252-0027

Einführung in die Programmierung

Übungen

Woche 2: Eclipse und EBNF

Timo Baumberger
Departement Informatik
ETH Zürich

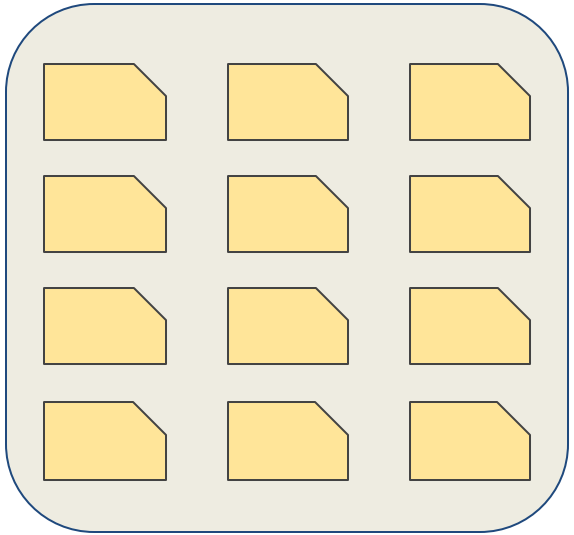
Organisatorisches

- Mein Name: Timo Baumberger
- Bei Fragen: tbaumberger@ethz.ch
- Neue Aufgaben: **Dienstag Abend** (im Normalfall)
- Abgabe der Übungen bis **Dienstag Abend (23:59)** Folgewoche
  - Abgabe immer via Git
  - Lösungen in separatem Projekt auf Git

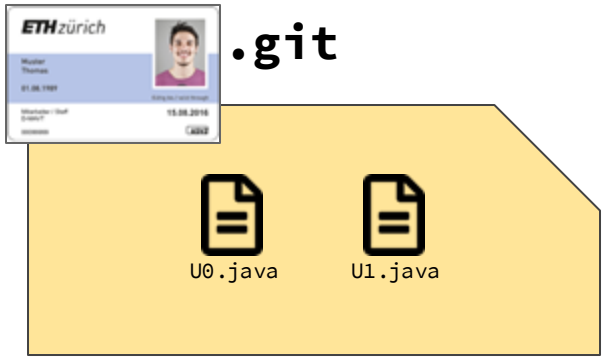
Inhalt

- Git Einführung
- EBNF Beispiele
- EBNF Rekursion
- Ableitungsbaum / Ableitungstabelle
- EBNF Aufgaben

Git Repository

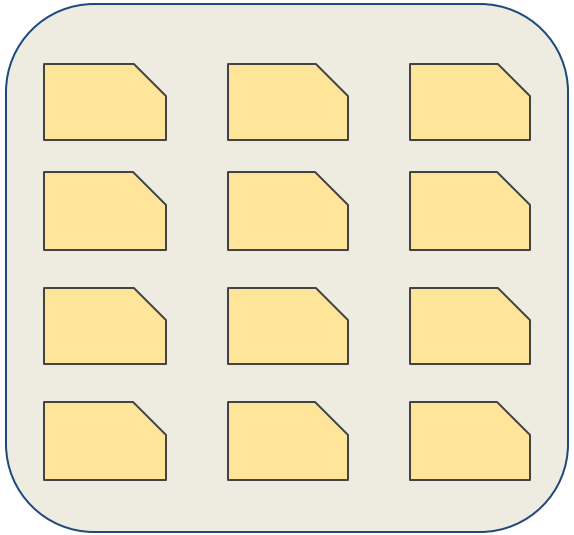


ETH Git-Server

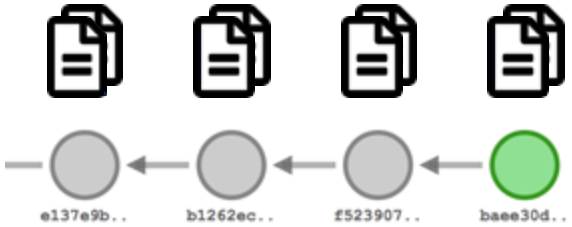


Jedes Repository auf dem Git-Server ist privat

Git Repository

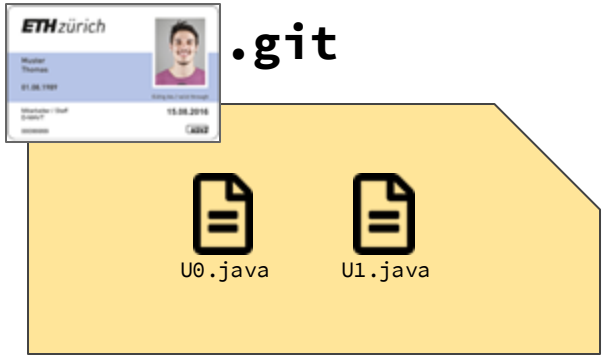


ETH Git-Server

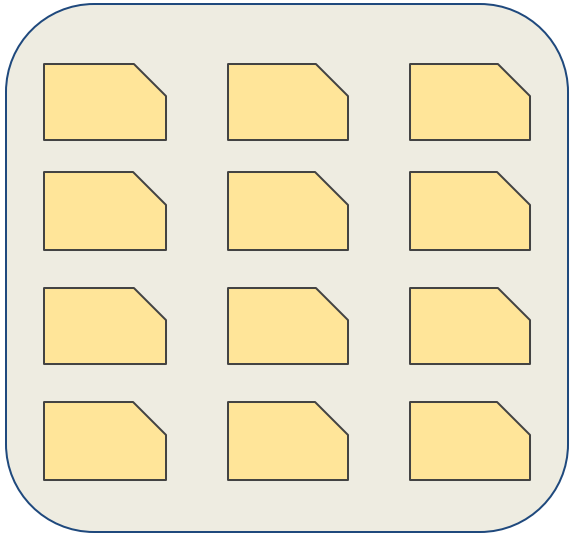


Ältester Commit Neuester Commit

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



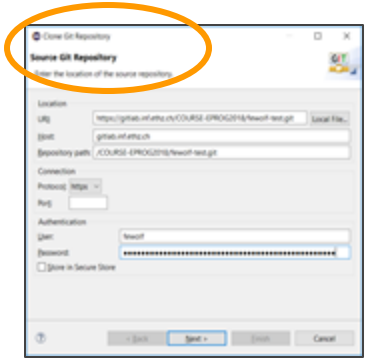
Git Clone: Einmaliges Einrichten

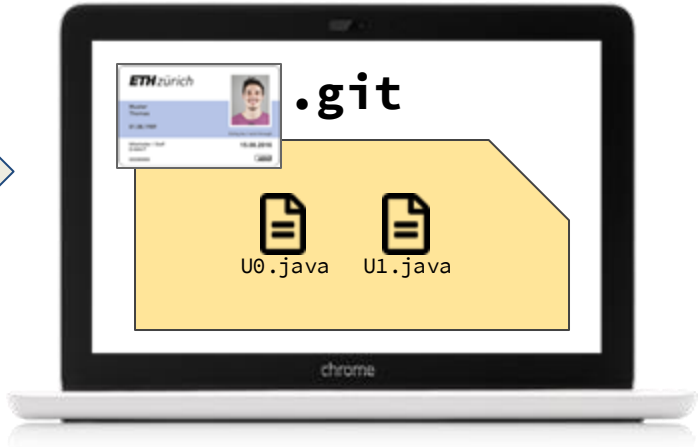


ETH Git-Server

Clone

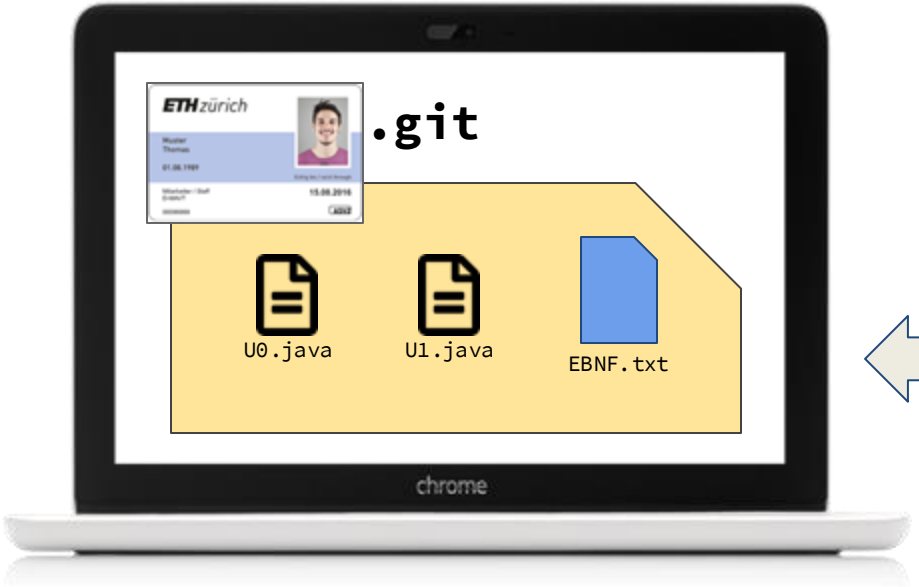
Kopiert das ganze Repository auf den eigenen Computer

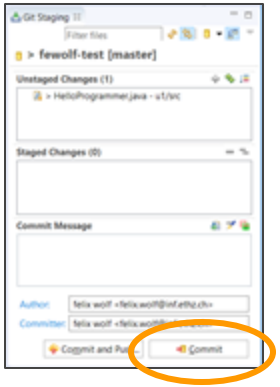




Lokales Git-Repository

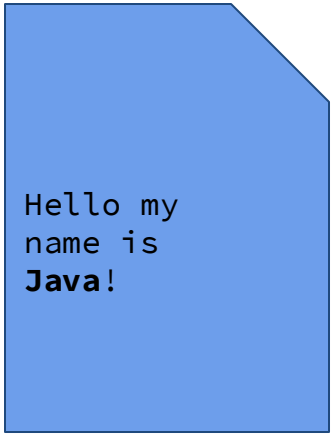
Git Commit: Fortschritt speichern





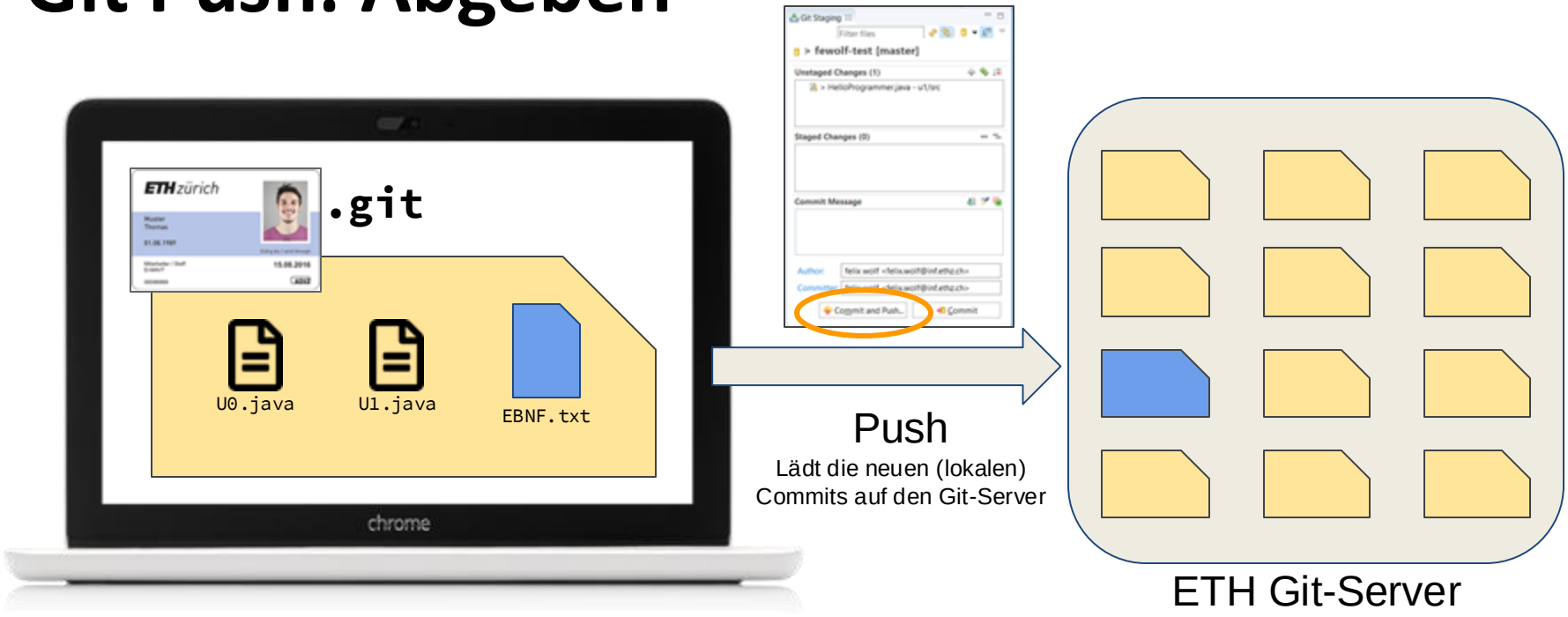
Commit

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

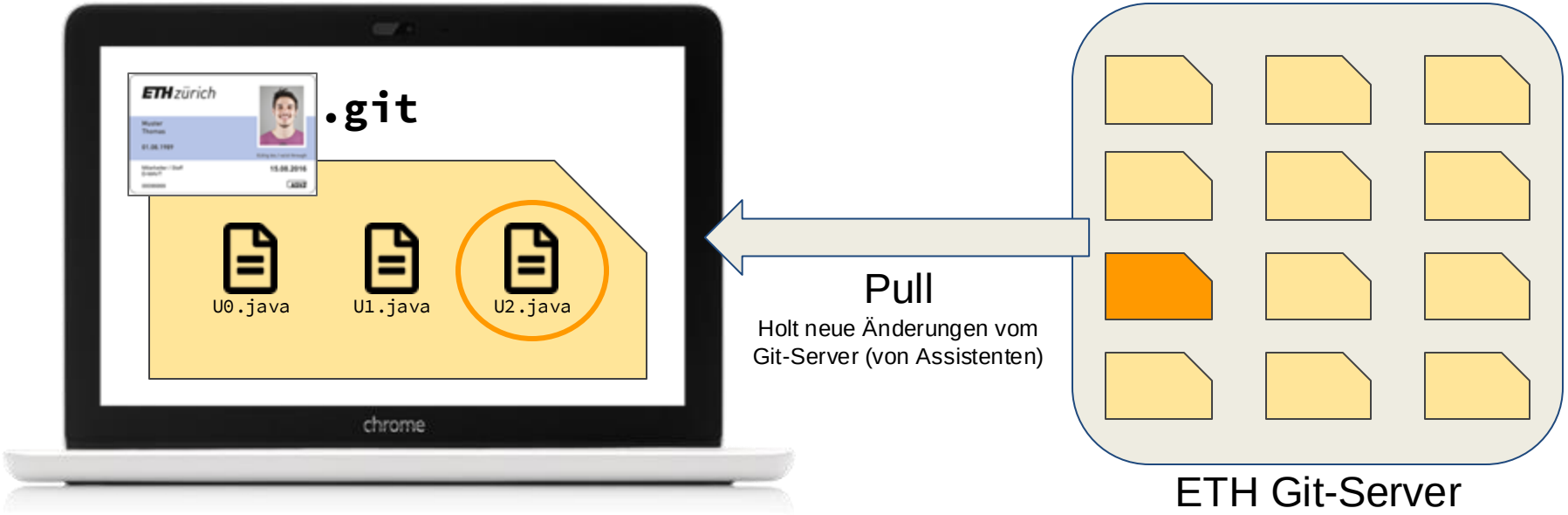


EBNF.txt

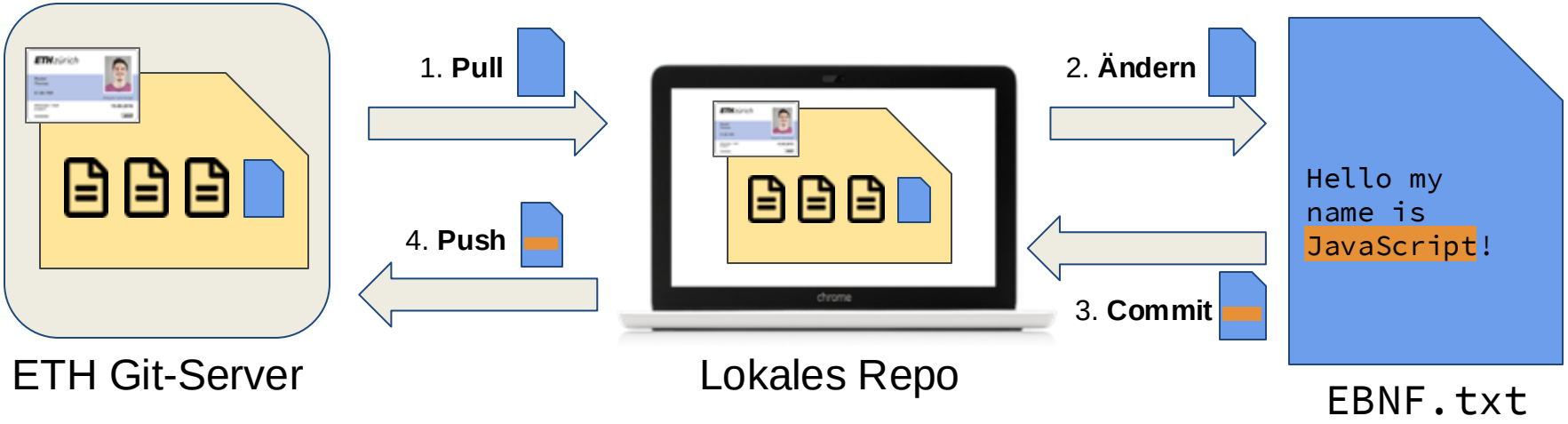
Git Push: Abgeben



Git Pull: Aufgaben / Feedback laden

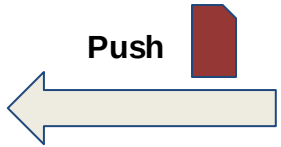


Git Pull/Push-Workflow

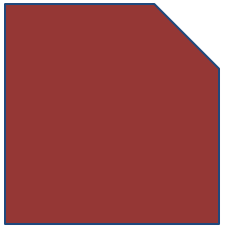


Git Repository ändert sich!

ETH Git-Server

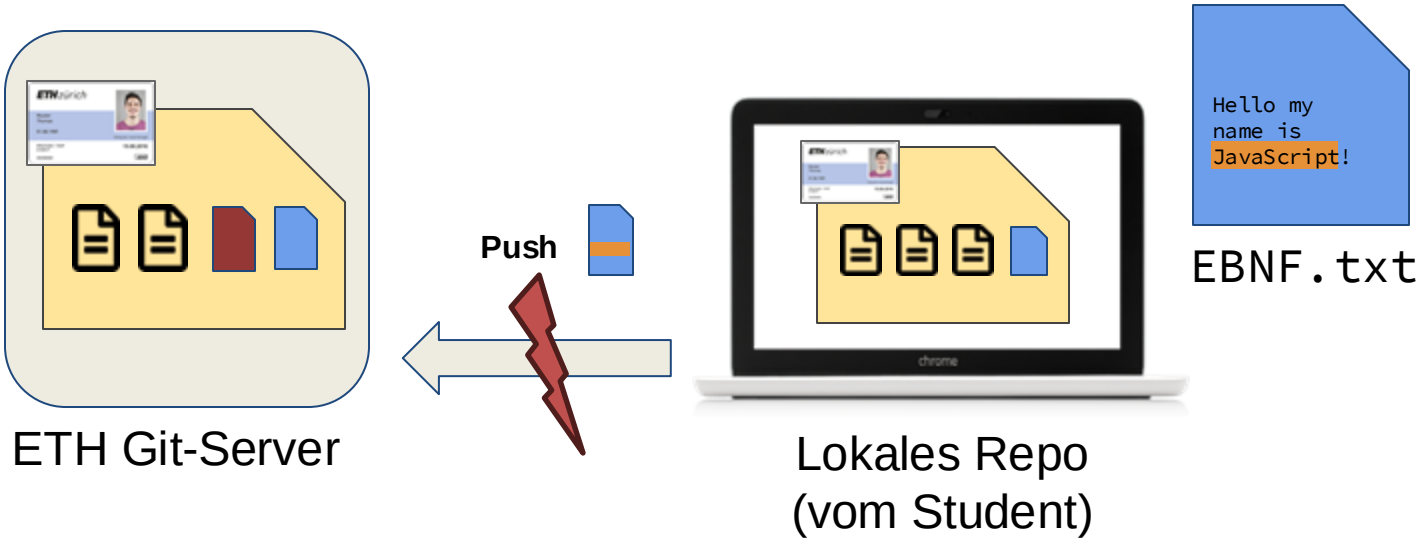


Lokales Repo
(von eurem TA)

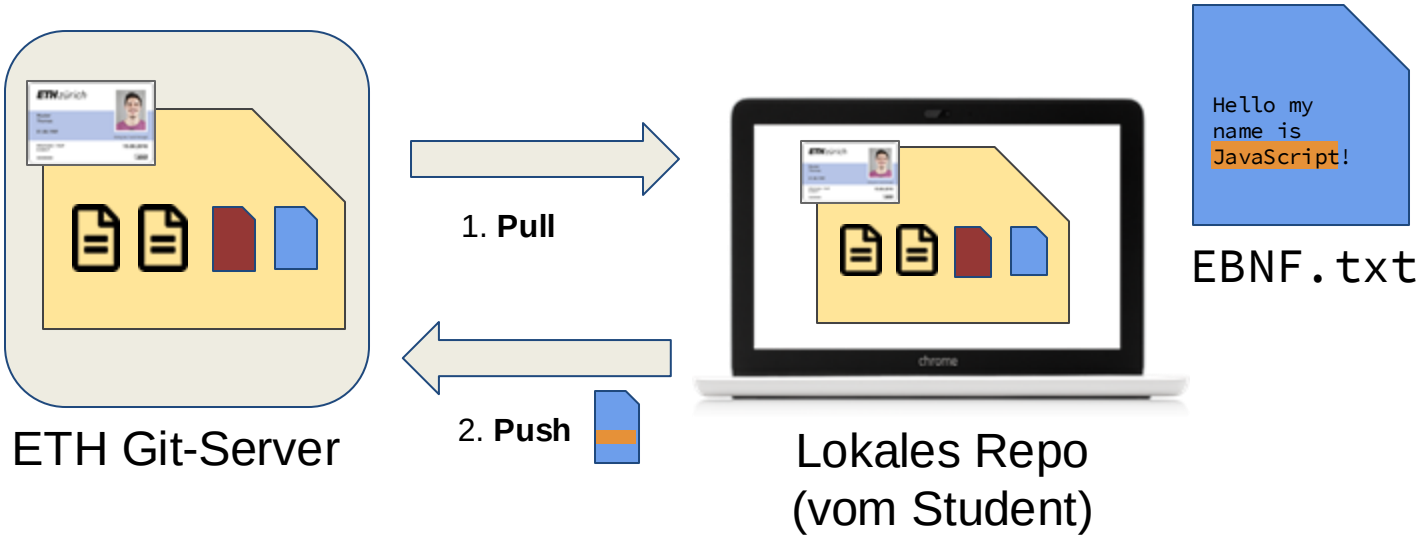


`Feedback.txt`

Git Merge Conflict



Git Merge Conflict verhindern



Immer zuerst pull dann push!

Interesse an Git geweckt?

- Fortgeschrittene Konzepte können hier trainiert werden (https://learngitbranching.js.org/)
- Allgemeine Git Dokumentation (https://git-scm.com/docs/user-manual)

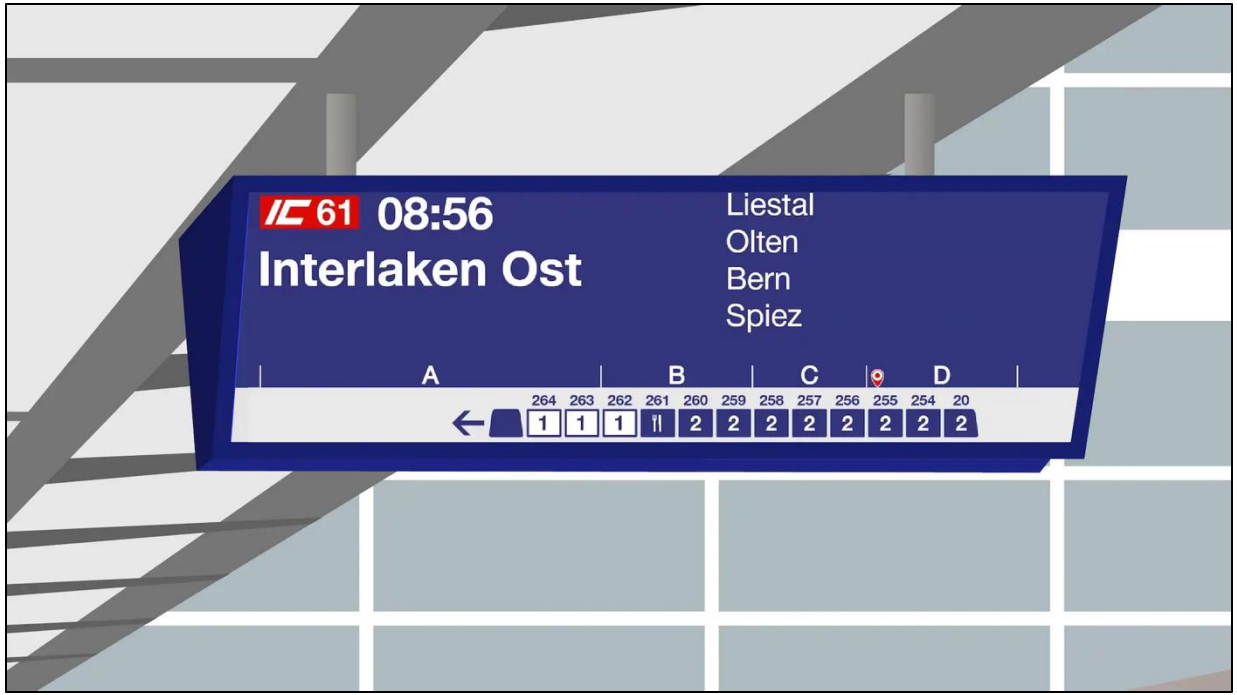
Git: Clone, Aus- und Einchecken

(Demo)

Häufige Git-Fehler vermeiden

- Vor erster Benutzung auf gitlab.inf.ethz.ch anmelden
- In U01 Aufgabe 4, <nethz-account> im Link inklusive <> ersetzen
- In U01 Aufgabe 4, bei Problemen Link per Hand abtippen
- Files/Directories/Projects/… immer in Eclipse löschen bzw. ändern (nie im File Explorer/Finder)
- Immer Pull und dann Push
- “Commit und Push” statt nur “Commit”
  - Falls Sie auf Commit geklickt haben, drücken Sie manuell per Rechtsklick auf das Repository und wählen Sie “Push” oder “Push to origin” aus
- Importieren bei neuen Projects nicht vergessen!

EBNF – Einfaches Beispiel – Bahnhof



Quelle: SBB

EBNF – Einfaches Beispiel – Bahnhof

EBNF – Einfaches Beispiel – Bahnhof

EBNF-Beschreibung von Anzeigetafel

<zugbezeichnung> <= IC61 | RE | S3 | EC
<abfahrztszeit> <= <stunde> : <minute>
<zielbahnhof> <= Zürich HB | Bern | Basel | Wil | Chur | Interlaken Ost
<stunde> = ?
<minute> = ?
<anzeigetafel> <= <zugbezeichnung> <abfahrtszeit> <zielbahnhof>

EBNF – Beispiel Programmiersprache

```
<programm> <= PROGRAM <bezeichner>
              BEGIN
               { <zuweisung> ; }
               END
<zuweisung> <= <bezeichner> := ((<zahl>|<bezeichner>)|<string>)
<bezeichner> <= <buchstabe> {<buchstabe>|<ziffer>}
<zahl> <= [-|+] <ziffer> {<ziffer>}
<string> <= "{<buchstabe>|<ziffer>}"
<buchstabe> <= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<ziffer> <= 0|1|2|3|4|5|6|7|8|9
```

Wie prüfe ich, was legal ist?

1. Informeller Beweis
2. Tabellen
3. Ableitungsbaum
4. Graphische Darstellung

JETZT

Quelle:
Wikipedia

```
<programm> <= PROGRAM <bezeichner>
              BEGIN
               { <zuweisung> ; }
               END
<zuweisung> <= <bezeichner> := ((<zahl>|<bezeichner>)|<string>)
<bezeichner> <= <buchstabe> {<buchstabe>|<ziffer>}
<zahl> <= [-|+] <ziffer> {<ziffer>}
<string> <= "{<buchstabe>|<ziffer>}"
<buchstabe> <= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<ziffer> <= 0|1|2|3|4|5|6|7|8|9
```

```
PROGRAM DEMO1
BEGIN
        A0:=3;
        B:=+45;
        H:=-100023;
        C:=A;
        D123:=B34A;
        ESEL:=GIRAFFE;
        TEXTZEILE:=”HALLO";
END
```

```
<programm> <= PROGRAM <bezeichner>
              BEGIN
               { <zuweisung> ; }
               END
<zuweisung> <= <bezeichner> := ((<zahl>|<bezeichner>)|<string>)
<bezeichner> <= <buchstabe> {<buchstabe>|<ziffer>}
<zahl> <= [-|+] <ziffer> {<ziffer>}
<string> <= "{<buchstabe>|<ziffer>}"
<buchstabe> <= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<ziffer> <= 0|1|2|3|4|5|6|7|8|9
```

```
PROGRAM DEMO1
BEGIN
        A0:=3;
        B:=+45;
        H:=-100023;
        C:=A;
        D123:=B34A;
        ESEL:=GIRAFFE;
        TEXTZEILE:=”HALLO";
END
```

```
<programm> <= PROGRAM <bezeichner>
            BEGIN
              { <zuweisung> ; }
              END
<zuweisung> <= <bezeichner> := ((<zahl>|<bezeichner>)|<string>)
<bezeichner> <= <buchstabe> {<buchstabe>|<ziffer>}
<zahl> <= [-|+] <ziffer> {<ziffer>}
<string> <= "{<buchstabe>|<ziffer>}"
<buchstabe> <= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<ziffer> <= 0|1|2|3|4|5|6|7|8|9
```

```
PROGRAM DEMO2
BEGIN
        myVariable3 := 5
        2GOOD2BETRUE := 6ER
END
```

```
<programm> <= PROGRAM <bezeichner>
              BEGIN
               { <zuweisung> ; }
               END
<zuweisung> <= <bezeichner> := ((<zahl>|<bezeichner>)|<string>)
<bezeichner> <= <buchstabe> {<buchstabe>|<ziffer>}
<zahl> <= [-|+] <ziffer> {<ziffer>}
<string> <= "{<buchstabe>|<ziffer>}"
<buchstabe> <= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<ziffer> <= 0|1|2|3|4|5|6|7|8|9
```

```
PROGRAM DEMO2
BEGIN
                myVariable3 := 5
                2GOOD2BETRUE := 6ER
END
```

```
<programm> <= PROGRAM <bezeichner>
            BEGIN
             { <zuweisung> ; }
             END
<zuweisung> <= <bezeichner> := ((<zahl>|<bezeichner>)|<string>)
<bezeichner> <= <buchstabe> {<buchstabe>|<ziffer>}
<zahl> <= [-|+] <ziffer> {<ziffer>}
<string> <= "{<buchstabe>|<ziffer>}"
<buchstabe> <= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<ziffer> <= 0|1|2|3|4|5|6|7|8|9
```

```
PROGRAM DEMO3
BEGIN
        GOODNAME := +-5
        MARK := 5.5
END
```

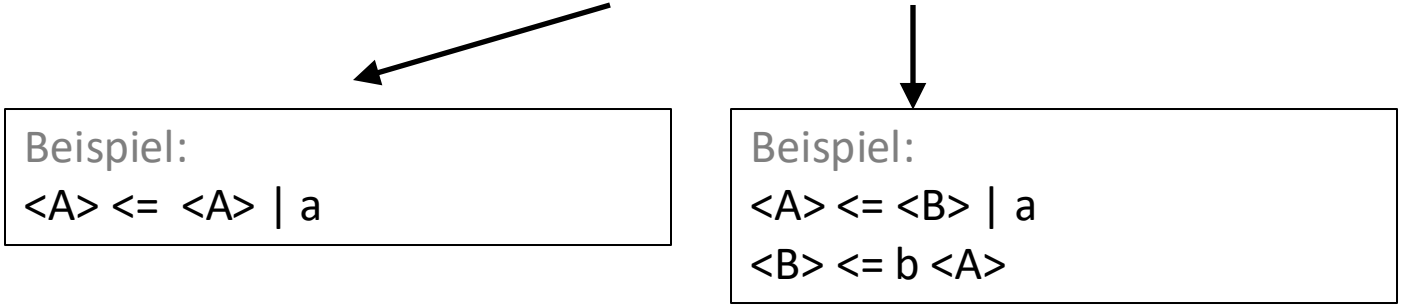
```
<programm> <= PROGRAM <bezeichner>
              BEGIN
               { <zuweisung> ; }
               END
<zuweisung> <= <bezeichner> := ((<zahl>|<bezeichner>)|<string>)
<bezeichner> <= <buchstabe> {<buchstabe>|<ziffer>}
<zahl> <= [-|+] <ziffer> {<ziffer>}
<string> <= "{<buchstabe>|<ziffer>}"
<buchstabe> <= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<ziffer> <= 0|1|2|3|4|5|6|7|8|9
```

```
PROGRAM DEMO3
BEGIN
                GOODNAME := +-5
                MARK := 5.5
END
```

Rekursion

Rekursion ist, wenn sich eine Regel selbst aufruft bzw. definiert.

Wir unterscheiden dabei zwischen direkter und indirekter Rekursion.



Sinvolle/endliche Rekursion brauchst Abbruchsoption (“base case”)

Rekursion Beispiele

EBNF-Beschreibung von List

<letter> <= a | b | c
<list> <= (<letter> , <list>) |<letter>

Legal?
a
a,b
a,b,c
a,a,a,a,a,a,a,a,a
a,bb,c

Rekursion Beispiele

Erstellen Sie eine Grammatik, die eine einfache Dateiverzeichnisstruktur beschreibt. In dieser Struktur besteht ein Verzeichnis entweder aus einem einzelnen Ordner oder aus einer Hierarchie von Ordnern, die durch Schrägstriche (/) getrennt sind.

Die folgenden Ordnernamen sind zulässig: dokumente, bilder, musik, downloads, desktop.

Das Verzeichnis kann aus beliebig vielen verschachtelten Ordnern bestehen. Beispielhafte gültige Verzeichnisse sind:

- dokumente
- bilder/musik
- downloads/bilder/dokumente

Repetition: Ableitungen

- Ableitungstabelle
  - Erste Zeile ist Startregel
  - Letzte Zeile ist Zeichenfolge
  - Übergang zwischen zwei Zeilen entspricht Ableitungsschritt

- Ableitungsbaum
  - Wurzel ist Namen der Startregel
  - Blätter sind Zeichen
  - Verbindungen stehen für einen Ableitungsschritt

Beispiel: Ableitung von -127 als Baum

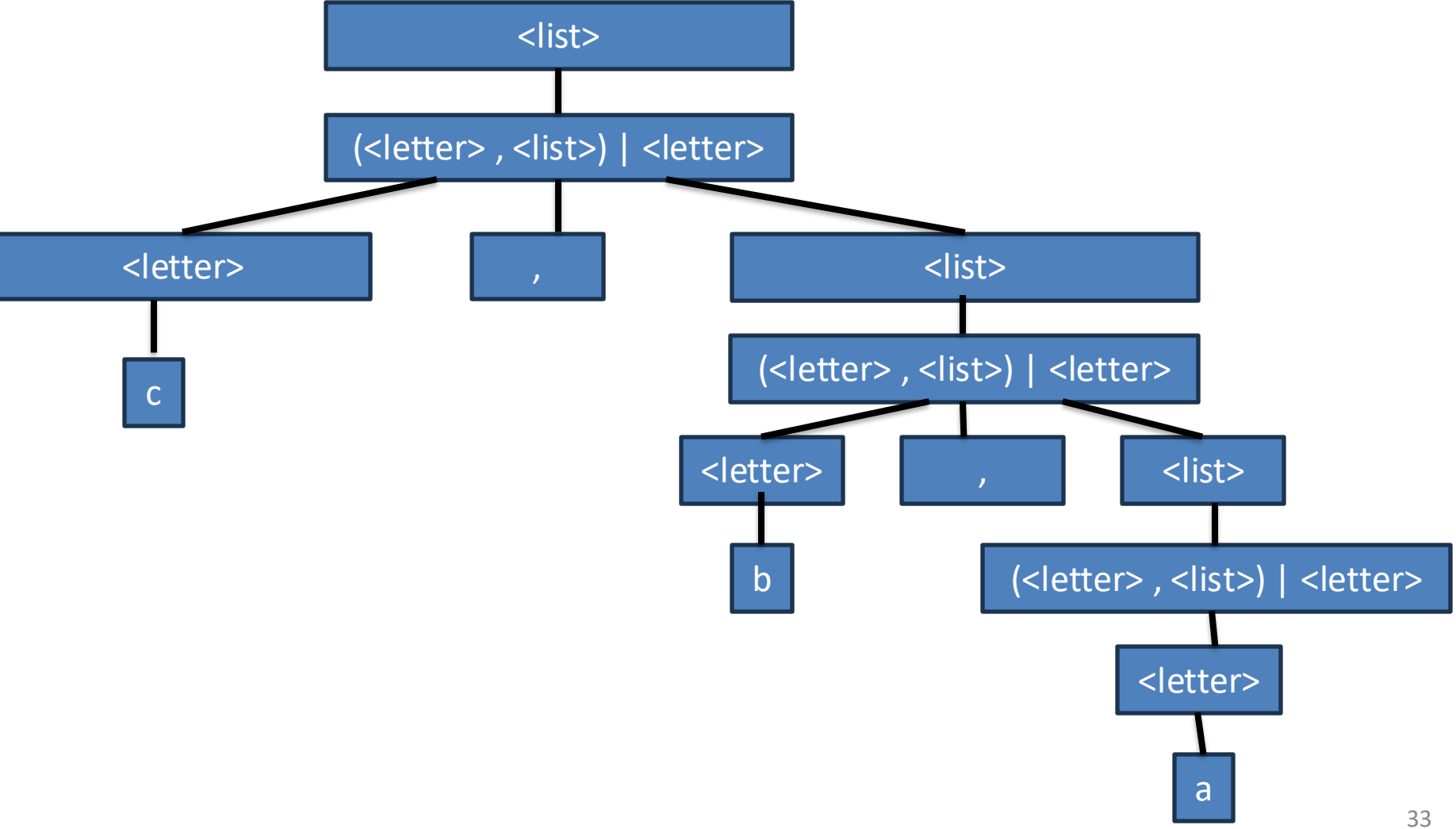
<digit>	⟵	0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<integer>	⟵	[+ | -] <digit> {<digit>}

Besipiel: Ableitung von c,b,a als Baum

EBNF-Beschreibung von List

<letter> <= a | b | c (R1)

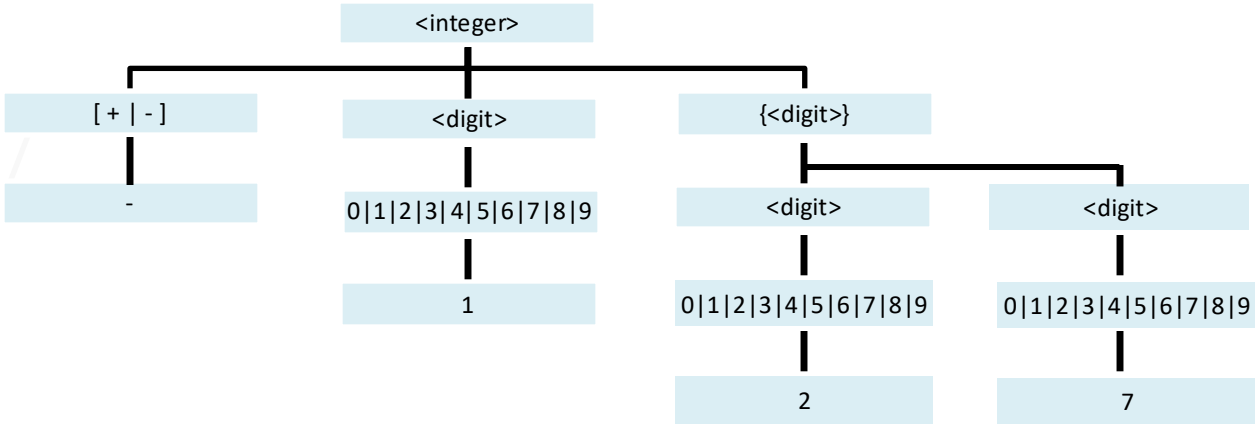
<list> <= (<letter> , <list>) |<letter> (R2)


<list>
(<letter> , <list>) | <letter>
<letter>
,
<list>
c
(<letter> , <list>) | <letter>
<letter>
,
<list>
b
(<letter> , <list>) | <letter>
<letter>
a

Beispiel: Ableitung von -127 als Baum

<digit> ⟵ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<integer> ⟵ [+ | -] <digit> {<digit>}



Beispiel: Ableitung von -127 als Tabelle

(R1)	<digit>	⟵	0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
(R2)	<integer>	⟵	[+ | -] <digit> {<digit>}

Beispiel: Ableitung von -127 als Tabelle

(R1)	<digit>	⟵	0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
(R2)	<integer>	⟵	[+ | -] <digit> {<digit>}

<integer>	⟵	[+ | -] <digit> {<digit>}	(R2)
	⟵	+ | - <digit> {<digit>}	Option gewählt
	⟵	- <digit> {<digit>}	- gewählt
	⟵	- <digit> <digit> <digit>	2 mal wiederholt
	⟵	- 1 <digit> <digit>	(R1) und 1 gewählt
	⟵	- 1 2 <digit>	(R1) und 2 gewählt
	⟵	- 1 2 7	(R1) und 7 gewählt

Beispiel: Ableitung von c,b,a als Tabelle

EBNF-Beschreibung von List

<letter> <= a | b | c (R1)

<list> <= (<letter> , <list>) |<letter> (R2)

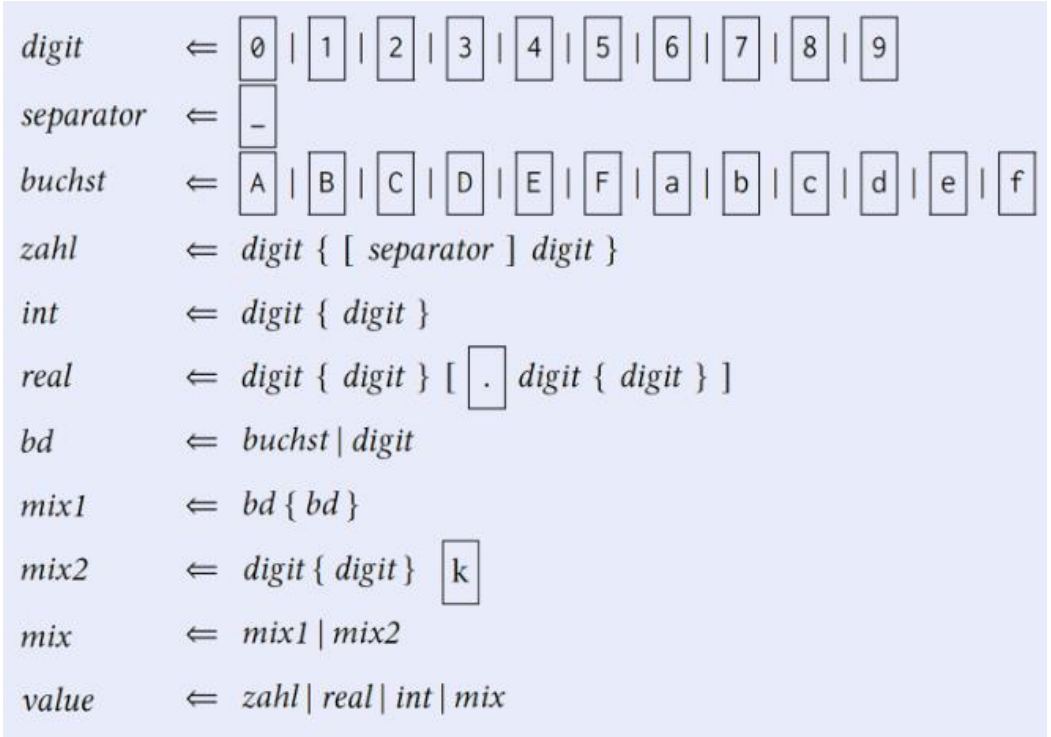
EBNF Notation

- In alten Prüfungen wird oft kursiv verwendet für EBNF-Regeln.
- *digit* statt <digit>
- Ab diesem Semester ist eine EBNF-Regel nur korrekt, wenn sie durch < > gekennzeichnet ist.
- Ebenfalls verwenden wir <- statt <=, beides wird aber als korrekt bewertet.

EBNF: Legal / Nicht Legal

Gegeben sei die EBNF-Beschreibung von *value*

- **2023**
- **_2023**
- **Back**
- **1a2b3c**
- **1a2b3k**
- **27_09_2023**
- **09.2023k**
- **12cd.34**
- **09.2023**
- **face**
- **123k**
- **0010K**
- **27_9.2023**
- **27.9.2023**



Kahoot

https://create.kahoot.it/share/eprog-u2-baumberger/b5c6cdf1-36e8-40c3-b5d5-aa4153e00fb1

https://create.kahoot.it/share/eprog-u2-zusatz/e953915f-7865-4bdc-b357-9117def8cf3f

Zusatzaufgaben

- Erstellen 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>`, , die alle Zahlen enthält, in denen die Ziffer 8 ungerade oft vorkommt. Beispiele sind 8, 128, 8881

Übungsblatt EBNF