CSE & INSO Collaboration

Meeting 2 – Advanced git and starting a Java Lab Repository

Goals

- Keep working with the new software
- Understanding the git workflow
- Creating the Java Lab Repository structure

Program overview

- Recap of last meeting
- Advanced git
- Live Demo together
- Issues
- Java Lab Repository

Tools

- Zoom
- GitHub Desktop and GitHub Account
- IntelliJ Community
- (TeamViewer)

Recap of the git vocabulary

- Repository
- Clone
- Add
- Commit
- Push
- Pull

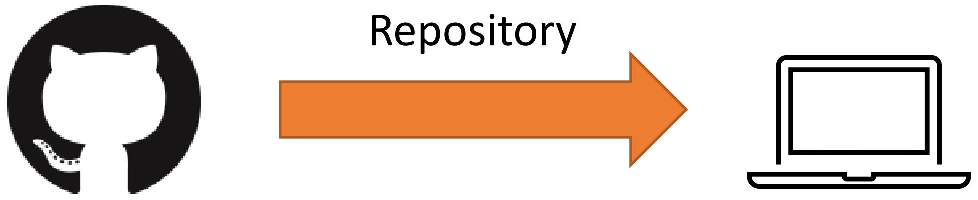
Repository

- Rough explanation: contains all your data (source code files) and their versions (history)
- Remote repository -> on the server
- Local repository -> on the client

git clone

- Clones (creates a copy) a repository from the server to the client (laptop, desktop computer, smartphone, …)

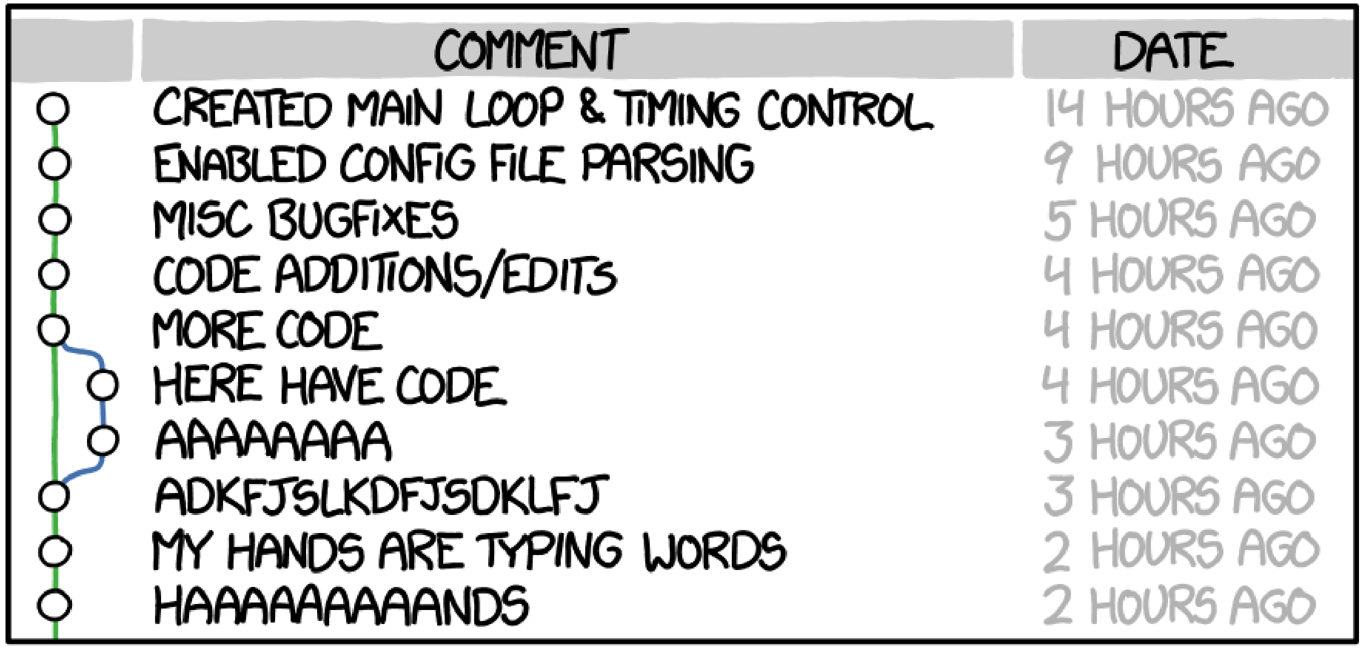
```
git clone https://github.com/RIT-at-SSE/git_tutorial
```



git commit

- Makes (changed/added/removed) files ready on the local repository to push to the remote repository.
- Adds a message to those files for explaining collaborators (your colleagues) what was changed/implemented/fixed

```
git commit -m "short and
precise commit message"
```



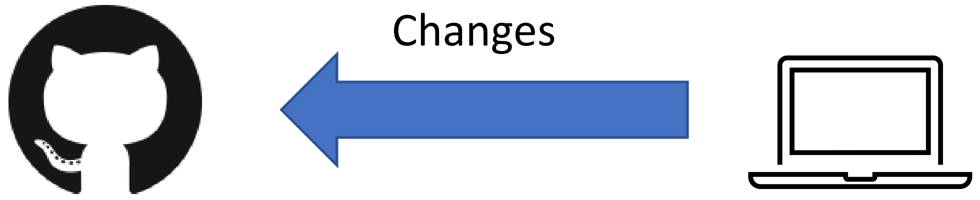
	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

git push

- Pushes (synchronizes) all changes of the current branch from the local repository to the remote repository

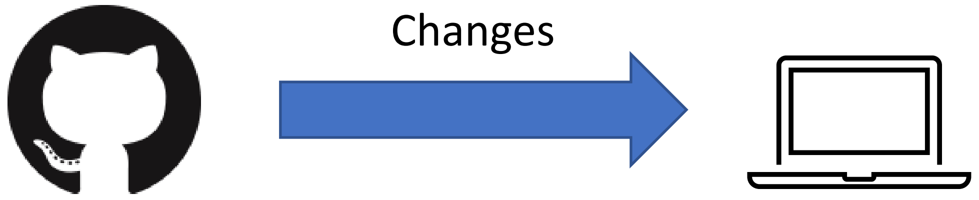
```
git push
```



git pull

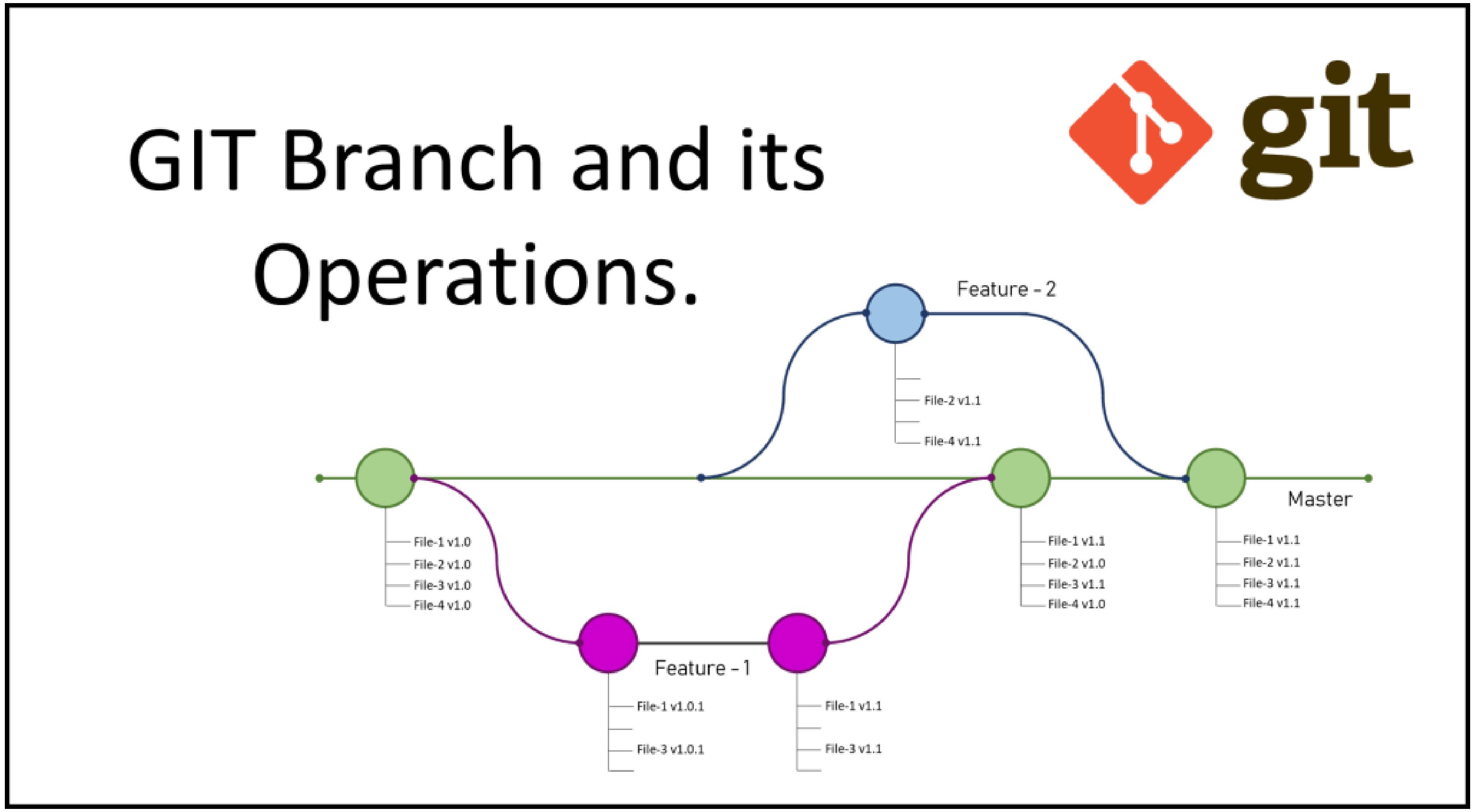
- Pulls (synchronizes) all changes of the current branch from the remote repository to the local repository

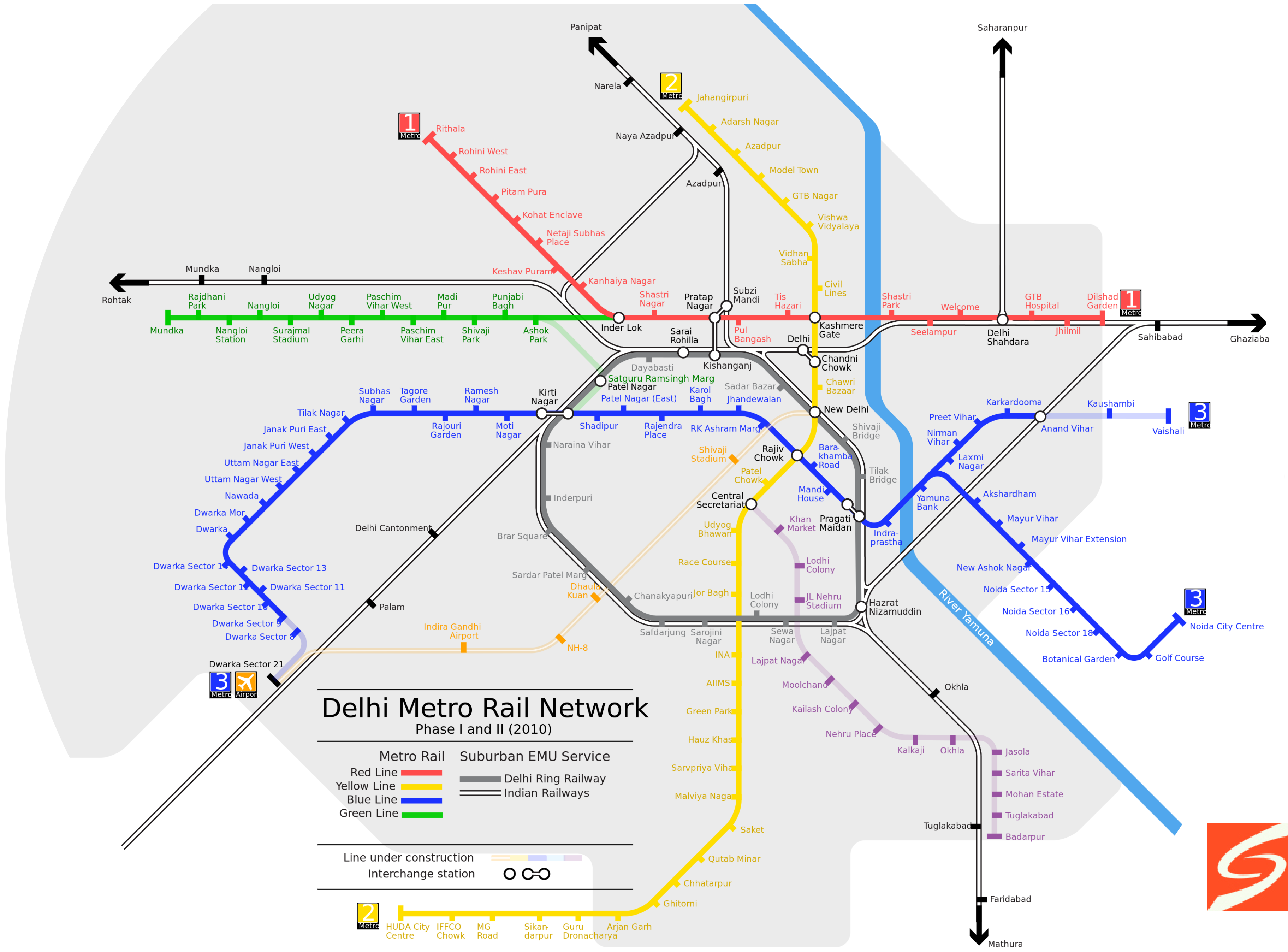
```
git pull
```



Advanced git

- **Branch**
- **Merge**




Delhi Metro Rail Network
Phase I and II (2010)
Metro Rail
Red Line
Yellow Line
Blue Line
Green Line
Suburban EMU Service
Delhi Ring Railway
Indian Railways
Line under construction
Interchange station
Panipat
Narela
Naya Azadpur
Azadpur
Saharanpur
Rohtak
Mundka
Nangloi
Sahibabad
Ghaziaba
Delhi Cantonment
Palam
Okhla
Tuglakabad
Faridabad
Mathura
River Yamuna
Rithala
Rohini West
Rohini East
Pitam Pura
Kohat Enclave
Netaji Subhas Place
Keshav Puram
Kanhaiya Nagar
Inder Lok
Shastri Nagar
Pratap Nagar
Subzi Mandi
Pul Bangash
Tis Hazari
Kashmere Gate
Shastri Park
Seelampur
Welcome
Delhi Shahdara
GTB Hospital
Jhilmil
Dilshad Garden
Jahangirpuri
Adarsh Nagar
Azadpur
Model Town
GTB Nagar
Vishwa Vidyalaya
Vidhan Sabha
Civil Lines
Chandni Chowk
Chawri Bazaar
New Delhi
Rajiv Chowk
Patel Chowk
Central Secretariat
Udyog Bhawan
Race Course
Jor Bagh
INA
AIIMS
Green Park
Hauz Khas
Malviya Nagar
Sarvpriya Vihar
Saket
Qutab Minar
Chhatarpur
Ghitorni
Arjan Garh
Guru Dronacharya
Sikan darpur
MG Road
IFFCO Chowk
HUDA City Centre
Mundka
Rajdhani Park
Nangloi Station
Nangloi
Surajmal Stadium
Udyog Nagar
Peera Garhi
Paschim Vihar West
Paschim Vihar East
Madi Pur
Shivaji Park
Punjabi Bagh
Ashok Park
Satguru Ramsingh Marg
Dwarka Sector 21
Dwarka Sector 8
Dwarka Sector 9
Dwarka Sector 10
Dwarka Sector 11
Dwarka Sector 12
Dwarka Sector 13
Dwarka Sector 14
Dwarka
Dwarka Mor
Nawada
Uttam Nagar West
Uttam Nagar East
Janak Puri West
Janak Puri East
Tilak Nagar
Subhas Nagar
Tagore Garden
Rajouri Garden
Ramesh Nagar
Moti Nagar
Kirti Nagar
Shadipur
Patel Nagar
Patel Nagar (East)
Rajendra Place
Karol Bagh
Jhandewalan
RK Ashram Marg
Barakhamba Road
Mandi House
Pragati Maidan
Indraprastha
Yamuna Bank
Laxmi Nagar
Nirman Vihar
Preet Vihar
Karkardooma
Anand Vihar
Kaushambi
Vaishali
Akshardham
Mayur Vihar
Mayur Vihar Extension
New Ashok Nagar
Noida Sector 15
Noida Sector 16
Noida Sector 18
Botanical Garden
Golf Course
Noida City Centre
Sarai Rohilla
Dayabasti
Kishanganj
Delhi
Sadar Bazar
Shivaji Bridge
Tilak Bridge
Hazrat Nizamuddin
Naraina Vihar
Inderpuri
Brar Square
Sardar Patel Marg
Chanakyapuri
Safdarjung
Sarojini Nagar
Lodhi Colony
Sewa Nagar
Lajpat Nagar
Shivaji Stadium
Dhaula Kuan
Indira Gandhi Airport
NH-8
Khan Market
Lodhi Colony
JL Nehru Stadium
Lajpat Nagar
Moolchand
Kailash Colony
Nehru Place
Kalkaji
Okhla
Jasola
Sarita Vihar
Mohan Estate
Tuglakabad
Badarpur
1 Metro
2 Metro
3 Metro
Airport

git branch

- List all local branches

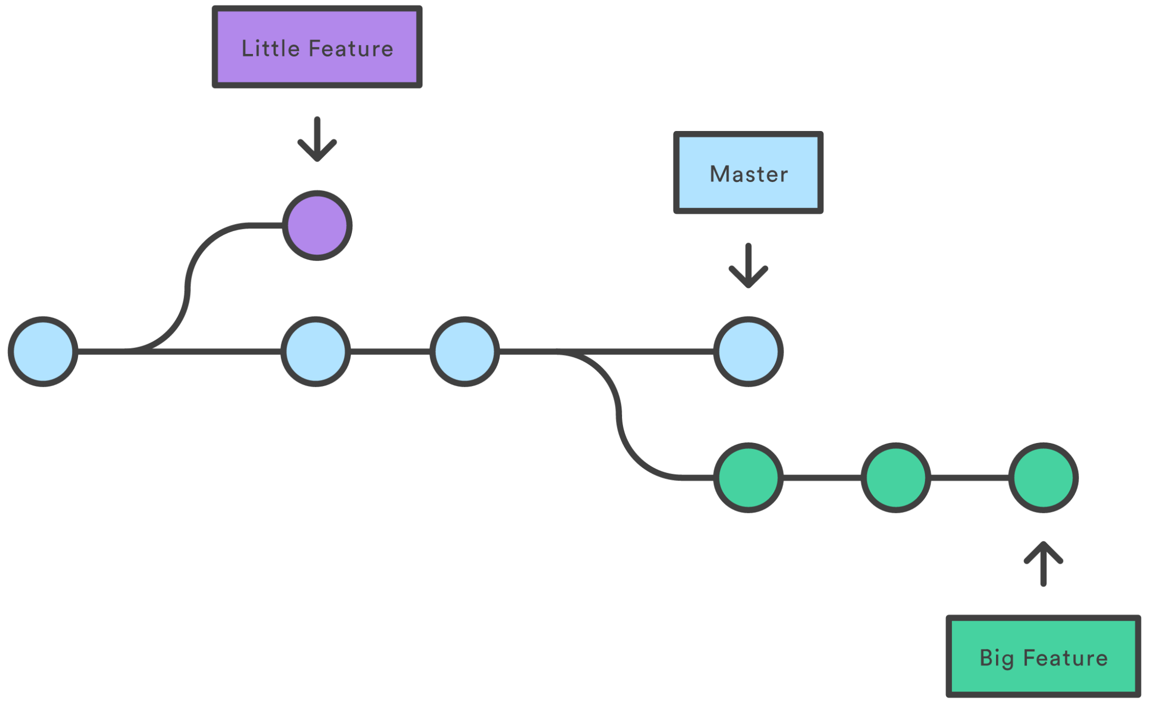
```
git branch
```

- List all remote branches

```
git branch -r
```

- List all branches

```
git branch -a
```



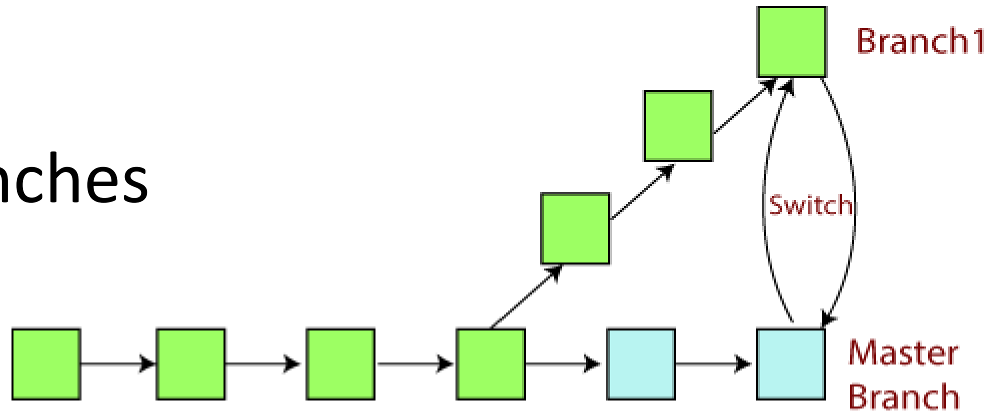
git checkout

- Create a new branch

```
git checkout -b "branch-name"
```

- Switch to a branch / switch between branches

```
git checkout "branch-name"
```



Git Checkout

git merge

- Merge one branch into the current branch

- Switch to the branch into which you want to merge

```
git checkout “to-branch”
```

- Merge the branch into the current branch

```
git merge “from-branch”
```

Live Demo together

Let’s try it out

Tutorial 1:

- Branching and Merging in GitHub

1. Create your own feature branch
2. Make at least 3 commits to your branch
    1. Add a file called <YourName>.md -> commit
    2. Add information about yourself to the file -> commit
    3. Commit any picture to your branch
3. Checkout the main branch -> your changes should not be visible now

Be sure to include helpful commit messages!

Branching and Merging

- Create a branch for each feature you want do develop
- Implement the feature together with your colleagues/collaborators
- Merge the feature to the main/master branch when finished implementing
- Often there is also a dev/development branch.

Dos and Don’ts when branching and merging

- Never ever (4 real) push directly to the main/master branch.
- The main/master branch should be a stable version of your software at ALL time after checkout/cloning.
- Use branches for implementing features.

Pull Request / Merge Request

- Merging on GitHub
- After your implementation on a branch is finalized -> open a PR (Pull Request)
- After PR was reviewed, merge the changes to the main/master branch

Live Demo together

Let’s try it out

Tutorial 2:

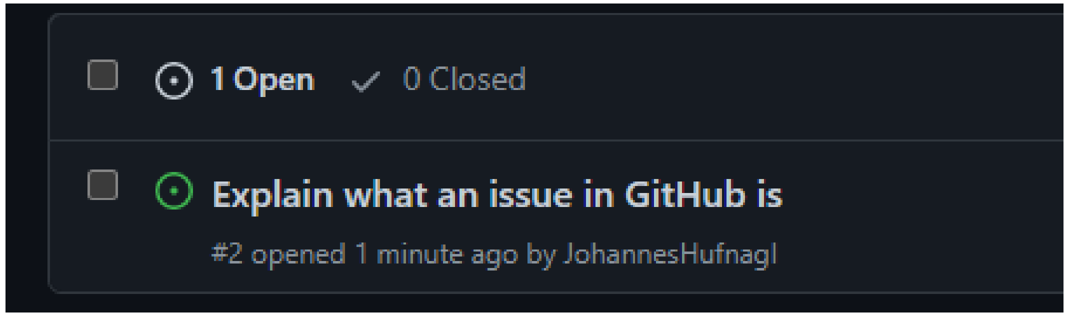
- Create a Pull request and let someone else review it

1. Checkout your own branch again
2. Create a Pull Request
3. Assign someone to review the Pull Request
4. Review your assigned Pull Request
5. Merge

Issues

- To Do's for your project
- Can be assigned to team members

https://github.com/RIT-at-SSE/git_tutorial/issues



1 Open ✓ 0 Closed

Explain what an issue in GitHub is

#2 opened 1 minute ago by JohannesHufnagl

Takeaways

- Recap of git basics
- git branching and merging
- Issues and Pull requests

Next time

- Working with the Java Lab Repository
- Looking at an example implementation of the week 1 of Java Lab
- Collaborative implementing week 2 of Java Lab

More information / useful ressources available here:

- https://github.com/skills/introduction-to-github
- https://www.toptal.com/developers/gitignore/
- https://www.w3schools.com/git/default.asp?remote=github
- https://docs.github.com/en/desktop
- YouTube is a great resource

Credits

- Raimund Rittnauer
- https://www.w3schools.com/git/default.asp?remote=github

- https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html