

LFCX:Tooling

1.) Benötigte Software

[→ ZP:Sheet:2]

1. **Powershell 7.5.x** :- weil Linux- und Windowsbefehle zur Netzwerkkonfiguration syntaktisch unterschiedliche Aufrufe und Ergebnisse liefern.
2. **Markdown-Editor** (*VSCODE* bzw. *VSCODIUM* mit Extension):- weil Sie Dokumentationen einfach und schnell erstellen können sollen.
3. **Python3** :- weil Sie Netzwerk und Datenanalysen programmieren können sollten.
4. **drawio** :- weil Sie Netzwerke, Protokolle und Prozesse einfach und untereinander austauschbar erstellen können sollen.
5. **Linux mit bash und net-tools** :- weil Sie auch remote über ssh und VPNs agieren können sollen
6. **Freeplane** :- zur Konzeption von Systemen, Ontologien, etc.

Diese Tools können Sie auf unterschiedlichen Wegen installieren:

A) Wer einen Windows-11-Rechner hat, kann

1. sich einen zweiten Rechner mit einem Linux anschaffen
2. seinen Windows-11-Rechner als Dual-Boot-System aufsetzen.
3. seinen Windows-11-Rechner mit einer Virtualisierungssoftware zu einem Host für Linux machen.
4. in seinem Windows-11-Rechner das Windows-Subsystem-Linux (WSL 2!) aktivieren

Es reicht dann aus, wenn Sie auf dem Windows-11-Rechner *VSCODE* (bzw. *VSCODIUM*), die *Powershell* 7.5 und *drawio* installieren. Sie könn(t)en diese Tools aber auch auf beiden Rechnern installieren.

B) Wer einen MAC-Rechner hat, kann

1. seinen MAC-Rechner mit einer Virtualisierungssoftware zu einem Host für Linux machen.

Es reicht dann aus, wenn Sie auf dem MAC-Rechner die entsprechenden *VSCODE* (bzw. *VSCODIUM*)-Binaries, die *Powershell* 7.5 mit .NET-Framework und *drawio* installieren. Sie könn(t)en diese Tools aber auch auf beiden Rechnern installieren.

C) Wer einen Linux-Rechner hat, kann

1. sich einen zweiten Rechner mit einem Windows-11 anschaffen
2. seinen Linux-Rechner als Dual-Boot-System aufsetzen.
3. seinen Linux-Rechner mit einer Virtualisierungssoftware zu einem Host für Windows-11 machen.

Es reicht dann aus, wenn Sie auf dem Linux-Rechner die entsprechenden *VS CODE* (bzw. *VSCodium*)-Binaries, die *Powershell 7.5* mit .NET-Framework und *drawio* installieren. Sie müssen zudem wenigstens die *Powershell 7.5* unter Windows-11 installieren.

Aus der Praxis:

- Wer einen Windows-11-Rechner hat (und übergangshalber an Admin-Rechte herankommt), sollte die Lösung *WSL 2* wählen.
- Wer einen MAC-Rechner nutzt, sollte die Virtualisierungslösung angehen.
- Wer einen Linux-Rechner (Ubuntu 24.04 etc.) nutzt, könnte auf einen Windows-Rechner verzichten, wenn er alle anderen Tools bei sich aktiviert.

2.) **Powershell 7.5.x als Tool für LF03, LF09, LF11**

Nutzungsbegründung:

- Nur Powershell 7 erlaubt Nutzung des WSL 2
- Powershell 7 bietet als .NET-Implementierungen Netzwerkanalyse- und konfigurationstools

A) → Zur Geschichte:

Windows

- enthielt ursprünglich nur die Command-Shell ([cmd](#)):
 - Suche in Windowssuche nach *cmd* oder *command*.
 - Klicke angezeigte *Eingabeaufforderung* an
 - Windows startet die Command-Shell zum Starten einfacher Programme
- Im Jahr 2003 hat Windows (Unix-)Admins eingeladen, um ihre Wünsche an eine bessere Admin-Shell zu erfahren. Ergebnis: So, wie unter UNIX/Linux.
- Daraufhin entstand die Powershell als eine auf den .NET-Framework aufsetzende Shell, auch mit eingebauten Befehlen.
 - Version 1.0 im Jahr 2006
 - Version 2.0 im Jahr 2008
 - Version 3.0 im Jahr 2012
 - Version 4.0 im Jahr 2013
 - Version 5.0 im Jahr 2015 zusammen mit Windows 10 (für ältere in Version 5.1 nachgereicht)
 - Version 6.0 im Jahr 2018 ist eine neue Open-Source-Version [Powershell-Core] (MIT) gehostet auf GitHub, die auch für LNX erhältlich ist.
 - Version 7.+x ist die aktuelle, 5.1 kriegt nur noch updates, 6 wird nicht weiterentwickelt, Basis ist jetzt die Version 7.x plattformübergreifend.

Hinweis: Linuxshells und Powershell unterscheiden sich trotzdem grundsätzlich:

- PWSH arbeitet .NET basiert. Die meisten Kommandos sind als NET-Funktionalitäten im PWSH-Code implementiert.
- Linux-Shells rufen externe Tools auf.

B) Installation unter Windows 11

Powershell 7 ersetzt nicht vollständig Powershell 5. Deshalb existiert Powershell 5 auch noch nach der Installation von Powershell 7 und wird oft als defaultmäßig gestartet.

- Ermitteln der eigenen Version:
 - Powershell starten
 - > `$psVersionTable` eingeben
- Ggf. Powershell Version 7.5 installieren:
 - per Windowsstrore ohne Adminrechte.
 - * suche darin nach Powershell 7.
 - * lasse es installieren
 - per → [GitHub-Repository](#)
 - * suche `PowerShell-7.5.2-win-x64.exe` unter <https://github.com/PowerShell/PowerShell/releases/tag/v7.5.2>
 - * Lade es runter in Deinen privaten App-Ordner.
 - * Lasse dafür Link in Panel / Start anlegen
- Installation verifizieren:
 - Starte Powershell 5:
 - * In Appsuche nach *Powershell* (sic!) suchen und starten
 - * > `$psVersionTable` eingeben
 - Starte Powershell 7:
 - * In Appsuche nach *pwsh* (sic!) suchen und starten oder
 - * In gestarter Powershell 5 `pwsh` aufrufen
 - * > `$psVersionTable` eingeben

Hinweis:

- Powershell 7 (= pwsh) gibt es als Binaries auch für Linux und Mac. Allerdings: Die Powershell unter Microsoft arbeitet mit .NET-Funktionen, während die und Linux (und Mac?) meistens Tools des Betriebssystems aufrufen. Deshalb ist die `pwsh` streng gesehen nicht betriebssystemübergreifend.

- Mittlerweile kann man das .NET-Framework (Core) auch für MAC und Linux installieren. Ob die Powershell unter Linux/Mac damit völlig gleich funktioniert, wie die Powershell unter Windows, müssten wir testen.
- Wichtig zu wissen ist, dass die .NET-implementierten Funktionen oft andere syntaktische Antworten generieren als die Tools unter Linux, selbst wenn sie inhaltlich auf dasselbe abzielen. Deshalb ist es wichtig, die Unterschiede wenigstens zu kennen.

C) Installation unter MAC

1. Homebrew (àla Mac installieren): <https://brew.sh>
2. Powershell gemäß <https://learn.microsoft.com/de-de/powershell/scripting/install/installing-powershell-on-macos?view=powershell-7.5>
3. "Powershell als App öffnen"
 - `pwsh` im Terminal eingeben
 - darin `brew install dotnet` eingeben

D) Installation unter Linux/Ubuntu

1. Installationsanleitung von Microsoft folgen <https://learn.microsoft.com/de-de/powershell/scripting/install/installing-powershell-on-linux?view=powershell-7.5>
2. Für Ubuntu <https://learn.microsoft.com/en-us/powershell/scripting/install/install-ubuntu?view=powershell-7.5>
3. .NET gemäß <https://learn.microsoft.com/de-de/dotnet/core/install/linux> installieren
 - Für Ubuntu
 - `sudo apt-get update && sudo apt-get install -y aspnetcore-runtime-9.0 ODER`
 - `sudo apt-get install -y dotnet-runtime-9.0`

ÜBUNG LFCX:Tooling:01

- Installieren Sie die aktuellste Powershell auf Ihrem Windowsrechner.
 - Verifizieren Sie ihre Installation
-

3) Linux als Tool für LF03, LF09, LF11

Nutzungsbegründung:

- Zu LF09 u. zu LF03 gehören Fragen nach bestimmten Netzwerkanalysetools. Die sollten LF09-Schülerinnen in der Linux- und in der Windowsversion kennen.
- Zu LF09 gehört das Thema '(asymmetrische) Verschlüsselung'. Die können LF09-Schülerinnen sich gut an und mit der Nutzung von [ssh](#) verdeutlichen. Unter Linux ist das einfacher zu aktivieren.

A) Varianten der Bereitstellung:

1. auf einem zweiten Rechner/Laptop

- Ubuntu-24.04.03 LTS ISO-Datei von <https://ubuntu.com/download/desktop> herunterladen.
- Mit Windowsmitteln einen bootfähigen USB-Stick erzeugen.
- Rechner von USB-Stick booten.
- Installationsanleitung folgen

2. Dual Boot Installation

- Verfahren wie unter (1).
- Bei Installation 'neben Windows installieren' auswählen

3. in einer virtuellen Maschine unter Windows

- Virtualisierungssoftware auf Host Windows installieren
 - Empfohlen: [→Virtual Box]{<https://www.virtualbox.org/>}. Ist GPL v3 lizenziert = echte freie Software → https://www.virtualbox.org/wiki/Licensing_FAQ
 - Abgeraten: VMware Workstation Player (kurz VMplayer) weil
 - * → [Vmware gekauft von Broadcom](#).
 - * Broadcom monetarisiert die vorher (für den privaten Gebrauch) "freie" Software.
 - * Lizenzen für VMware Workstation sind über diese Produktnummer nicht mehr erhältlich. Die Nachfolge-Lizenz war VMware Desktop Hypervisor Pro, diese ist aber auch im November 2024 nicht mehr erhältlich → <https://www.software-express.de/hersteller/vmware/workstation/player>
 - * Antwort der Community: GitHub-Projekt zum Sammeln existierender Keys. → <https://github.com/hegdepavankumar/VMware-Workstation-Pro-17-Licence-Keys>.
 - * Aber trotzdem keine seriöse Variante mehr.
- Virtualisierungssoftware auf Host MAC von → <https://ubuntu.com/download/desktop> installieren
- Ubuntu-24.04.03 LTS ISO-Datei von → <https://ubuntu.com/download/desktop> herunterladen.

- In Virtualisierungssoftware als Start-/Bootmedium aktivieren
- Virtuelle Maschine booten.
- Installationsanleitung folgen

4. in einer virtuellen Maschine unter MacOs

- Virtualisierungssoftware auf Host MacOs installieren
 - mögliche: [→ Virtual Box]{<https://www.virtualbox.org/>}. Ist GPL v3 lizenziert = echte freie Software → https://www.virtualbox.org/wiki/Licensing_FAQ
 - getestet: [→ iUTM]{<https://mac.getutm.app/>}. Ist Apache-v2 lizenziert = echte freie Software → <https://github.com/utmapp/UTM>. Binaries auf Github und im AppleStore, dort aber angeblich gegen Geld.

5. als Windows Subsystem Linux

- Über Windowssuchzeile nach *Features* suchen lassen
- Button *Windows-Features aktivieren ...* anklicken
- Im Dialog *Windows Subsystem für Linux* anwählen/aktivieren
- Neustart
- Powershell 7.5.x nach Windowsanleitung installieren → <https://learn.microsoft.com/en-us/powershell/scripting/install/installing-powershell-on-windows?view=powershell-7.5>
- Powershell (Adminmode) aufrufen
- gemäß → [Microsoftanleitung WSL-Aktivierung](#) verfahren - bzw.
 - powershell: `wsl --install`
 - powershell: `wsl --set-version UBUNTU 2`

B) Verifikation:

- LNX (als virtuelle Maschine oder Dual-Boot etc.)
 - Linuxshell bzw. öffnen
 - > `ping 8.8.8.8` eingeben
 - Wenn Rückgabe *64 bytes from 8.8.8.8, dann*
 - Zugriff auf Linuxtool *ping* (Linux läuft)
 - und Netzzugriff auf Google-Server (Netzzugang per NAT aktiviert)
- WSL auf Windows 11 Rechner:
 - Powershell öffnen
 - > `pwsh` eingeben (= Wechsel zu Powershell 7)

- > `ping 8.8.8.8` eingeben (= Netzzugang verifizieren)
- > `ipconfig` eingeben (= Windowsanalyse Netzwerkinterface)
- > `wsl` eingeben (= WSL aktivieren)
- > `ifconfig` eingeben (= Linuxanalyse Netzwerkinterface)

Hinweis: *Falls `ifconfig` oder gar `ping` unter LNX oder WSL fehlt, bitte in der Shell eingeben: `sudo apt-get install net-tools`

ÜBUNG LFCX:Tooling:02

- Installieren Sie eine Ihnen genehme Linux-Variante.
 - Verifizieren Sie Ihre Installation
-

4) VS CODE als Markdowntool für LF09

Nutzungsbegründung: Markdown

- ist das Standardformat für Dokumentation in GitHub
- erleichtert das schnelle Schreiben augenfällig gestaltete Texte.

A) Hintergrund: VS CODE

- wird von Microsoft zusammen mit der Community unter der MIT-Lizenz auf GitHub entwickelt [<https://github.com/microsoft/vscode>]
- ist echte Open-Source-Software
- wird auch als Binärpaket für Windows, Linux und Mac bereitgestellt → <https://code.visualstudio.com/download>
- kann mit Extensionen funktionell erweitert werden.

Hinweise:

- Die Binärpakete sind mit einer gesonderten Microsoft-Lizenz verbunden → <https://code.visualstudio.com/license>. Die erlaubt Windows, ihre Binärpakete mit 3rd-Party-Komponenten anzureichern. Außerdem sind Extensionen u.U. anders lizenziert.
- Wer Extensionen aus dem Windowsmarketplace installiert, stimmt den → [Visual-Studio-Marketplace-Terms-of-Use](#). Die besagen:

“Marketplace Offerings are intended for use only with In-Scope Products and Services and you may not install, reverse-engineer, import or use Marketplace Offerings in products and services except for the In-Scope Products and Services.” → S.2

Das bedeutet, dass Extensionen aus dem Windowsmarketplace nur in die von Microsoft bereitgestellten Binaries eingebunden werden dürfen.

Die Community stellt aber unter dem Namen → VSCODIUM eine Version ohne die Microsoft integrierten 3rd-Party-Komponenten. Deren Extension müssen konsequenterweise aus einem gesonderten Repository [→ <https://open-vsx.org/>] stammen.

B) Installation

- VS CODE (Visual Studio Code) für
 - → Windows (<https://code.visualstudio.com/docs/setup/windows>)
 - → Linux
 - * Linux-Binaries vom <https://code.visualstudio.com/download> = https://code.visualstudio.com/docs/?dv=linux64_deb downloaden.
 - * Anleitung von <https://code.visualstudio.com/docs/setup/linux> folgen
 - → VSCODIUM für
 - * → Windows
 - * → Linux
 - VS Plugins/Extensions jeweils über *Preferences/Extensions* installieren lassen. D.h. für
 - * VS CODE erweitert aus → Visual Studio Marketplace:
 - → markdown-all-in-one v. yzhang
 - → markdown converter v. manuth
 - → LTeX+
 - * VSCODIUM :
 - → markdown-all-in-one v. yzhang
 - → markdown converter v. manuth
 - → LTeX

C) Verification

- Aus Github-Repository <https://github.com/kreincke/proMdTutorials>
 - die Datei `markdown-selftutorial.md` in einen gesonderten Ordner herunterladen
 - den Ordner `img` mit Inhalt in denselben Ordner neben die Datei `markdown-selftutorial.md` herunterladen

- *markdown-selftutorial.md* in VS CODE bzw. VSCODIUM öffnen
 - Quellcode sollte mit *Syntax-Highlighting* dargestellt werden
 - Previewmode sollte die Seite ungefähr wie <https://github.com/kreincke/proMdTutorials> darstellen
- Ist die Extension LTeX+ installiert, ist die Rechtschreibkontrolle automatisch aktiviert.

D) PDF Druck

- STR SHFT P drücken
 - in Eingabezeile *Markdown: Convert Document* eingeben und anklicken
 - Anfrage zur Installation von *chromium* bestätigen (der Interpreter)
 - Dokument anzeigen lassen
-

ÜBUNG LFCX:Tooling:03

- Installieren Sie *Visual Studio Code* auf Ihrem Windows- oder Linuxrechner.
 - Installieren Sie die genannten Extensionen.
 - Erzeugen Sie einen Arbeitsordner *md . test* in Ihrem Homeverzeichnis.
 - Erzeugen Sie darin den Ordner *img*
 - Laden Sie von → <https://github.com/kreincke/proMdTutorials> die Datei → <https://github.com/kreincke/proMdTutorials> *selftutorial.md* in den generierten Ordner *md . test* herunter.
 - Laden Sie die beiden Bilddateien aus → <https://github.com/kreincke/proMdTutorials/tree/main/img> in den generierten Ordner *md . test / img* herunter.
 - Öffnen Sie die Datei *md . test / markdown-selftutorial . md* in Ihrer VS CODE/VSCODIUM-Instanz
 - Überzeugen Sie sich, dass der Code mit Syntaxhighlighting dargestellt wird.
 - Öffnen Sie die Previewseite (Lupe re. oben).
 - Überzeugen Sie sich, dass der Inhalt der Datei jetzt in etwa wie die Readme-Datei aus → <https://github.com/kreincke/proMdTutorials/> dargestellt wird.
 - Erzeugen Sie aus Ihrer *Visual Studio Code*-Version eine PDF-Datei davon
-
-

ÜBUNG LFCX:Tooling:04

- Schreiben Sie einen kurzen neuen deutschen Markdowntext.
- Schreiben Sie ein Wort falsch
- Fügen Sie oben die Zeile <!-- `LTEx: Language=de-DE` --> ein gefolgt von einer Leerzeile.
- Sichern Sie die Datei.

Danach sollte das falsch geschriebene Wort mit einer geschlängelten Linie markiert werden. Gehen Sie mir der Maus darauf, werden Ihnen unter der Glühbirne Alternativen angeboten.

Drawio als Dokumentationstool für LF03, LF09, LF11

Nutzungsbegründung:

- ist freie Software.
- stellt für alle Diagrammtypen standardisierte Symbole bereit.

Hinweis: draw.io gibt es als Desktopvariante (kann offline genutzt werden) oder als Onlineservice
<https://app.diagrams.net/>

- draw.io für
 - → [Windows](#) aus Windows-Marketplace installieren lassen
 - oder per Installer von <https://github.com/jgraph/drawio-desktop/releases/tag/v28.0.6>
 - → [Linux/Ubuntu](#) als zip herunterladen und manuell integrieren
 - oder [per snap installieren lassen](#)
 - → [Browser](#)
-

ÜBUNG LFCX:Tooling:05

- Installieren Sie die für Sie passende draw.io-Version
 - Erzeugen Sie ein neues Netzwerk-Diagramm
 - Suchen Sie die Netzwerk-Symbole.
 - Dokumentieren Sie darin und damit ein Netzwerk bestehend aus [Laptop](#) → [Switch](#) → [Router](#) → [Loadbalancer](#) → [Server](#) und Linien als Kabel
 - Laden Sie Ihr Ergebnis unter einem Dateinamen nach dem Muster [lf09-ueb-YYYYMMDD-ihrname](#) in den Ordner [uebungen](#) hoch. (Dabei stehe [YYYYMMDD](#) für das Datum im [ISO 8601 Format](#). Alles andere bitte in Kleinbuchstaben.)
-

Freeplane als Konzeptionstool für LF03, LF09, LF11

Nutzungsbegründung:

- ist freie Software.
- läuft unter Java.
- erstellt Mindmaps in XML-Format, Ergebnisse sind daher leicht austauschbar.
- Freeplane
 - Windows:
 - * [Freeplane-Setup-X.Y.Z.exe](https://github.com/freeplane/freeplane/releases) von <https://github.com/freeplane/freeplane/releases> downloaden & starten
 - * GPL License akzeptierten = Installer laufen lassen
 - Linux(Ubuntu/Debian)
 - * [freeplane_xxx~upstream-1_all.deb](#) Datei herunterladen
 - * [sudo apt-get install ./freeplane_xxx~upstream-1_all.deb](#) (Menueintrag wird unter Kategorie *Office* angelegt)