

## Notes for the tutorial

- Spend a lot of time on workflow
- Spend some time on setup
- ~~Create a disposable email addr.~~ ✓
- Do everything in front of them
- Include the "Plan"
- Explain that Git is "hard to grasp" because of a "new" way of thinking about digital projects



- Explain "What is Git" diagram
    - "Master" = any computer, e.g. GitHub
    - "Your Work" = your pc
    - "S. Elses Work" = Grigori's pc
- (They should imagine this)

↓ continue

- Explain why Git / Linux is big deal
- Mention that Linux has a fantastic / passionate community
- Git shares / inherits Linux - philosophy
- Even if Microsoft / Apple / Google disappear, Git will stay
- Explain open source
- Mention that Linus used his experience from BitKeeper to design Git (best in the world)
- ~~Commit message: capital + action~~ ✓

- Mention that branches can diverge / fork without ever reuniting

## Plan : technical part

- Explain work flow (long)
- Explain setup (short)
- Practice setup (at least once)
  - Create account
  - Create repository
  - Clone repository
  - Add (stage?)
  - Commit
  - Push
- Practice workflow (at least once)
  - Clone my repository
  - Make changes + commit
  - Try push ←
  - Success: done
  - Not up to date: pull → merge

# Plan: non-technical part

## • History

- 1991 - 2002

Linux developed by passing files around

- 2002 - 2005

Community adopts proprietary version control system

- 2005

Company asks for money  
Linus makes Git

## • Philosophy


- Safe

- Fast

- Powerful (merge, history, ...)

- Distributed

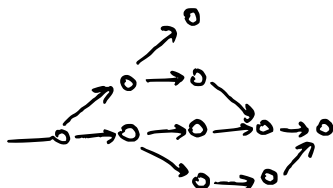
## Workflow

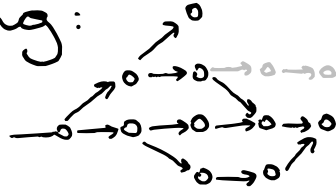
- Clone or pull
  - Make changes
  - Stage changes
  - Commit
  - Try push
  - Maybe merge (auto or manual)
- 

## Terms

- Clone = download 1-st time
- Pull = update your branch
- Stage = select which changes to commit = Add
- Commit = create a permanent snapshot of project
- Push = make another "agent" pull your branch
- Merge = duh

## Deeper terms

- Checkout = Jump : 
- Pull request = Ask "agent" to pull
- Pull = Fetch + Merge
- Fetch = Just download other's work (branch) without merging:



- Linus Torvalds = genius/god

## Best practices

- Small frequent commits
- Commit messages:
  - Short title (<50 characters)
  - Short body (<100 characters) (optional)
  - Descriptive
  - Imperative mood
  - Capital + no period
  - Good examples:
    - "Add logging"
    - "Simplify input parsing"
    - "Remove dependency on X"
  - Bad examples:
    - "Added logging"
    - "I made some changes"
    - "Debugging"



## To do

- ~~Switch screen~~ ✓
- ~~Check "slides"~~ ✓
- ~~Read notes~~ ✓
- Transfer "slides" to pc
- ~~Commit to repo~~ ✓
- ~~Add xkcd commits~~ ✓
- ~~Open Google forms~~ ✓

