Termux Grundbefehle

- Guide 01: Getting Started (Erste Schritte)
- Guide 02: Package Management (Paketverwaltung)
- Guide 03: File System Navigation (Dateisystem)
- Guide 04: Text Editors (Texteditoren)
- Guide 05: Storage Management (Speicherverwaltung)
- Guide 06: Termux API (Android-Schnittstelle)
- Guide 07: Remote Access with SSH (Fernzugriff)
- Guide 08: Running Linux Distros (Linux-Distributionen)
- Guide 09: Python Environment (Python-Umgebung)Neue Seite
- Guide 10: User Customization mit ZSH (Personalisierung)
- Guide 11: Must-Have Packages (Essenzielle Pakete)
- Guide 12: Commands Every User Should Know (Grundbefehle)
- Guide 13: Customize Your Termux (Personalisierung)
- Guide 14: Top Keyboard Shortcuts (Tastatur-Kurzbefehle)
- Guide 15: Networking Essentials (Netzwerk-Grundlagen)
- Guide 16: Automate Everything (Automatisierung Basis)
- Guide 19: Git & GitHub in Termux (Versionsverwaltung)
- Guide 20: Secure Your Termux Like a Pro (Sicherheit) Seite
- Guide 21: Automate Like a Boss with Scripts & Tasks (Skripte)
- Guide 22: Optimize Termux Like a Pro (Optimierung)

Guide 01: Getting Started (Erste Schritte)

Der Einstieg in die Linux-Welt auf deinem Android-Gerät.

- **Was ist Termux?** Ein leistungsstarker Terminal-Emulator für Android, der eine vollwertige Linux-Umgebung bietet – komplett ohne Root-Rechte.
- **Wichtiger Installationshinweis:** Lade Termux **nicht** aus dem Google Play Store herunter (da veraltet und nicht mehr unterstützt). Nutze stattdessen **F-Droid** oder das offizielle **GitHub-Repository**.
- **Erste Befehle nach der Installation:**
  - System & Paketquellen aktualisieren: `pkg update && pkg upgrade`
  - Zugriff auf den Telefonspeicher gewähren: `termux-setup-storage`
- **Essentielle Navigationsbefehle:**
  - Aktuelles Verzeichnis anzeigen: `pwd`
  - Dateien und Ordner auflisten: `ls`
  - Verzeichnis wechseln: `cd <ordnername>`
  - Terminal-Bildschirm leeren: `clear`

Guide 02: Package Management (Paketverwaltung)

Programme suchen, installieren und sauber halten.

- **Der Paketmanager (`pkg` vs. `apt`):** Nutze in Termux vorzugsweise immer den Befehl `pkg`, da er im Hintergrund automatisch die passenden Android-Plattformkonfigurationen vornimmt.
- **Die wichtigsten Befehle:**
  - Paket suchen: `pkg search <paketname>`
  - Paket installieren: `pkg install <paketname>`
  - Paket deinstallieren: `pkg uninstall <paketname>`
  - Paketdetails anzeigen: `pkg show <paketname>`
  - Temporären Cache bereinigen: `pkg clean`
- **Optionale Repositories freischalten:**
  - Für grafische Anwendungen (X11): `pkg install x11-repo`
  - Für Root-spezifische Tools: `pkg install root-repo`

Guide 03: File System Navigation (Dateisystem)

Verstehe die Verzeichnisstruktur von Termux.

- **Wichtige Systempfade:**
  - `$HOME` (Dein persönlicher Startordner): `/data/data/com.termux/files/home`
  - `$PREFIX` (Systemdateien, Bibliotheken & Binaries): `/data/data/com.termux/files/usr`
- **Befehle zur Dateiverwaltung:**
  - Ordner erstellen: `mkdir <ordnername>`
  - Leeren Ordner löschen: `rmdir <ordnername>`
  - Leere Datei erstellen: `touch <dateiname>`
  - Datei löschen: `rm <dateiname>`
  - Ordner samt Inhalt unwiderruflich löschen: `rm -rf <ordnername>`
  - Datei/Ordner kopieren: `cp <quelle> <ziel>`
  - Datei/Ordner verschieben oder umbenennen: `mv <quelle> <ziel>`

Guide 04: Text Editors (Texteditoren)

Dateien direkt im Terminal bearbeiten.

- **Nano (Perfekt für Einsteiger):**
  - Installation: `pkg install nano`
  - Starten: `nano datei.txt`
  - *Shortcuts:* Speichern mit `CTRL + O`, Beenden mit `CTRL + X`.
- **Micro (Modern & Intuitiv):**
  - Installation: `pkg install micro`
  - Starten: `micro datei.txt`
  - *Vorteil:* Unterstützt vertraute PC-Shortcuts wie `CTRL + C` (Kopieren), `CTRL + V` (Einfügen) und `CTRL + S` (Speichern).
- **Vim (Mächtig für Fortgeschrittene):**
  - Installation: `pkg install vim`
  - Starten: `vim datei.txt`
  - *Bedienung:* `i` drücken für den Schreibmodus (Insert). `ESC` drücken, um den Modus zu verlassen. Speichern & Schließen mit `:wq`, Schließen ohne Speichern mit `:q!`.
- **Emacs (Extrem erweiterbar):**
  - Installation: `pkg install emacs`

Guide 05: Storage Management (Speicherverwaltung)

Verknüpfung zwischen Termux und deinem Android-Dateisystem.

- **Aktivierung:** Führe zuerst `termux-setup-storage` aus und bestätige die Android-Berechtigung.
- Dadurch wird das Verzeichnis `~/storage` in deinem Home-Ordner angelegt. Es enthält folgende symbolische Links zu deinen echten Android-Ordnern:
  - `~/storage/shared` $\rightarrow$ Das Hauptverzeichnis deines internen Speichers (`/sdcard`)
  - `~/storage/downloads` $\rightarrow$ Dein standardmäßiger Download-Ordner
  - `~/storage/dcim` $\rightarrow$ Fotos und Kameraaufnahmen
  - `~/storage/pictures` $\rightarrow$ Bilder-Ordner
  - `~/storage/music` $\rightarrow$ Musik-Ordner
  - `~/storage/movies` $\rightarrow$ Video-Ordner

Guide 06: Termux API (Android-Schnittstelle)

Steuere deine Hardware und Android-Funktionen über Code.

- **Vorbereitung:**
  1. Installiere die App **Termux:API** aus F-Droid.
  2. Installiere das Gegenstück im Terminal: `pkg install termux-api`
- **Praktische API-Befehle:**
  - Batteriestatus auslesen: `termux-battery-status`
  - Foto über die Kamera schießen: `termux-camera-photo <ausgabe.jpg>`
  - Text aus der Android-Zwischenablage lesen: `termux-clipboard-get`
  - Text in die Zwischenablage schreiben: `termux-clipboard-set "Dein Text"`
  - Taschenlampe an- und ausschalten: `termux-torch on` / `termux-torch off`
  - Eine Android-Toast-Nachricht einblenden: `termux-toast "Hallo Welt"`
  - Smartphone vibrieren lassen: `termux-vibrate -d 1000` *(Dauer in ms)*
  - Sprachausgabe (Text-to-Speech): `termux-tts-speak "Hallo vom Terminal"`

Guide 07: Remote Access with SSH (Fernzugriff)

Verbinde dich vom PC aus mit deinem Smartphone-Terminal.

- **Installation & Steuerung:**
  - SSH installieren: `pkg install openssh`
  - SSH-Server starten: `sshd`
  - SSH-Server stoppen: `pkill sshd`
- **Verbindungsdaten auslesen:**
  - Deinen aktuellen User abfragen: `whoami`
  - Ein Passwort für den Login festlegen: `passwd`
  - Die IP-Adresse deines Smartphones ermitteln: `ifconfig` oder `ip a`
- **Vom PC aus einloggen:**
  - *Hinweis:* Termux nutzt standardmäßig den Port **8022** statt der üblichen 22.
  - Befehl: `ssh <dein_username>@<deine_ip_adresse> -p 8022`

Guide 08: Running Linux Distros (Linux-Distributionen)

Betreibe vollwertige Linux-Systeme via PRoot (kein Root erforderlich).

- **Befehlsübersicht:**
  - PRoot-Manager installieren: `pkg install proot-distro`
  - Liste aller verfügbaren Linux-Systeme anzeigen: `proot-distro list`
  - Eine Linux-Distribution installieren (Beispiel Ubuntu): `proot-distro install ubuntu`
  - In die installierte Linux-Umgebung einloggen: `proot-distro login ubuntu`
- **Unterstützte Distributionen:** Ubuntu, Debian, Alpine Linux, Arch Linux, Fedora, Manjaro, Void Linux.

Guide 09: Python Environment (Python-Umgebung)Neue Seite

Skripte schreiben und ausführen direkt auf dem Handy.

- **Grundlagen:**
  - Python installieren: `pkg install python`
  - Installierte Version prüfen: `python --version`
- **Paketverwaltung mit PIP:**
  - PIP aktualisieren: `pip install --upgrade pip`
  - Bibliotheken installieren: `pip install <paketname>` *(z.B. requests, numpy)*
- **Skriptausführung:**
  - Python-Datei starten: `python skript.py`
  - Interaktive Python-Konsole öffnen: `python` *(Verlassen mit `exit()`)*
- **Virtuelle Umgebungen (Virtual Envs):**
  - Tool installieren: `pkg install python-virtualenv`
  - Umgebung erstellen: `python -m venv meinenv`
  - Umgebung aktivieren: `source meinenv/bin/activate`

Guide 10: User Customization mit ZSH (Personalisierung)

Verpasse deinem Terminal ein Upgrade mit Oh My Zsh, Themes und Plugins.

Voraussetzungen installieren:

```
pkg install zsh git curl
```

Oh My Zsh installieren:

```
sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

Zsh als Standard-Shell festlegen:

```
chsh -s zsh
```

Nützliche Plugins hinzufügen (Syntax-Highlighting & Autosuggestions):

- *Highlighting:*

```
git clone https://github.com/zsh-users/zsh-syntax-highlighting.git ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-syntax-highlighting
```

- *Autosuggestions:*

```
git clone https://github.com/zsh-users/zsh-autosuggestions ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-autosuggestions
```

Plugins in der Konfiguration aktivieren:

Öffne die Datei `~/.zshrc` mit einem Editor und passe die Plugin-Zeile wie folgt an:

```
plugins=(git zsh-syntax-highlighting zsh-autosuggestions)
```

Guide 11: Must-Have Packages (Essenzielle Pakete)

Supercharge your Termux with these essential packages for every user.

- **NEOFETCH:** Zeigt Systeminformationen schön an.
  - Command: `pkg install neofetch`
- **FIGLET:** Erstellt ASCII-Art-Banner.
  - Command: `pkg install figlet`
- **NMAP:** Netzwerk-Scanner & Sicherheits-Auditing-Tool.
  - Command: `pkg install nmap`
- **GIT:** Versionsverwaltungssystem für Entwickler.
  - Command: `pkg install git`
- **PYTHON:** Mächtige Programmiersprache für Skripte.
  - Command: `pkg install python`
- **HTOP:** Interaktiver Prozess-Viewer (besser als `top`).
  - Command: `pkg install htop`
- **RIPGREP:** Superschnelles Suchwerkzeug für Text.
  - Command: `pkg install ripgrep`
- **SQLITE:** Befehlszeilen-Datenbankverwaltung.
  - Command: `pkg install sqlite`

> **Pro-Tipp:** Halte deine Pakete immer aktuell mit: `pkg update && pkg upgrade`
> **Extra-Tipp:** Nutze `termux-change-repo`, um zu schnelleren Spiegelservern (Mirrors) zu wechseln.

Guide 12: Commands Every User Should Know (Grundbefehle)

Master the basics. Build the foundation. Level up your Termux game.

- **ls:** Listet Dateien und Verzeichnisse auf. -> `ls -lah`
- **cd:** Verzeichnis wechseln. -> `cd /sdcard`
- **cat:** Dateiinhalt anzeigen. -> `cat file.txt`
- **grep:** Nach Textmustern suchen. -> `grep "root" file.txt`
- **pkg:** Pakete verwalten. -> `pkg update && pkg upgrade`
- **cp:** Dateien oder Verzeichnisse kopieren. -> `cp file.txt /sdcard/`
- **rm:** Dateien oder Verzeichnisse löschen. -> `rm file.txt`
- **chmod:** Dateiberechtigungen ändern. -> `chmod 755 script.sh`
- **clear:** Terminal-Bildschirm leeren. -> `clear`
- **history:** Befehlshistorie anzeigen. -> `history`

Guide 13: Customize Your Termux (Personalisierung)

Make Termux truly yours. Themes, prompts, aliases, and configs.

1. **APPLY A THEME:** Nutze das `termux-style`-Repository, um Themes zu installieren.
2. **POWERLINE PROMPT:** Macht die Eingabeaufforderung sauberer und informativer.
3. **ALIASES:** Erstelle Kurzbefehle für deine Lieblingskommandos.
4. **CUSTOM .BASHRC:** Bearbeite `~/.bashrc`, um Aliases und Umgebungsvariablen zu setzen.
5. **FONTS & COLORS:** Ändere Schriftarten und Farbschemata.
6. **MOTD & BANNER:** Füge eine "Message of the Day" für den echten Hacker-Vibe hinzu.
7. **CONFIG BACKUP:** Sichere deine Konfigurationen in der Cloud.

Beliebte Themes

- Matrix
- Dracula
- Nord
- Gruvbox

Praktische Aliases für die `~/.bashrc`

Bash

```
alias ll='ls -lah --color=auto'
alias gs='git status'
alias update='pkg update && pkg upgrade'
alias cls='clear'
alias ..='cd ..'
```

Guide 14: Top Keyboard Shortcuts (Tastatur-Kurzbefehle)

Boost your speed and productivity. Learn the essential Termux shortcuts.

Shortcut	Funktion
CTRL + A	Cursor an den Anfang der Zeile bewegen
CTRL + E	Cursor an das Ende der Zeile bewegen
CTRL + C	Aktuellen Prozess abbrechen (Kill)
CTRL + V	Aus der Zwischenablage einfügen
CTRL + L	Terminal-Bildschirm leeren
CTRL + Z	Aktuellen Prozess pausieren (Suspend)
CTRL + D	Ausloggen / Session beenden
CTRL + R	Rückwärtssuche in der Befehlshistorie
TAB	Autovervollständigung für Befehle & Dateien
ESC, ESC	Letztes Argument einfügen (entspricht `.`)

> **Zusatz-Tipps:**
>
> - **ESC lange drücken:** Öffnet extra Tasten (F1-F12, Pfeiltasten, Home, End).
> - **Lautstärketasten nutzen:** Lautstärke hoch (`Vol+`) = Page Up | Lautstärke runter (`Vol-`) = Page Down.

Guide 15: Networking Essentials (Netzwerk-Grundlagen)

Diagnose, scan, and secure your connection like a pro.

- **PING:** Prüfen, ob ein Host erreichbar ist. -> `ping google.com`
- **TRACEROUTE:** Route zu einem Host verfolgen. -> `traceroute google.com`
- **IP INFO:** IP & Standort-Info anzeigen. -> `curl ifconfig.me`
- **NETSTAT:** Netzwerkverbindungen anzeigen. -> `netstat -tuln`
- **NMAP (PORT SCAN):** Offene Ports eines Hosts scannen. -> `nmap -sS -T4 127.0.0.1`
- **DNS LOOKUP:** DNS-Einträge einer Domain auflösen. -> `dig google.com`
- **WGET:** Dateien aus dem Web herunterladen. -> `wget example.com/file`
- **CURL:** Daten von/zu Servern übertragen. -> `curl -O example.com/file`
- **SS (SOCKET STAT):** Modernes Tool zur Socket-Analyse. -> `ss -tuln`
- **OPENSSL:** SSL/TLS-Verbindungen testen. -> `openssl s_client -connect example.com:443`

Guide 16: Automate Everything (Automatisierung Basis)

Work less. Automate more. Let Termux do the boring stuff.

1. **CRON JOBS:** Aufgaben planen und im Hintergrund ausführen. -> `crontab -e`
2. **BASH SCRIPTS:** Befehle, Backups und Updates automatisieren. -> `nano script.sh`
3. **TASKER + TERMUX:** Termux-Aufgaben automatisiert über Android (Tasker) triggern. -> `termux-task`
4. **AUTO BACKUPS:** Wichtige Dateien täglich in die Cloud sichern. -> `rclone sync`
5. **NOTIFICATIONS:** Benachrichtigungen bei fertigen Aufgaben erhalten. -> `termux-notification -c`
6. **API REQUESTS:** Daten von APIs abrufen und protokollieren. -> `curl -s api.url`
7. **DATA PROCESSING:** Daten parsen, filtern und analysieren wie ein Profi. -> `jq | grep | awk`

Guide 19: Git & GitHub in Termux (Versionsverwaltung)

Version control. Collaborate. Deploy. All from your phone.

- **Repository klonen:** `git clone https://github.com/user/repo.git`
- **Status prüfen:** `git status`
- **Änderungen hinzufügen:** `git add .`
- **Commit erstellen:** `git commit -m "message"`
- **Zu Remote hochladen:** `git push origin main` *(Hinweis: Auf der Infografik-Tabelle ist ein Tippfehler "pit push", im Terminal-Screen rechts steht es korrekt mit "git")*
- **Updates herunterziehen:** `git pull origin main`
- **Branch-Liste anzeigen:** `git branch -a`
- **Branch erstellen:** `git branch new-branch`
- **Branch wechseln:** `git checkout new-branch`
- **Branch zusammenführen:** `git merge new-branch`
- **Datei entfernen:** `git rm file.txt`
- **Verlauf anzeigen:** `git log --oneline --graph`
- **Änderungen einsehen:** `git diff`
- **GitHub Auth (Token):** `git config --global credential.helper store`

Guide 20: Secure Your Termux Like a Pro (Sicherheit) Seite

Harden your mobile setup. Protect your data. Stay private. Stay in control.

1. **UPDATE EVERYTHING:** Pakete aktuell halten. -> `pkg update && pkg upgrade -y`
2. **STRONG PASSWORDS:** Sichere Passwörter verwenden. -> `passwd`
3. **SSH ACCESS:** Sicherer Zugriff auf das Gerät. -> `pkg install openssh`
4. **ENABLE FINGERPRINT:** `termux-auth` für zusätzliche Sicherheit nutzen. -> `pkg install termux-auth`
5. **ENCRYPT STORAGE:** Wichtige Dateien verschlüsseln. -> `pkg install cryptsetup`
6. **HIDE YOUR ACTIVITY:** Verlauf und Metadaten löschen. -> `history -c && clear`
7. **USE FIREWALL:** Unerwünschte Verbindungen blockieren. -> `pkg install termux-firewall`
8. **SCAN FOR MALWARE:** Das System regelmäßig scannen. -> `pkg install clamav`
9. **KEEP BACKUPS:** Wichtige Daten sichern. -> `rsync -avz ~/ /sdcard/backup/`
10. **SECURE DELETE:** Dateien sicher und unwiederbringlich löschen. -> `shred -u file.txt`

Guide 21: Automate Like a Boss with Scripts & Tasks (Skripte)

Write once. Run anywhere. Save time.

1. **CREATE A SCRIPT:** Skriptdatei erstellen. -> `nano myscript.sh`
2. **MAKE IT EXECUTABLE:** Ausführungsrechte vergeben. -> `chmod +x myscript.sh`
3. **RUN YOUR SCRIPT:** Skript ausführen. -> `./myscript.sh`
4. **SCHEDULE WITH CRON:** Crontab bearbeiten. -> `crontab -e`
5. **BACKGROUND EXECUTION:** Langlaufende Aufgaben im Hintergrund ausführen. -> `./myscript.sh &`
6. **LOG OUTPUT:** Ausgaben in einer Logdatei speichern. -> `./myscript.sh > log.txt 2>&1`
7. **RUN AT BOOT (ANDROID 7+):** Skripte beim Gerätestart ausführen. -> `termux-boot /sdcard/boot.sh`
8. **MONITOR SERVICES:** Dienste im Auge behalten. -> `termux-service list`
9. **NOTIFICATIONS:** Benachrichtigung bei Skript-Ende senden. -> `termux-notification -c "Done!"`
10. **VERSION CONTROL:** Skripte mit Git tracken. -> `git init && git add . && git commit -m "init"`

Guide 22: Optimize Termux Like a Pro (Optimierung)

Speed up your workflow. Customize everything. Make Termux truly yours.

1. **UPDATE & UPGRADE:** System frisch halten. -> `pkg update && pkg upgrade -y`
2. **CLEAN CACHE:** Speicherplatz freigeben. -> `pkg clean && rm -rf $PREFIX/var/cache/*`
3. **USE PROOT DISTROS:** Vollwertige Linux-Distributionen ausführen. -> `pkg install proot-distro`
4. **MULTIPLEX TERMINALS:** Sitzungen verwalten wie ein Profi. -> `pkg install tmux` -> `tmux`
5. **MONITOR RESOURCES:** Performance überwachen. -> `pkg install htop` -> `htop`
6. **CREATE ALIASES:** Lange Befehle verkürzen. -> `nano ~/.bashrc` -> `alias ll='ls -lah'`
7. **SMART SEARCH:** Dateien blitzschnell finden. -> `pkg install ripgrep` -> `rg 'pattern'`
8. **BACKGROUND JOBS:** Befehle im Hintergrund laufen lassen. -> `command &` -> `jobs -l`
9. **SYNC & BACKUP:** Daten geräteübergreifend abgleichen. -> `pkg install rsync`
10. **SECURE ACCESS:** SSH-Schlüssel statt Passwörter nutzen. -> `ssh-keygen -t ed25519`
11. **EXTEND WITH PLUGINS:** Die Shell aufwerten. -> `pkg install zsh` -> `chsh -s zsh`
12. **CUSTOMIZE LOOK:** Optik verschönern. -> `nano ~/.termux/colors.properties`