



Gitting gud

Presented by the terminal master™

Collaborator

- Kom even langs mijn pc om je gebruikersnaam van GitHub in te vullen

Warning

- Gebruik de terminal (Git bash, terminal.app, iTerm, whatever)
- Als ik iemand een GUI zie gebruiken, slinger ik je laptop uit het raam



1: Cloning

- Kopieer een "remote" repo naar jouw "lokale" pc
- Remember: git repo = folder
- Remote = repo = folder
- Local = repo = folder

1: Cloning

```
$ cd /path/to/folder
```

```
$ git clone [url]
```

1: Cloning

```
$ git clone
```

```
https://github.com/Lucky-  
Loek/git-gud.git
```

1: ~~Cloning~~ forking

- Als iemand je niet als collaborator heeft toegevoegd
- Letterlijk een clone command, alleen dan aan GitHub's kant

2: Making changes

- Voeg je naam toe aan de README

3: Staging changes

```
$ git status
```

3: Staging changes

- Wat zie je?

3: Staging changes

```
$ git add README.md
```

```
$ git status
```

3: Staging changes

- Wat zie je?

4: Committing changes

```
$ git commit -m "Add name to  
README"
```

```
$ git status
```

```
$ git log
```

4: Committing changes

- Wat zie je?

4: Committing changes

- Commit altijd in tegenwoordige tijd
- Nooit langer dan 80 karakters (waarom?)
- Beschrijf wat je hebt gedaan, niet blablabla commit of zoidontknow

4: Committing changes

- Add -> voor toevoegingen
- Fix -> voor bugfixes
- Update -> voor updates
- Change -> voor veranderingen
- Remove -> voor verwijderingen

5: *Let the hunger games begin*

```
$ git push
```

5: The hunger games

- Wie was eerst?
- Waarom "doet hij het niet"?

5: The hunger games

- Er zijn twee waarheden

5: The hunger games

- Trust me: je wilt dat "hij het niet doet"

6: History was found on conflict

```
$ git pull
```

```
$ git status
```

```
$ git diff
```

6: Paniek paniek paniek paniek paniek paniek

- Wat gebeurt er?

6: Paniek paniek paniek paniek paniek paniek

- Er zijn twee waarheden

6: Paniek paniek paniek paniek paniek paniek

- Trust me: je wilt dat "hij het niet doet"

7: Resolving conflicts

- Open de README

7: Resolving conflicts

<<<<<<< HEAD

Lokale naam

=====

Remote naam

>>>>>>> commit hash

7: Resolving conflicts

- Verwijder alle "rare" dingen die niet jouw naam zijn
- <<< HEAD ===== remote >>>> hash

7: Resolving conflicts

```
$ git status
```

```
$ git diff
```

```
$ git commit -m "Fix merge  
conflict"
```

8: *GOTO* considered harmful

```
$ git push
```

Rinse and repeat from slide
16: “Let the hunger games
begin”

Noodmaatregelen

```
$ cd ..
```

```
$ rm -rf git-gud
```

```
$ git clone
```

```
https://github.com/Lucky-Loek/git-gud.git
```

```
$ cd git-gud
```

9: Give up

- Dit kan makkelijker

9: Give up

- Twee manieren:
 - Nooit hetzelfde bestand editen
 - Branches gebruiken

9: Give up

- Let op!
- Branches voorkomen geen merge conflicts
- Ze verminderen de hoeveelheid alleen

10: Branching

- Wat is een "branch"?

10: Branching

- Een aftakking vanaf een commit

10: Branching

- Twee smaakjes:
 - Feature branch
 - Developer branch

10.1: Feature branch

- Voor elke taak/user story een nieuwe branch
- Na oplossen in "master" mergen

10.2: Developer branch

- Voor elke sprint een branch per developer
- Na sprint in "master" mergen

10: Branching

- Feature branch:
 - Vaker werk tijdens sprint
 - (bijv. 10 merges van 100 LoC)

10: Branching

- Developer branch:
 - Meer werk na sprint
 - (bijv. 1 merge van 1000 LoC)

10: Branching

- Geen goed of fout
- Kies iets en kijk hoe het werkt

11: Creating branches

```
$ git branch [je naam]
```

```
$ git checkout [je naam]
```

OF

```
$ git checkout -b [je naam]
```

11: Creating branches

```
$ git status
```

12: Making changes

- Maak een HTML bestand met jouw naam als bestandsnaam
- Zet je naam in een `<h1>` tag

12: Creating branches

```
$ git status
```

12: Making changes

- Maak een HTML bestand met jouw naam als bestandsnaam
- Zet je naam in een `<h1>` tag

3: Staging changes

```
$ git add [naam].html
```

```
$ git status
```


4: Committing changes

```
$ git commit -m "Add html  
file with my name"
```

```
$ git status
```

```
$ git log
```

13: Making a difference

```
$ git diff master
```

13: Making a difference

- Wat zie je?

14: When push comes to shove

```
$ git push
```

6: Paniek paniek paniek paniek paniek paniek

- Wat gebeurt er?

6: Paniek paniek paniek paniek paniek paniek

- Twee keuzes:
 - Goed nadenken en alles uittypen
 - "git push" en dan suggestie kopiëren

15: Please pull my fin-

- Ga naar <https://github.com/Lucky-Loek/git-gud>
- Open een pull request vanaf jouw branch naar master

16: Clean your own stuff

- Ik kan nu niet meer mergen
- En nu?

16: Cleaning up

```
$ git fetch master
```

```
$ git merge origin/master
```

```
$ git push
```

16: Cleaning up v2

```
$ git checkout master
```

```
$ git pull
```

```
$ git checkout [je naam]
```

```
$ git merge master
```

16: Clean your own stuff

- Het "merge" commando zorgt volgens sommigen voor een onnodige commit
- Kan ook met "rebase"
- <https://www.atlassian.com/git/tutorials/merging-vs-rebasing>

16: Clean your own stuff

- Ik kan nu weer mergen!

Moraal van het verhaal

- "git status" is je beste vriend
- Merge conflict = git saving you
- A "git pull" a day keeps the conflicts away
- Clean your own stuff

Practice makes perfect

- Dit leer je niet even
- Gewoon honderdduizend keer doen
- <https://learngitbranching.js.org/>