



Git



Git



- Git ist ein Programm/Software zum dezentralen Verwalten und Versionieren von (Code-)Dateien.
- Es ermöglicht auch das gleichzeitige Arbeiten von mehreren Personen an denselben Dateien eines Projekt
- Das Repository ist ein verwaltetes Verzeichnis/Ordner auf dem Server und man kann es sich als „Lager/Depot“ vorstellen.
- Die Repositories können nur von mir für die entsprechenden Projekte angelegt werden.
- <http://git.joachimhofmann.org/git/projektname/DeinName.git>





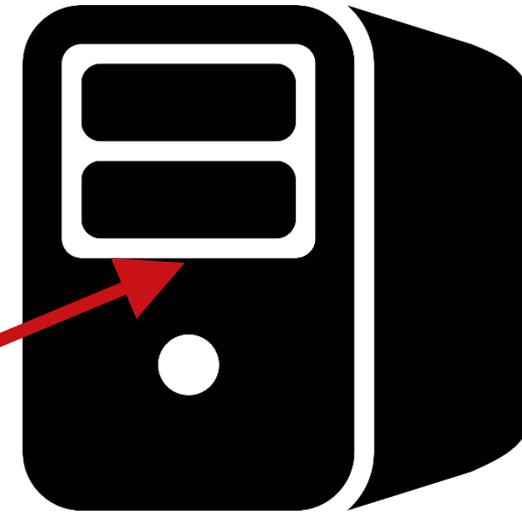
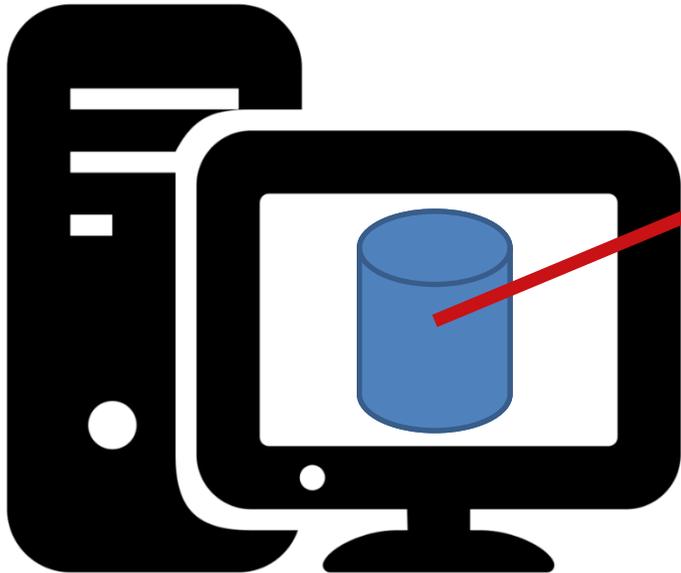
Initialisierung



Initialisieren eines Repositorys mit einem Projekt

<http://git.joachimhofmann.org/git/projektname/DeinName.git>

Dein Rechner

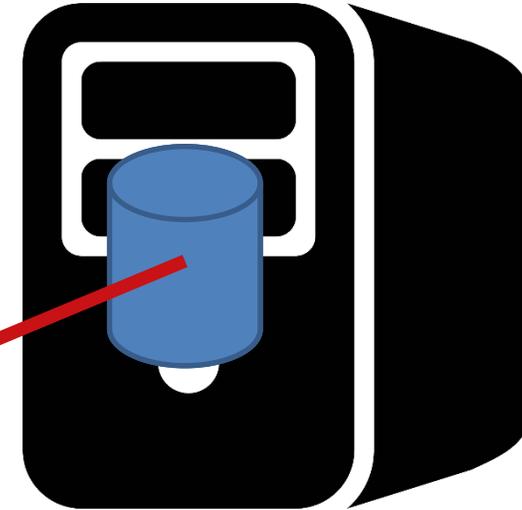
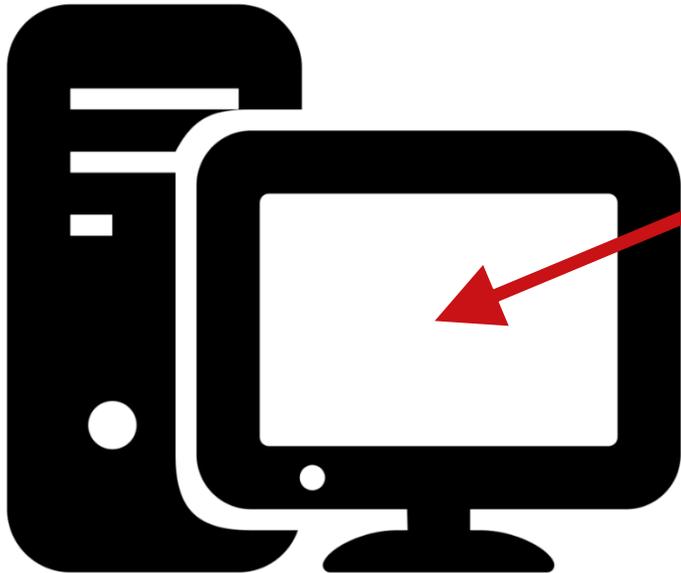


Werkzeuge ->
Teamarbeit->
Share this Project

Initialisieren deines Rechners mit einem Projekt

<http://git.joachimhofmann.org/git/projektname/DeinName.git>

Dein Rechner



Werkzeuge ->
Teamarbeit->
Arbeitskopie erstellen

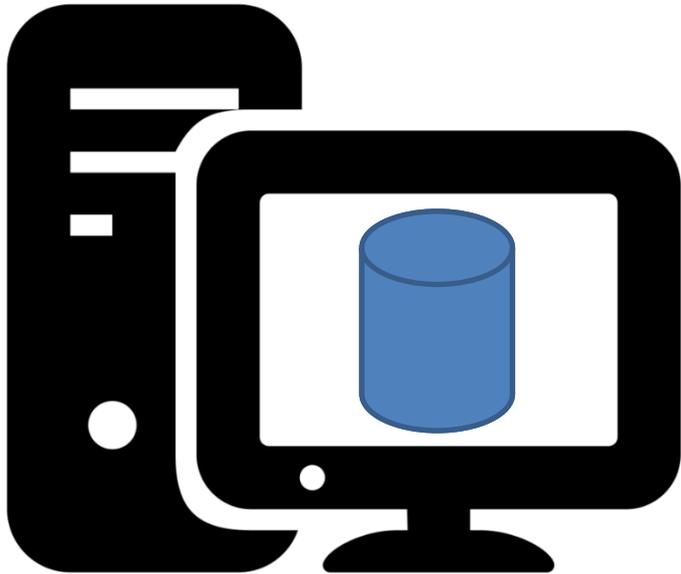
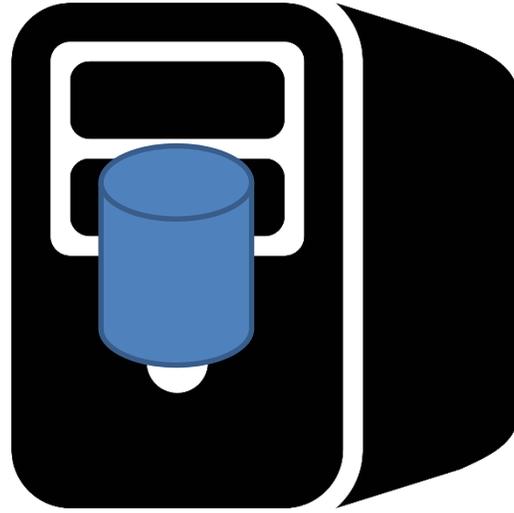


Arbeiten mit Git



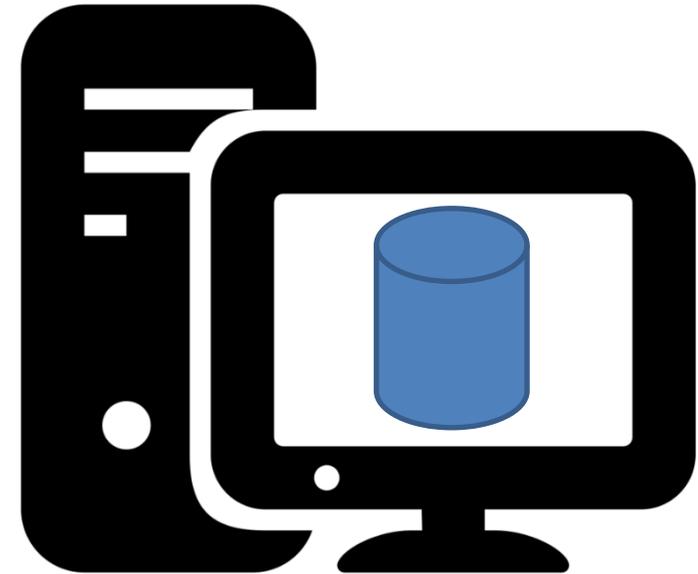
Arbeiten mit Git

Werkzeuge ->
Teamarbeit->
commit/push



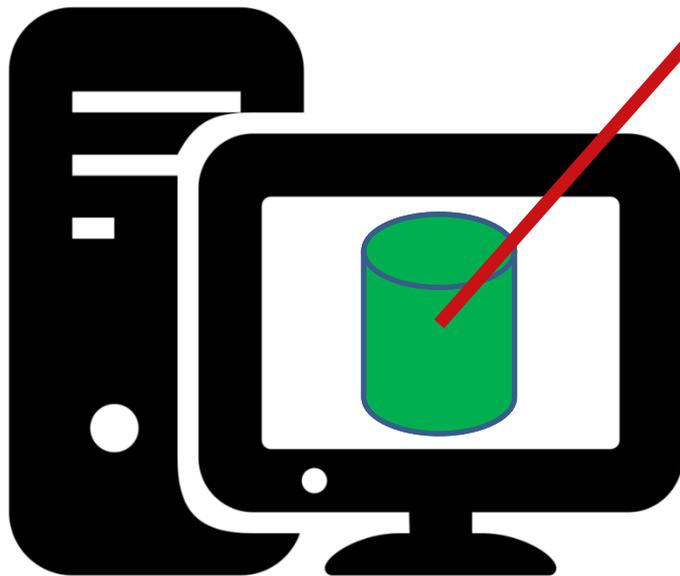
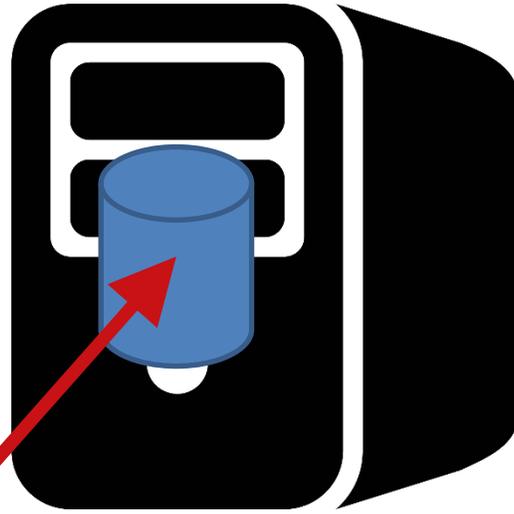
Schulrechner

Heimrechner



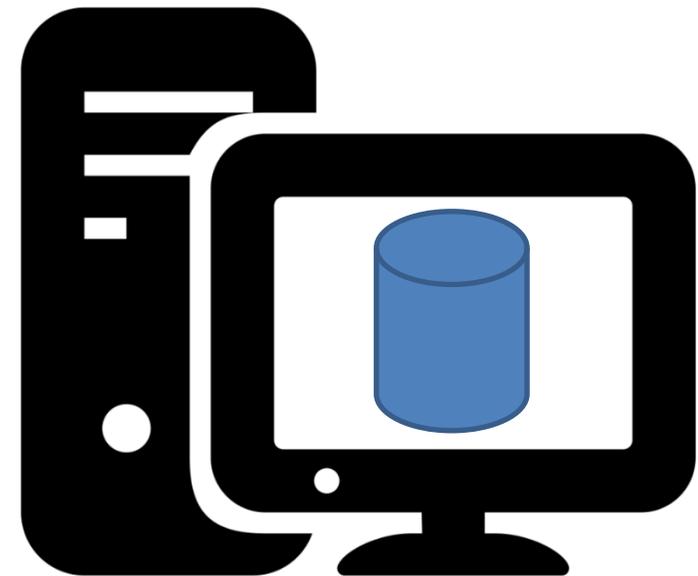
Arbeiten mit Git

Werkzeuge ->
Teamarbeit->
commit/**push**

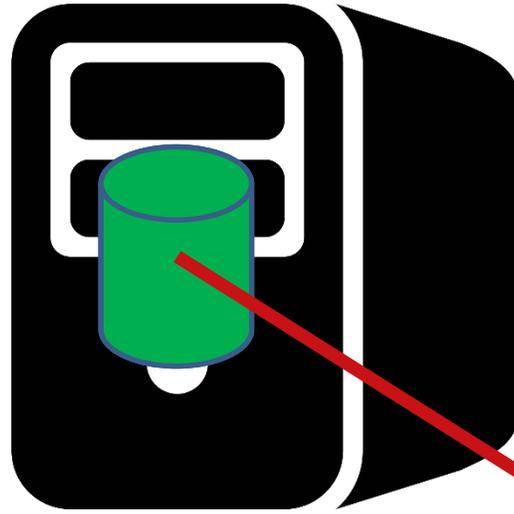


Schulrechner

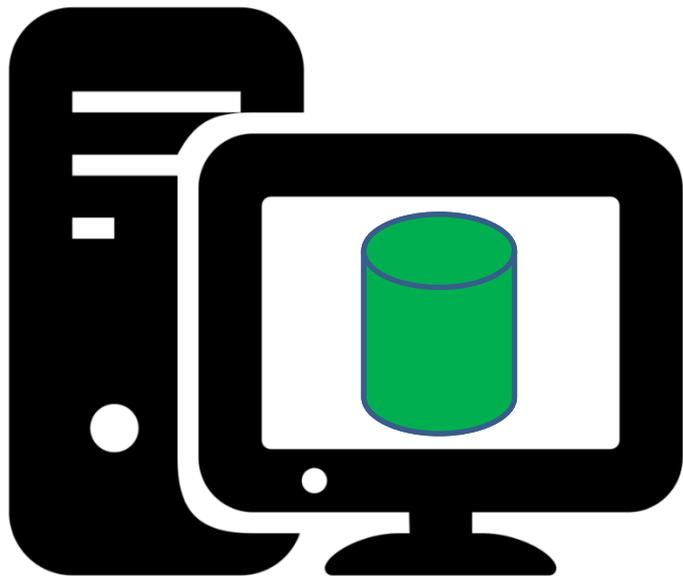
Heimrechner



Arbeiten mit Git

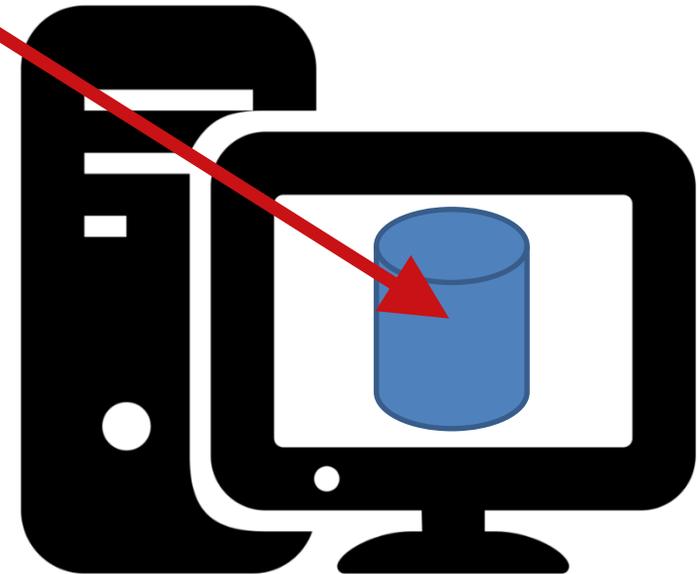


Werkzeuge ->
Teamarbeit->
Aktualisieren/pull

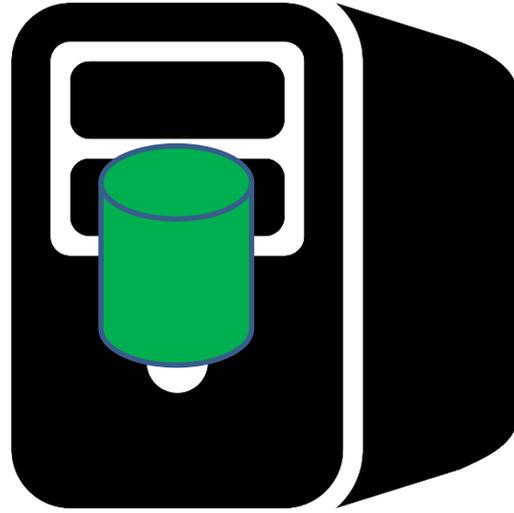


Schulrechner

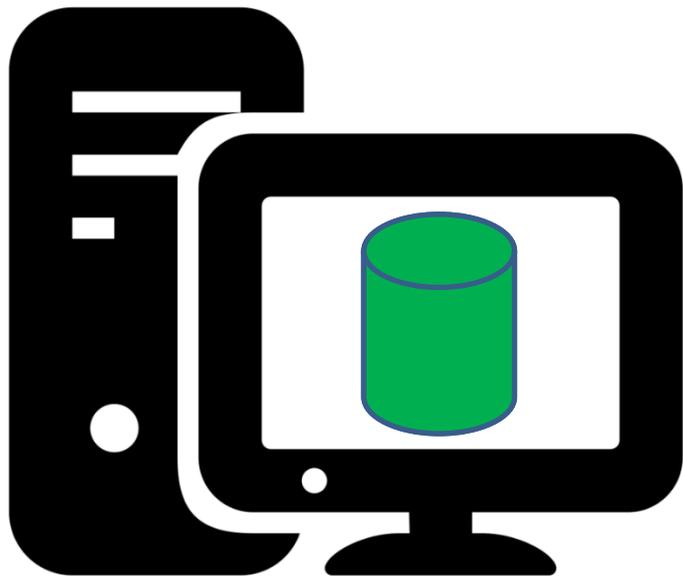
Heimrechner



Arbeiten mit Git

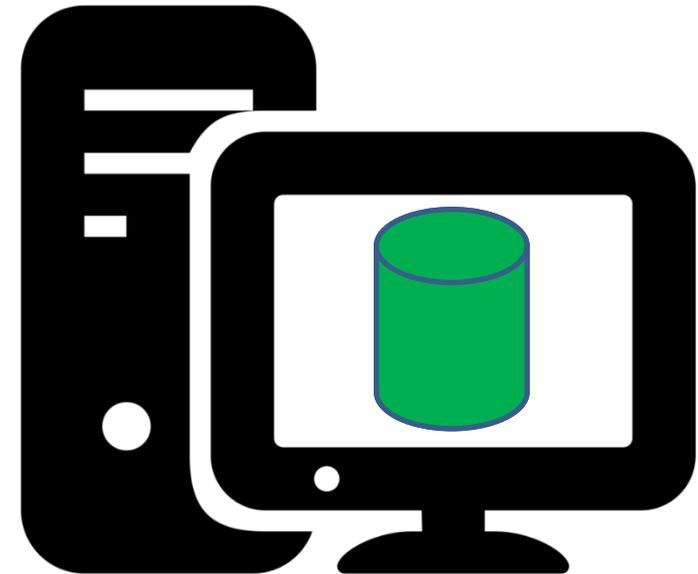


Werkzeuge ->
Teamarbeit->
commit/push

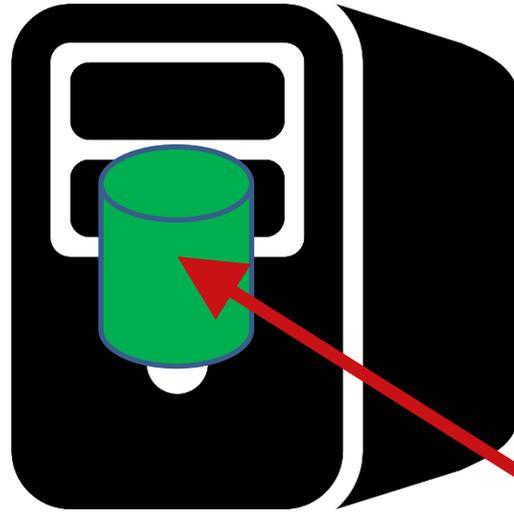


Schulrechner

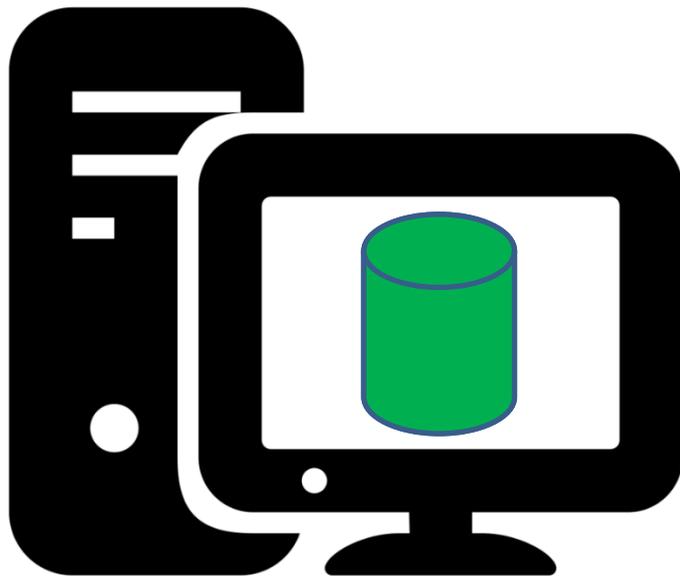
Heimrechner



Arbeiten mit Git

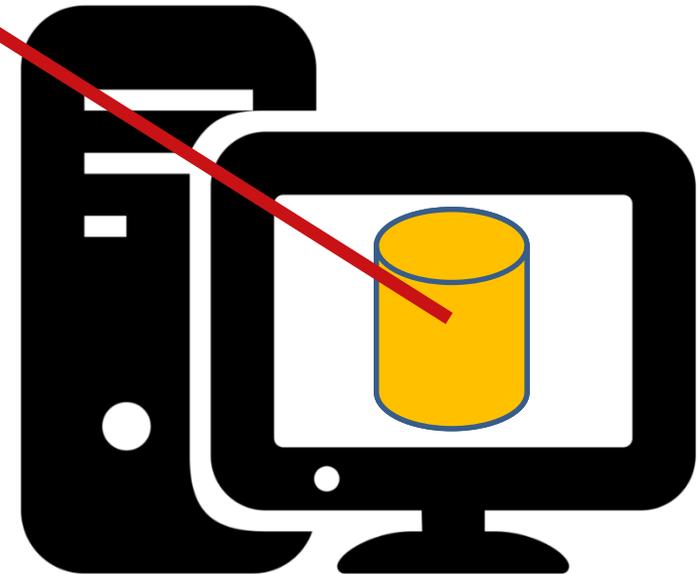


Werkzeuge ->
Teamarbeit->
commit/**push**



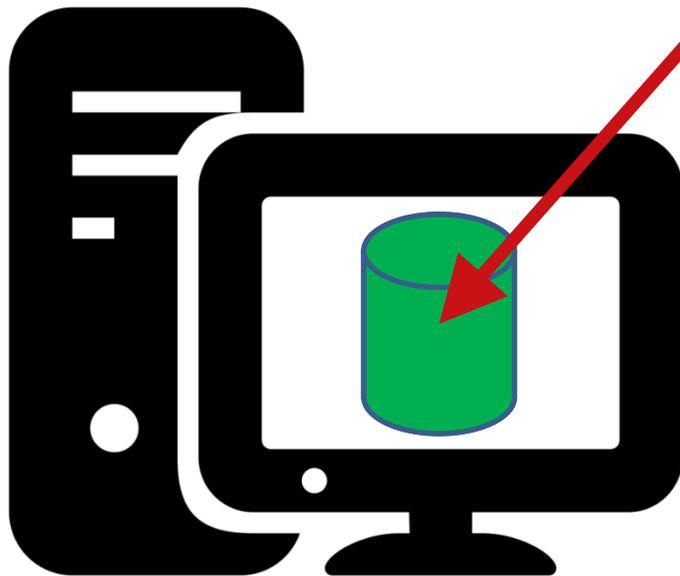
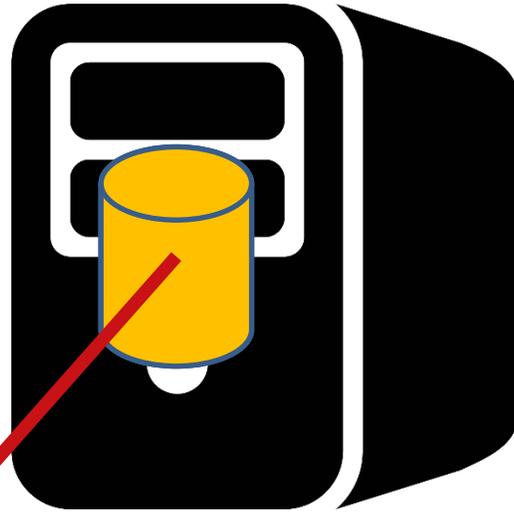
Schulrechner

Heimrechner



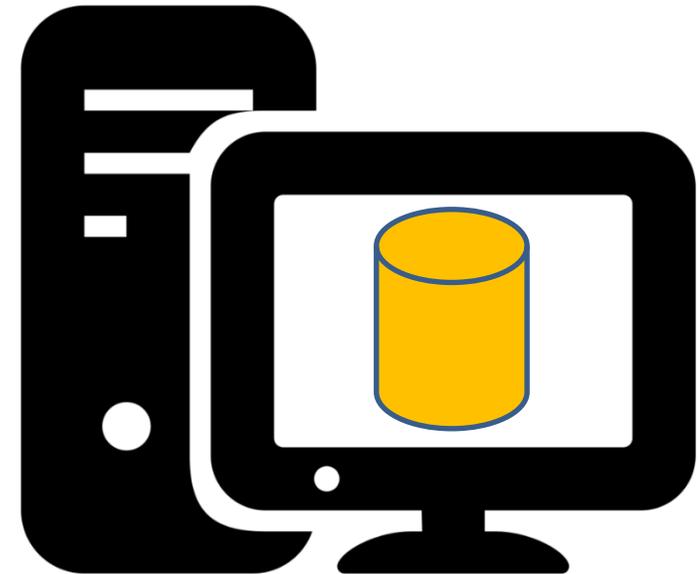
Arbeiten mit Git

Werkzeuge ->
Teamarbeit->
Aktualisieren/pull



Schulrechner

Heimrechner



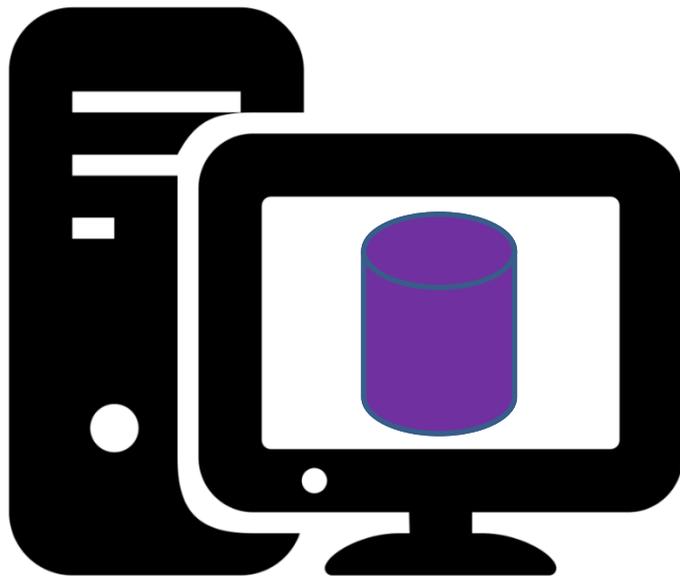
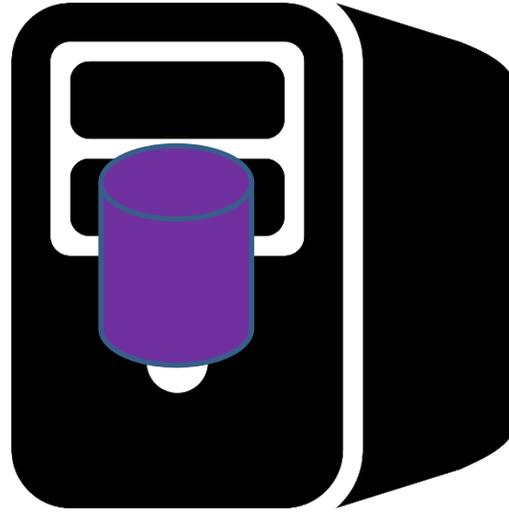


Teamarbeit mit Git (Merge)



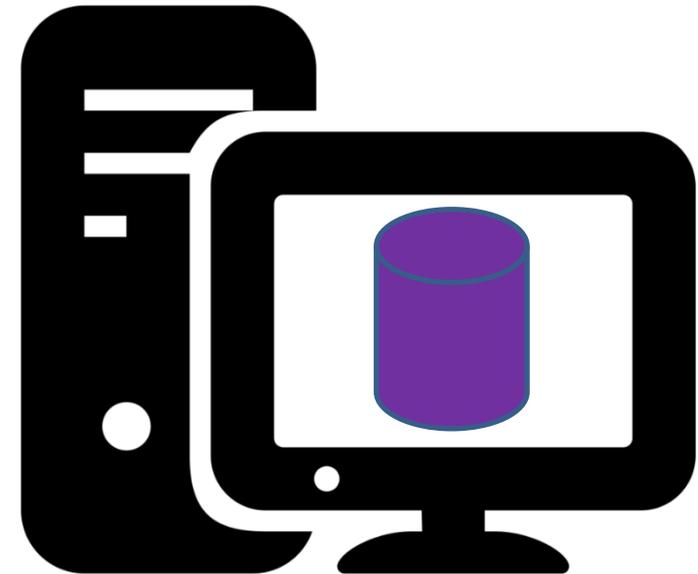
Teamarbeit mit Git

Werkzeuge ->
Teamarbeit->
commit/push



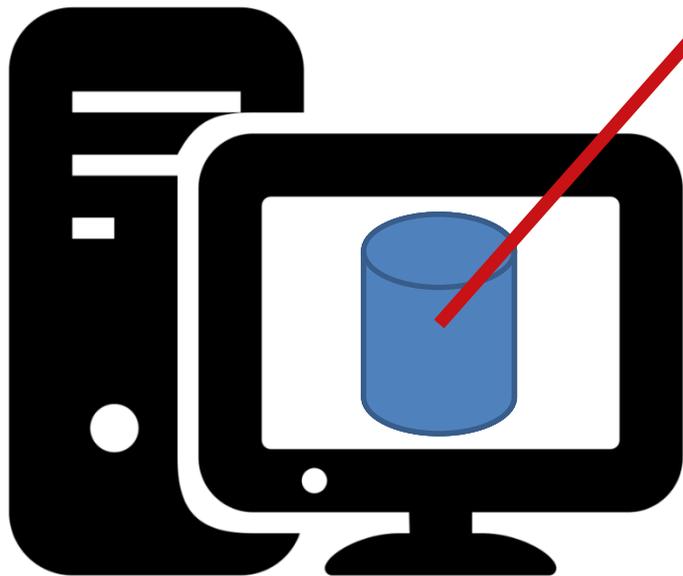
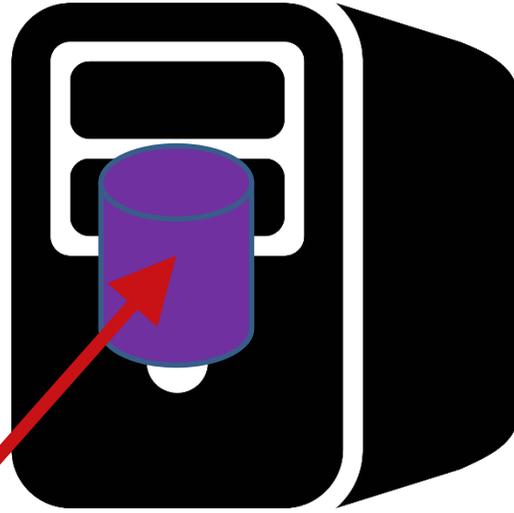
Person A

Person B



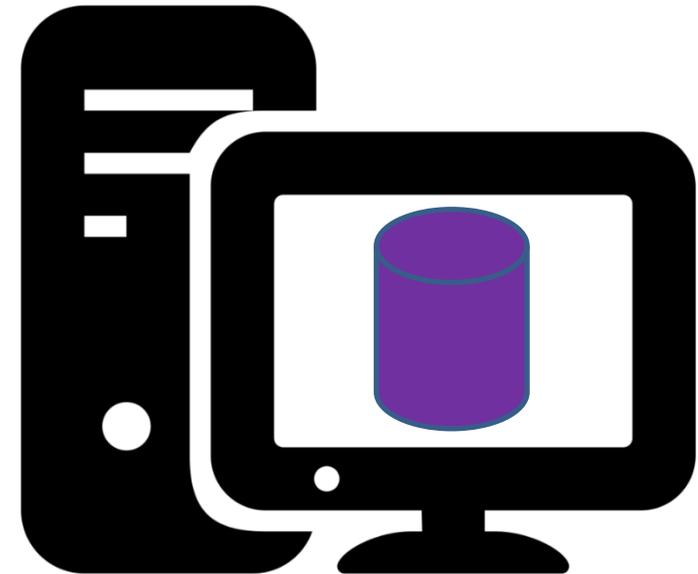
Teamarbeit mit Git

Werkzeuge ->
Teamarbeit->
commit/**push**

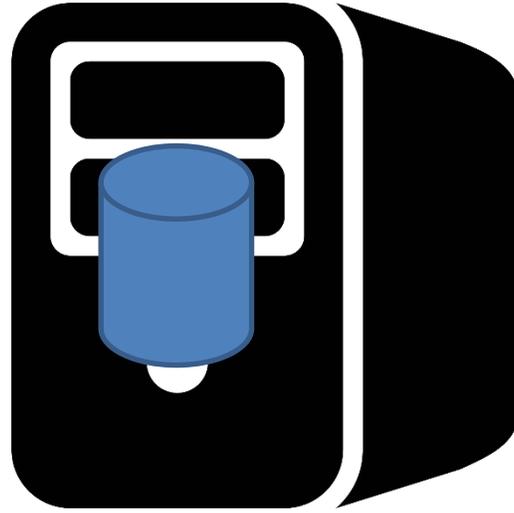


Person A

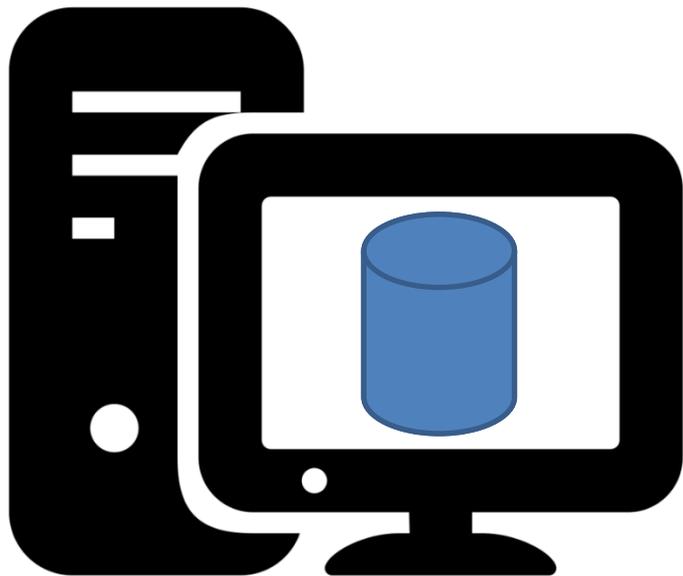
Person B



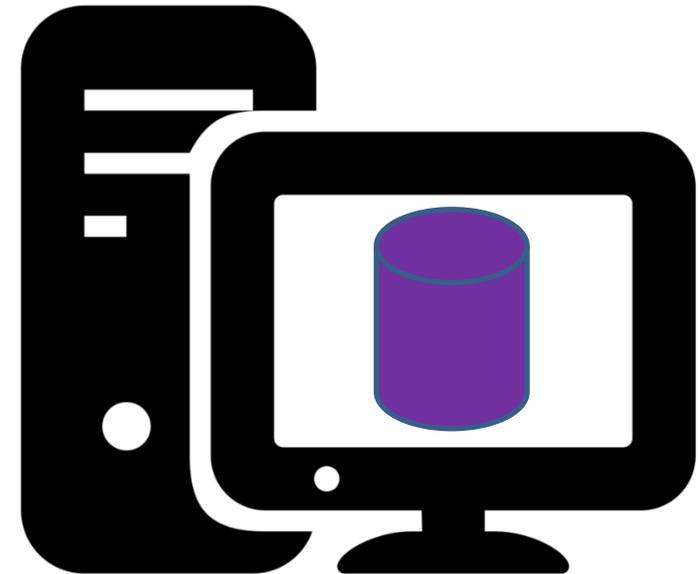
Teamarbeit mit Git



Werkzeuge ->
Teamarbeit->
commit/push

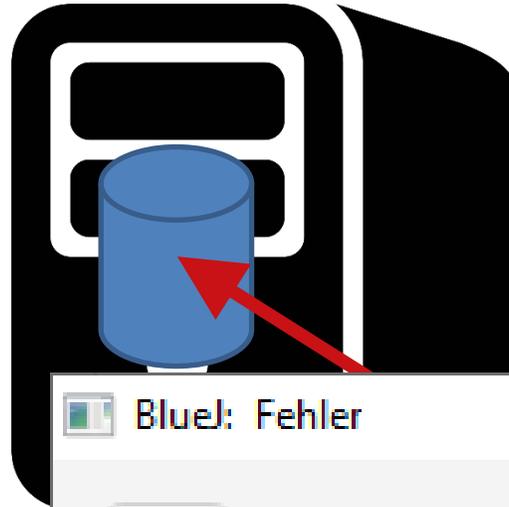


Person A

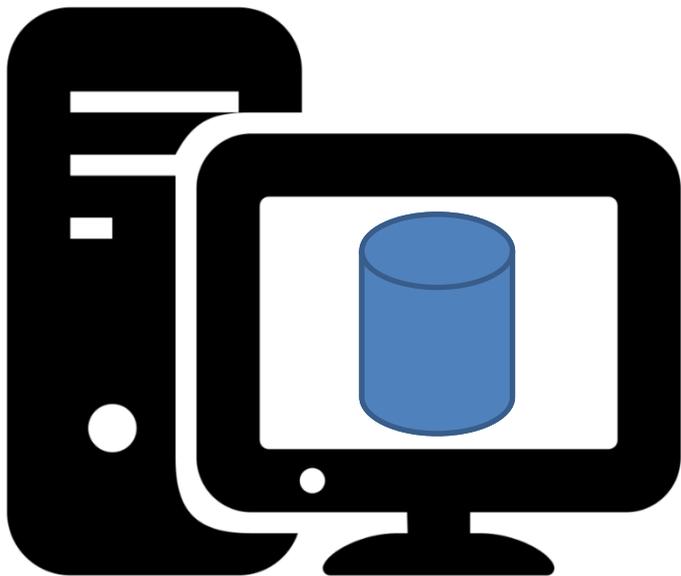
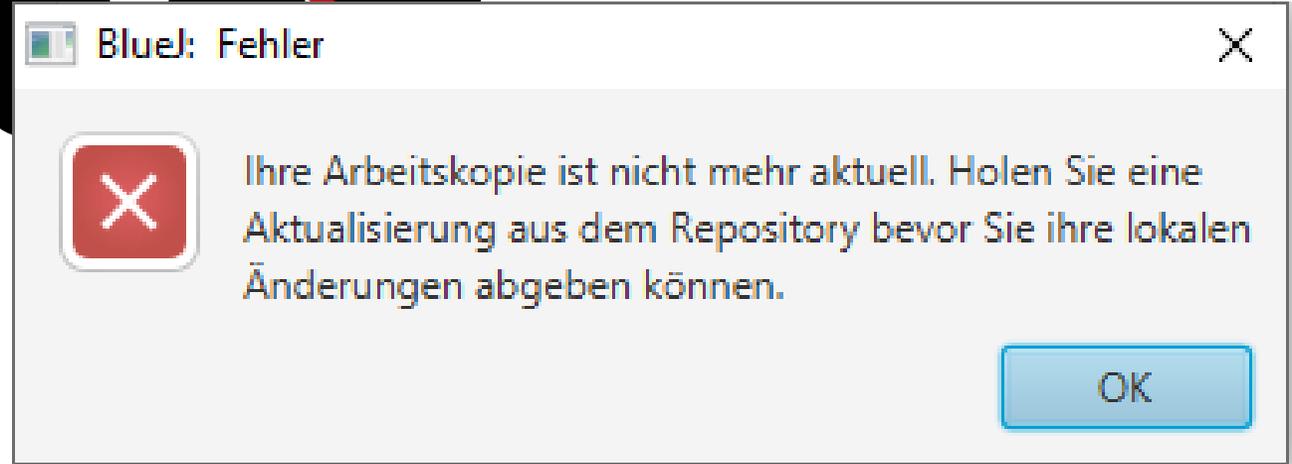


Person B

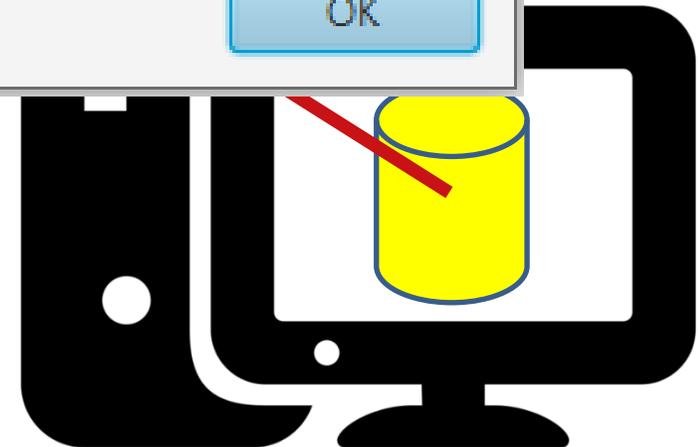
Teamarbeit mit Git



Werkzeuge ->
Teamarbeit->
commit/**push**

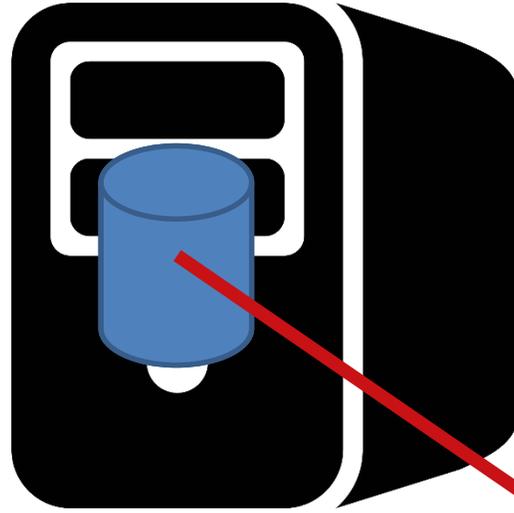


Person A

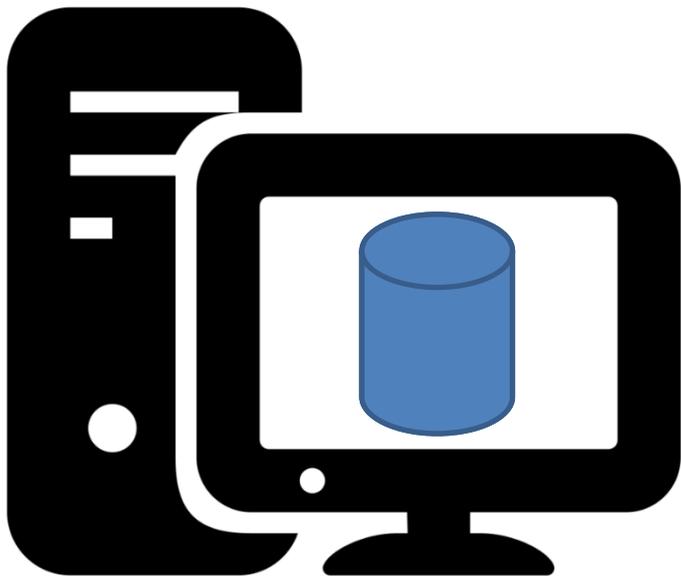


Person B

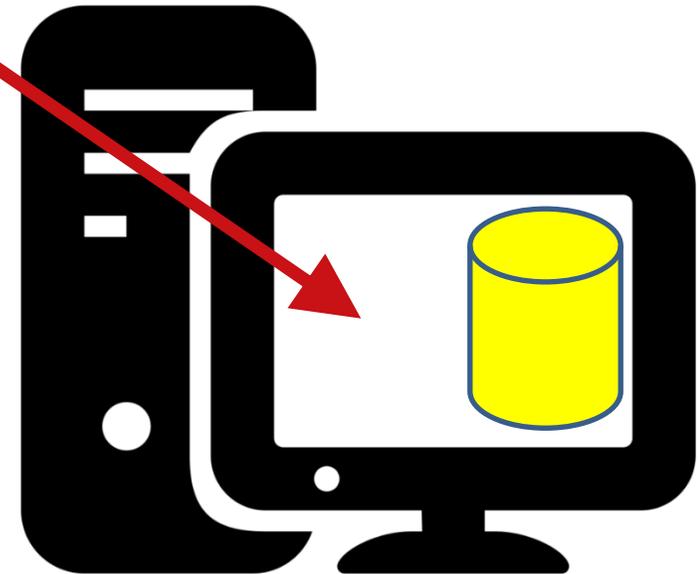
Teamarbeit mit Git



Werkzeuge ->
Teamarbeit->
Aktualisieren/pull



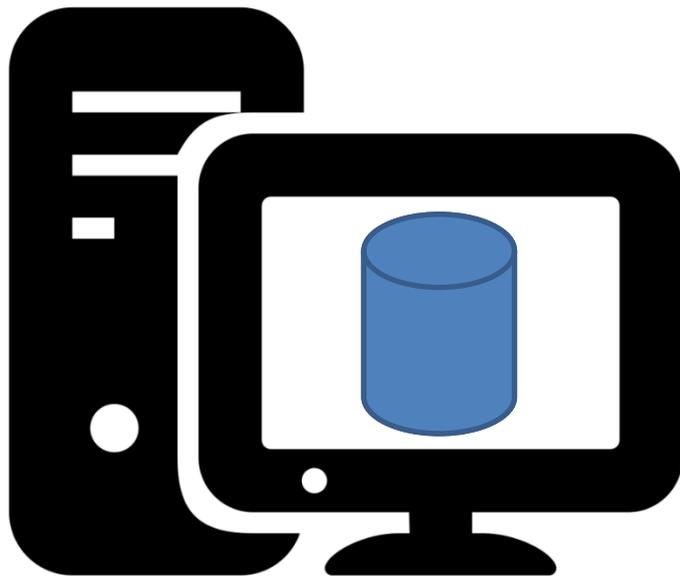
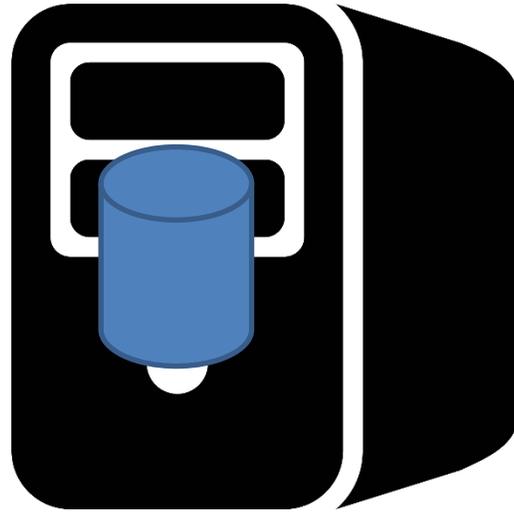
Person A



Person B

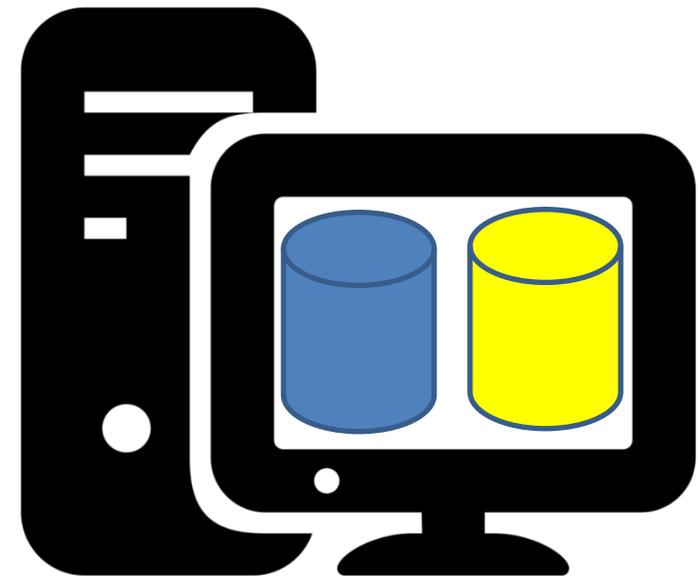
Teamarbeit mit Git

Fall 1: Änderungen bedingen sich **nicht** -> automatische Zusammenführung (**merge**) möglich -> **push** sofort möglich



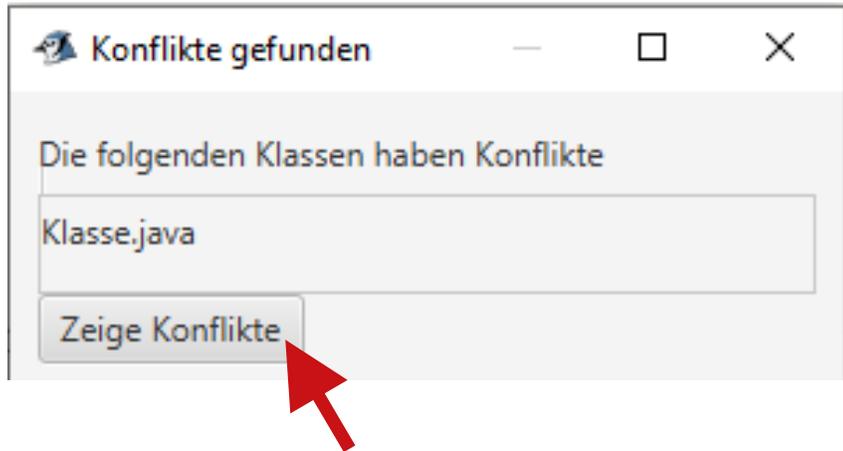
Person A

Fall 2: (hier abgebildet) Änderungen bedingen sich -> **Konflikt** -> Zusammenführung muss **händisch** durch Person B erfolgen -> **Commit** einer neuen gemeinsamen Version



Person B

Teamarbeit mit Git



Mein Code

Code vom Partner

```
public void werBistDu(){  
    <<<<<<< HEAD  
        System.out.println("Ich bin Person B");  
    =====  
        System.out.println("Ich bin Person A");  
    >>>>>>> 5ae73c1a2b519115025c33ad889ebce91a04a70d  
}
```

Teamarbeit mit Git

- Übernehmen des eigenen Codes oder

The diagram illustrates a Git commit history. A vertical bar on the left shows the progression of commits. The top commit is labeled 'HEAD' and contains the code: `public void werBistDu(){ System.out.println("Ich bin Person B");`. Below it is a commit with the code: `System.out.println("Ich bin Person A");`. A red dashed line indicates a merge operation from the lower commit back to the 'HEAD' branch. The commit hash `5ac73e1a2b519115025e33ad889ebcc91a04a70d` is shown below the merge line. The code blocks are highlighted in yellow, and the merge line is highlighted in red.

```
public void werBistDu(){  
    System.out.println("Ich bin Person B");  
    System.out.println("Ich bin Person A");  
}
```

5ac73e1a2b519115025e33ad889ebcc91a04a70d

Teamarbeit mit Git

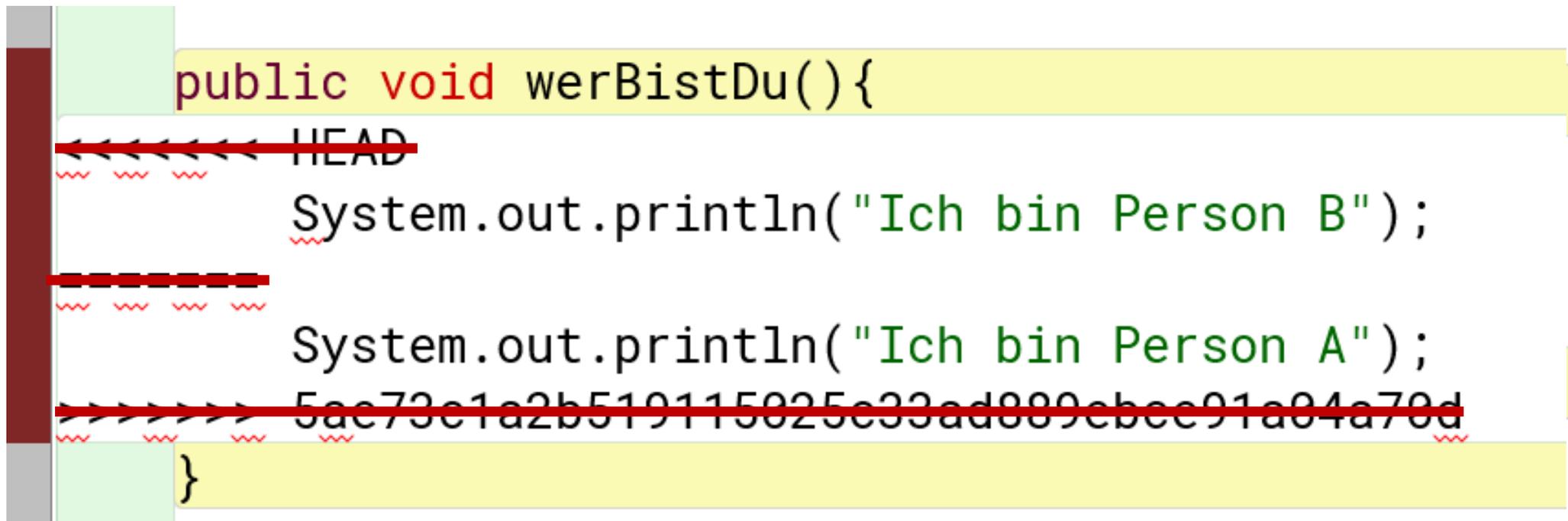
- Übernehmen des eigenen Codes oder
- Übernehmen des Codes vom Partner oder

```
public void werBistDu(){  
←←←←← HEAD  
System.out.println("Ich bin Person B");  
-----  
System.out.println("Ich bin Person A");  
→→→→→ 5ac73e1a2b519115025e33ad889ebcc91a04a70d  
}
```

Teamarbeit mit Git

- Übernehmen des eigenen Codes oder
- Übernehmen des Codes vom Partner oder
- Code von beiden übernehmen oder anders zusammenführen

```
public void werBistDu(){  
    System.out.println("Ich bin Person B");  
    System.out.println("Ich bin Person A");  
    5ac73e1a2b519115025e33ad889ebcc91a04a70d  
}
```



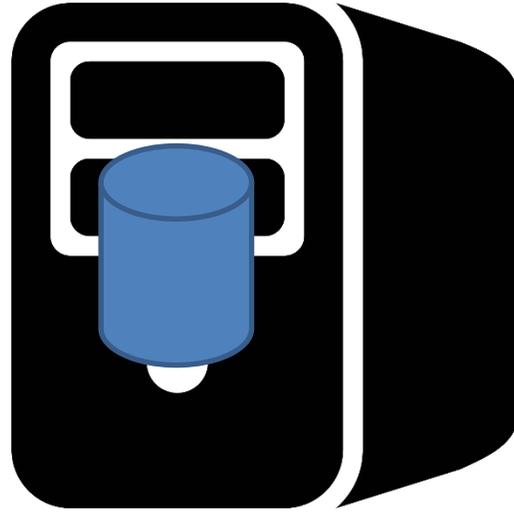
Teamarbeit mit Git

- Übernehmen des eigenen Codes oder
- Übernehmen des Codes vom Partner oder
- Code von beiden übernehmen oder anders zusammenführen

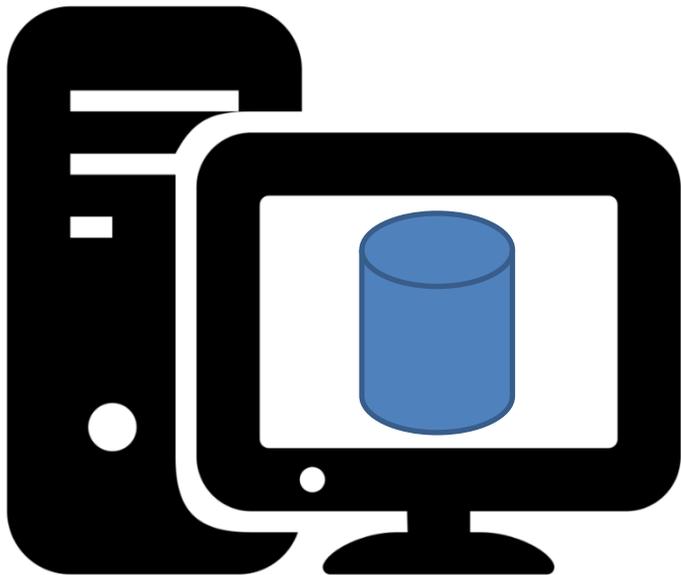
```
public void werBistDu(){  
    System.out.println("Wir sind ein Team");  
}
```

Teamarbeit mit Git

Der zusammengeführte Code muss ebenfalls als neue Version des Projekts gespeichert werden.

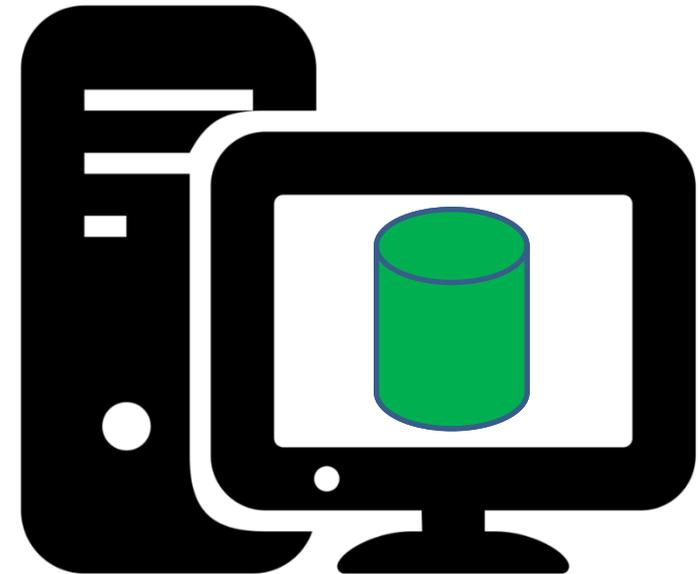


Werkzeuge ->
Teamarbeit->
commit/push



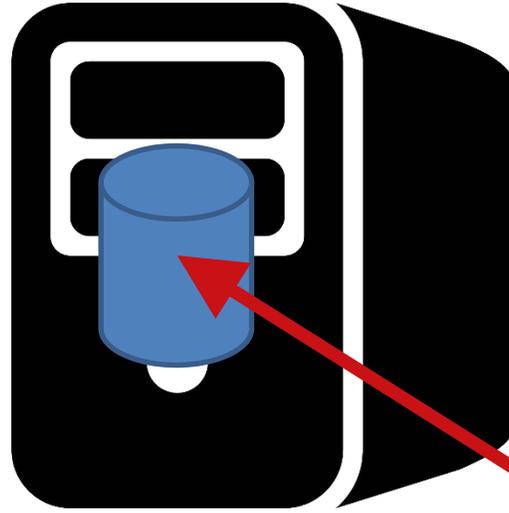
Person A

Person B

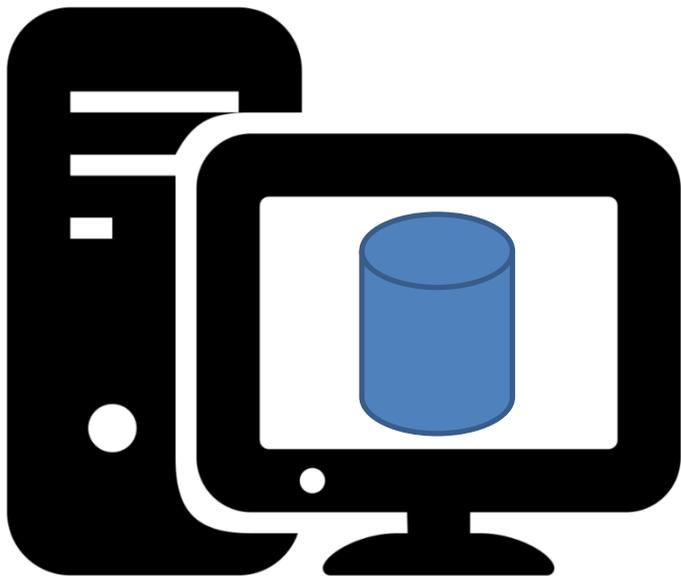


Teamarbeit mit Git

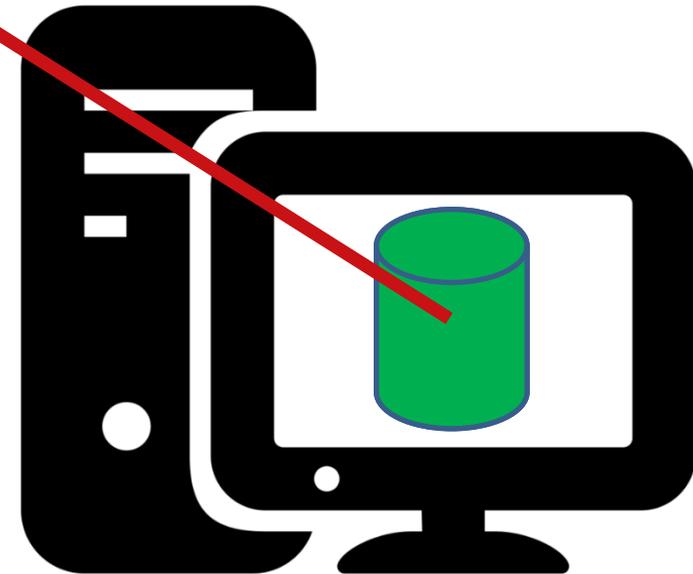
Der zusammengeführte Code muss ebenfalls als neue Version des Projekts gespeichert werden.



Werkzeuge ->
Teamarbeit->
commit/**push**



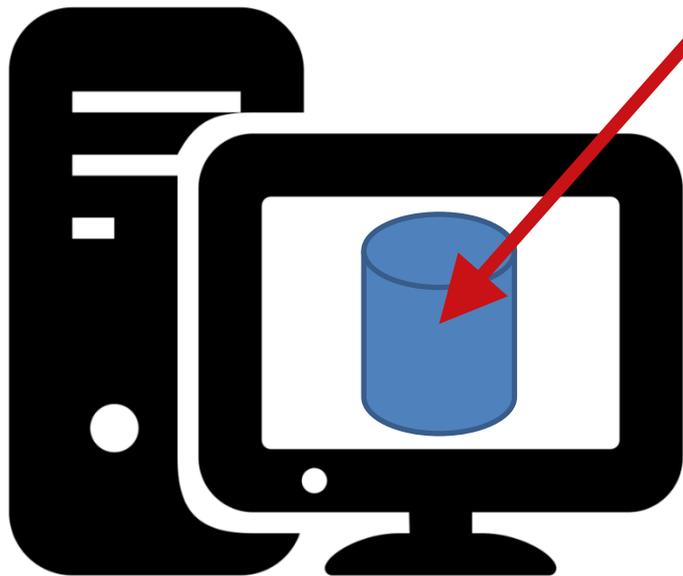
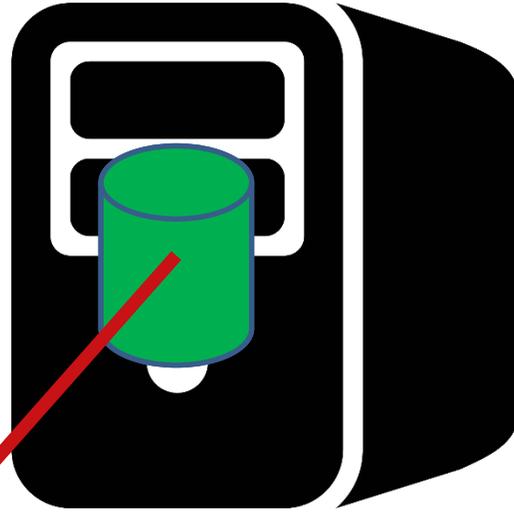
Person A



Person B

Teamarbeit mit Git

Werkzeuge ->
Teamarbeit->
Aktualisieren/pull



Person A

Person B

