Raspberry Pi Cheat Sheet

A quick reference guide to Raspberry Pi, covering setup, common commands, GPIO pins, and troubleshooting tips. This cheat sheet is designed to help both beginners and experienced users get the most out of their Raspberry Pi.

Getting Started

Initial Setup

1. Download Raspberry Pi Imager:
Download the official Raspberry Pi Imager from the Raspberry Pi website for your operating system (Windows, macOS, Ubuntu).

Raspberry Pi Imager Download

2. Install the Imager:
Run the downloaded installer and follow the on-screen instructions.

3. Prepare SD Card:
Insert an SD card (at least 8GB recommended) into your computer.

4. Choose OS and SD Card:
Open Raspberry Pi Imager, select the desired OS (e.g., Raspberry Pi OS), and choose the connected SD card.

5. Write to SD Card:
Click 'Write' to flash the OS onto the SD card. This process will erase all existing data on the card.

6. Boot Raspberry Pi:
Insert the SD card into your Raspberry Pi, connect peripherals (keyboard, mouse, display), and power it on.

7. Initial Configuration:
Follow the on-screen prompts to set up your Raspberry Pi (e.g., set the country, language, keyboard layout, and Wi-Fi).

8. Update the System:
Open a terminal and run the following commands to update the package lists and upgrade installed packages:

```
sudo apt update
sudo apt upgrade
```

Basic Commands

`sudo reboot`	Reboots the Raspberry Pi.
`sudo shutdown now`	Shuts down the Raspberry Pi immediately.
`sudo apt update`	Updates the package lists.
`sudo apt upgrade`	Upgrades installed packages.
`sudo apt install `	Installs a new package.
`sudo apt remove `	Removes a package.
`df -h`	Shows disk space usage.
`free -m`	Shows memory usage.

File Management

`ls`	List files and directories in the current directory.
`cd `	Change the current directory.
`mkdir `	Create a new directory.
`rm `	Remove a file.
`rmdir `	Remove an empty directory.
`cp `	Copy a file.
`mv `	Move or rename a file.
`nano `	Open a file in the Nano text editor.

GPIO Programming

GPIO Pinout

Refer to the official Raspberry Pi documentation for the specific GPIO pinout of your model.

Raspberry Pi Pinout

Key considerations:

- **Power Pins:** 3.3V, 5V, and Ground (GND).
- **GPIO Pins:** General Purpose Input/Output pins.
- **I2C Pins:** SDA (Data) and SCL (Clock).
- **SPI Pins:** MOSI, MISO, SCLK, CE0, CE1.
- **UART Pins:** TXD and RXD.

RPI.GPIO Library (Python)

`import RPi.GPIO as GPIO`	Import the RPi.GPIO library.
`GPIO.setmode(GPIO.BCM)`	Set the GPIO numbering mode to BCM (Broadcom SOC channel) or `GPIO.BOARD` (physical pin numbering).
`GPIO.setup(pin, GPIO.IN)`	Set a GPIO pin as an input.
`GPIO.setup(pin, GPIO.OUT)`	Set a GPIO pin as an output.
`GPIO.output(pin, GPIO.HIGH)`	Set a GPIO pin HIGH (3.3V).
`GPIO.output(pin, GPIO.LOW)`	Set a GPIO pin LOW (0V).
`GPIO.input(pin)`	Read the state of a GPIO pin (returns `GPIO.HIGH` or `GPIO.LOW`).
`GPIO.cleanup()`	Clean up GPIO resources when the script ends.

Example: Blinking LED

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.OUT)

try:
    while True:
        GPIO.output(18, GPIO.HIGH)
        time.sleep(0.5)
        GPIO.output(18, GPIO.LOW)
        time.sleep(0.5)
except KeyboardInterrupt:
    GPIO.cleanup()
```

Networking

Network Configuration

`ifconfig`	Display network interface configurations.
`iwconfig`	Display wireless network configurations.
`ping `	Test network connectivity by sending ICMP echo requests.
`hostname -I`	Display the IP address of the Raspberry Pi.
`sudo nano /etc/network/interfaces`	Edit network interface configurations (use with caution).
`sudo nano /etc/wpa_supplicant/wpa_supplicant.conf`	Edit Wi-Fi configuration (SSID and password).

SSH Access

Enable SSH:

By default, SSH is disabled. To enable it, use `sudo raspi-config` and navigate to 'Interface Options' -> 'SSH' and enable it. Alternatively, create an empty file named `ssh` in the boot partition of the SD card.

Connect via SSH:

Use an SSH client (e.g., PuTTY on Windows, Terminal on macOS/Linux) to connect to the Raspberry Pi using its IP address.

```
ssh pi@<raspberry_pi_ip_address>
```

The default username is `pi` and the default password is `raspberry` (change this immediately after initial setup).

VNC Access

Install VNC Server:

Install a VNC server on the Raspberry Pi.

```
sudo apt update
sudo apt install tightvncserver
```

Start VNC Server:

Start the VNC server and set a password.

```
vncserver :1
```

Connect via VNC Client:

Use a VNC client on your computer to connect to the Raspberry Pi using its IP address and the display number (e.g., `<raspberry_pi_ip_address>:1`).

Troubleshooting

Common Issues

1. No Boot:

- Ensure the SD card is properly inserted and flashed with a valid OS image.
- Check the power supply (use a 5V 2.5A power adapter).
- Check the activity LED (green) for blinking patterns indicating boot progress.

2. Network Connectivity Issues:

- Verify Wi-Fi credentials in `/etc/wpa_supplicant/wpa_supplicant.conf`.
- Check if the Raspberry Pi is obtaining an IP address using `ifconfig`.
- Ensure the network is functioning correctly.

3. Permission Errors:

- Use `sudo` to run commands that require elevated privileges.
- Check file permissions using `ls -l` and modify them using `chmod` if necessary.

4. Package Installation Errors:

- Run `sudo apt update` to update the package lists before installing new packages.
- Ensure there is enough disk space available.

5. GPIO Issues:

- Double-check the wiring and pin connections.
- Ensure the correct GPIO numbering mode is used (`GPIO.BCM` or `GPIO.BOARD`).
- Verify that GPIO pins are properly configured as inputs or outputs.

Log Files

`/var/log/syslog`	System log file containing general system messages.
`/var/log/auth.log`	Authentication log file recording login attempts and authorization events.
`/var/log/daemon.log`	Daemon log file containing messages from various system services.
`/var/log/kern.log`	Kernel log file recording kernel-related messages.

Useful Commands for Troubleshooting

`dmesg`	Display kernel messages, useful for identifying hardware-related issues.
`vcgencmd get_throttled`	Check for CPU throttling due to overheating or insufficient power.
`top`	Display real-time system resource usage (CPU, memory).
`journalctl`	Query the systemd journal for logs.