



120+ BEFEHLE

# Raspberry Pi Befehle Cheat Sheet

Die wichtigsten Terminal-Befehle für Raspberry Pi OS –  
von den ersten Schritten über GPIO und Kamera bis zu  
Backup & Bootloader.

raspberrypi.tips · Stand Juli 2026

Dieses PDF darf unter Angabe der Quelle (raspberrypi.tips) vervielfältigt, genutzt und weitergegeben werden.

# Inhalt

---

1. Erste Schritte & System-Basics
2. System-Info & Diagnose
3. Dateien & Verzeichnisse
4. Netzwerk & SSH
5. Paketverwaltung (APT)
6. Prozesse & Dienste (systemd)
7. GPIO & Hardware-Schnittstellen
8. Kamera-Modul
9. Bootloader, Firmware & Updates
10. Storage, SD-Karte & Backup
11. Benutzer & Rechte
12. Tastenkombinationen & Terminal-Tricks

Ob frisch aufgesetzter Pi oder jahrelanger Dauerbetrieb: Irgendwann steht man vor dem Terminal und weiß den Befehl nicht mehr auswendig. Dieses **Raspberry Pi Befehle Cheat Sheet** sammelt über 120 Terminal-Kommandos für Raspberry Pi OS an einem Ort – von den ersten Schritten über GPIO und Kamera bis zu Bootloader-Updates und SD-Karten-Backup. Kein Tutorial zum Durchlesen, sondern eine Referenz zum Nachschlagen und Kopieren.

### Was ist ein Raspberry Pi Cheat Sheet?

Ein Raspberry Pi Cheat Sheet ist eine kompakte Referenzliste der wichtigsten Terminal-Befehle für Raspberry Pi OS, sortiert nach Themen wie System, Netzwerk, GPIO oder Backup. Es ersetzt kein Tutorial, sondern dient als schnelles Nachschlagewerk für Befehle, die man sich nicht dauerhaft merken will oder muss.

**So nutzt du dieses Cheat Sheet:** Alle Befehle sind nach Themen sortiert und direkt zum Kopieren gedacht. Platzhalter wie `<Pfad>` oder `<Dienst>` ersetzt du durch deine eigenen Werte. Befehle mit vorangestelltem `sudo` brauchen Admin-Rechte.

### Lieber zum Ausdrucken?

Alle 120+ Befehle als PDF zum Herunterladen – kostenlos, ohne Anmeldung.

[PDF herunterladen](#)

## Erste Schritte & System-Basics

Das Grundgerüst, das auf jedem frisch aufgesetzten Pi als Erstes gebraucht wird.

Befehl	Was er macht
<code>sudo raspi-config</code>	Öffnet das zentrale, menügeführte Konfigurationstool: Spracheinstellung, Boot-Verhalten, Schnittstellen (SSH, I2C, SPI, Kamera), Übertaktung – die meisten Grundeinstellungen laufen hierüber.
<code>passwd</code>	Ändert das Passwort des aktuell angemeldeten Benutzers – fragt zuerst das alte, dann zweimal das neue Passwort ab.
<code>sudo reboot</code>	Startet den Pi sauber neu, alle laufenden Dienste werden vorher ordentlich beendet.
<code>sudo shutdown -h now</code>	Führt den Pi sofort und sicher herunter (h = halt) – wichtig vor dem Ziehen des Stromkabels, sonst drohen SD-Karten-Schäden.
<code>sudo shutdown -r +5</code>	Kündigt allen angemeldeten Nutzern einen Neustart in 5 Minuten an, danach läuft er automatisch.
<code>whoami</code>	Gibt den Benutzernamen des aktuell aktiven Terminal-Nutzers aus – praktisch nach einem <code>su</code> oder <code>sudo -i</code> .
<code>hostname</code>	Zeigt nur den Gerätenamen (ohne Domain), unter dem sich der Pi im Netzwerk meldet.

Befehl	Was er macht
<code>hostname -I</code>	Listet alle dem Pi zugewiesenen IPv4-/IPv6-Adressen (LAN, WLAN, VPN) in einer Zeile – siehe auch unser <a href="#">ausführlicher Guide zur IP-Adresse</a> .
<code>uname -a</code>	Gibt Kernel-Name, Version, Build-Datum und Architektur (z. B. aarch64) in einer Zeile aus – wichtig für Kompatibilitäts-Checks.
<code>cat /etc/os-release</code>	Zeigt Name, Versionsnummer und Codename des installierten Raspberry Pi OS – zuverlässiger als <code>uname</code> für die OS-Version.
<code>date</code>	Gibt aktuelles Datum und Uhrzeit inkl. Zeitzone aus, z. B. zum Abgleich mit Cronjob- oder Log-Zeitstempeln.
<code>timedatectl</code>	Zeigt lokale Zeit, UTC, Zeitzone und NTP-Sync-Status; mit <code>sudo timedatectl set-timezone</code> lässt sich die Zeitzone direkt ändern.
<code>uptime</code>	Zeigt Laufzeit seit dem letzten Boot sowie die Systemlast der letzten 1/5/15 Minuten (Load Average).

## System-Info & Diagnose

`vcgencmd` ist Pi-exklusiv (Broadcom-Firmware-Schnittstelle) und funktioniert nicht auf normalem Debian.

Befehl	Was er macht
<code>vcgencmd measure_temp</code>	Liest die SoC-Temperatur direkt vom Sensor aus, z. B. <code>temp=42.8'C</code> – ab 80 °C drosselt der Pi automatisch den Takt, ab 85 °C wird zusätzlich throttled.
<code>vcgencmd get_throttled</code>	Prüft, ob der Pi wegen zu wenig Strom oder zu hoher Temperatur gedrosselt wurde. <code>0x0</code> heißt: alles unauffällig – jeder andere Wert bedeutet, dass das aktuell oder seit dem letzten Start schon einmal passiert ist.
<code>vcgencmd measure_volts</code>	Zeigt standardmäßig die Spannung der Kernkomponente <code>core</code> ; per Parameter auch <code>sdram_c</code> / <code>sdram_i</code> / <code>sdram_p</code> für die RAM-Bereiche.
<code>vcgencmd measure_clock arm</code>	Gibt die aktuelle CPU-Taktfrequenz in Hz aus – zum Abgleich mit dem in <code>raspi-config</code> eingestellten Übertaktungswert.
<code>vcgencmd get_config int</code>	Listet alle als Ganzzahl gesetzten <code>config.txt</code> -Parameter (z. B. <code>arm_freq</code> , <code>over_voltage</code> ) mit ihrem aktuell aktiven Wert.
<code>vcgencmd version</code>	Zeigt Versionsnummer und Build-Datum der GPU-Firmware/Bootloader-Kombination – hilfreich beim Melden von Bugs.
<code>free -h</code>	Zeigt RAM- und Swap-Belegung in lesbaren Einheiten (h = MB/GB statt Bytes) inkl. verfügbarem und gepuffertem Speicher.
<code>df -h</code>	Zeigt belegten/freien Platz aller gemounteten Dateisysteme in lesbaren Einheiten – nützlich, um eine volle SD-Karte frühzeitig zu erkennen.

Befehl	Was er macht
<code>du -sh &lt;Ordner&gt;</code>	Ermittelt die Gesamtgröße eines Ordners (s = summiert, h = lesbar) – gut zum Aufspüren von Speicherfressern vor dem Aufräumen.
<code>top</code>	Zeigt laufende Prozesse live, sortiert nach CPU-Auslastung – mit <code>q</code> beenden, mit <code>P / M</code> nach CPU bzw. RAM neu sortieren.
<code>htop</code>	Farbige, scrollbare Alternative zu <code>top</code> mit Maus-Bedienung und Balkenanzeige pro CPU-Kern (ggf. <code>sudo apt install htop</code> ).
<code>lscpu</code>	Zeigt Kernanzahl, Architektur, Taktbereich und Cache-Größen der CPU in strukturierter Form.
<code>vccgncmd get_camera</code>	Gibt <code>supported=1 detected=1</code> aus, wenn ein Kameramodul korrekt erkannt und aktiviert wurde – erste Anlaufstelle bei Kamera-Problemen.

**Praxis-Tipp:** Für Dauerbetrieb und Backups (siehe Punkt 10) lohnt sich eine Karte mit hoher Schreibfestigkeit, z. B. die [Samsung Pro Endurance microSD](#) – normale Consumer-Karten sind für 24/7-Logging nicht ausgelegt. Mehr dazu in unserem [SD-Karten-Vergleich](#).

## Dateien & Verzeichnisse

Befehl	Was er macht
<code>ls -la</code>	Listet alle Dateien inkl. versteckter ( <code>-a</code> , beginnend mit Punkt) im Langformat ( <code>-l</code> : Rechte, Besitzer, Größe, Datum).
<code>cd &lt;Pfad&gt;</code>	Wechselt das Arbeitsverzeichnis; <code>cd ..</code> geht eine Ebene hoch, <code>cd</code> ohne Argument zurück ins Home-Verzeichnis.
<code>pwd</code>	Gibt den vollständigen, absoluten Pfad des aktuellen Verzeichnisses aus (print working directory).
<code>cp &lt;Quelle&gt; &lt;Ziel&gt;</code>	Kopiert eine einzelne Datei – ein bereits existierendes Ziel wird ohne Rückfrage überschrieben.
<code>cp -r &lt;Quelle&gt; &lt;Ziel&gt;</code>	Kopiert einen kompletten Ordner mit allem, was darin liegt – auch verschachtelte Unterordner (dafür sorgt das Zusatz-Flag <code>-r</code> ). Ohne <code>-r</code> bricht <code>cp</code> bei Ordnern mit einer Fehlermeldung ab.
<code>mv &lt;Quelle&gt; &lt;Ziel&gt;</code>	Verschiebt eine Datei/Ordner oder benennt sie um, wenn Quelle und Ziel im selben Verzeichnis liegen.
<code>rm &lt;Datei&gt;</code>	Löscht eine Datei unwiderruflich – es gibt keinen Papierkorb und kein Standard-Undo.

Befehl	Was er macht
<code>rm -r &lt;Ordner&gt;</code>	Löscht einen Ordner mit allem, was darin liegt, auch verschachtelte Unterordner ( <code>-r</code> ) – in Kombination mit <code>-f</code> (force) erfolgt keine Rückfrage mehr, also besonders vorsichtig verwenden.
<code>mkdir &lt;Name&gt;</code>	Legt einen neuen, leeren Ordner an; mit <code>-p</code> werden fehlende übergeordnete Ordner gleich mit erstellt.
<code>touch &lt;Datei&gt;</code>	Erstellt eine leere Datei, falls sie nicht existiert, oder aktualisiert nur den Zeitstempel einer bestehenden Datei.
<code>nano &lt;Datei&gt;</code>	Öffnet die Datei im einsteigerfreundlichen Terminal-Editor – speichern mit <code>Strg+O</code> , beenden mit <code>Strg+X</code> .
<code>cat &lt;Datei&gt;</code>	Gibt den kompletten Dateiinhalt am Stück aus – bei langen Dateien unübersichtlich, dafür eignet sich <code>less</code> besser.
<code>less &lt;Datei&gt;</code>	Öffnet die Datei seitenweise zum Durchblättern (Pfeiltasten, <code>/</code> zum Suchen), ohne sie komplett in den Speicher zu laden.
<code>find &lt;Pfad&gt; -name "&lt;Muster&gt;"</code>	Durchsucht einen Ordner inklusive aller Unterordner nach Dateinamen, Muster erlaubt Platzhalter wie <code>*.log</code> für "alles, was auf .log endet".
<code>grep -r "&lt;Text&gt;" &lt;Pfad&gt;</code>	Durchsucht einen Ordner samt aller Unterordner ( <code>-r</code> ) nach einem Textstück und zeigt für jeden Treffer Dateiname plus die passende Zeile an.
<code>tar -czvf archiv.tar.gz &lt;Ordner&gt;</code>	Packt einen Ordner in ein komprimiertes Archiv ( <code>c</code> = create, <code>z</code> = gzip, <code>v</code> = verbose, <code>f</code> = Dateiname).
<code>tar -xzvf archiv.tar.gz</code>	Entpackt ein gzip-komprimiertes tar-Archiv ( <code>x</code> = extract) ins aktuelle Verzeichnis.
<code>chmod +x &lt;Datei&gt;</code>	Setzt das Ausführungsrecht für Besitzer, Gruppe und Andere – notwendig, damit Skripte direkt per <code>./skript.sh</code> gestartet werden können.
<code>chown pi:pi &lt;Datei&gt;</code>	Ändert Besitzer ( <code>pi</code> ) und Gruppe ( <code>pi</code> ) einer Datei – häufig nötig, wenn Dateien z. B. per FTP als root angelegt wurden.

## Netzwerk & SSH

Befehl	Was er macht
<code>ip a</code>	Zeigt alle Netzwerkschnittstellen ( <code>eth0</code> , <code>wlan0</code> , <code>lo</code> ) mit IPv4-/IPv6-Adressen und Status (UP/DOWN) – detaillierter als <code>hostname -I</code> .
<code>ping &lt;Adresse&gt;</code>	Sendet ICMP-Pakete an ein Ziel und misst die Antwortzeit – mit <code>Strg+C</code> beenden, mit <code>-c 4</code> auf 4 Pakete begrenzen.

Befehl	Was er macht
<code>nmcli device wifi list</code>	Listet alle in Reichweite sichtbaren WLAN-Netzwerke mit Signalstärke und Verschlüsselungstyp auf.
<code>nmcli device wifi connect "&lt;SSID&gt;" password "&lt;Passwort&gt;"</code>	Verbindet sich direkt per Kommandozeile mit einem WLAN, ohne <code>raspi-config</code> zu öffnen.
<code>iwgetid</code>	Zeigt die SSID des aktuell verbundenen WLANs – schnellster Check, ob überhaupt eine WLAN-Verbindung besteht.
<code>sudo raspi-config nonint do_ssh 0</code>	Aktiviert SSH nicht-interaktiv per Skript (0 = an, 1 = aus) – ideal, um Headless-Setups zu automatisieren, siehe auch unser <a href="#">SSH-Tutorial</a> .
<code>ssh pi@&lt;IP-Adresse&gt;</code>	Baut eine verschlüsselte Terminal-Verbindung zum Pi auf – beim ersten Verbinden muss der Host-Fingerprint bestätigt werden.
<code>scp &lt;Datei&gt; pi@&lt;IP&gt;:&lt;Zielpfad&gt;</code>	Kopiert eine Datei verschlüsselt über SSH auf den Pi, Syntax wie <code>cp</code> , nur mit Host-Adresse davor.
<code>ssh-keygen -t ed25519</code>	Erzeugt ein modernes, sicheres SSH-Schlüsselpaar – Public Key anschließend mit <code>ssh-copy-id</code> auf den Pi übertragen für passwortlosen Login.
<code>curl -I &lt;URL&gt;</code>	Ruft nur die HTTP-Response-Header ab, ohne den Seiteninhalt herunterzuladen – schneller Test, ob ein Server erreichbar ist und welchen Status er liefert.
<code>wget &lt;URL&gt;</code>	Lädt eine Datei direkt ins aktuelle Verzeichnis herunter, unterstützt mit <code>-c</code> auch das Fortsetzen abgebrochener Downloads.
<code>sudo raspi-config nonint get_hostname</code>	Gibt den aktuell gesetzten Hostnamen aus – Gegenstück zu <code>do_hostname</code> zum Setzen eines neuen Namens.

## Paketverwaltung (APT)

Befehl	Was er macht
<code>sudo apt update</code>	Lädt die aktuellen Paketlisten von den konfigurierten Repositories – installiert noch nichts, ist aber Voraussetzung für <code>upgrade/install</code> .
<code>sudo apt upgrade -y</code>	Installiert für bereits vorhandene Pakete verfügbare neue Versionen, ohne neue Abhängigkeiten zu installieren oder Pakete zu entfernen ( <code>-y</code> bestätigt automatisch).

Befehl	Was er macht
<code>sudo apt full-upgrade -y</code>	Wie upgrade, darf aber zusätzlich Pakete entfernen oder neu installieren, wenn das für die Auflösung von Versionskonflikten nötig ist – empfohlen bei größeren OS-Updates.
<code>sudo apt install &lt;Paket&gt;</code>	Installiert ein neues Paket samt aller benötigten Abhängigkeiten.
<code>sudo apt remove &lt;Paket&gt;</code>	Deinstalliert das Programm, lässt aber Konfigurationsdateien in <code>/etc</code> unangetastet – praktisch bei einer späteren Neuinstallation.
<code>sudo apt purge &lt;Paket&gt;</code>	Entfernt Programm und Konfigurationsdateien vollständig – für einen wirklich sauberen Neustart der Installation.
<code>sudo apt autoremove</code>	Entfernt automatisch installierte Abhängigkeiten, die von keinem verbleibenden Paket mehr benötigt werden.
<code>apt list --installed</code>	Zeigt alle über apt/dpkg installierten Pakete mit Versionsnummer – gut kombinierbar mit <code>  grep &lt;Name&gt;</code> .
<code>apt-cache search &lt;Begriff&gt;</code>	Durchsucht Paketnamen und Kurzbeschreibungen der Repositories nach einem Stichwort, bevor man den genauen Paketnamen kennt.
<code>dpkg -l   grep &lt;Name&gt;</code>	Prüft direkt über dpkg (die Ebene unter apt), ob und in welcher Version ein Paket installiert ist.
<code>sudo apt clean</code>	Löscht heruntergeladene .deb-Installationsdateien aus <code>/var/cache/apt/archives</code> – gibt bei knappem Speicher schnell Platz frei.

## Prozesse & Dienste (systemd)

Befehl	Was er macht
<code>systemctl status &lt;Dienst&gt;</code>	Zeigt, ob ein Dienst aktiv/inaktiv/fehlgeschlagen ist, seit wann, und die letzten Log-Zeilen direkt mit dazu.
<code>sudo systemctl start &lt;Dienst&gt;</code>	Startet einen Dienst sofort, ändert aber nichts am Autostart-Verhalten beim nächsten Boot.
<code>sudo systemctl stop &lt;Dienst&gt;</code>	Beendet einen laufenden Dienst sofort, ohne ihn dauerhaft zu deaktivieren.
<code>sudo systemctl restart &lt;Dienst&gt;</code>	Beendet und startet einen Dienst neu in einem Schritt – Standardvorgehen nach Konfigurationsänderungen.
<code>sudo systemctl enable &lt;Dienst&gt;</code>	Sorgt dafür, dass der Dienst bei jedem Systemstart automatisch mitgestartet wird.
<code>sudo systemctl disable &lt;Dienst&gt;</code>	Entfernt den Autostart-Eintrag – der Dienst muss danach manuell gestartet werden, läuft aber ggf. noch bis zum nächsten Neustart.
<code>systemctl list-units --type=service</code>	Listet alle aktuell geladenen Dienste mit Status (active/inactive/failed) auf.

Befehl	Was er macht
<code>ps aux</code>	Zeigt alle laufenden Prozesse aller Benutzer mit PID, CPU-/RAM-Auslastung und vollem Befehl.
<code>ps aux   grep &lt;Name&gt;</code>	Filtert die Prozessliste auf einen Namen – schnellster Weg, um die PID eines Programms zu finden.
<code>kill &lt;PID&gt;</code>	Bittet einen Prozess höflich, sich zu beenden – er darf vorher noch aufräumen (z. B. Dateien speichern), reagiert aber nicht immer sofort.
<code>kill -9 &lt;PID&gt;</code>	Beendet einen Prozess sofort und ohne Rücksicht, ohne ihm Zeit zum Aufräumen zu lassen – nur nutzen, wenn <code>kill</code> allein nicht reicht.
<code>killall &lt;Name&gt;</code>	Beendet alle Prozesse mit exakt diesem Namen gleichzeitig, ohne vorher die PID nachschlagen zu müssen.
<code>journalctl -xe</code>	Zeigt die letzten System-Logs mit erweiterten Erklärungen ( <code>-x</code> ) direkt am Ende des Logs ( <code>-e</code> ) – erster Schritt bei jedem Fehler nach dem Boot.
<code>journalctl -u &lt;Dienst&gt; -f</code>	Zeigt das Log eines einzelnen Dienstes live ( <code>-f</code> = follow, wie <code>tail -f</code> ) – ideal zum Debuggen während des Starts.

## GPIO & Hardware-Schnittstellen

**Tip:** Für die Pin-Belegung selbst musst du dir nichts merken – unser interaktives [GPIO-Pinout-Tool](#) zeigt auf einen Klick Pin-Nummer, Funktion und Board-Position für Pi 5, 4, 3 und Zero. Die Befehle unten sind für den direkten Zugriff im Terminal.

**Hinweis:** Der klassische Befehl `gpio readall` aus der WiringPi-Bibliothek gilt als veraltet und ist in aktuellen Raspberry Pi OS-Repos nicht mehr enthalten. Für neue Projekte `raspi-gpio` verwenden. Mehr Hintergrund im [GPIO-Einsteiger-Tutorial](#).

Befehl	Was er macht
<code>raspi-gpio get</code>	Zeigt für alle Pins Funktion (Modus), Pull-Up/Down-Widerstand und aktuellen Level (0/1) in einer kompakten Übersicht.
<code>raspi-gpio get &lt;Pin&gt;</code>	Zeigt Modus, Pull-Widerstand und Level nur für den angegebenen Pin, z. B. <code>raspi-gpio get 17</code> .
<code>raspi-gpio set &lt;Pin&gt; op dh</code>	Konfiguriert den Pin als Ausgang ( <code>op</code> ) und schaltet ihn direkt auf High/3,3V ( <code>dh</code> = drive high).
<code>raspi-gpio set &lt;Pin&gt; op dl</code>	Konfiguriert den Pin als Ausgang und schaltet ihn auf Low/0V ( <code>dl</code> = drive low).

Befehl	Was er macht
<code>raspi-gpio funcs</code>	Listet für jeden Pin alle verfügbaren alternativen Hardwarefunktionen (z. B. I2C, SPI, UART, PWM) auf.
<code>sudo raspi-config nonint do_i2c 0</code>	Aktiviert die I2C-Schnittstelle inkl. nötiger Kernel-Module, ohne das Menü zu öffnen (0 = an, 1 = aus).
<code>sudo raspi-config nonint do_spi 0</code>	Aktiviert die SPI-Schnittstelle nicht-interaktiv – Voraussetzung für viele Displays und ADC-Module.
<code>i2cdetect -y 1</code>	Scannt den I2C-Bus 1 (Standard bei aktuellen Modellen) und zeigt die Adressen aller erkannten Geräte in einer Tabelle.
<code>sudo raspi-config nonint do_serial_hw 0</code>	Aktiviert die Hardware-UART ( <code>ttyAMA0</code> ) für zuverlässige serielle Kommunikation, z. B. mit GPS-Modulen oder RTC-Boards.

## Kamera-Modul

**Hinweis:** Aktuelle Raspberry Pi OS-Versionen nutzen den `rpicam-*`-Befehlssatz (Nachfolger von `libcamera-*` und dem alten `raspistill`). Alte Befehlsnamen funktionieren teils noch, weil sie im Hintergrund automatisch zu den neuen weitergeleitet werden – in neuen Skripten sollte man sie trotzdem nicht mehr verwenden.

Befehl	Was er macht
<code>rpicam-hello</code>	Öffnet für einige Sekunden eine Live-Vorschau der Kamera auf dem angeschlossenen Display – einfachster Funktionstest.
<code>rpicam-still --list-cameras</code>	Listet alle vom System erkannten Kameramodule mit Sensor-Namen und unterstützten Auflösungen auf.
<code>rpicam-still -o bild.jpg</code>	Nimmt nach kurzem Autofokus-/Belichtungsabgleich ein einzelnes Foto auf und speichert es unter dem angegebenen Namen.
<code>rpicam-vid -t 10000 -o video.h264</code>	Nimmt ein Video mit der in <code>-t</code> angegebenen Dauer in Millisekunden auf (hier: 10 Sekunden).
<code>vcgencmd get_camera</code>	Schneller Software-Check, ob die Kamera erkannt ( <code>detected=1</code> ) und im System aktiviert ( <code>supported=1</code> ) ist, ohne ein Bild aufzunehmen.

## Bootloader, Firmware & Updates

Pi-exklusiv: EEPROM-Bootloader-Verwaltung gibt es nur auf echter Raspberry-Pi-Hardware (Pi 4/5).

Befehl	Was er macht
<code>vcgencmd bootloader_version</code>	Zeigt Versionsdatum und Release-Kanal (critical/stable/beta) des aktuell geflashten EEPROM-Bootloaders.
<code>sudo rpi-eeeprom-update</code>	Vergleicht die installierte mit der neuesten verfügbaren Bootloader-Version, ohne etwas zu verändern (reiner Check).
<code>sudo rpi-eeeprom-update -a</code>	Spielt automatisch die neueste verfügbare EEPROM-Version ein – wird beim nächsten Neustart durch <code>recovery.bin</code> geschrieben.
<code>sudo raspi-config nonint do_expand_rootfs</code>	Vergrößert die Root-Partition auf die volle Kapazität der SD-Karte/SSD – bei den meisten aktuellen Images bereits automatisch beim ersten Boot erledigt.
<code>sudo apt update &amp;&amp; sudo apt full-upgrade -y</code>	Kombinierter Befehl für ein komplettes System-Update inkl. Kernel- und Firmware-Paketen in einem Rutsch.

## Storage, SD-Karte & Backup

**Vorsicht bei `dd`:** `if=` (Input) und `of=` (Output) dürfen nicht vertauscht werden – ein Zahldreher überschreibt die falsche Platte unwiderruflich. Laufwerksnamen vorher immer mit `lsblk` prüfen. Eine ausführliche Anleitung gibt es in unserem [Backup-Tutorial](#).

Befehl	Was er macht
<code>lsblk</code>	Zeigt alle Blockgeräte als Baumstruktur mit Partitionen, Größen und Mountpunkten – der sicherste erste Schritt vor jedem <code>dd</code> -Befehl.
<code>sudo fdisk -l</code>	Zeigt detaillierte Partitionstabellen (Typ, Start-/End-Sektor, Größe) aller erkannten Laufwerke.
<code>sudo mount /dev/sda1 /mnt</code>	Hängt die Partition <code>sda1</code> manuell in das (leere) Verzeichnis <code>/mnt</code> ein, danach über <code>/mnt</code> zugreifbar.
<code>sudo umount /mnt</code>	Hängt ein zuvor gemountetes Laufwerk sauber aus – vor dem physischen Abziehen unbedingt nötig, um Datenverlust zu vermeiden.
<code>sudo dd if=/dev/sdX of=backup.img bs=4M status=progress</code>	Erstellt ein exaktes 1:1-Abbild der gesamten Karte inklusive Partitionstabelle; <code>bs=4M</code> beschleunigt die Übertragung, <code>status=progress</code> zeigt den Fortschritt live.

Befehl	Was er macht
<code>sudo dd if=backup.img of=/dev/sdX bs=4M status=progress</code>	Schreibt ein zuvor erstelltes Image zurück auf eine Karte/SSD – überschreibt dabei restlos alle vorhandenen Daten auf dem Ziel.
<code>rsync -avh &lt;Quelle&gt; &lt;Ziel&gt;</code>	Gleicht zwei Ordner ab und überträgt nur, was neu oder geändert ist – deutlich schneller als jedes Mal alles komplett neu zu kopieren. <code>a</code> übernimmt dabei auch Rechte und Zeitstempel, <code>v</code> zeigt jede Datei einzeln an, <code>h</code> macht die Größenangaben lesbar (MB/GB statt Bytes).

## Benutzer & Rechte

Befehl	Was er macht
<code>sudo adduser &lt;Name&gt;</code>	Legt interaktiv einen neuen Benutzer inkl. Home-Verzeichnis an und fragt dabei Passwort sowie optionale Angaben ab.
<code>sudo deluser &lt;Name&gt;</code>	Entfernt einen Benutzer-Account; das Home-Verzeichnis bleibt standardmäßig erhalten (mit <code>--remove-home</code> komplett löschen).
<code>sudo usermod -aG sudo &lt;Name&gt;</code>	Fügt den Benutzer zur sudo-Gruppe hinzu, ohne ihn aus seinen bisherigen Gruppen zu entfernen (das bewirkt <code>-a</code> ) – ohne dieses Flag würden alle anderen Gruppen-Mitgliedschaften verloren gehen.
<code>groups &lt;Name&gt;</code>	Zeigt alle Gruppen, in denen ein Benutzer Mitglied ist – z. B. um zu prüfen, ob er zu <code>gpio</code> oder <code>i2c</code> gehört.
<code>who</code>	Listet alle aktuell angemeldeten Benutzer mit Terminal und Login-Zeit auf, auch bei mehreren gleichzeitigen SSH-Sitzungen.
<code>sudo -i</code>	Öffnet eine Sitzung, in der du dauerhaft als Administrator (root) arbeitest, statt jedem einzelnen Befehl <code>sudo</code> voranzustellen – bis du sie mit <code>exit</code> wieder verlässt. Mit Vorsicht nutzen, hier gibt es keine Sicherheitsbremse mehr.

## Tastenkombinationen & Terminal-Tricks

Kombination	Was sie macht
<code>Strg + C</code>	Bricht den gerade laufenden Befehl sofort ab – der Not-Aus-Knopf im Terminal, wenn etwas hängt oder man es sich anders überlegt.
<code>Strg + Z</code>	Pausiert den Prozess und legt ihn in den Hintergrund – mit <code>fg</code> wieder in den Vordergrund holen, mit <code>bg</code> im Hintergrund weiterlaufen lassen.
<code>Strg + R</code>	Startet eine interaktive Rückwärtssuche durch die Befehlshistorie – Tippen filtert live, Enter führt aus.

Kombination	Was sie macht
<code>Tab</code>	Vervollständigt Befehle, Dateinamen und Pfade automatisch – zweimal drücken zeigt bei Mehrdeutigkeit alle Optionen.
<code>history</code>	Zeigt eine nummerierte Liste aller zuletzt ausgeführten Befehle – mit <code>!123</code> lässt sich Befehl Nummer 123 direkt erneut ausführen.
<code>!!</code>	Führt den zuletzt ausgeführten Befehl exakt noch einmal aus, ohne ihn erneut eintippen zu müssen.
<code>sudo !!</code>	Wiederholt den letzten Befehl mit vorangestelltem <code>sudo</code> – praktisch, wenn ein Befehl gerade wegen fehlender Rechte fehlgeschlagen ist.
<code>clear</code>	Leert die sichtbare Terminal-Ausgabe, der Scroll-Verlauf bleibt aber über die Bildlaufleiste erhalten.
<code>exit</code>	Beendet die aktuelle Shell-Sitzung – bei einer SSH-Verbindung wird dadurch auch die Verbindung zum Pi getrennt.

Wer regelmäßig neue Skripte auf dem Pi laufen lässt, sollte auch einen Blick auf unseren [Autostart-Guide](#) werfen – damit Programme nach jedem Neustart automatisch starten, ohne dass ein Befehl von Hand nötig ist.

### Wie finde ich die IP-Adresse meines Raspberry Pi per Terminal heraus?

Am schnellsten mit dem Befehl `hostname -I` direkt am Pi – er zeigt alle zugewiesenen IP-Adressen auf einen Blick. Alternativ liefert `ip a` eine detailliertere Übersicht aller Netzwerkschnittstellen.

### Warum funktioniert der Befehl `gpio readall` bei mir nicht mehr?

Das `gpio`-Tool aus der WiringPi-Bibliothek wurde aus den offiziellen Raspberry Pi OS-Repositories entfernt, da WiringPi als Projekt lange nicht mehr aktiv gepflegt wurde. Als Ersatz dient `raspi-gpio`: `raspi-gpio get` zeigt den Zustand aller Pins, `raspi-gpio funcs` die verfügbaren alternativen Funktionen.

### Wie prüfe ich, ob mein Raspberry Pi durch Unterspannung gedrosselt wird?

Mit `vcgencmd get_throttled` prüfen. `0x0` bedeutet: alles unauffällig. Jeder andere Wert zeigt, dass der Pi aktuell oder seit dem letzten Start schon einmal wegen zu wenig Strom gedrosselt wurde – meist ein Hinweis auf ein zu schwaches Netzteil oder ein schlechtes Kabel.

### Was ist der Unterschied zwischen `apt upgrade` und `apt full-upgrade`?

`apt upgrade` aktualisiert nur bereits installierte Pakete und installiert dabei keine neuen oder entfernt bestehende. `apt full-upgrade` darf zusätzlich Pakete entfernen oder neu installieren, wenn das für die Auflösung von Abhängigkeiten bei größeren Updates nötig ist.

### Wie erstelle ich ein komplettes Backup meiner SD-Karte per Kommandozeile?

Mit `sudo dd if=/dev/sdX of=backup.img bs=4M status=progress` lässt sich ein 1:1-Image der kompletten Karte erstellen. Wichtig: Vorher mit `lsblk` das richtige Laufwerk identifizieren, da `if=` und `of=` bei Verwechslung das falsche Ziel überschreiben.

## Mehr auf [raspberrypi.com](https://raspberrypi.com)

Tutorials, Vergleiche und Tools rund um den Raspberry Pi – von GPIO über Smart Home bis Media Center.

[raspberrypi.com](https://raspberrypi.com)