

# Übungsstunde 1

Einführung in die Programmierung

# Organisatorisches

Unsere Namen: **Maxim Huber & Josua Bürki**

E-Mail-Adressen für Fragen zu den Übungen:

**maximhuber@ethz.ch**

**jbuerki@ethz.ch**

Unsere Webseite: **<https://n.ethz.ch/~maxihuber/>** mit Tipps, Links, den Slides und Kahoots

# Vorstellungsrunde

# Organisatorisches

Übungsabgabe immer via Git

Individuelles Feedback zu einzelnen Aufgaben via Git im File corrections

Musterlösung wird am Tag nach der Abgabe in Git veröffentlicht

- Separates Projekt uX-sol; gleich auschecken wie Vorlage (pullen)

# Aufbau der Lektionen

Besprechung der Übungen

Theorie-Input

Live Coding

Vorbesprechungen der neuen Übungen

Kahoot

Tipps und Tricks

# Warum sich EProg lohnt

Chance für gute Note

Programmieren macht Spass

Für das weitere Studium

Thomas 

# Lernziele heute

Eclipse eingerichtet

Git eingerichtet

EBNF

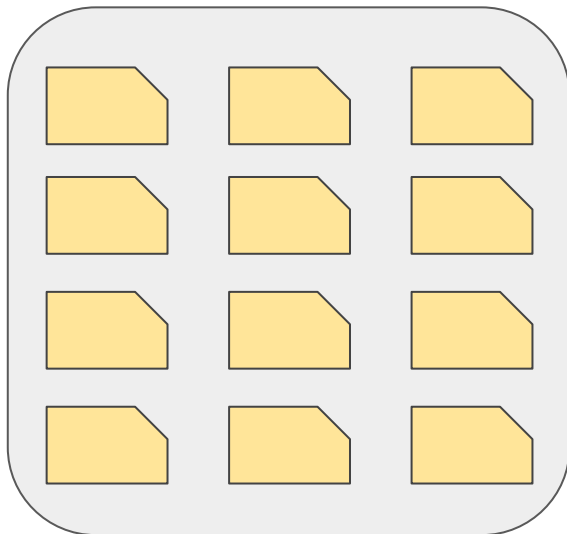
# Nachbesprechung Übung 0



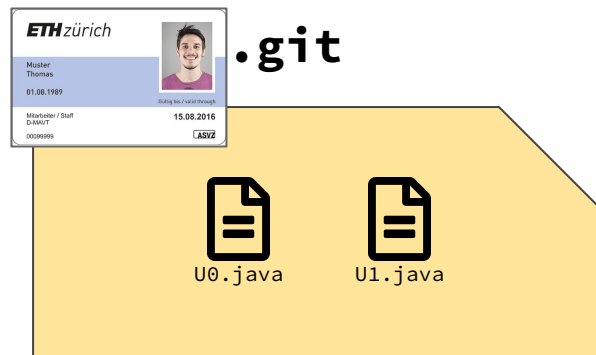
# Fragen zu Eclipse & Tipps

# Vorbesprechung Übung 1

# Git Repository

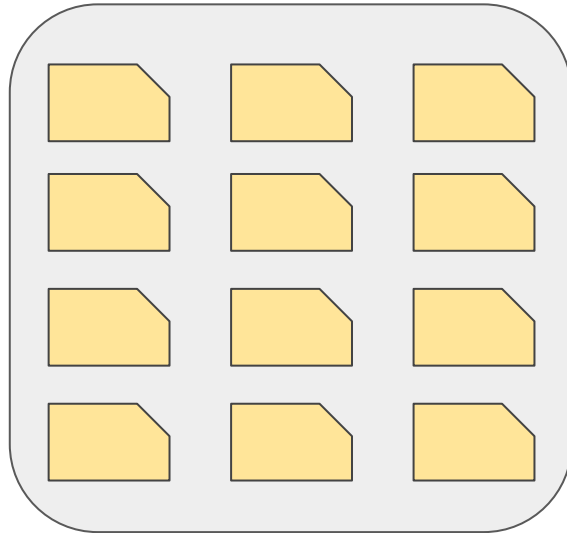


ETH Git-Server

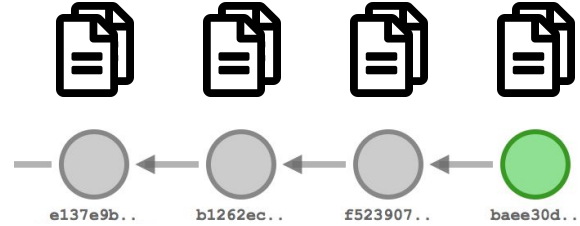


Jedes Repository auf dem Git-Server ist privat

# Git Repository



ETH Git-Server



Ältester Commit

Neuester Commit

Jedes Repository auf dem Git-Server  
Enthält eine Folge von **Commits** (die **History**)



.git

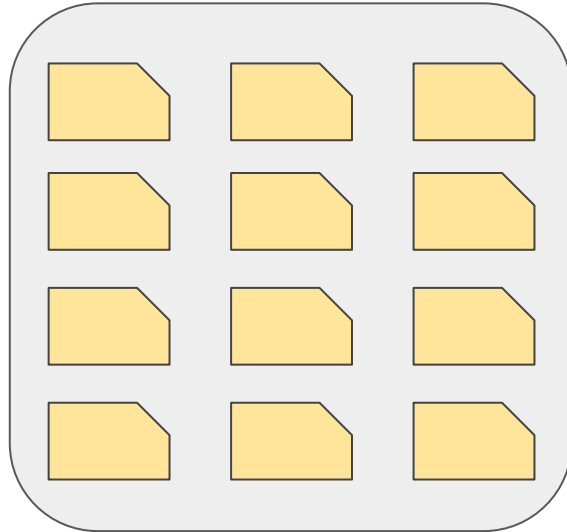


U0.java



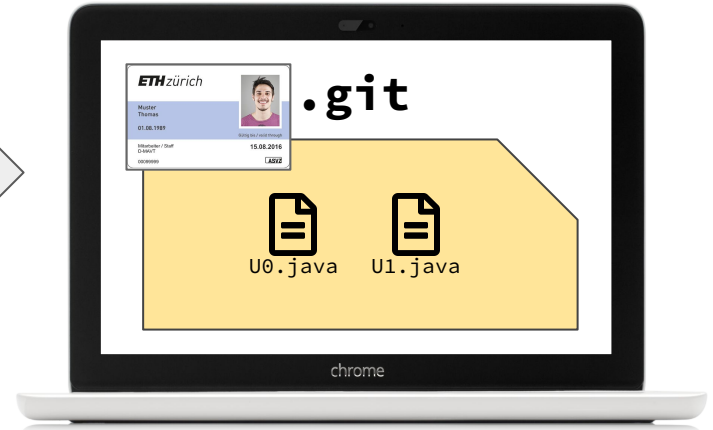
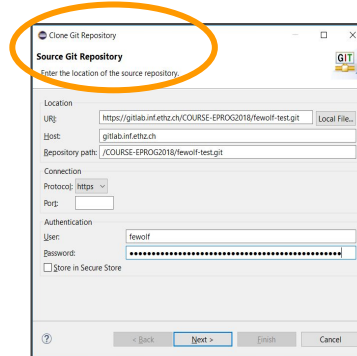
U1.java

# Git Clone: Einmaliges Einrichten



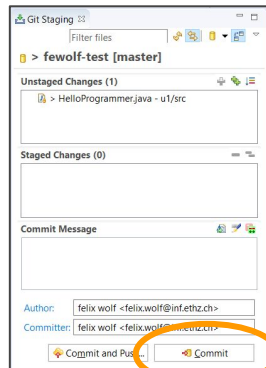
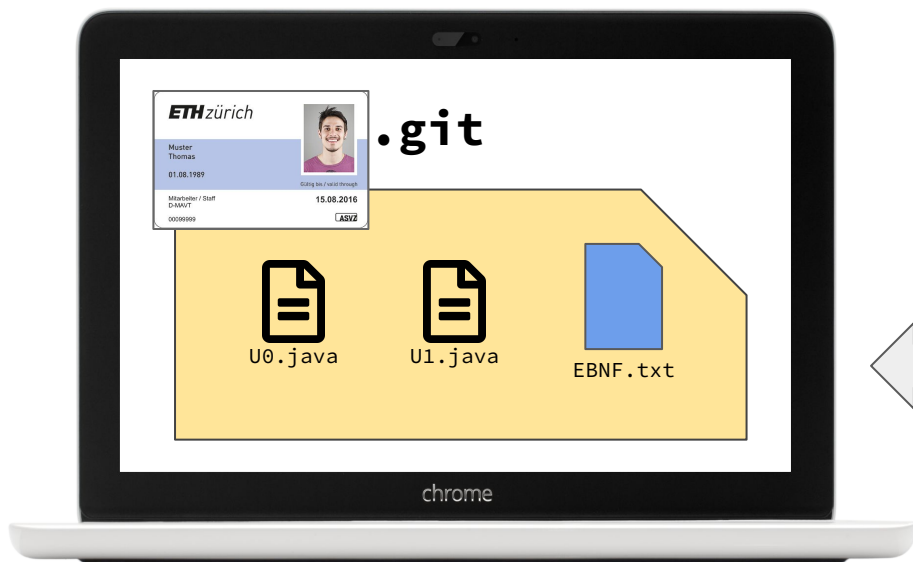
ETH Git-Server

**Clone**  
Kopiert das ganze Repository  
auf den eigenen Computer



Lokales Git-Repository

# Git Commit: Fortschritt speichern

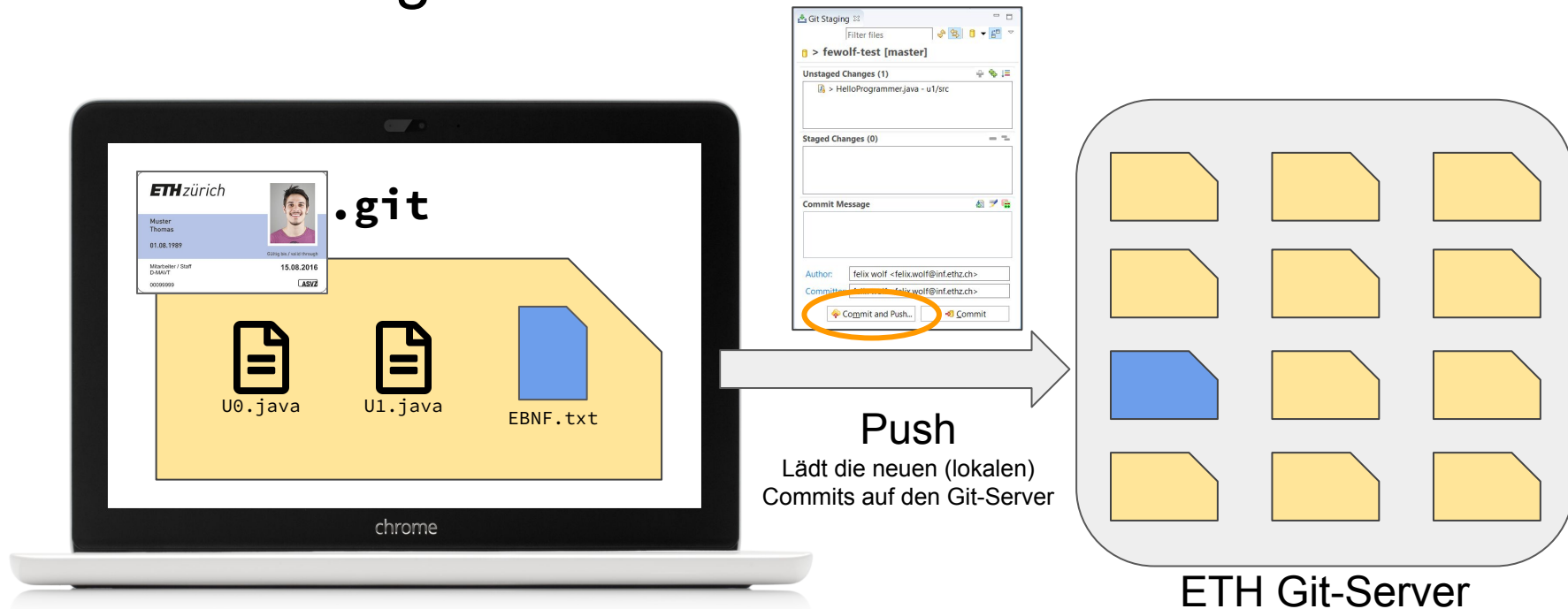


**Commit**  
Fügt neuen Commit mit  
Änderungen/neuen Dateien  
der *lokalen* History hinzu



**EBNF.txt**

# Git Push: Abgeben

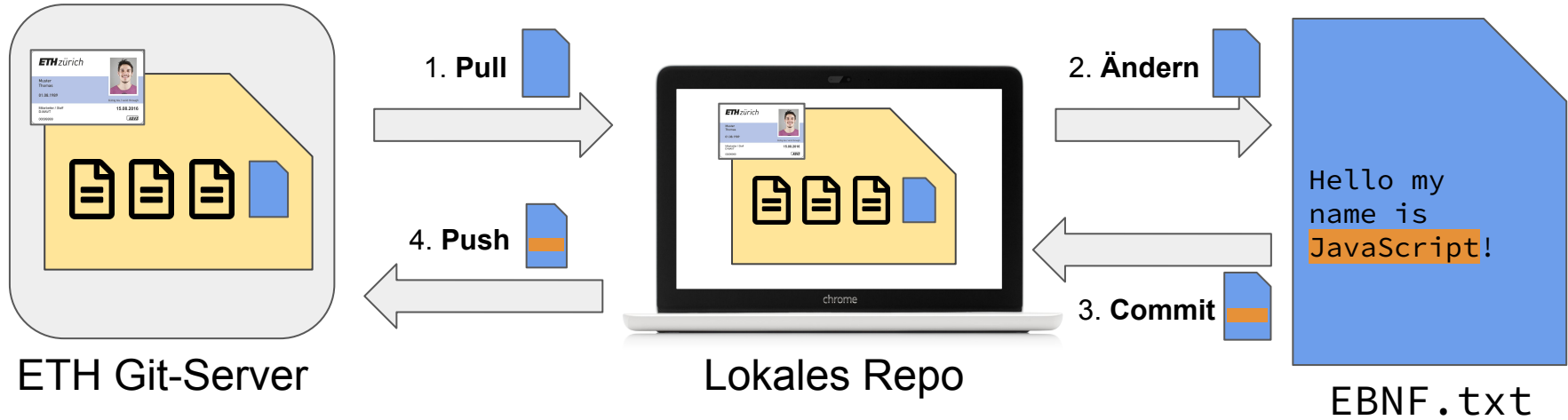


# Git Pull: Neue Aufgaben/Feedback herunterladen





# Git Pull/Push-Workflow



Git: Clone, Aus- und Einchecken

# EBNF Aufgaben

# Kombinationsmöglichkeiten

- Aufreihung                       $\langle \text{zahl1} \rangle \leq 1\ 2\ 3$
- Auswahl                         $\langle \text{zahl2} \rangle \leq 1\ |\ 2\ |\ 3\ |$
- Option                          $\langle \text{zahl3} \rangle \leq [3]$
- Wiederholung                 $\langle \text{zahl4} \rangle \leq \{4\}$
- Rekursion                      $\langle \text{zahl5} \rangle \leq [\langle \text{zahl5} \rangle]\ 5$

## Aufgabe 4 a)

Erstellen Sie eine Beschreibung `<geradezahl>`, die als legale Symbole alle geraden Zahlen (d.h. Zahlen, die ohne Rest durch 2 teilbar sind) zulässt. Beispiele sind 2, 4, 10, -20.

## Aufgabe 4 b)

Zeigen Sie in einer Tabelle, dass Ihre Beschreibung das Symbol “28” als gerade Zahl erkennt.

# Zusatzaufgaben

- Erstellen Sie eine Beschreibung `<palindrom>`, welche als legale Symbole alle Zahlen zulässt, die von Vorne und Hinten gleich gelesen werden und die nur die Ziffern von 1 bis 4 verwenden. Beispiele sind 11, 232, 444
- Erstellen Sie eine Beschreibung `<five>`, welche alle Summen von positiven Zahlen zulässt, welche 5 ergeben. Beispiele sind “1 + 4”, “2 + 1 + 1 + 1”, “5”
- Erstellen Sie eine Beschreibung `<oddEight>`, welche alle Zahlen zulässt, die die Ziffer 8 eine ungerade Anzahl mal enthalten. Beispiele sind 8, 128, 8881

Kahoot!



\*\*\*\*\*

Gegeben sei die EBNF-Beschreibung von *value*

*digit*  $\Leftarrow$   $\boxed{0} \mid \boxed{1} \mid \boxed{2} \mid \boxed{3} \mid \boxed{4} \mid \boxed{5} \mid \boxed{6} \mid \boxed{7} \mid \boxed{8} \mid \boxed{9}$

*separator*  $\Leftarrow$   $\boxed{-}$

*char*  $\Leftarrow$   $\boxed{A} \mid \boxed{B} \mid \boxed{C} \mid \boxed{D} \mid \boxed{E} \mid \boxed{F} \mid \boxed{a} \mid \boxed{b} \mid \boxed{c} \mid \boxed{d} \mid \boxed{e} \mid \boxed{f}$

*num*  $\Leftarrow$  *digit* { [ *separator* ] *digit* }

*int*  $\Leftarrow$  *digit* { *digit* }

*real*  $\Leftarrow$  *digit* { *digit* } [  $\boxed{.}$  *digit* { *digit* } ]

*cd*  $\Leftarrow$  *char* | *digit*

*hexa1*  $\Leftarrow$  *cd* { *cd* }

*hexa2*  $\Leftarrow$  *digit* { *digit* }  $\boxed{h}$

*hexa*  $\Leftarrow$  *hexa1* | *hexa2*

*value*  $\Leftarrow$  *num* | *real* | *int* | *hexa*

# Gültig oder nicht (für *value* )

- 1245
- 00972
- 00100h
- 1a00
- 1a00h
- 1\_000\_000
- 209AB
- 001ab.001h
- 4.9901
- deadface
- 4\_000.0
- 00100H
- 001ab.001
- 0x0ABC