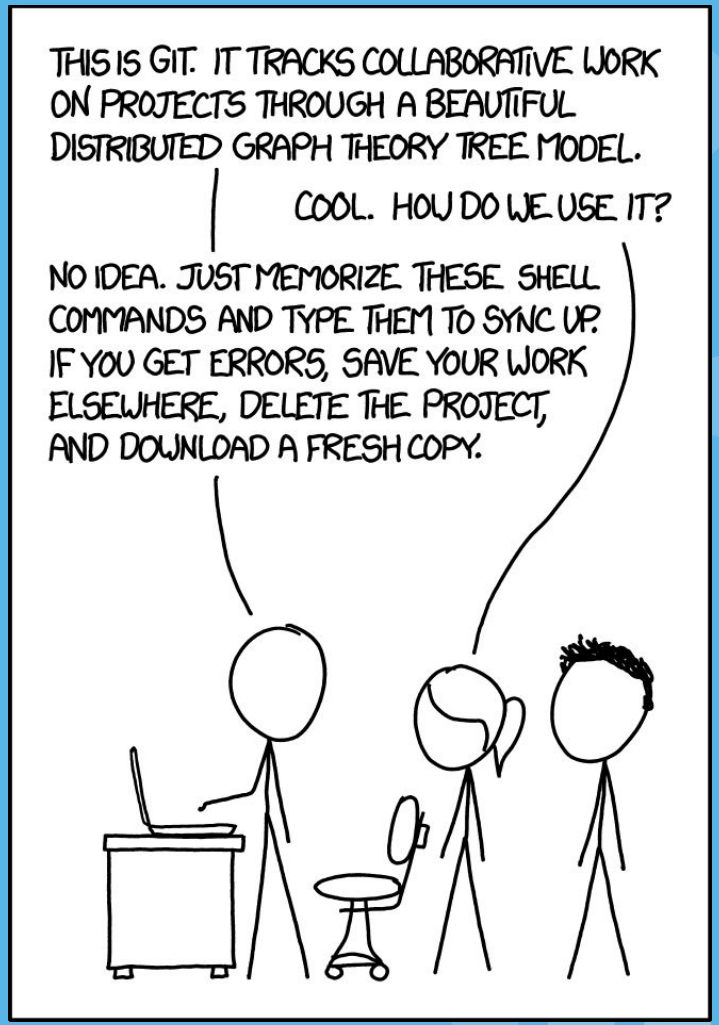
Workshop GIT

Wesley Klop

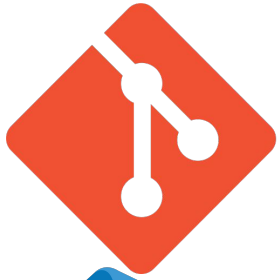


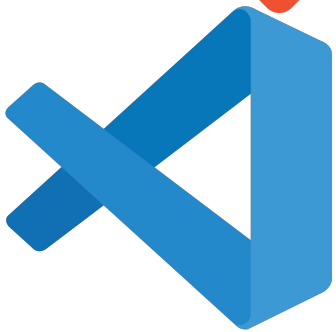
Dit ben ik

- Software Engineering student
- Al ±8 jaar aan het programmeren
-
- Mail: wesley19097@gmail.com
- GitHub: https://github.com/WesleyKlop
- Ook te vinden op Teams voor vragen

Dit heb je nodig

- Git CLI (https://git-scm.com)
- Een editor / IDE naar keuze bijv.
  - VSCode
  - Vim
  - Jetbrains
- Een GitHub account
  - Of Gitlab
  - Of Bitbucket
  - Of whatever



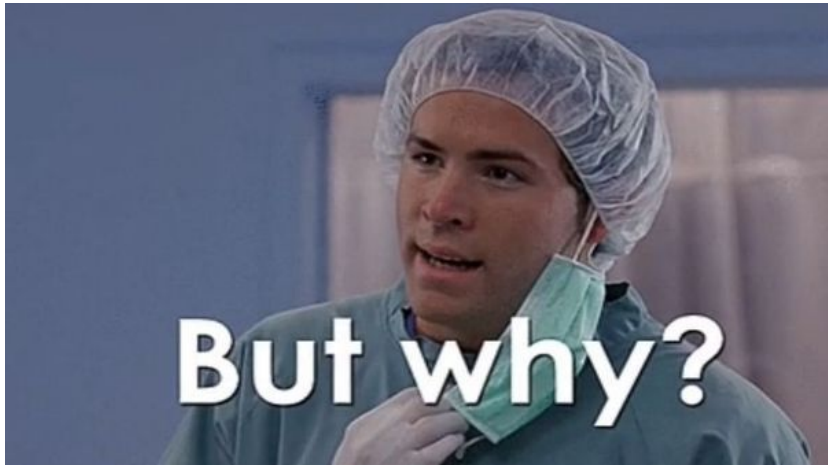


Dit gaan we behandelen

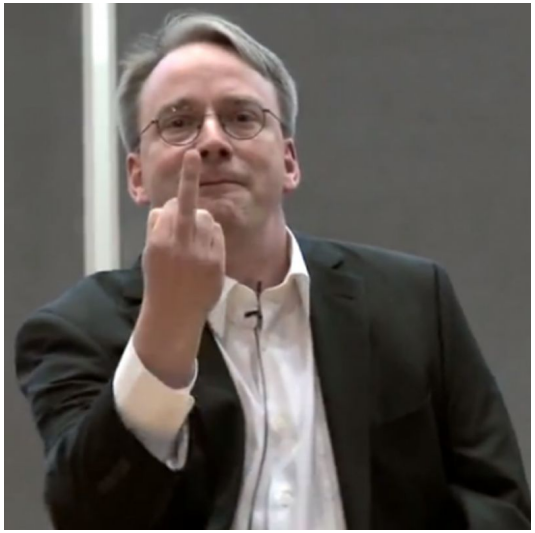
- Waarom deze workshop?
- Wat is git?
- Zo moet het niet
- Zo kan het wel
- Aan de slag
  - Mini mini crash course git commando’s
  - De opdracht(en)
- Tips & Tricks

Waarom deze workshop

- Veel programmeurs gezien die geen git kunnen of maar wat doen
- Easy to learn, difficult to ~~master~~ main
- Essentieel in het bedrijfsleven en tijdens je studie
- Beter samenwerken
- Dit mist in het curriculum



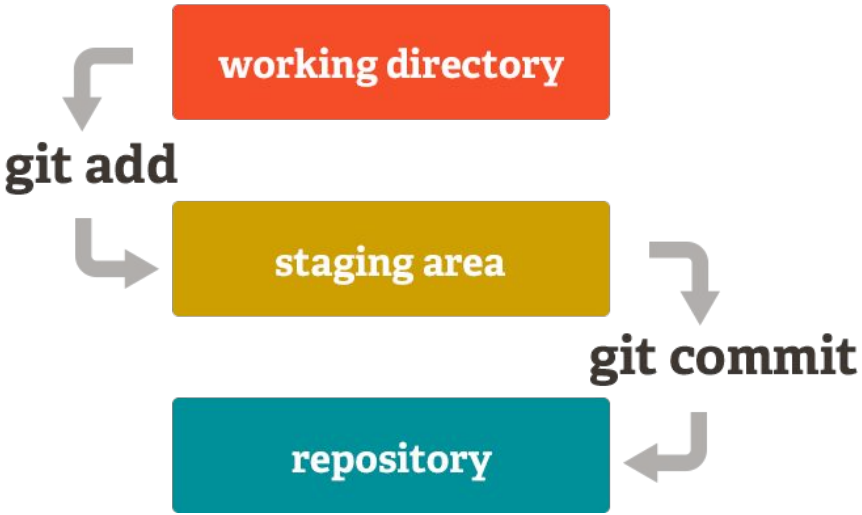
Wat is Git?



- Versiebeheersysteem
- Gedistribueerd
- Gemaakt door Linus Torvalds
  - Op de foto maakt hij zijn mening over NVIDIA duidelijk
- Onafhankelijk van elkaar aanpassingen maken en samenvoegen
- Een must-have skill voor alle IT’ers (zoals jij!)

Basis basis git

- Working directory
- Staging
- Local repository
- Remote repository
- Branches

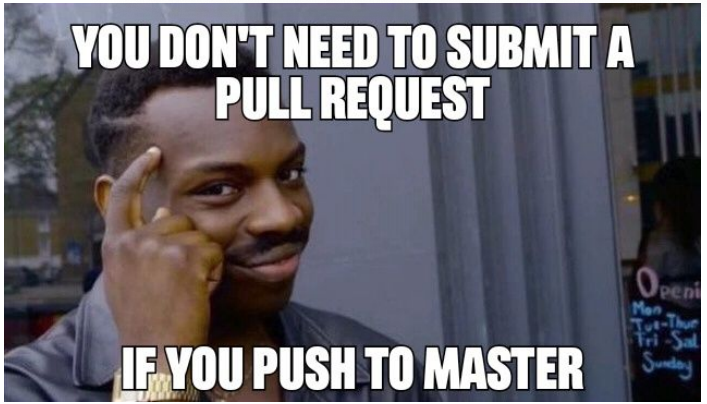


ZO MOET HET NIET!!!11!

Voorbeelden

- Branches zijn geen mensen
- Direct committen op de hoofd branch
- `git commit -am “changes”`
- `git push -f`
- Multiple talen
- Heel veel verschillende dingen in 1 commit die niks met elkaar te maken hebben
- Ongerelateerde code style veranderingen

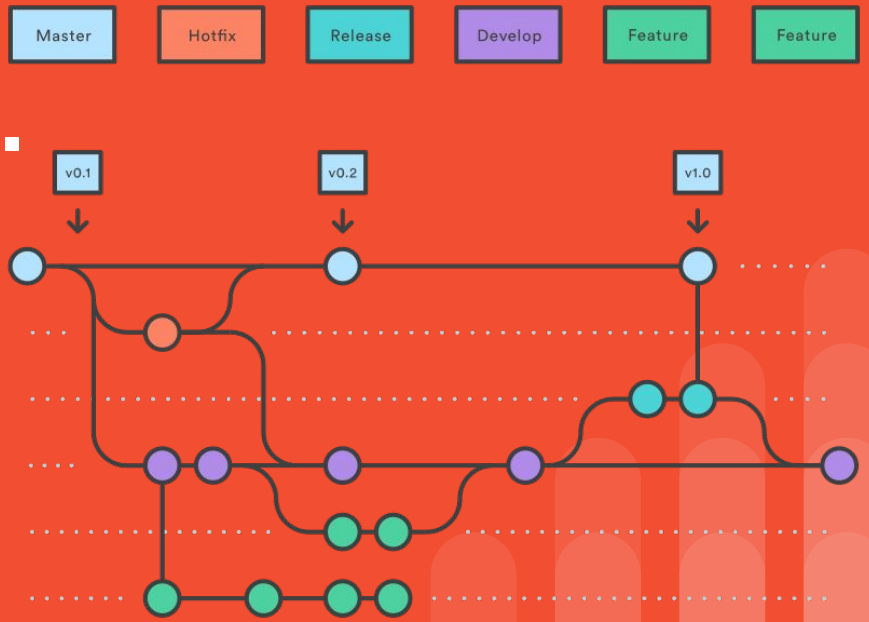
```
wesley@snijplank:~/Projects/Web/git-...
wesley in snijplank in git-workshop on  main
❯ git branch
  ben-jij-al-syntax-lid-btw
  guus
  homepage
* main
  thom
  veranderingen

wesley in snijplank in git-workshop on  main
❯
03/12, 3:35 PM   -zsh   main
```



WesleyKlop wants to merge 442 commits into `develop`

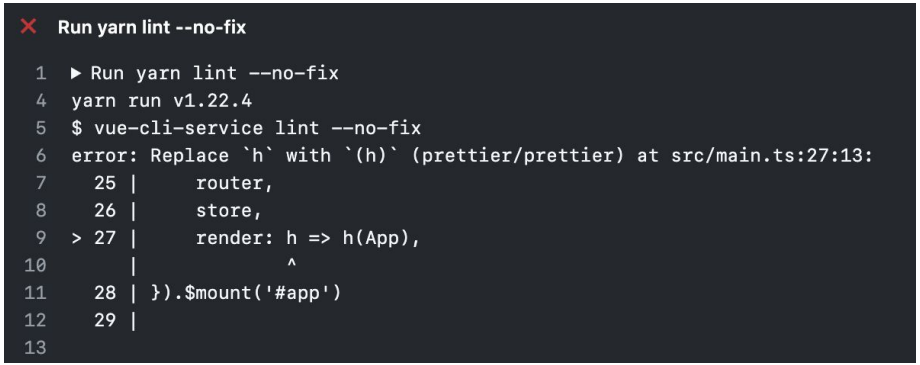
ZO KAN HET WEL.

feat. Git Flow



Over het algemeen

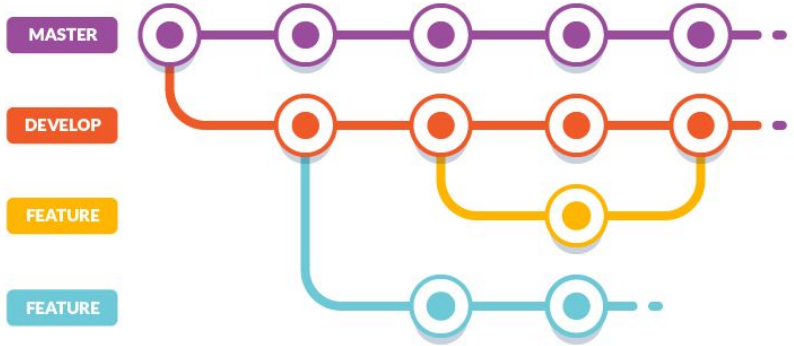
- Zorg dat je main branch altijd werkt en branch protection heeft
- Spreek een taal af waarin je je commit messages schrijft(NL, EN)
- Schrijf in toekomstige tegenwoordige tijd “Implement creation of users”
- Branch naar een doel (user stories)
- Commit **vaak** (het is gratis!)
- Gebruik een `.gitignore`!
  - `.vscode, .idea`
  - `node_modules, vendor`
- Dwing dezelfde code style af
  - .editorconfig (basics)
  - PSR-12, PEP-8, etc...
  - Prettier



Over het algemeen (2)

- Gebruik Pull Requests (PRs)
- Doe aan Code Reviews!
- Kleine branches
  - Anders een losse “epic” branch
- Vergeet niet te pullen voordat je aan je volgende feature / bugfix begint
- Maar het belangrijkste:
  - Wees **consistent** met je team

Git Flow

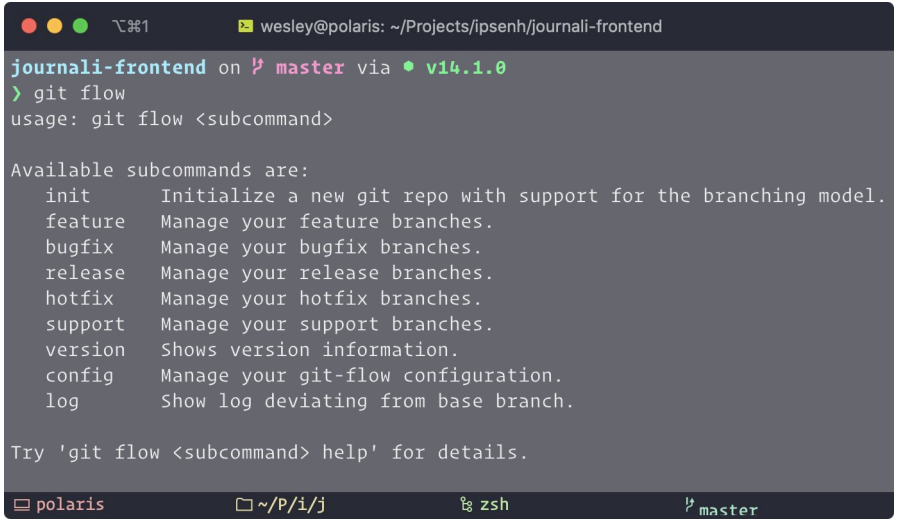


- 2 branches
  - Main (production)
  - Develop (development)
- Main zou altijd moeten werken
- Branch op basis van develop
  - `feature/implement-auth`
  - `feature/42-update-logo`
  - `bugfix/user-nullpointer-error`
  - `bugfix/1337-missing-translation`
- Release? Merge develop in main
  - Meerdere environments (OTAP)
  - Rollbacks

Geweldige bron: https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

Voordelen

- Minder merge conflicts
- Meer duidelijkheid
- Mogelijkheden meer automatisering
- Geen ruzie meer
- Betere code schrijven
- Hogere ontwikkelsnelheid
- Grip op je project
- Flexibel
- Makkelijker debuggen



Een miniem stukje theorie

Basis git

- 4 “omgevingen”
  - working directory - De bestanden die je ziet in je git project
  - stage - De veranderingen die je gaat registreren met “`git commit`”
  - local repository - De lijn van veranderingen die lokaal opgeslagen staan
  - remote repository - Centrale versie van alle gepushde veranderingen
- Basic workflow
  - `git init / git clone` – Initialiseer of kloon een project
  - `git add` – Voeg gemaakte veranderingen toe aan je stage
  - `git commit` – Registreer je stage naar de lokale repository
  - `git push` – Push de veranderingen in je lokale repository naar het centrale git versiebeheersysteem
- Meer commando’s
  - `git status` – Zie de huidige status van je project
  - `git checkout (-b)` – Wissel van branch (en maak deze eventueel aan) of bezoek een commit hash
  - `git branch` – Zie een lijst van branches
  - `git log` – Zie een log van commits

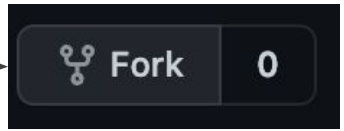
Uitleg repository maken

1. Maak een mapje aan waar je project in komt
2. Open je project in een terminal / git bash
3. Initialiseer je project met git init
4. Maak wijzigingen zoals een README aanmaken
5. Voeg je gemaakte wijzigingen toe aan de stage (`git add .`) voegt alle bestanden in je huidige directory (`.`) recursief aan je stage toe
6. Commit je gemaakte wijzigingen (gebruik `git commit -m “Voeg README toe”` om meteen een bericht mee te geven)
7. Maak een github/gitlab/bitbucket repository aan
8. Registreer je repository als remote (`git remote add origin [url van je repo]`)
9. Push naar de repository (`git push -u origin main`)

Aan de slag

Een kleine web applicatie

- Ik heb een git repo gemaakt: https://github.com/WesleyKlop/git-workshop/
- Voeg iets toe aan de repo!
  - Je naam aan de public/assets/contributors.json file
  - Fix een issue (plaats eerst een reactie om dubbel werk te voorkomen)
  - Als je een beter idee hebt, ga je gang!
- Maar hoe?
  - Maak je eigen branch
  - Let op naming conventions!
  - Doe wijzigingen
  - Maak een pull request



Tips & Tricks

Tips & Tricks

- git blame
  - zie wie iets gedaan heeft
- git fetch origin
  - Haal de remote repository genaamd origin op
- git bisect
  - Zoek naar de commit die een bug geïntroduceerd heeft
- git rebase
  - Herbaseer jouw branch op een andere branch, “herhaalt” de commits (ongeveer)
- git cherry-pick
  - Haal een enkele commit van een andere branch af
- git revert
  - Maak een commit ongedaan
- github actions etc.
  - auto labels
  - testen
  - builden
  - deployen
  - changelogs
- git tags
  - Checkpoints op bepaalde commits
- commit hooks
  - husky
- git flow plugin
  - `brew install git-flow(-avh)`
  - `apt install git-flow`
  - Ingebouwd met Windows git client

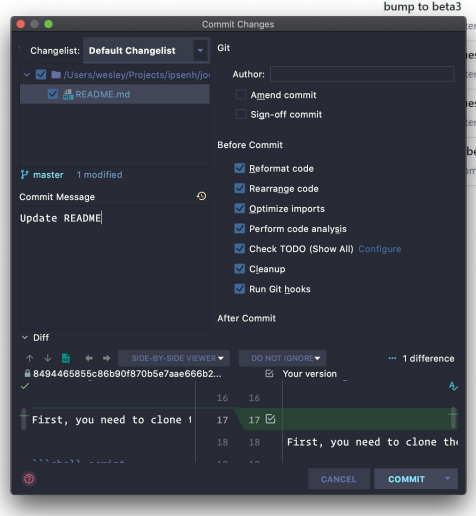
Vaak oefenen, dan git je vanzelf gud

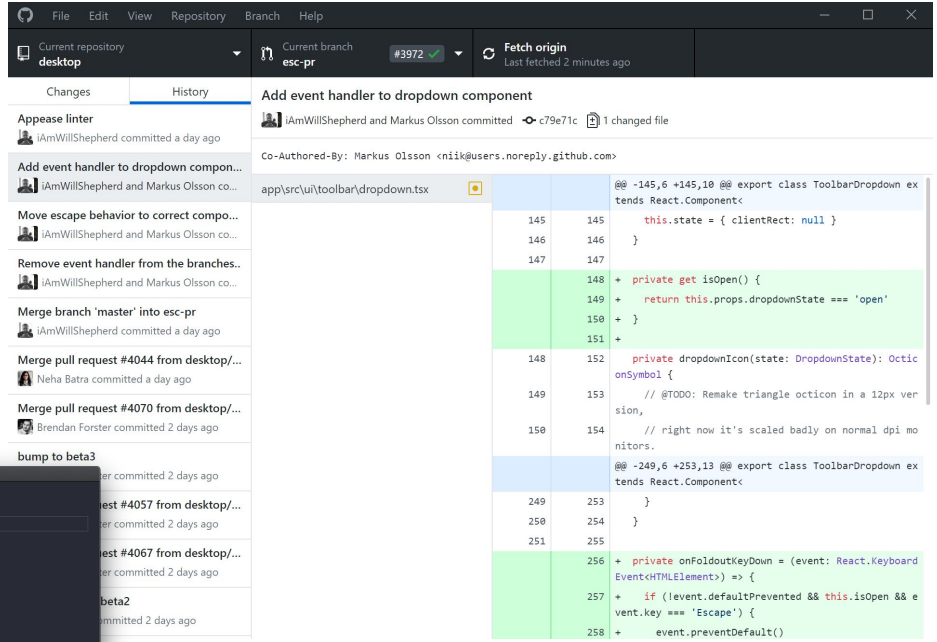
Geen CLI fan?

Veel IDE's hebben git support ingebouwd

Git Kraken

Github desktop





Vragen?