

A person wearing a black hoodie is leaning over a desk, pointing at a computer keyboard. Another person is sitting at the desk, looking at the keyboard. The scene is dimly lit with a blueish tint.

# vim-Tutorial: Mit wenig Zeichen viel erreichen

Benedict Reuschling  
benedict@reuschling.org

16. März 2024  
Chemnitzer Linuxtage

# Trainings-Start

- Viele Benutzer kennen nur die vim-Basics
- Verpassen viele nützliche Built-in Funktionen
- Hier setzen wir an
- Geeignet für Anfänger, die gerade mit vim gestartet sind
- Ziel: Zeit und Tipparbeit sparen



**Quick win: Zahlen ändern**

# Zahlen verändern

langsamer Weg:

Insert Modus, Weglöschen, neue Zahl tippen, ESC drücken

Schneller Weg: Cursor vor der Zahl

Erhöhen: **CTRL-a** (add)

Verringern: **CTRL-x** (subtract)

Kann sogar negativ/positiv werden

Um mehr als nur +/-1: Zahl davor

**100 CTRL-a** = Erhöht um 100

**100 CTRL-x** = Verringert um 100

Funktioniert auch mit Hex (z.B. 0x1a)



**Autovervollständigung lernen  
und anwenden**

# Wörter vervollständigen

## Warum sich wiederholen?

- Bereits im Text vorhandene Wörter erneut einfügen
- Beispielttext im Editor: `Linuxtage`
- Insert-Modus, Anfangsbuchstabe des Wortes tippen: `L`
- Dann **CTRL+P** (Previous)
- Vervollständigung oder Menü mit Auswahl bei Mehrdeutigkeit



# Warum nur Wörter?

## Vervollständige auch...

- **Zeilen:**  
CTRL-x, CTRL-l (Line)
- **Pfade/Dateien:**  
CTRL-x, CTRL-f (File)
- **Rechtschreibung:**  
CTRL-x, CTRL-k  
:set spell  
spelllang=de\_de

Cursor auf Wort, dann z= für  
Wortvorschläge



# Omni-Completion

~/.vimrc:

```
set nocompatible  
filetype plugin on  
set omnifunc=syntaxcomplete#Complete
```

Quelltext vervollständigen:

**CTRL-x, CTRL-o** (omni)

z.B. eine CSS-Datei mit:

```
h1 { co
```

Zeigt nur an dieser Stelle gültige Elemente an



**Häufige Tippfehler beseitigen**

# Rechtschreibfehler

## Automatische Korrektur

- Abbreviations nutzen
- In der `.vimrc` eine persönliche Liste mit häufigen Verschreibern führen:

```
iab taht that
```

- Im Insert Modus (`iab` = Insert Abbreviation) wird dann beim Schreiben `taht` automatisch in `that` geändert



**Editor-Text durch die Shell filtern**

# Shell-Filter

- Mit `:%!Kommando` wird der Inhalt des Buffers (Editor-Fenster) durch ein Shell-Kommando gefiltert
- Beispiel: Zahlen sortieren  
189  
567  
234  
`:%!sort`
- Nützlich:  
`:%!column -t`  
`:%!uniq`  
`:%!fold -sw30`  
`:%TOhtml`



**Eingabesequenzen mit Makros  
automatisieren**

# Eingaben aufzeichnen

- Makros zeichnen Eingabesequenzen auf und speichern diese zur späteren Verwendung in Registern
- Aufzeichnung starten und in Register `r` speichern: `qr`
- Beliebige Aktionen ausführen (Text eingeben, Kommandos), z.B.  
`iHallo Welt<Esc>`
- Aufzeichnung stoppen: `q`



# Makros anzeigen und wieder ausführen

- Aufgezeichnete Makros anzeigen:  
:registers oder  
:reg oder  
:reg r
- Ausführen: Cursor an gewünschte Stelle setzen, dann @ gefolgt von der Registernummer: @r
- Beliebige Anzahl Ausführungen:  
<Zahl>@r, z.B. 42@r



# **action-range-object Notation**

# vim Anweisungen geben

## Aktion-Bereich-Objekt Notation

- Kommandomodus erlaubt Anweisungen, die menschlicher Sprache ähneln
- Beispiel: delete 3 words: `d3w`
- Start: die Aktion (was ist zu tun)
- Mitte: der Bereich (wo, wieviele)
- Ende: das Objekt (worauf)
- Beliebig kombinier- und austauschbar



# Action-Range-Objekt in Aktion

- `echo "Hallo Welt"` im Editor
- Ziel: Ändern von `Hallo Welt`
- Cursor innerhalb der `"` (beliebig)
- `ci` (change inside `"`)
- Löscht alles innerhalb der `"` und versetzt in den Insert-Modus
- Nur Löschen: `di` (delete inside `"`)
- Grosse Tippersparnis



## Action/Verb

- c** - **change**
- d** - **delete**
- y** - **yank** (copy)
- g u** - lowercase
- g U** - uppercase
- =** - filter/C-indenting
- g ?** - ROT13 encoding
- >** - shift right
- <** - shift left

## Range/Modifier/Operator

- i** - **inside**
- a** - **around**
- t** - **till**, search something and stop before it
- /** - find string or regex
- NUMBER** - a number like 2, 5, 11

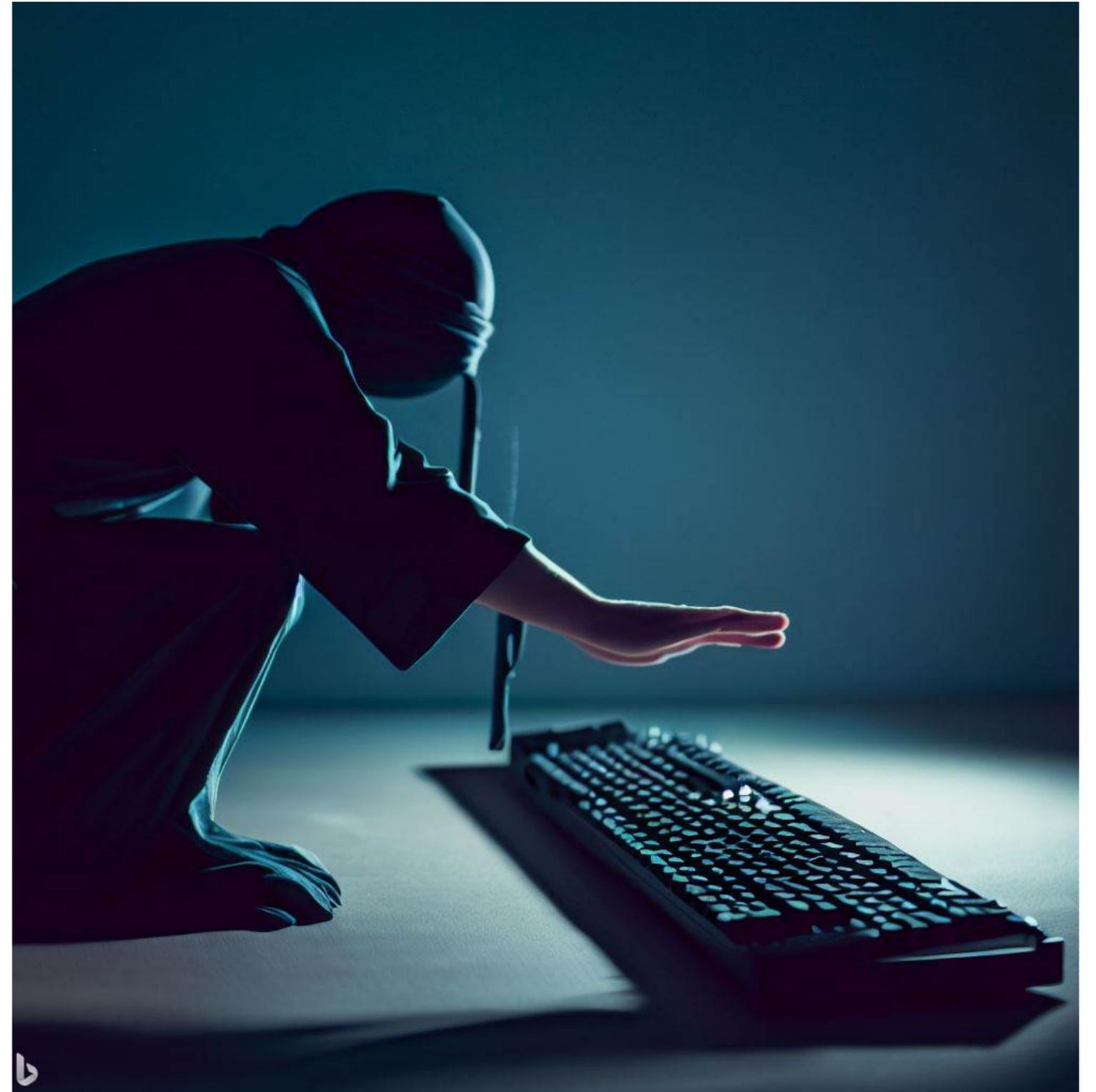
## Object/Noun

- w** - **word**
- s** - **sentence**
- p** - **paragraph**
- t** - **tag** (HTML/XML)
- b** - **block** (in programming)
- 0** - start of line
- \$** - end of line

# Übung

Wie lauten die Action-Range-Object Kürzel?

- Lösche die nächsten zwei Sätze
- Ändern bis zum nächsten Komma
- Ganzen Satz in Grossbuchstaben ändern
- ROT13 für den ganzen Absatz
- Ändere den Text innerhalb der {}
- Ändere den Text inklusive der () darum
- Lösche alles bis zum Zeilenende



# Trainings-Ende

Üben

Wiederholen

Anwenden

Zeitersparnis ernten

Danke und bis zum nächsten  
Mal!

