










# Was ist Git?

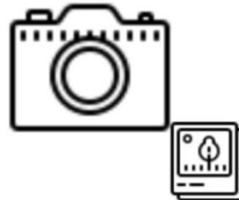
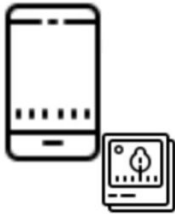
Tom, Hinrik, Aytac, Julian

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

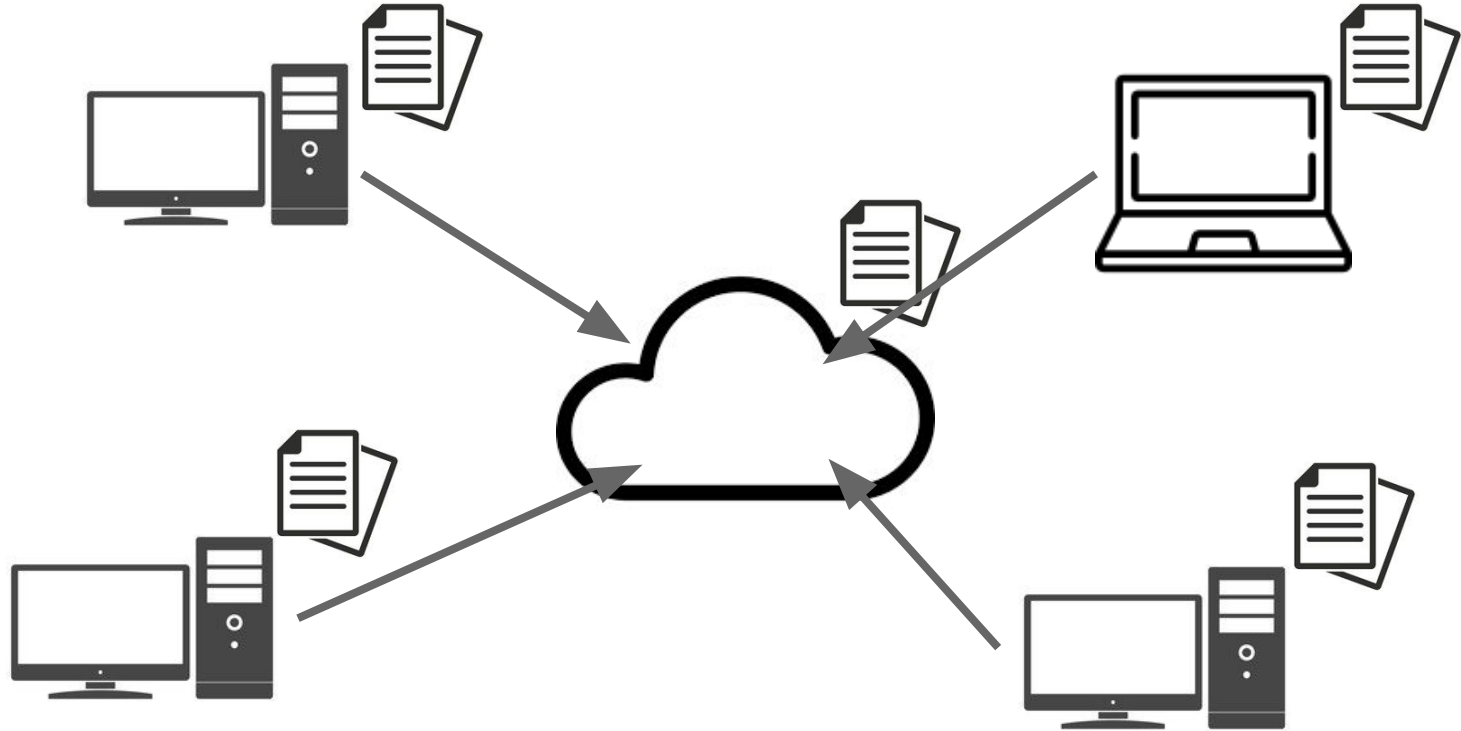
# Problem: Versionierung

Name	Änderungsdatum	Typ
 Präsentation - fertig - DIE HIER ABGEBEN(2) - Kopie.pptx	02.10.2020 13:21	Microsoft PowerP...
 Präsentation - fertig - DIE HIER ABGEBEN(2).pptx	02.10.2020 13:21	Microsoft PowerP...
 Präsentation - fertig - DIE HIER ABGEBEN.pptx	02.10.2020 13:21	Microsoft PowerP...
 Präsentation - fertig - Kopie - Kopie.pptx	02.10.2020 13:21	Microsoft PowerP...
 Präsentation - fertig - Kopie.pptx	02.10.2020 13:21	Microsoft PowerP...
 Präsentation - fertig.pptx	02.10.2020 13:21	Microsoft PowerP...
 Präsentation - Kopie (2).pptx	02.10.2020 13:21	Microsoft PowerP...
 Präsentation - Kopie.pptx	02.10.2020 13:21	Microsoft PowerP...
 Präsentation.pptx	02.10.2020 13:21	Microsoft PowerP...

# Problem: Dezentralisierung



# Problem: Synchrone Nutzung



# Die Lösung: Git

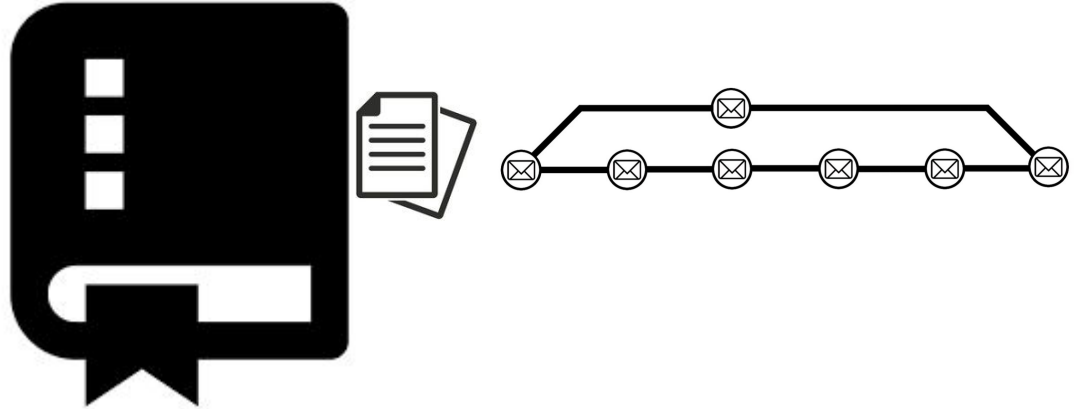
- Versionskontrollsystem
- zentrale Verwaltung von Code
- Versionshistorie
- gleichzeitig multiple Versionen
- arbeiten mit mehreren Personen
- Benachrichtigung über Änderungen



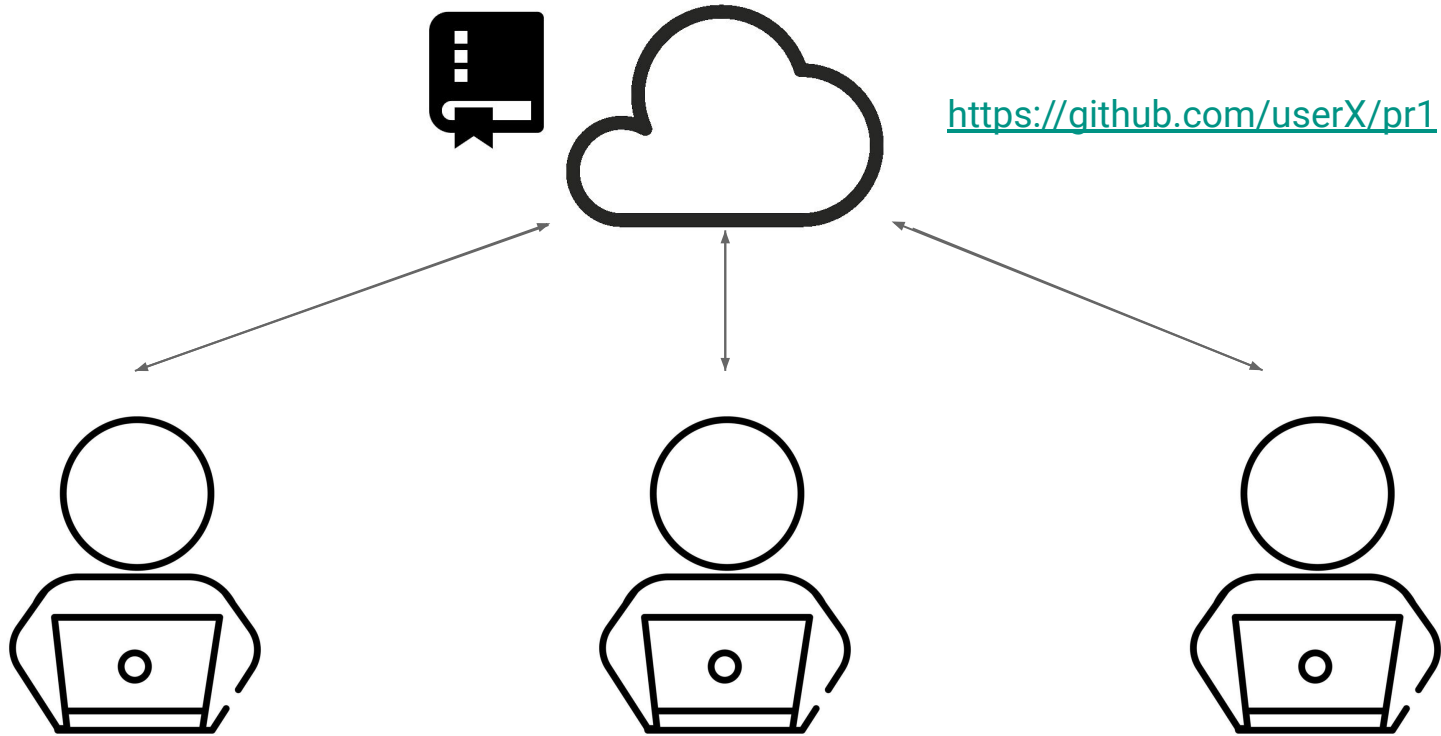
# Repository

Zentraler Speicher für

- Dateien
- Historie
- Branches



# Distributed Version Control



# Versionsverwaltungssoftware

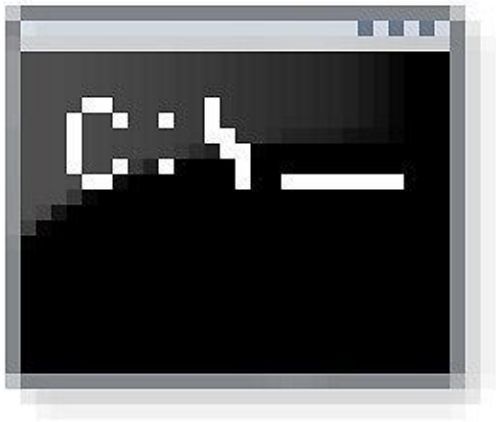


GitLab





# Anwendungssoftware



- Changes | History
- Select branch to compare...
- Merge branch 'master' into designHinrik  
h.ehrenfried • Dec 17, 2021
- patch  
h.ehrenfried • Dec 17, 2021
- Merge branch 'fixing' into 'master'  
TimothyCryman • Dec 16, 2021
- fixed merge conflicts  
TimothyCryman • Dec 16, 2021
- Merge branch 'deploy' into 'master'  
HollowPurity • Dec 16, 2021
- Deploy  
HollowPurity • Dec 16, 2021
- fixed drag and drop verschiebung  
Julian Gegner • Dec 16, 2021
- assetEdge für julian  
h.ehrenfried • Dec 16, 2021

modal fixes

h.ehrenfried fa88bac 6 changed files +85 -37

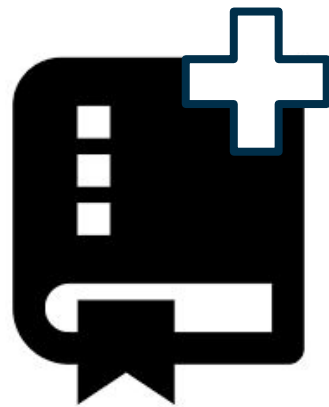
src\components\drawing\AddPic.js	22	23	<div className={classes.back}>
src\components\...\AddPic.module.css	23	24	<IconButton onClick={modalHandler}>
src\components\...\DownloadOptions.js	24	25	<CloseIcon></CloseIcon>
src\components\proj...\CreateProject.js	25	26	</IconButton>
src\components\proj...\CreateProject.js	26	-	</div>
src\compo...\CreateProject.module.css	27	-	<div className={classes.layoutBase}>
src\compo...\LoadProject.module.css	28	-	<div className={classes.header}>
	29	-	<h1>{translation.MESSAGES.SELECT_PROJECT}</h1>
	30	27	</div>
	31	-	<div className={classes.bodyProjects}>
	32	-	<form>
	33	-	<div className={classes.upload}>
	34	-	<div className={classes.gridTop}>
	35	-	<div className={classes.insideGrid}>
	36	-	<p className={classes.uploadText}>{translation.BUTTONS.UPLOAD_PICTURE}</p>
	37	-	<UploadIcon> </UploadIcon>
		28	+
		29	+
			<div className={classes.layoutBase}>

# Repository erstellen

```
git init
```

```
git remote add [alias] [url]
```

```
git clone [url]
```



# Übung 1

1. cloned das Repository unter [https://gitty.informatik.hs-mannheim.de/2012537/PR1\\_Git\\_Uebung](https://gitty.informatik.hs-mannheim.de/2012537/PR1_Git_Uebung) in Eclipse
2. importiert das Repository als Projekt

# Änderungen vornehmen

```
git add <filename>
```

```
git commit -m <MESSAGE>
```

```
git push
```

```
git pull
```

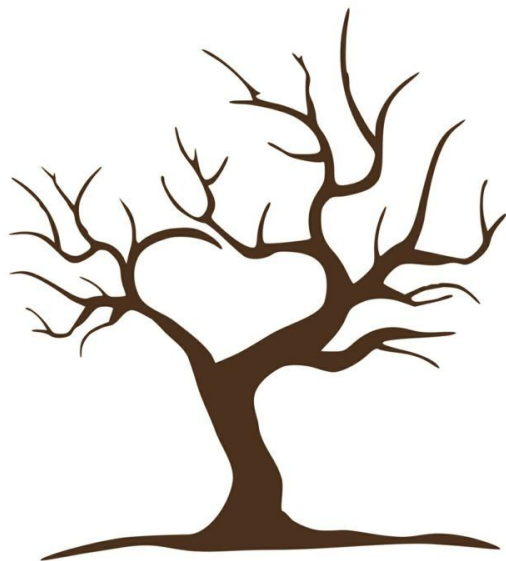
# Übung 2

1. Führt einen Pull auf das Repository aus, um die neueste Version zu erhalten
2. Erstellt eine neue Datei <Matrikelnummer>.txt
3. Schreibt euren Namen in die Datei
4. Ladet die Datei auf das Repository hoch

# Branches

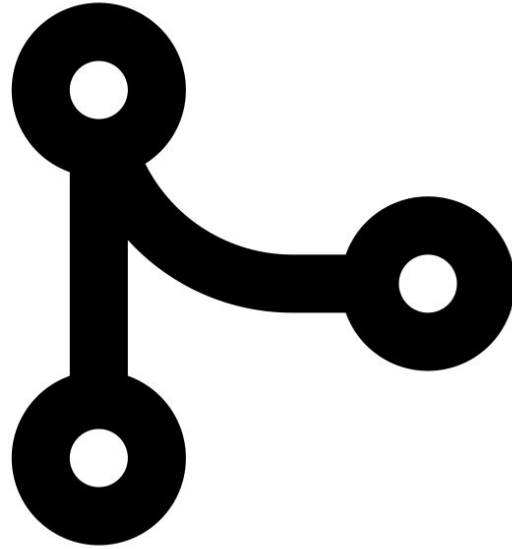
```
git branch <branchname>
```

```
git switch <branchname>
```



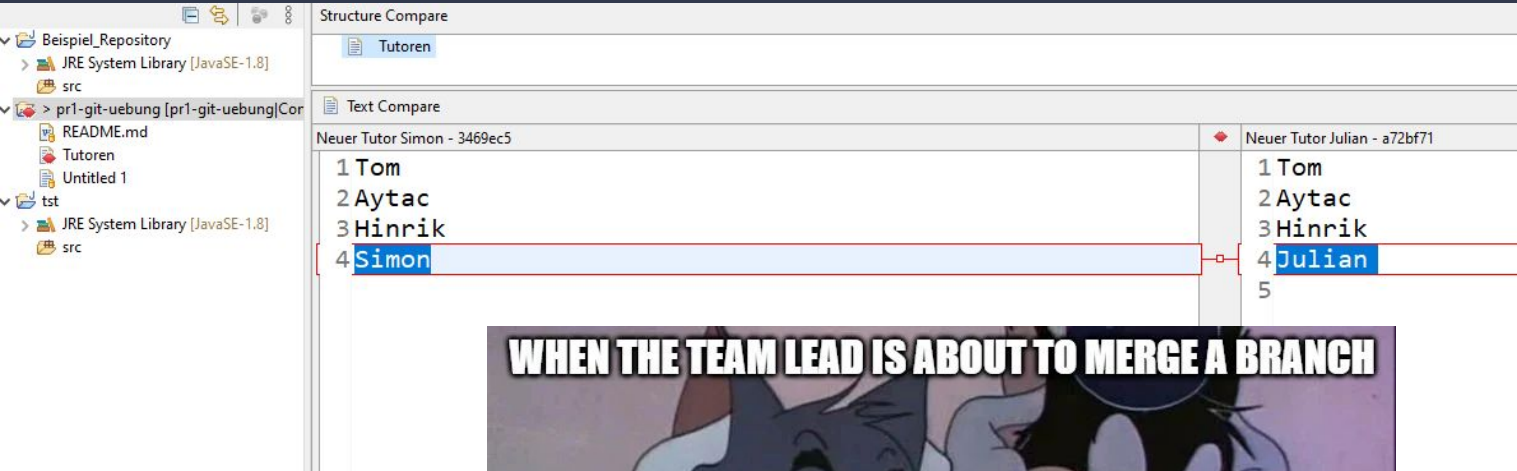
# Merges

```
git merge <branch>
```





# Merge Konflikte



# Übung 3

1. Erstellt Euch ein eigenes Repository
2. Legt eine lokale Datei an und fügt Eure Änderung in Euer Repository hinzu
3. Sichert das Repository auf Gitty
4. Ermöglicht Euren Teammitgliedern, an Euren Repositorys zu arbeiten
5. Nehmt Änderungen an den Repositorys Eurer Teammitglieder vor
6. Fügt in Eurem eigenen Repo alles zu einem Main-Branch zusammen
7. Welches Teammitglied hat welche Änderungen in Eurem Repo vorgenommen?

# Hilfreiche Links und Quellen

Git CLI downloaden: <https://git-scm.com/downloads>

Git CheatSheet: <https://education.github.com/git-cheat-sheet-education.pdf>

Was ist Git?: <https://www.atlassian.com/de/git/tutorials/what-is-git>

Vortrag von Halver Kelm:

[https://services.informatik.hs-mannheim.de/~fischer/lectures/PR1\\_Files/video/05\\_git\\_introduction.mp4](https://services.informatik.hs-mannheim.de/~fischer/lectures/PR1_Files/video/05_git_introduction.mp4)

Folien von Lilly Schumann und Lena Oeser:

[https://services.informatik.hs-mannheim.de/~fischer/lectures/PR1\\_Files/PR1\\_Git.pdf](https://services.informatik.hs-mannheim.de/~fischer/lectures/PR1_Files/PR1_Git.pdf)

# Kontakt



[aytac.kocyigit@stud.hs-mannheim.de](mailto:aytac.kocyigit@stud.hs-mannheim.de)

Aytac#1465

[hinrik.ehrenfried@stud.hs-mannheim.de](mailto:hinrik.ehrenfried@stud.hs-mannheim.de)

Reis#9460

[julian.gegner@stud.hs-mannheim.de](mailto:julian.gegner@stud.hs-mannheim.de)

Crondung#0186

[tom-marcel.roessler@stud.hs-mannheim.de](mailto:tom-marcel.roessler@stud.hs-mannheim.de)

fried\_snickers#4307