directory
`pwd`	Print the current working directory
`mkdir <directory>`	Create a new directory
`cp <source> <destination>`	Copy files or directories
`mv <source> <destination>`	Move or rename files or directories
`rm <file>` or `rm -r <dir>`	Remove files or directories (use `-r` for recursive deletion)
`cat <file>`	Display the contents of a file
`grep <pattern> <file>`	Search for a pattern in a file
`top` or `htop`	Display real-time system resource usage
`man <command>`	Display the manual page for a command
`history`	View a list of previously executed commands
`ctrl + c`	Interrupt or terminate the current command or process
`touch <file>`	Creates a new file

Terminal / Console

- **Linux**
  - Terminal / Terminator
- **Windows**
  - Bash Shell / Git Bash (Recommended)
- **Mac**
  - Terminal

2.2 Git

- Who already worked with Git?
- For what did you use it?

How does it look like?

- previously:
  - `main.py, main_2.py, main_3_notworking.py, main_final.py, main_final_final.py`

Small Overview

- previously:
  - `main.py, main_2.py, main_3_notworking.py, main_final.py, main_final_final.py`

- **Git**
  - version control system for tracking software changes
  - efficiently tracks code history and allows easy reverting to previous versions
  - supports branching for independent feature development
  - facilitates merging of code changes back into the main codebase
  - provides command-line and GUI (e.g. git-scm.com, gitkraken.com, VS Studio)

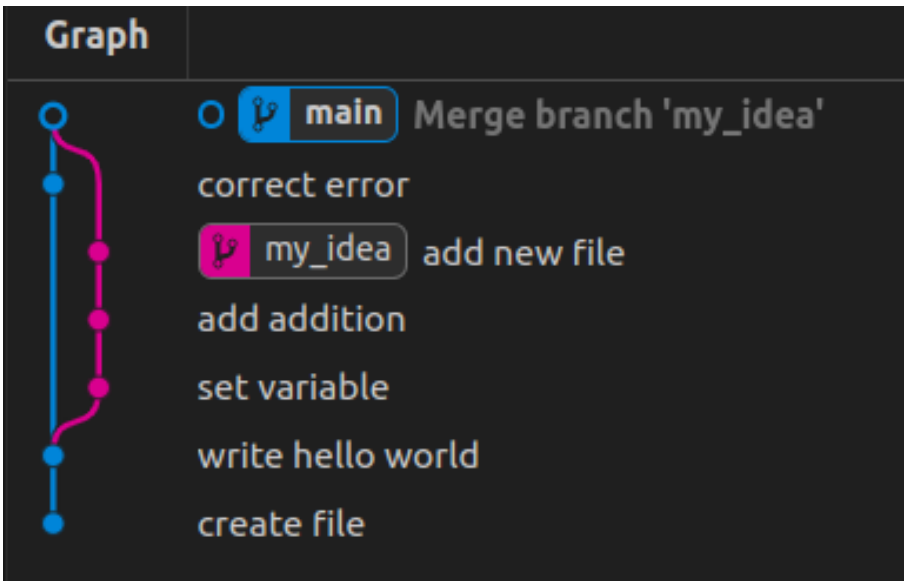


Image: VS using Git Graph

Command	Description
`git init`	Initializes a new Git repository
`git add .`	Adds all changes the staging area
`git commit -m "[message]"`	Commits changes with a descriptive message
`git checkout -b [branch`	create new branch and switch to it
`git checkout [branch]`	Switches to a different branch
`git branch`	Lists branches in the repository
`git merge [branch]`	Merges [branch] into the current branch
`git status`	Displays the current repository status
`git log`	Shows the commit history

Hint: https://www.w3schools.com/git/

See example here (13:16min).

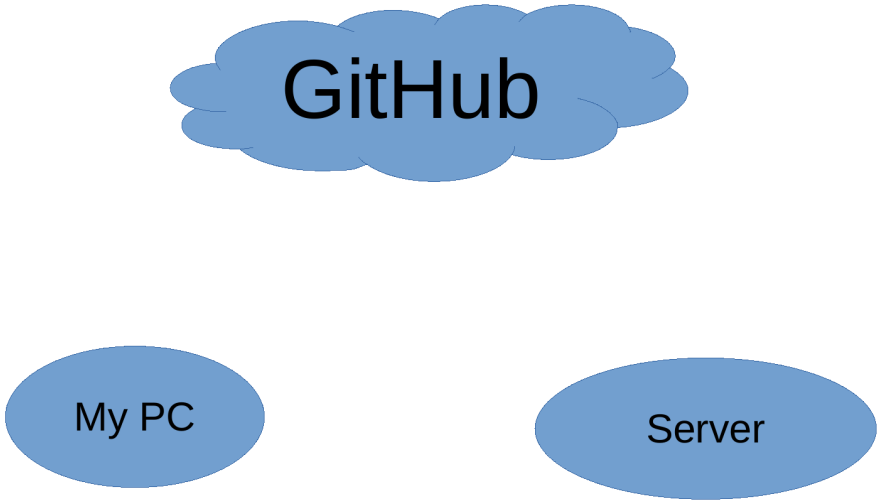
2.3 GitHub / GitLab

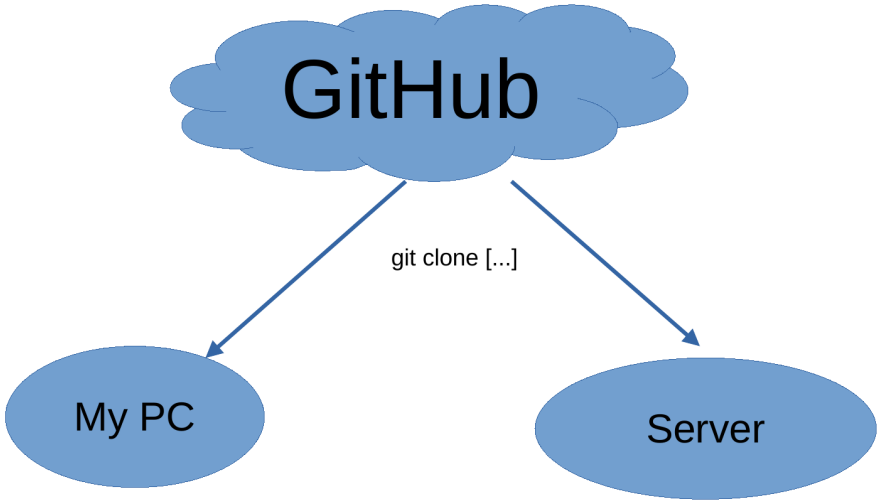
Why do you use GitHub / GitLab?

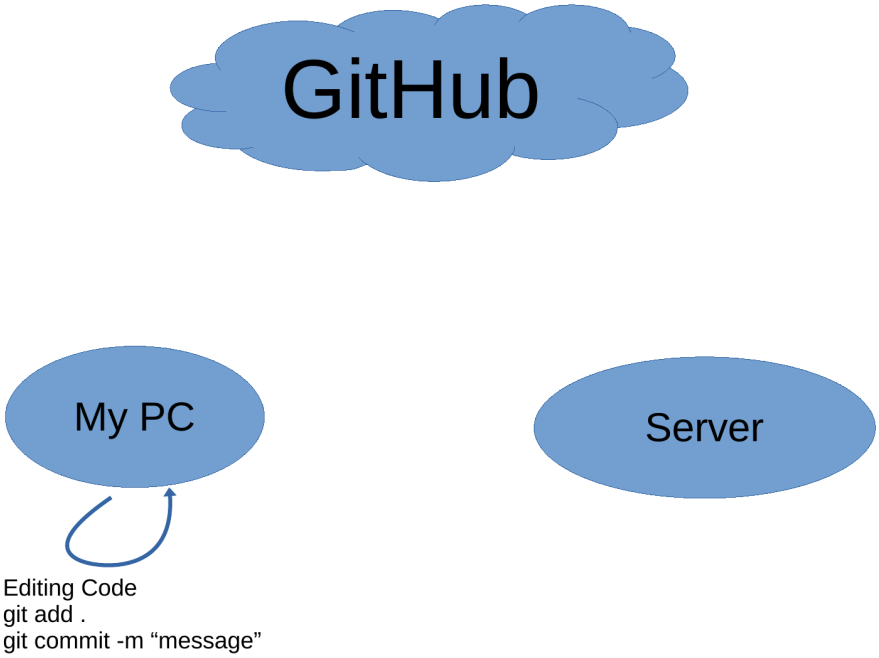
- share code with other people (open source)
  - get code from other people
- save your code on the cloud
  - access your data from different computers
- develeop your program with different developers

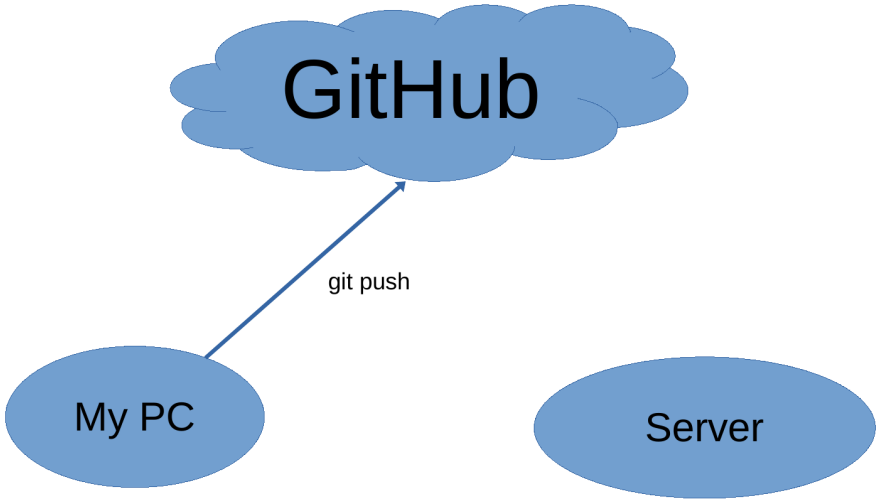
How to use it for yourself?

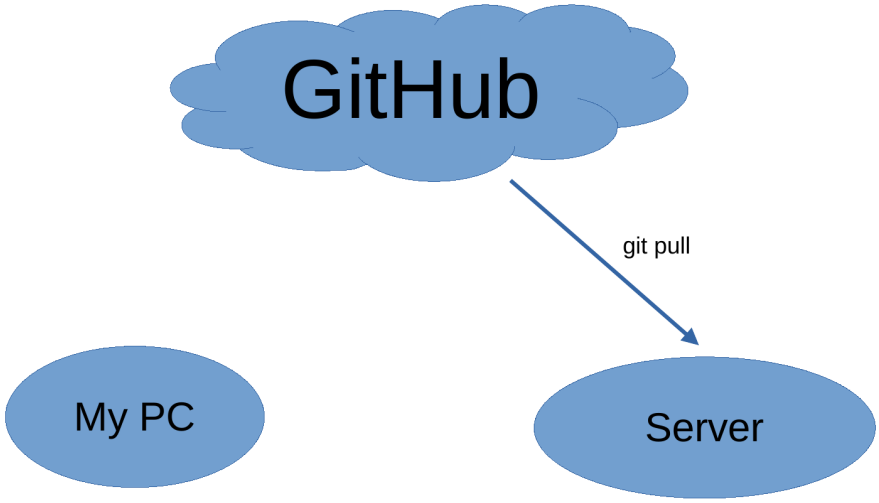
Command	Description
`git clone [repository]`	Creates a local copy of a remote repository
`git pull`	Fetches and merges changes from a remote repository
`git push`	Pushes local commits to a remote repository



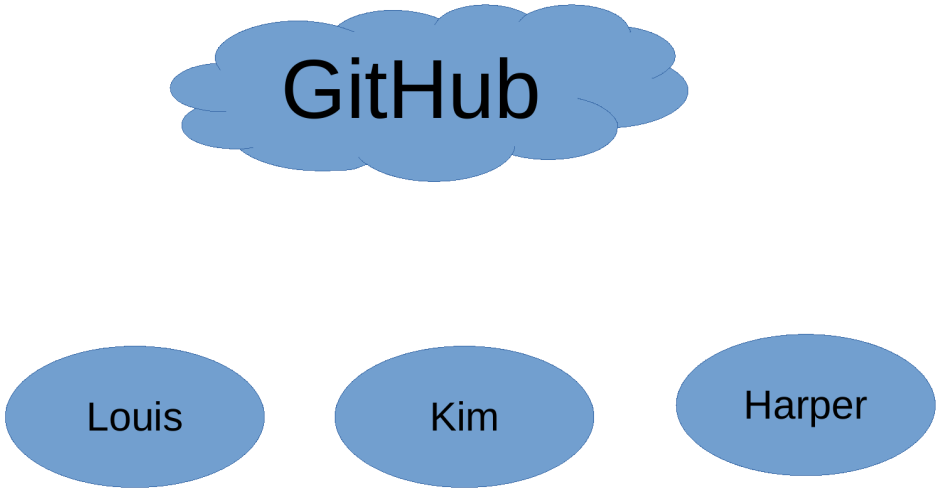

GitHub
git clone [...]
My PC
Server

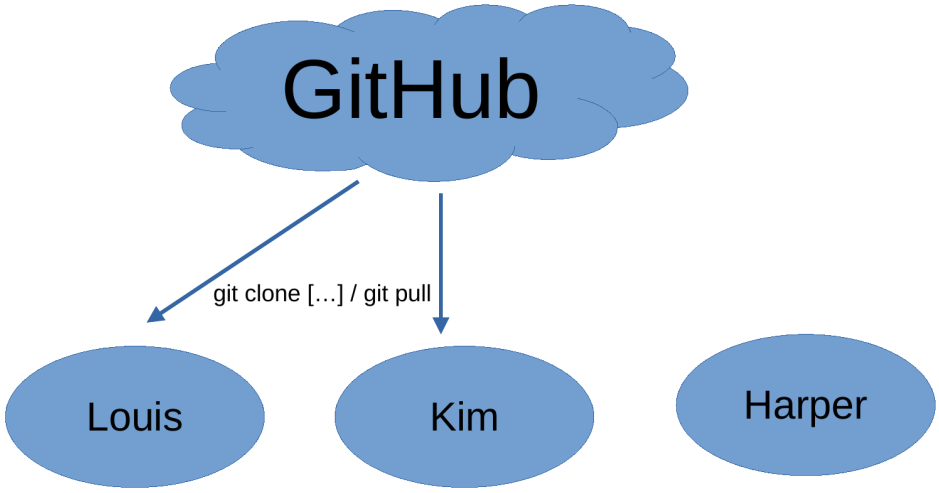

GitHub
My PC
Server
Editing Code
git add .
git commit -m “message”

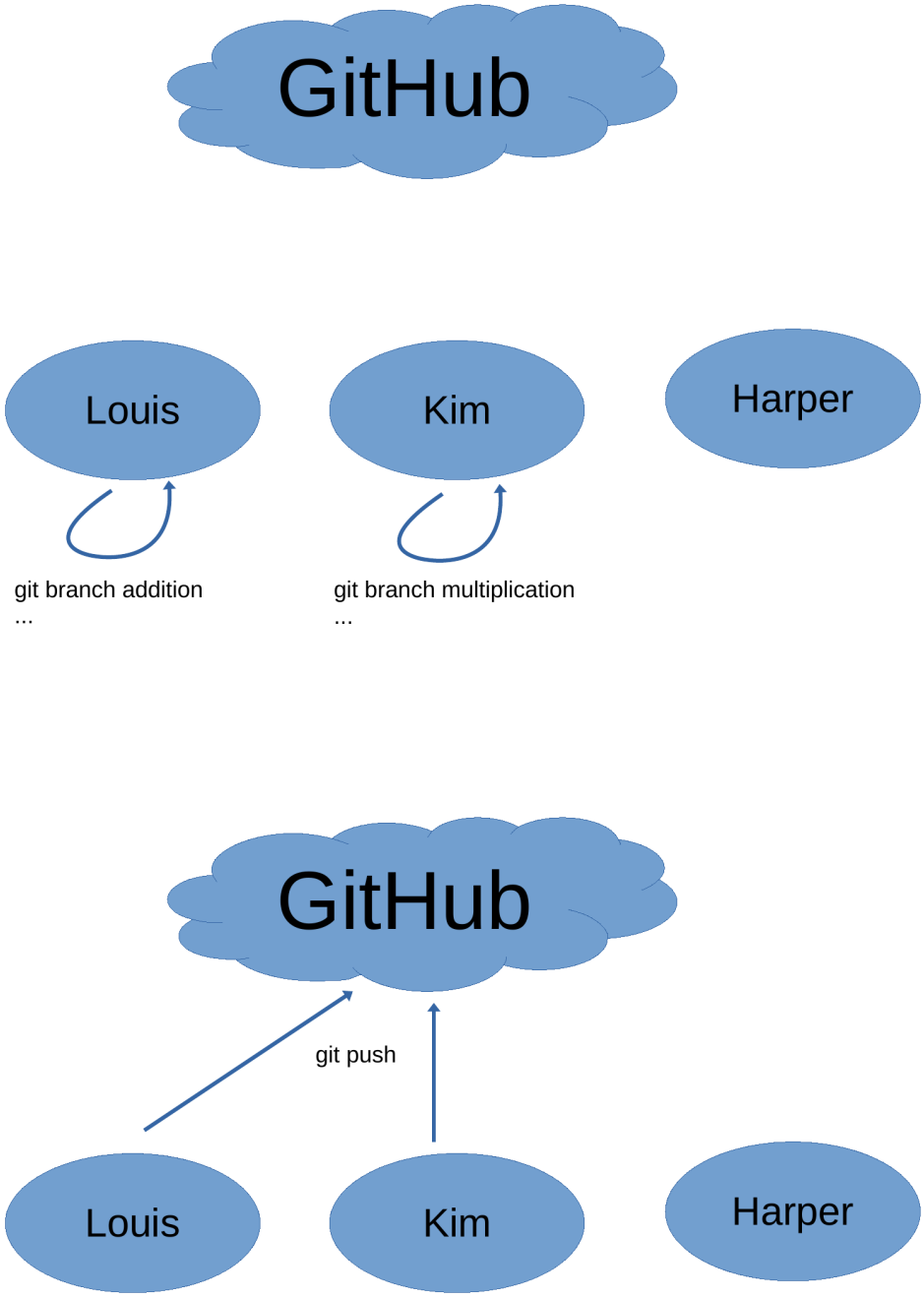




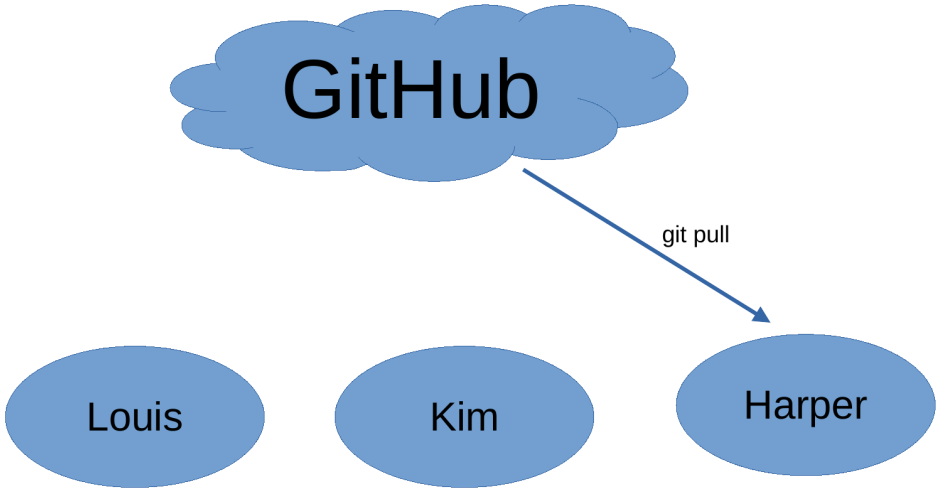
How to use it with other developers?

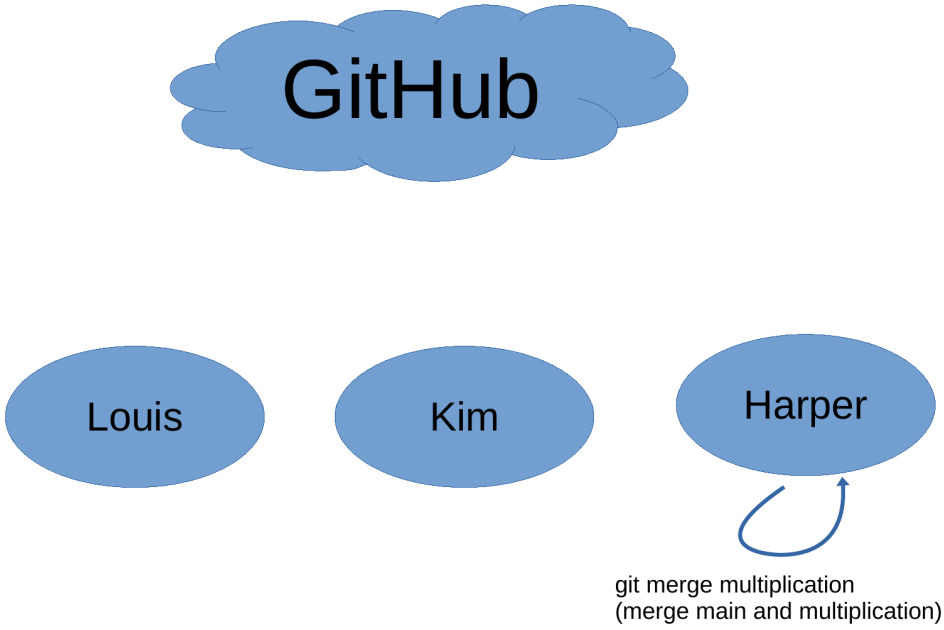

GitHub
Louis
Kim
Harper

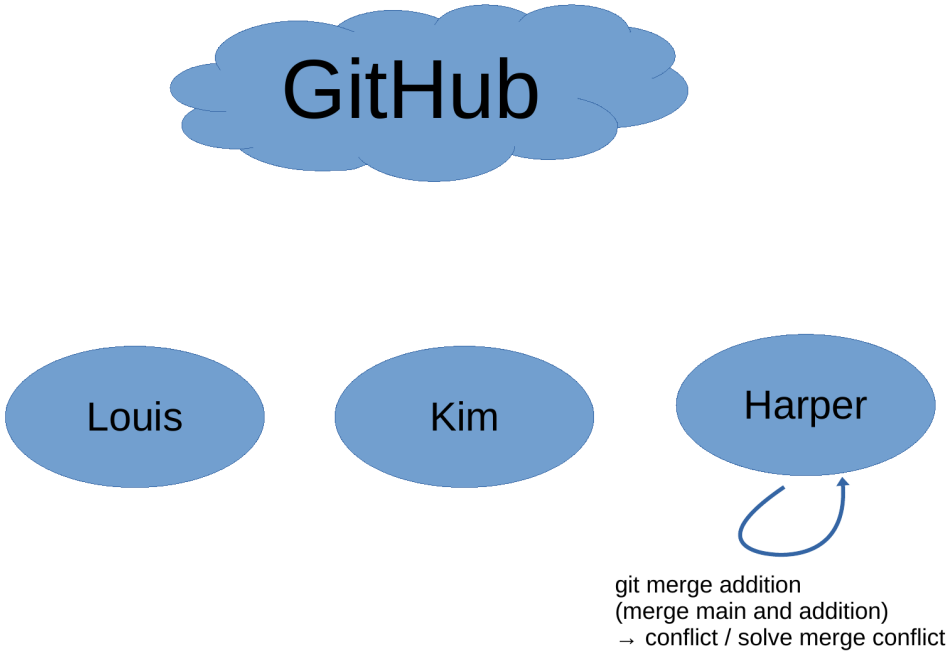

GitHub
git clone […] / git pull
Louis
Kim
Harper

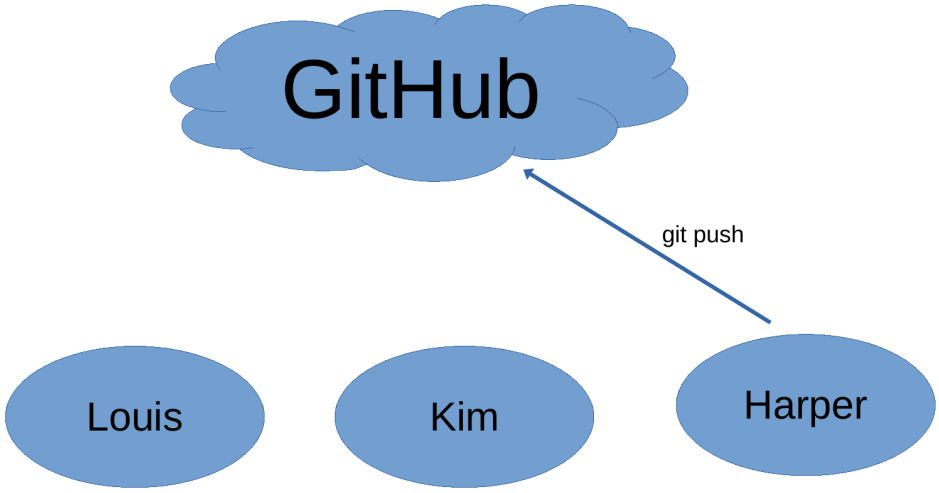

GitHub
Louis
Kim
Harper
git branch addition
...
git branch multiplication
...

GitHub
git push
Louis
Kim
Harper


GitHub
git pull
Louis
Kim
Harper


GitHub
Louis
Kim
Harper
git merge multiplication
(merge main and multiplication)





Remark

- you will hand in your excersie via GitHub
- important: set up everything so you can use it
  - install Git
  - set up SSH Key (4:26min)
  - register for the GitHub class and clone your repository on your computer (example (5:10min))

    - see your points on the automated grading system (under Actions, see example (0:49min))

1.4 Examples Summary

- Console / Terminal (12:11min)
- Git (13:16min)
- set up SSH Key (4:26min)
- GitHub Classes (5:10min)
- Automated Grading System (0:49min)