

Versionskontrolle mit



git

Agenda

- Was ist Versionskontrolle
- Was ist Git
- Wie installieren wir Git
- Wie Arbeiten wir mit Git



Warum ist Versionskontrolle
wichtig?



```
main.py > ...  
1 def function_1():  
2     print("Rock")  
3  
4 def function_2():  
5     print("Paper")  
6  
7 def function_3():  
8     print("Scissors")  
9  
10
```



```
main.py > ...
1 def function_1():
2     print("Rock")
3
4 def function_2():
5     print("Paper")
6
7 def function_3():
8     print("Scissors")
9
10
```

```
main.py > ...
1 def function_1():
2     print("Rock")
3
4 def function_2():
5     print("Paper")
6
7 def function_3():
8     print("Scissors")
9
10 #####
11 def main(input):
12
13     if (input == "a"):
14         return function_1()
15     if input == "b":
16         return function_2()
17     if input == "c":
18         return function_3()
19
20
```

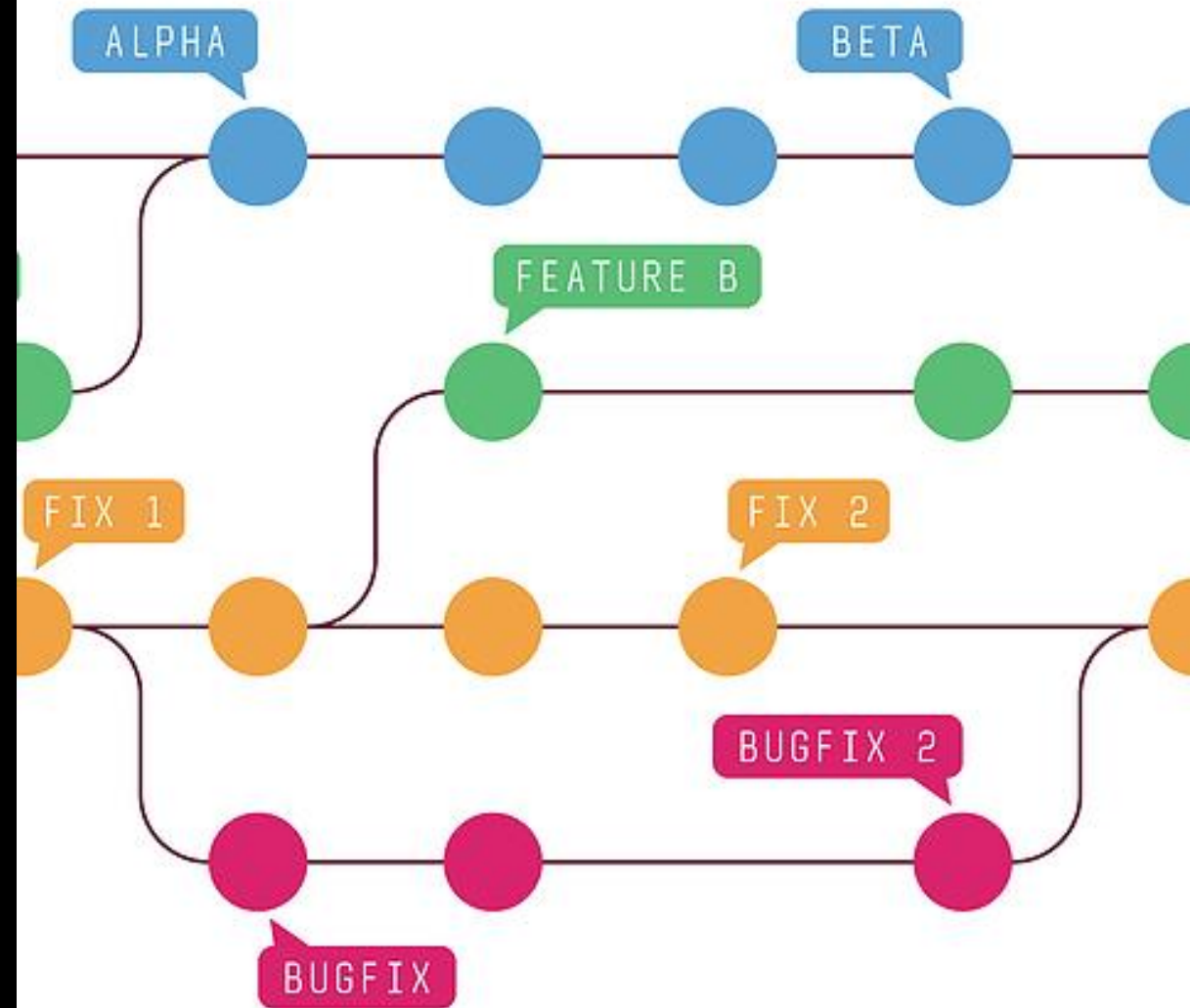
```
main.py > ...
1 def function_1():
2     print("Rock")
3
4 def function_2():
5     print("Paper")
6
7 def function_3():
8     print("Scissors")
9
10
```

```
main.py > ...
1 def function_1():
2     print("Rock")
3
4 def function_2():
5     print("Paper")
6
7 def function_3():
8     print("Scissors")
9
10 #####
11 def main(input):
12
13     if (input == "a"):
14         return function_1()
15     if input == "b":
16         return function_2()
17     if input == "c":
18         return function_3()
19
20
```

```
main.py > ...
1 def function_1():
2     print("Rock")
3
4 def function_2():
5     print("Paper")
6
7 def function_3():
8     print("Scissors")
9
10 #####
11 def main(input):
12
13     if (input == "a"):
14         return function_1()
15     if input == "b":
16         return function_2()
17     if input == "c":
18         return function_3()
19
20 #####
21 main(input("please enter a, b or c: "))
```

Was ist Versionskontrolle?

- Speichert Änderungen
- Backups
- Nachvollziehbarkeit
- Oft in der Cloud
- Vereinfacht Kollaboration



Bekannte Versionsverwaltungssysteme



Apache Subversion (SVN)



Concurrent Versions System (CVS)



Bitkeeper



git



Git

- 2005 ursprünglich von Linus Torvalds entwickelt, da BitKeeper seine Lizenz geändert hatte.
- Ziele:
 - 1. Unterstützung verteilter, BitKeeper-ähnlicher Arbeitsabläufe
 - 2. Sehr hohe Sicherheit gegen sowohl unbeabsichtigte als auch böswillige Verfälschung
 - 3. Hohe Effizienz

■ „Git“ bedeutet so viel wie Blödmann

- *„Ich bin ein egoistischer Mistkerl, und ich benenne all meine Projekte nach mir. Zuerst ‚Linux‘, jetzt eben ‚Git‘.“*
- – Linus Torvalds

Installation

- <https://git-scm.com/>



Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.



About

Documentation

Downloads

[GUI Clients](#)

[Logos](#)

Community

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Downloads



macOS



Windows



Linux/Unix

Older releases are available and the [Git source repository](#) is on GitHub.



GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)

Installation

1. Wähle die gewünschte Sprache für die Installation aus und klicke auf "OK".
2. Akzeptiere die Lizenzvereinbarung, indem Du auf "Next" klickst.
3. Lasse die Standardoptionen für Installationsort, Startmenü-Ordner und Pfadumgebungsvariable unverändert und klicke auf "Next".
4. Wähle "Git from the command line and also from 3rd-party software" für die Pfadumgebungsvariable aus und klicke auf "Next".
5. Wähle "Use OpenSSH" für SSH-Executable und klicke auf "Next".
6. Lasse "Checkout Windows-style, commit Unix-style line endings" für Line Ending Conversion ausgewählt und klicke auf "Next".
7. Wähle "Use Windows' default console window" als Terminal-Emulator und klicke auf "Next".

Konfiguration

```
git config --global user.name "dein Name"  
git config --global user.email "deine E-Mail-Adresse"
```

```
git config --list
```

Let's build from here

The world's leading AI-powered developer platform.

Email address

Sign up for GitHub

Start a free enterprise trial >

Trusted by the world's leading organizations 

3M

KPMG

 Mercedes-Benz

SAP

P&G

 TELUS

- <https://github.com/>

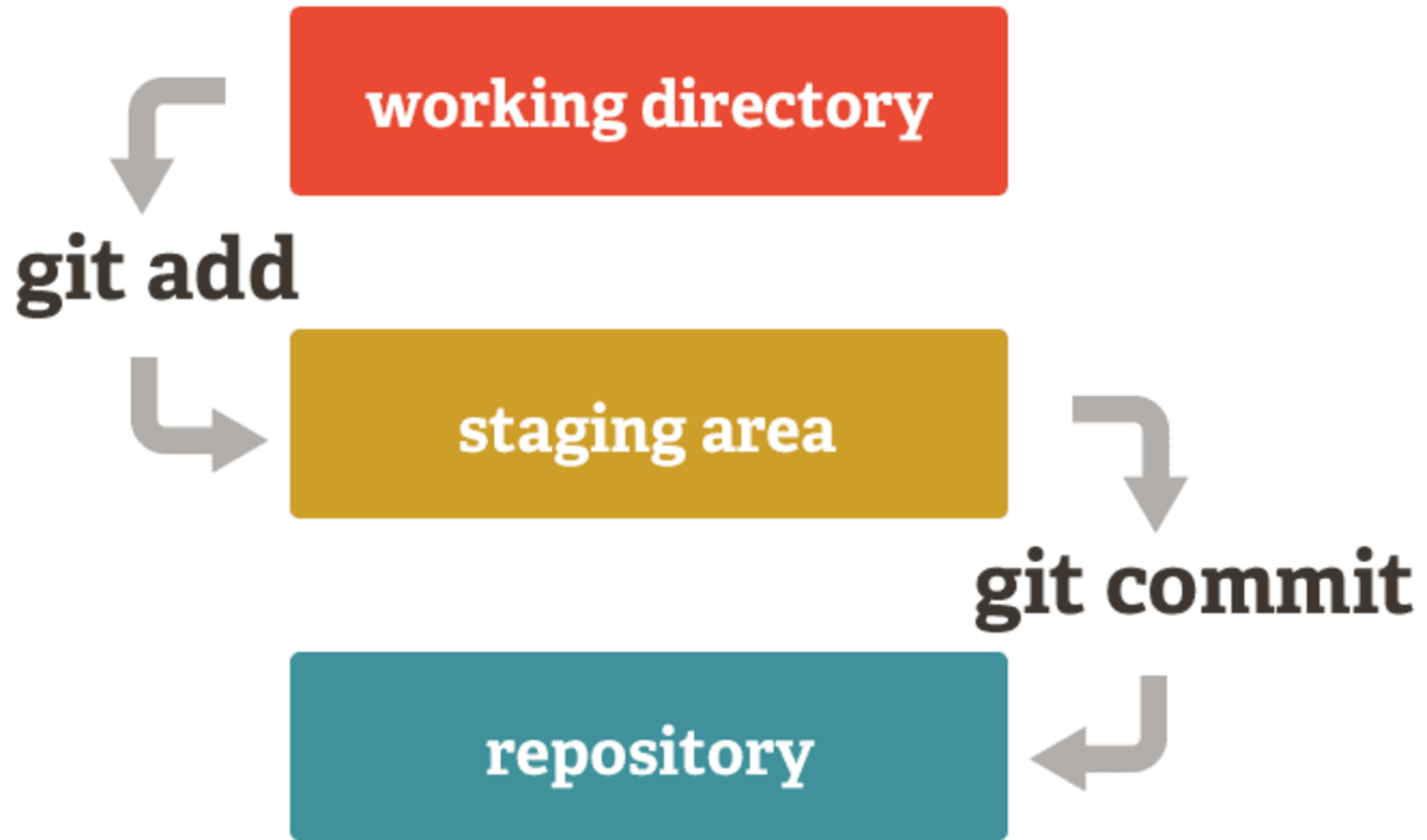
Begriffe

- Repository
- Branch
- Remote Repository
- Konflikte
- Staging-Bereich
- .gitignore

Erste Schritte

- `git add`
- `git commit -m „add file“`
- `git remote add origin <Link>`
- `git push -u origin master`





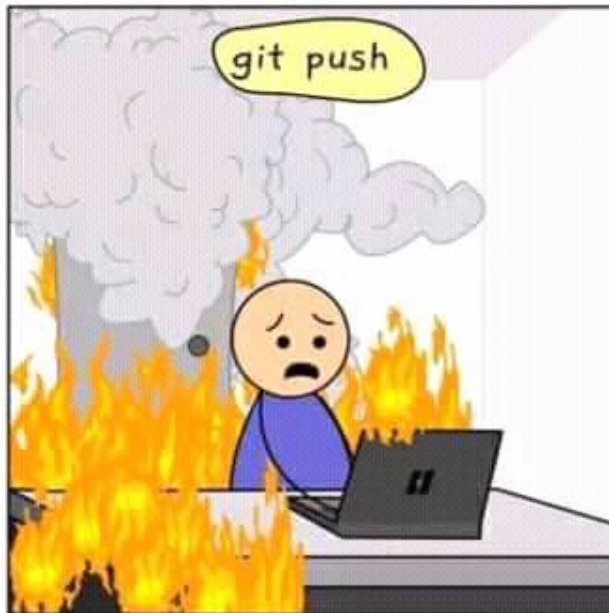
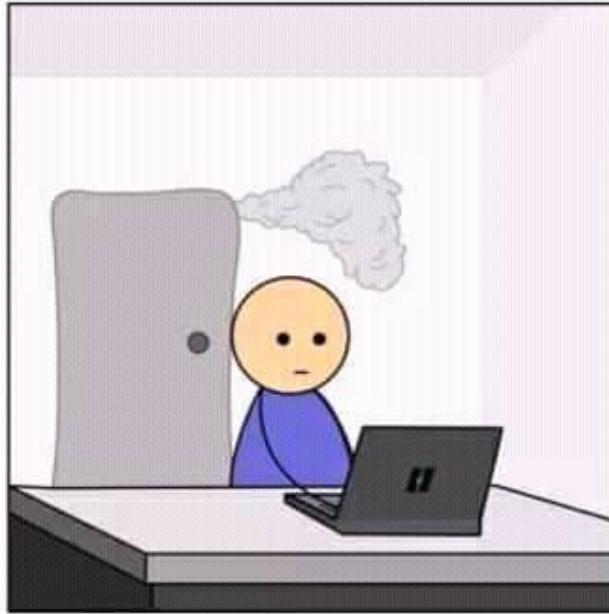
MAKES A GIT COMMIT



**WRITES AN ACTUAL GOOD
COMMIT MESSAGE**

makeameme.org

In case of fire



- git clone
- git status



TEAM A SCRUM BOARD

LEGEND

● Casey Choi

● Nadeem Hazir

● Nick Fitzpartcik

● Zoey Green

★ High Priority

BACKLOG

Translating landing page to Spanish

Researching business users

Experiment for Q4

TO DO

Publish comment section

Publishing pricing page

Add new category

Add new illustrations

Update user request sheet

IN PROGRESS

Make payment modal more visible

Implement team event

Experiment premium plan

Publishing pricing page

REVIEW / QA

Fix discrepancies with tool

Release new business onboarding

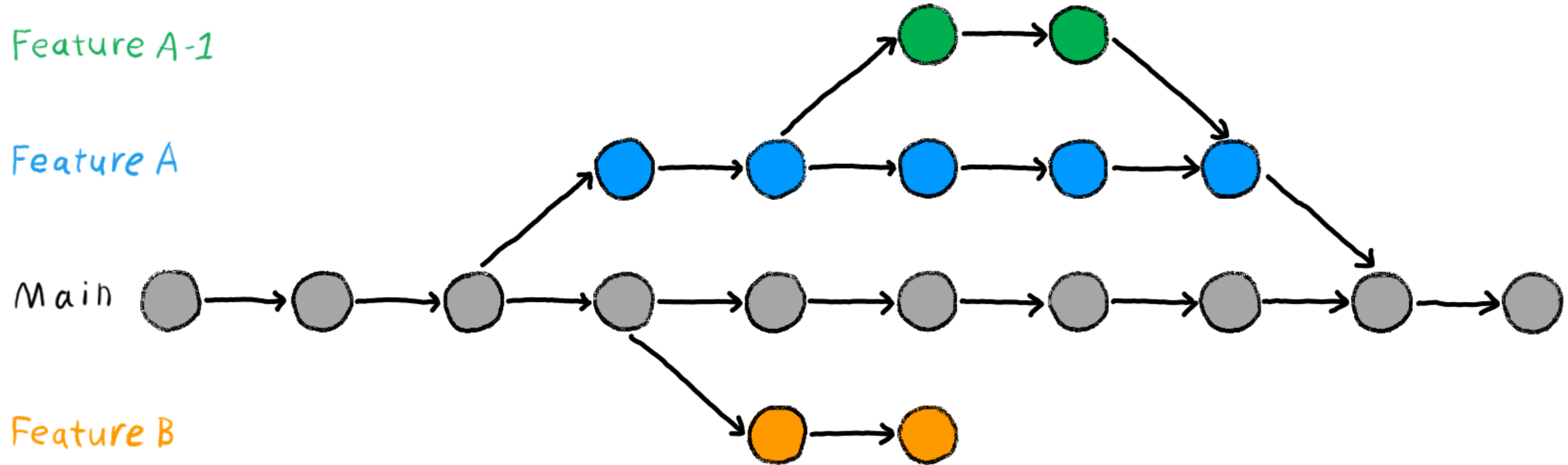
Enable tracking

DONE

Updating tags for tracking

Downgrade student capabilities

Git Branch



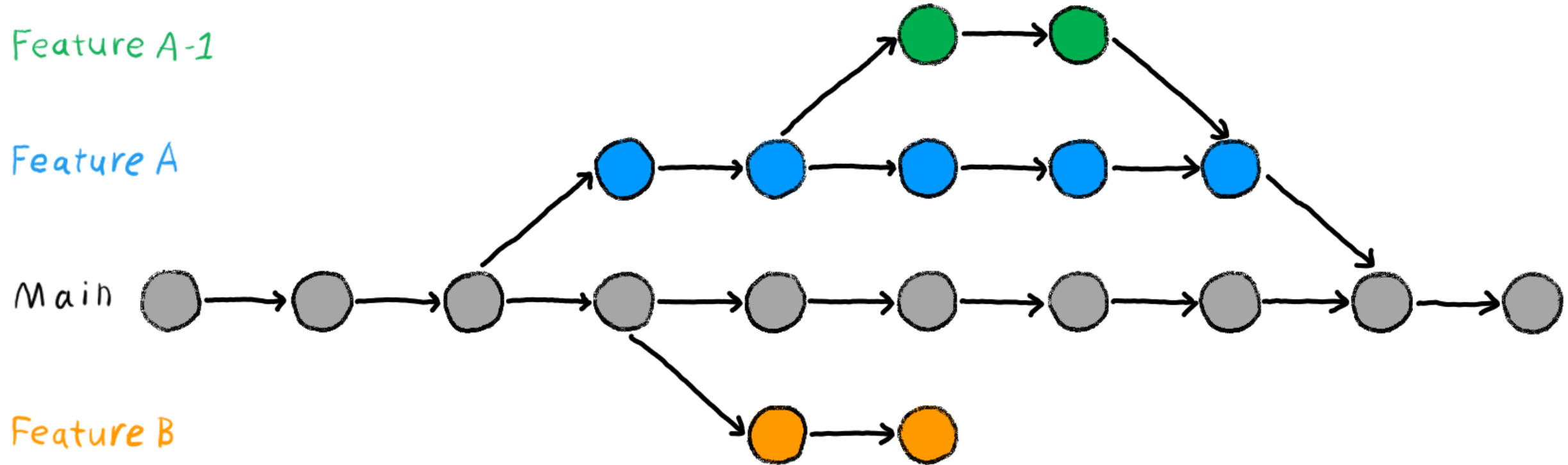
When your coworker asks you which git branch you're currently working on



Git Branch

- `git branch <name>`
- `git checkout <branchname>`
- `git branch`
- `git branch -d <branchname>`
- `git merge <branchname>`
- Neuen Branch erzeugen
- Auf Branch wechseln
- Alle anzeigen
- Löschen
- In den aktuellen zusammenführen

Git Branch



**IF YOU DON'T GIT PULL BEFORE
MAKING CHANGES**



YOU'RE GONNA HAVE A BAD TIME

Merge conflict

When you are gonna merge two branches after long time



GIT MERGE





[Overview](#) [Release Notes](#) [Help](#)

GitHub Desktop

Focus on what matters instead of fighting with Git. Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.

[Download for Windows \(64bit\)](#)

[Feeling brave?](#)

Try new features in the [Beta Channel](#) before they're released.

[Prefer the MSI?](#)

Download for [Windows \(MSI\)](#)

[macOS?](#)

Download for [macOS](#)

By downloading, you agree to the [Open Source Applications Terms](#).

• <https://desktop.github.com/>