



Technische  
Universität  
Braunschweig



# Programmieren 1 – Übung #1

Arne Schmidt

# Heute

SSH

Git

Wiederholung

# SSH (Wiederholung)

# SSH

Protokoll zur sicheren Kommunikation in Netzwerken.

Findet Verwendung in:

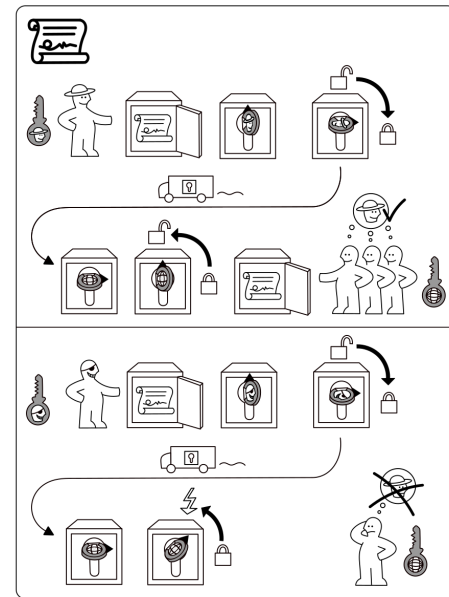
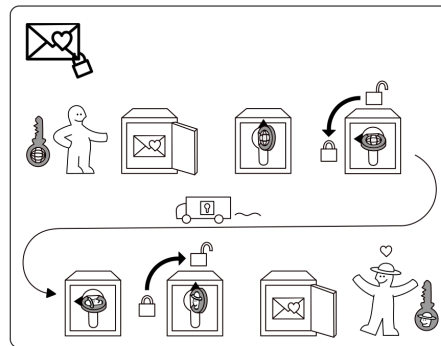
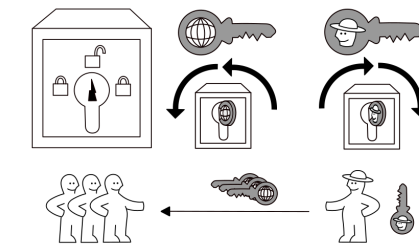
- Remote Administration (ssh)
- Daten über das Netzwerk kopieren (scp)
- Sichere FTP Verbindungen (sftp)

Basierend auf einem **Schlüsselpaar**

*Public Key*: Wird auf dem Server hinterlegt

*Private Key*: Nur lokal gespeichert, dient zur Identifikation

## PUBLIK KEY KRYPTO



idea-instructions.com/public-key/  
v1.2, CC by-nc-sa 4.0

IDEA

# Erzeugen des Schlüsselpaars

ssh-keygen

Dann kommt Aufforderung ein Passwort einzugeben  
(Optional, verschlüsselt den privaten Schlüssel)

Speicherort **nicht ändern**, wenn man sich nicht mit  
SSH näher auseinander setzen will.

```
$ cat ~/.ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAklOUpkDHrfHY17SbrmTIpNLTGK9Tjom/BWDSU  
GPl+nafzLHDTYW7hdI4yZ5ew18JH4JW9jbhUFRviQzM7xLELEVf4h9lFX5QVkbPppSwg0cda3  
Pbv7k0dJ/MTyBlWXFCR+HAo3FXRitBqxiX1nKhXpHAZsMciLq8V6RjsNAQwdsdMFvSlVK/7XA  
t3FaoJoAsncM1Q9x5+3V0Ww68/eIFmb1zuUFljQJKprX88XypNDvjYNby6vw/Pb0rwert/En  
mZ+AW40ZPnTPI89ZPmVMLuayrD2cE86Z/il8b+gw3r3+1nKatmIkjn2so1d01QraTlMqVSsbx  
NrRFi9wrf+M7Q== schacon@mylaptop.local
```

*Beispiel  
Public key*

Private key liegt dann in ~/.ssh/id\_rsa

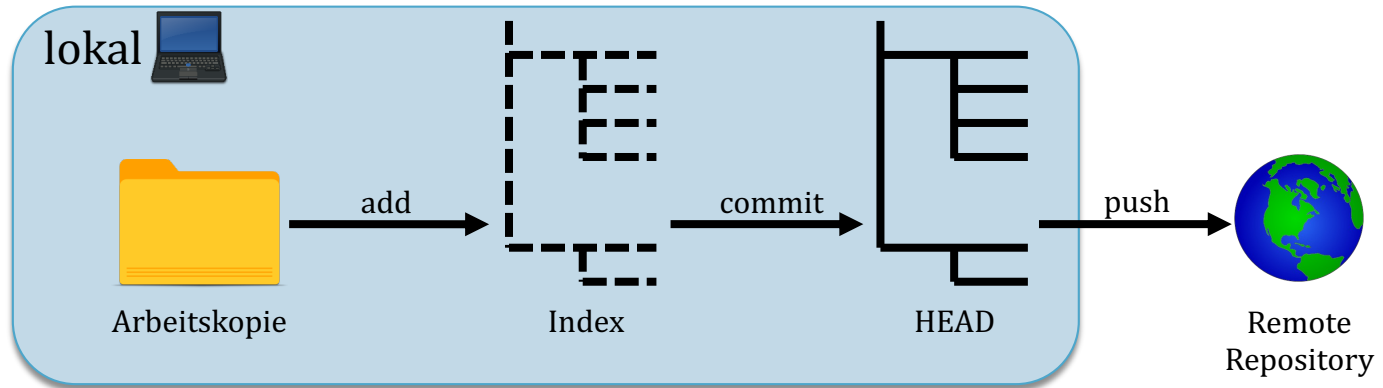
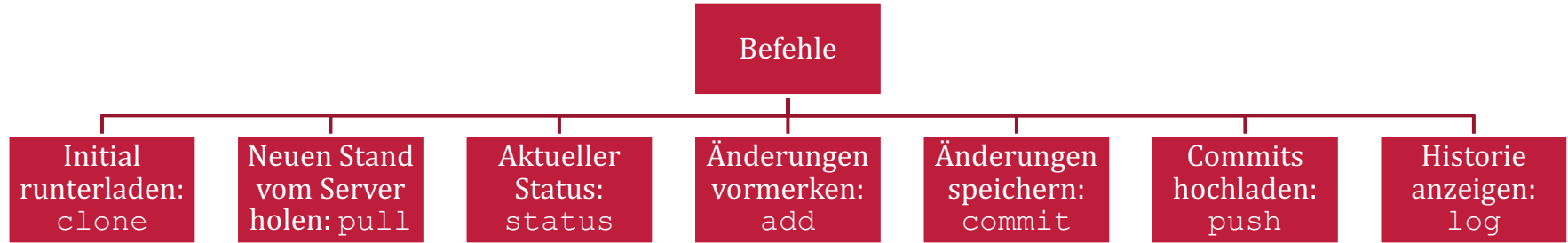
# Git

# Git – Installation

## Versionskontrollsystem (VCS)

- Gemeinsam an Projekten arbeiten
- Nachvollziehen was warum getan wurde
- Änderungen rückgängig machen
- Parallele Entwicklungsstränge
- Installation
  - Linux (Ubuntu/Debian): `apt install git`
  - macOS: z.B. via [homebrew](#) mit `brew install git`
  - Windows: z.B. via [git for windows](#)

# Git – Befehle





# Git Cheat Sheet ([blinry](#) / [CC BY-SA 4.0](#))

## Setup:

```
git config --global user.name "Your Name"
git config --global user.email "x@y.com"
```

Name und Email einstellen (einmalig).

**git init**

Erzeugt neues Repository im aktuellen Verzeichnis.

**git clone url [dir]**

Dupliziert das angegebene Repository ins Verzeichnis dir.

## Nützliches:

**git checkout datei**

Verwirft Änderungen an *datei* seit dem letzten Commit.

**git blame datei**

Wer hat diese Zeile in *datei* verbraucht?

## Hilfe:

**git help cmd**

Zeigt Hilfe für den Git-Befehl *cmd* an.

<http://git-scm.com/documentation>

Tutorials und Videos auf der offiziellen Website

## Workflow:

(Dateien bearbeiten)

Nicht zu viel auf einmal machen: Kleine Commits schaffen mehr Übersicht und verursachen weniger Konflikte.

**git status / git diff**

Welche Dateien sind noch nicht vorgemerkt? Was wurde genau geändert?

**git add datei**

Merkt aktuellen Zustand von *datei* für den nächsten Commit vor.

**git commit**

Speichert vorgemerkte Änderungen ab.

## History:

**git log [--graph]**

Zeigt die Versionsgeschichte an. Beenden mit *q*.

**git checkout name**

Wechselt zu Branch oder Commit *name*.

**git branch [-d] name**

Erzeugt [löscht] den Branch *name*.

**+ Spick-Zettel!**

**git commit -a**

Cooler Abkürzung: Addet automatisch alle nicht-neuen Dateien.

## Teamwork:

**git pull**

Immer zuerst Änderungen vom Server holen.

Konflikt?

(Konflikte beheben,  
Konfliktmarker entfernen)

**git add dateien**

**git commit**

**git push**

Schiebt die neuen Commits auf den Server.

# Nutzen des IBR Gitlab

## Aktivieren der y-Nummer

### IBR Account Management: y-Account aktivieren

y-Account aktivieren

Passwort vergessen →

Passwort ändern →

Passwort key-basiert ändern →

IZ-Mitarbeiter-Account erzeugen →

Durch die Aktivierung Ihres GITZ-y-Accounts erhalten Sie einen gleichnamigen Account im IBR, mit dem Sie Dienste des IBR nutzen können.

Benutzer:

GITZ-Passwort:

Account aktivieren

<https://www.ibr.cs.tu-bs.de/passwd/activate.html>

## Anmelden



LDAP

Standard

Username


Password

☐ Remember me

Sign in

<https://gitlab.ibr.cs.tu-bs.de>

# Git - Repository Webinterface



+

A

📄

🔗

✉

🔍 Search or go to...

Project

U

unterlagen

📌 Pinned

▼

📄 Issues

0

👤 Manage

➤

📅 Plan

➤

📄 Code

➤

📊 Monitor

➤

⚙ Settings

➤

prog1 &gt; ... &gt; material &gt; unterlagen

U

**unterlagen** 

Project ID: 5263 

🔔

▼

☆ Star

1

🍴 Forks

0

🔗

 2 Commits 

🌿

 1 Branch 

🏷

 0 Tags 

💾

 3 KiB Project Storage

 **Update README.md**  
Arne Schmidt authored 1 month ago

30988446 

main ▼

unterlagen /

+

▼

History

Find file

Edit ▼

📄 ▼

Clone ▼

📄 README

➤ Add LICENSE

➤ Add CHANGELOG

➤ Add CONTRIBUTING

➤ Set up CI/CD

⚙ Configure Integrations

 README.md

## unterlagen

In diesem Git-Repository sind die Unterlagen zu Programmieren 1 im Winter 23/24 zu finden.

 Technische  
Universität  
Braunschweig

# Git – Repository Webinterface – Clone

The screenshot shows the Git repository web interface for a project named 'unterlagen'. The interface includes a sidebar on the left with navigation options like 'Project', 'Pinned', 'Issues', 'Manage', 'Plan', 'Code', 'Monitor', and 'Settings'. The main content area displays the repository details, including the project name 'unterlagen', its ID '5263', and statistics: 2 Commits, 1 Branch, 0 Tags, and 3 KiB Project Storage. A recent commit by Arne Schmidt is shown, titled 'Update README.md'. Below this, there are buttons for 'main', 'unterlagen /', and a '+ v' dropdown. A row of buttons allows adding new files (README, LICENSE, CHANGELOG, CONTRIBUTING) or setting up CI/CD and integrations. A table lists the repository's contents, showing the 'README.md' file with its last commit and update date. At the bottom, the repository's description is provided.


Clone-Link für lokale Kopie  
Per ssh oder  
https (erfordert regelmäßiges Anmelden)

Name	Last commit	Last update
README.md	Update README.md	1 month ago

**unterlagen**

In diesem Git-Repository sind die Unterlagen zu Programmieren 1 im Winter 23/24 zu finden.

# Git – Repository Webinterface – Dateien



+

A

📄

🔗


✉

Search or go to...

Project

- unterlagen
- Pinned
- Issues 0
- Manage
- Plan
- Code
- Monitor
- Settings

prog1 > ... > material > unterlagen



**unterlagen**


Project ID: 5263

🔔

☆ Star 1

🍴 Forks 0

🔗 2 Commits 🌿 1 Branch 🏷 0 Tags 📄 3 KiB Project Storage



**Update README.md**  
Arne Schmidt authored 1 month ago

30988446

main

 unterlagen / 

+

History

Find file

Edit

📄

Clone

📄 README

🔗 Add LICENSE

🔗 Add CHANGELOG

🔗 Add CONTRIBUTING

🔗 Set up CI/CD

⚙ Configure Integrations

Name	Last commit	Last update
<a href="#">README.md</a>	Update README.md	1 month ago

📄 README.md

## unterlagen

In diesem Git-Repository sind die Unterlagen zu Programmieren 1 im Winter 23/24 zu finden.

Übersicht aller Ordner und Dateien

# Git – Repository Webinterface – Issues

prog1 > ... > material > unterlagen

**unterlagen**

Project ID: 5263

Star 1 Forks 0

2 Commits 1 Branch 0 Tags 3 KiB Project Storage

**Issues** 0

**Übersicht von Issues**  
**Hier könnt ihr Fehler in den Slides melden!**

Update README.md 30988446

History Find file Edit Clone

Add CONTRIBUTING Set up CI/CD Configure Integrations

Name	Last commit	Last update
README.md	Update README.md	1 month ago

README.md

**unterlagen**

In diesem Git-Repository sind die Unterlagen zu Programmieren 1 im Winter 23/24 zu finden.

# Git – Repository Webinterface – Account

**Euer Account**  
**Dort könnt ihr auch euren**  
**public ssh key hinterlegen!**

prog1 > ... > material > unterlagen

Search or go to...

Project

- unterlagen
- Pinned
- Issues
- Manage
- Plan
- Code
- Monitor
- Settings

Storage

Update README.md  
Arne Schmidt authored 1 month ago

30988446

main unterlagen / +

History Find file Edit Clone

README Add LICENSE Add CHANGELOG Add CONTRIBUTING Set up CI/CD Configure Integrations

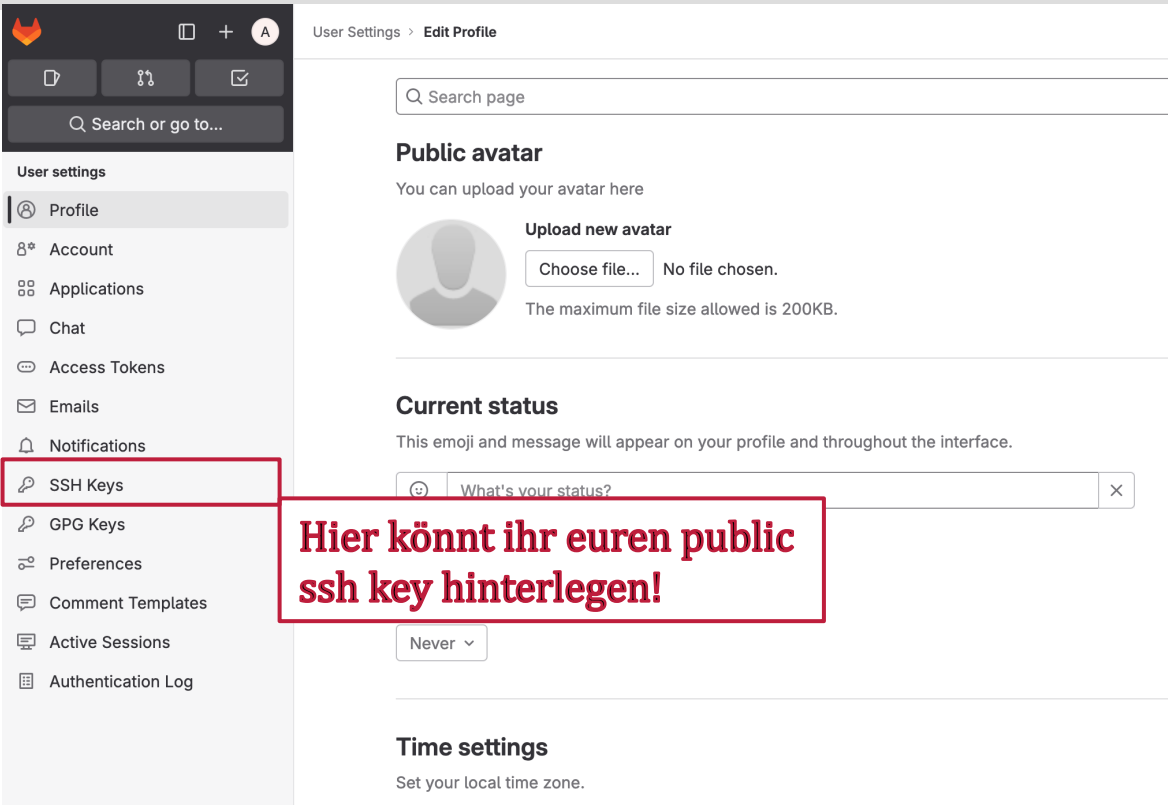
Name	Last commit	Last update
README.md	Update README.md	1 month ago

README.md

## unterlagen

In diesem Git-Repository sind die Unterlagen zu Programmieren 1 im Winter 23/24 zu finden.

# Git – Account Webinterface



User Settings > Edit Profile

Search page

## Public avatar

You can upload your avatar here

Upload new avatar

Choose file... No file chosen.

The maximum file size allowed is 200KB.

## Current status

This emoji and message will appear on your profile and throughout the interface.

What's your status?

Never

## Time settings

Set your local time zone.

**Hier könnt ihr euren public ssh key hinterlegen!**



# Git – Probleme?



Wenn ihr euch nicht im Git anmelden könnt:

- Y-Nummer freigeschaltet? (Dauert ggf 24h)  
Hier Probleme? Mail an [prog1@tu-braunschweig.de](mailto:prog1@tu-braunschweig.de)
- Daten korrekt?

Schaltet möglichst früh eure y-Nummer bei uns frei!

Je eher, desto besser können wir bei Problemen helfen  
und Stress vermeiden (z.B. Deadlines von Hausaufgaben).

# Git – Repositories für kleine Übungen

Eure Repositories werden während der kleinen Übungen erstellt, wenn die 2er Gruppen festgesetzt sind.

Sobald sie existieren, prüft bitte, ob ihr Zugriff habt.

Sollte jemand keinen Zugriff haben: Mail an [prog1@tu-braunschweig.de](mailto:prog1@tu-braunschweig.de)

# Regeln für die Git Repos

Sofern es nicht anders in der Aufgabe steht:

- Nur Source-Dateien (.py oder .java) hochladen!
- Keine .class Dateien hochladen!
- Ordnerstruktur einhalten:
  - Abgaben von Blatt [x], Aufgabe [y] müssen in den Ordner blatt[x]/pflichtaufgabe[y]/
  - Beispiel: Blatt 1 Aufgabe 2 muss in den Ordner blatt1/pflichtaufgabe2/
- Falls “.gitlab-ci.yml” im Repository existiert: Nicht ändern!

# Demo

# Wiederholung der Vorlesung

# Begriffe

## Algorithmen

- Berechnungsvorschrift
- Eigenschaften

## Datentypen

- Char
- Boolean
- Integer

## Paradigmen

- Imperativ
- Funktional
- Objektorientiert

## Programmiersprache

- Lexik
- Syntax
- Semantik

## Sprachklassen

- Maschinsprache
- Maschinenorientiert
- Problemorientiert

## Compiler

## Interpreter

# Python und Datentypen

## Pythons Lexik

- Zeichen
- Schlüsselwörter
- Operatoren

## Binärzahlen

$$(110101)_2 = 53$$

## Boolean

- Wahrheitswert
- Unäre und Binäreoperatoren

True / False

1 / 0

## Integer

- Beliebig groß
- Binäre Operatoren
- Bitweise Operatoren

3179, 5, 891, ...

## Float

- Beschränkt in Wertebereich **und** Präzision
- Vergleichsoperatoren

$1.1001101101 \cdot 2^5$

## Strings

- Beliebig lang
- Binäre Operatoren  
+ und \*

“Das ist ein String <(o.o<)”

# Binärzahlen

$$378 = 101111010$$

$$(10010)_2 = 18$$

$$42 = 101010$$

$$(10101)_2 = 21$$

$$1490 = 10111010010$$

$$(110100)_2 = 52$$

$$9315 = 10010001100011$$

$$(111111)_2 = 63$$



# Operatoren (Python)

$377 / 2$  liefert Wert \_\_\_\_\_ vom Typ \_\_\_\_\_

$5 // 2 == 2$  liefert Wert \_\_\_\_\_ vom Typ \_\_\_\_\_

"Ha " \* 3 liefert den Wert \_\_\_\_\_ vom Typ \_\_\_\_\_

$19 >> 2$  liefert den Wert \_\_\_\_\_ vom Typ \_\_\_\_\_

# Operatoren (Python)

$377 / 2$  liefert Wert 188.5 vom Typ Float

$5 // 2 == 2$  liefert Wert True vom Typ Boolean

`"Ha " * 3` liefert den Wert "Ha Ha Ha " vom Typ String

$19 >> 2$  liefert den Wert 4 vom Typ Integer

# Float

72.0 als Float

$$72 = (1001000)_2 = (1.001000)_2 \cdot 2^{(110)_2}$$

Also

Sign: 0, Exponent:  $(10000000101)_2$ , Fraction: 001000000...

1024.0 als Float

$$1024 = (1000000000)_2 = (1.0)_2 \cdot 2^{(1010)_2}$$

Also

Sign: 0, Exponent:  $(10000001001)_2$ , Fraction: 000000000...