

# **Prestudies Guide T-Systems MMS/ Mecom - DevOps Academy 2023 Guide**

## **Einleitung**

Der nachfolgende Leitfaden soll Dir einen Überblick geben, mit welchen Themen Du Dich in den Wochen der Prestudies beschäftigen solltest. Bitte recherchiere auch selbst nach weiteren Quellen für die Themen im Internet.

Ein kleiner Tipp: Versuche Dich nicht in den Abgründen des Internet zu verlieren. Wenn Du eine Webseite als Quelle für ein Thema gefunden hast, folge nicht mehr als maximal einem Link zu weiterführenden Seiten oder Begriffserklärungen. Ansonsten wirst Du das Gefühl bekommen, mit Informationen überschwemmt zu werden, und das Thema niemals in einigen Tagen bewältigen zu können.

Arbeite eng mit Deinem/Deinen Study-Buddy zusammen. Stimmt Euch regelmäßig ab und beantwortet Euch gegenseitig Fragen, bzw. findet gemeinsam die besten Quellen und Inhalte. Notiert Euch dabei auch Fragen, für die Ihr nicht auf Anhieb eine Erklärung findet, und bringt diese Fragen in unsere gemeinsamen wöchentlichen Fragerunden mit.

Viel Erfolg und Spaß bei der Bearbeitung der Themen!

## **Ziele der Prestudies**

Nach Abschluss der Prestudies-Phase solltest Du...

- Die Grundlegenden Tools für das Training installiert haben
- Die Tools auf Ihre funktion überprüft haben
- Ein grundlegendes Verständnis um die Rolle der Tools in der Softwareentwicklung haben
- Ein grundlegendes Verständnis um die Entwicklung einfacher Programme haben.
- Ein grundlegendes Verständnis von Containern haben

## **Allgemeine Hinweise**

- Für die meisten Abschnitte gibt es einen Hinweis darauf, wie lange Du Dich damit beschäftigen solltest. Nimm Dir diese Zeit und versuche Dich in dieser Zeit nicht ablenken zu lassen. Arbeite in 45 Minuten blöcken und nimm Dir Pausen weg vom Bildschirm – benutze einen Timer.
- Wenn Du auf Begriffe oder Konzepte stößt, die Dir unbekannt sind, dann nimm Dir auch die Zeit, diese Begriffe zu recherchieren.
- Bei schweren Themen hilft es manchmal am nächsten Tag nochmal drauf zu sehen.
- Wenn Du trotz Recherche etwas nicht verstehst, dann versuche gemeinsam mit Deinem Study Buddy einen Ansatz zum Verständnis zu finden.

- Wenn es dennoch offene Fragen zu Konzepten, Begriffen oder Themen gibt, dann notiere diese Punkte damit Du Sie später z.B. mit dem Trainer klären kannst.
- Achte darauf wie Du lernst und fokussiere darauf. Was Du lernst ändert sich in der IT schnell - wie Du lernst bleibt Dir erhalten.

## Aufgaben

### Keepass

Just use KeePass.



Ziel: Sicheres Verwalten von

- Benutzernamen und Passwörtern
- Tokens
- Keys



Time:

- 3 x 45 Minuten



Aufgaben:

- Installiere <https://keepass.info/>
- Wähle ein starkes Passwort welches Du Dir gut merken kannst z.B: “Bei meinem Freund Gerold habe ich zum ersten mal WOW gespielt. Ich habe Wildschweinfelle gesammelt, und wollte gar nicht mehr aufhören.” -> Bm-FGhiz1tenxWOWg.IhWg,uwgnma.
- Benutze dieses Passwort an keiner anderen Stelle.
- Vergiß dieses Passwort nicht.
- Überlege Dir eine Backup Strategie für Deinem Password Manager.
- Erstelle ein Backup
- Teste Dein Backup - Funktioniert Dein Passwort Manager nach einem Backup noch?
- Überlebt Dein Backup einen Totalverlust der Festplatte?
- Sieh Dir die Features von Keepass an und benutze Keepass für alle Secrets.



Dokumentation:

- [keepass features](#)



Tutorial:

- [keepass firststeps](#)



Videos:

- [The ULTIMATE Guide to KeePass](#)

## Office

### All Microsoft 365 Apps Explained in 6 Minutes



Ziel: MS-Teams für das online Training bereit zu halten.



Time:

- 1 x 45 Minuten



Aufgaben:

- Melde Dich mit Deinem @brightstraining.com Account auf <https://office.com> an.
- Stelle sicher Word, Excel, PowerPoint, Outlook, Teams sind installiert.
- Mache Dich mit MS-Teams vertraut.
- Richte Deinen
- Damit die weiteren Accounts nach dem Training erhalten bleiben solltest Du eine eigene EMail benutzen.



Dokumentation:

- [Microsoft 365 documentation](#)

## Cloud Guru

Frag nach ob Dir ein acloudguru account gesponsored wird.

## Trello

Make Work Vibrant | Trello



Ziel:

- Bequem mit Trello umgehen können



Time:

- 2 x 45 Minuten



Aufgaben:

- Leg Dir einen Account an <https://trello.com>
- Organisier Deine Prestudies in Trello.



Tutorial:

[trello 101](#)



Videos:

[Organize Your College Life with Trello](#) [Guide to Using Trello for University](#) [How to Use Trello to Organize Your Cleaning Schedule](#) [Ultimate Trello To-Do List](#)

## Command Line Basics

### Bash in 100 Seconds



Ziel:

- Ein Terminal öffnen können
- Befehle in ein Terminal eingeben können
- Verzeichnisstrukturen navigieren können
- Unterschied zwischen Command Line, Bash und Powershell verstehen
- Einfache Skripte schreiben können



Time:

- 10 x 45 Minuten
- Verteile die Übungen über die gesamten Prestudies
- Erledige die Aufgaben mit allen Terminals



Aufgaben:

- Öffne ein Terminal (Command Line, Bash oder Powershell)
- Wenn Du weitere Terminals hast, mach die Übung mit diesen nochmal.
- Beachte den wichtigen Unterschied welches Terminal Du benutzt.
- Kommandos und Syntax unterscheiden sich zwischen Terminals.
- Finde Dein user home dircetory
- Liste Dateien und Verzeichnisse
- Erstelle neue Verzeichnisse und Dateien
- Gib Inhalte von Textdateien im Terminal aus
- Schreibe ein Skript um text einzulesen und wieder auszugeben
- Schreibe ein Skript um eine zufallszahl auszugeben
- Schreibe ein Skript welches den Verzeichnisnamen ausgibt in dem sich das skript befindet



Dokumentation:

- [Windows CMD](#)
- [Windows PowerShell-Referenz](#)
- [Bash Reference Manual](#)



Tutorial:

- [bash-scripting-tutorial-for-beginners](#)
- [Windows Command Prompt in 15 Minutes](#)
- [Introduction to PowerShell](#)
- [Get started with Windows PowerShell](#)
- [Introduction to Bash](#)



Videos:

- [Customizing your Windows Terminal Hotkeys and Actions](#)
- [Windows Command Line Tutorial - 1 - Introduction to the Command Prompt](#)
- [40 Windows Commands you NEED to know \(in 10 Minutes\)](#)
- [Terminal vs. Bash vs. Command line vs. Prompt](#)

## WSL2

### Developing on Windows with WSL2



Ziel:

- Basis für Linux erfahrungen
- Basis für Container entwicklung



Time:

- 5 x 45 Minuten



Aufgaben:

- Installiere Ubuntu für WSL



Dokumentation:

- [windows/wsl](#)



Tutorial:

- [ubuntu on wsl2](#)
- [Installieren von Linux unter Windows mit WSL](#)



Videos:

- [BEST Web Dev Setup? Windows & Linux at the same time \(WSL\)](#)
- [Kali Linux APPS on Windows in 5min \(WSLg\)](#)
- [Developing for Linux on Windows - Scott Hanselman - NDC Porto 2022](#)
- [Linux Apps in Windows! - WSL Setup Guide](#)



## Docker

### Docker in 100 Seconds



Ziel:

- docker auf dem rechner installieren
- Funktion überprüfen damit es ab Tag 1 einsatzbereit ist.



Time:

- 5 x 45 Minuten



Aufgaben:

- Installiere docker <https://docs.docker.com/get-docker/>
- run `docker pull hello-world`
- run `docker run --rm hello-world`
- run `docker run --rm -d -p 80:80 docker/getting-started`
- Auf Deinem Rechner werden jetzt docs gehostet: <http://localhost:80>
- Wenn installation oder die Befehle oben probleme bereiten melde Dich frühzeitig



Dokumentation:

- [docs](#)
- [docs in container](#)
- [docker hub](#)



Tutorial:

- [docker-curriculum](#)
- [Guides Get started](#)
- [baeldung docker-compose](#)
- 



Videos:

- [Docker Tutorial for Beginners \[FULL COURSE in 3 Hours\]](#)
- [Docker Compose will BLOW your MIND!! \(a tutorial\)](#)
- [Docker Tutorial for Beginners 10 - How to Use DockerHub](#)

## Github

### What is GitHub?



Ziel:

- Deinen Github account für Deine restliche Karriere bereit haben.
- Einfache git Befehle zu verwenden.



Time:

- 3 x 45 Minuten
- Sieh dir die Aufgaben nochmal an wenn Du vscode installiert hast



Aufgaben:

- Installiere GIT <https://git-scm.com/downloads>
- Leg Dir einen Account an <https://github.com/>
- Leg Dir ein Privates repo an "MyDevDictionary" in dem Du Dir Notizen machen kannst.
- Clone dein neues remote repo auf deinen lokalen rechner.
- Nachdem vscode installiert ist - sieh Dir die vscode git integration an.



Dokumentation:

- [git reference](#)



Tutorial:

- [github hello world](#)



Videos:

- [Git for Everybody: How to Clone a Repository from GitHub](#)
- [Git It? How to use Git and Github](#)
- [Git vs. GitHub: What's the difference?](#)

## Stackoverflow

Stack Overflow is 10!



Ziel:

- Schneller Zugriff zum wichtigsten Frage und Antwortportal für Software Entwicklung.



Time:

- 2 x 45 Minuten



Aufgaben:

- Leg Dir einen Account an <https://stackoverflow.com/>
- Beschäftige Dich mit der Etiquette: [How To Use Stack Overflow](#)
- Beschäftige Dich mit dem Thema Minimal REproducibile Example: [minimal-reproducible-example](#)
- Verschaffe Dir einen Überblick über [stackexchange](#)



Dokumentation:

- [meta.stackexchange](#)



Tutorial:

- [How to Ask a Question on Stack Overflow](#)



Videos:

- [Stack Overflow](#)

## Python

### God-Tier Developer Roadmap



Ziel:

- Erste Softwareentwicklungs Erfahrungen.



Time:

- 7 x 45 Minuten



Aufgaben:

- Installiere Python [downloads](#)
- Schreibe ein Hello World Programm in Python
- Schreibe ein World Programm in Python mit PySimpleGUI
- Schreibe ein World Programm in Python mit Flask



Dokumentation:

- “Hello, World!” program
- [The Python Language Reference](#)
- [pip documentation](#)
- [PySimpleGUI](#)
- [Flask](#)



Tutorial:

- [Beginner’s Guide to Python](#)
- [Hello, World!](#)
- [The Python Requirements File and How to Create it](#)



Videos:

- [Hello World in Python](#)
- [PySimpleGUI tutorial 1: Installation and Simple program](#)
- [Create an advanced GUI app with PySimpleGUI](#)
- [Flask Tutorial #1 - How to Make Websites with Python](#)

## VSCode

Learn Visual Studio Code in 7min



Ziel:

- Eine einfache universale IDE für alle Aufgaben des Trainings.



Time:

- 3 x 45 Minuten



Aufgaben:

- Installiere Install visual studio code: [download](#)
- Installiere Plugins für git, python, yaml, draw.io, vscode-pets, and more if you like.



Dokumentation:

- [docs](#)
- [marketplace](#)



Tutorial:

- [terminal/basics](#)
- [languages/markdown](#)
- [languages/python](#)
- [containers/overview](#)
- [sourcecontrol/github](#)



Videos:

- [Visual Studio Code](#)

## Reddit

What Is Reddit? How Reddit became the front page of the internet



Ziel:

- Netzwerkverbindungen zu Software Entwicklern.



Time:

- 2 x 45 Minuten



Aufgaben:

- Leg Dir einen Account an <https://www.reddit.com>
- Starte einen Timer mit 60 Minuten.
- Sieh Dir subreddits an, deren typischen Beiträge und Regeln.



Dokumentation:

- [reddit 101](#)



reddit:

- [learnprogramming](#)
- [AskProgramming](#)
- [devops](#)
- [java](#)
- [learnjava](#)
- [javahelp](#)
- [Python](#)
- [learnpython](#)
- [pythontips](#)
- [docker](#)
- [KeePass](#)
- [Office365](#)
- [Trello](#)

## Postman

[We did this | Postman](#)



Ziel:

- Postman zur Verfügung haben
- Fremde APIs managen
- Eigene APIs managen



Time:

- 3 x 45 Minuten



Aufgaben:

- Führe einen query aus ([postman-echo.com/get](https://postman-echo.com/get))
- Schreibe einen postman Test für GET auf [google.com](https://google.com)



Dokumentation:

- [docs][<https://learning.postman.com/docs/getting-started/introduction/>]



Tutorial:

- [sending-the-first-request](#)
- [designing-and-developing-your-api/testing-an-api](#)



Videos:

- [postman](#)
- [Always Check for the Hidden API when Web Scraping](#)
- [Postman API Entwicklungsumgebung - ideal für REST-Schnittstellen](#)
- [Postman API Crash Course for Beginners \[2020\] - Learn Postman in 1 hour](#)
- [REST API & RESTful Web Services Explained | Web Services Tutorial](#)

## Youtube



Ziel:

- Der Youtube algorithmus schlägt bubble gerecht vor.
- Hier ist die devops bubble



Time:

- N x 45 Minuten
- Youtube Videos sollten weit unten auf der prioritätenliste sein.
- Nutze Zeiten in denen Du nichts anderes machen kannst.



Aufgaben:

- Leg Dir einen Account zu.
- Leg Dir eine “Watch Later” Liste an.
- Füge Interessante Videos dort hinzu.
- Schließe Youtube nach einem Video.
- Sieh Dir die Spielzeit der Videos in der “Watch Later” Liste an.
- Entscheide gezieht welche Videos Deine Zeit Wert sind.



Kanäle:

[TechWorldwithNana](#), [shanselman](#), [Cleancoders](#), [CodingTech](#), [Computerphile](#), [hak5](#), [mediacccde](#), [NDC Conferences](#), [DevOpsConference](#), [Fireship](#), [DenshiV-ideo](#), [AcloudGuru](#), [JohnHammond010](#), [ContinuousDelivery](#), [NetworkChuck](#), [Socratica](#), [DEFCONConference](#)



## Open-Ended

What is DevOps? REALLY understand it | DevOps vs SRE



Ziel:

- Use your remaining prestudy time to tackle some extra topics



Time:

- N x 45 Minuten



Aufgaben:

- Sieh Dir jedes Thema für 5 Minuten an und überlege wo das Thema in DevOps platziert ist.
- Arbeite frei mit den Themen - Such Dir Dokumentation, Tutorials und Inspiration für Projekte (Youtube)
- Keine Panik, Du sollst nicht in den Prestudies durch die Liste unten kommen



Dokumentation:

- [Summary of 'Clean code' by Robert C. Martin](#)
- [SOLID](#)
- [List\\_of\\_software\\_development\\_philosophies](#)
- [Hexagonal architecture](#)
- [The Three Ways: The Principles Underpinning DevOps](#)
- [BDD](#)
- [TDD](#)
- [Clean Architecture](#)
- [markdown](#)
- [draw.io](#)
- [jenkins](#)
- [kubernetes](#)
- [Logging operator](#)
- [ansible](#)
- [webhook](#)
- [Azure](#)
- [Icinga](#)
- [prometheus](#)
- [grafana](#)
- [opentracing](#)
- [opentelemetry](#)
- [Elastic Stack](#)
- [minikube](#)
- [ssh](#)

- [What is SSL, TLS and HTTPS?](#)
- [ACR](#)
- [kubectl](#)
- [kustomize](#)
- [k8slens](#)
- [aks](#)
- [argocd](#)
- [terraform](#)
- [kafka](#)
- [plug-in](#)
- [CDN](#)
- [groovy](#)
- [TCP vs. UDP: Understanding 10 Key Differences](#)
- [Cron](#)
- [crontab.guru](#)