Getting Started
Setting up your Raspberry Pi

url: https://www.raspberrypi.com/documentation/computers/getting-started.html

Edit this on GitHub
To get started with your Raspberry Pi computer you'll need the following accessories:
A computer monitor, or television. Most should work as a display for the Raspberry Pi, but for best results, you should use a display with HDMI input. You'll also need an appropriate display cable, to connect your monitor to your Raspberry Pi.

NOTE If your display uses an HDMI connection and has built-in speakers, you can use it to output sound.

A computer keyboard and mouse

- Any standard USB keyboard and mouse will work with your Raspberry Pi.
- Wireless keyboards and mice will work if already paired.
- For keyboard layout configuration options see raspi-config.

A good quality power supply.
We recommend the official Raspberry Pi Power Supply which has been specifically designed to consistently provide +5.1V despite rapid fluctuations in current draw. Those fluctuations in demand is something that happens a lot with when you're using peripherals with the Raspberry Pi, and something that other supplies—designed to provide consistent current for charging cellphones—usually don't cope with all that well. It also has an attached micro USB cable, which means that you don't accidentally use a poor quality cable—something that can be an issue.
For the Raspberry Pi 4 Model B and Raspberry Pi 400 you should use the type C power supply; for all other models you should use the micro USB power supply.

Finally you'll need an SD card; we recommend a minimum of 8GB micro SD card, and to use the Raspberry Pi Imager to install an operating system onto it.

Optional items
A network (Ethernet) cable to connect your Raspberry Pi to your local network and the Internet.
If you aren't using an HDMI monitor with speakers you might also need some form of sound hardware. Audio can be played through speakers or headphones by connecting them to the AV jack (not available on the Raspberry Pi 400). However speakers must have their own amplification since the output from your Raspberry Pi is not powerful enough to drive them directly.

Troubleshooting
You can get help with setting up your Raspberry Pi on our forums.

Installing the Operating System
Edit this on GitHub

Raspberry Pi recommend the use of Raspberry Pi Imager to install an operating system on your SD card. You will need another computer with an SD card reader to install the image.

NOTE Before you start, don't forget to check the SD card requirements.

IMPORTANT NOOBS, or New Out Of the Box Software to give it its full name, was an SD card-based installer for Raspberry Pi computers; we no longer recommend or support using NOOBS. Going forward, please use Raspberry Pi Imager.

Using Raspberry Pi Imager

Raspberry Pi have developed a graphical SD card writing tool that works on Mac OS, Ubuntu 18.04, and Windows called Raspberry Pi Imager; this is the easiest option for most users since it will download the image automatically and install it to the SD card.

Download the latest version of Raspberry Pi Imager and install it. If you want to use Raspberry Pi Imager from a second Raspberry Pi, you can install it from a terminal using sudo apt install rpi-imager. Then:

- Connect an SD card reader with the SD card inside.
- Open Raspberry Pi Imager and choose the required OS from the list presented.
- Choose the SD card you wish to write your image to.
- Review your selections and click on the Write button to begin writing data to the SD Card.

NOTE If using Raspberry Pi Imager on Windows 10 with controlled folder access enabled, you will need to explicitly allow Raspberry Pi Imager permission to write the SD card. If this is not done, the imaging process will fail with a "failed to write" error.

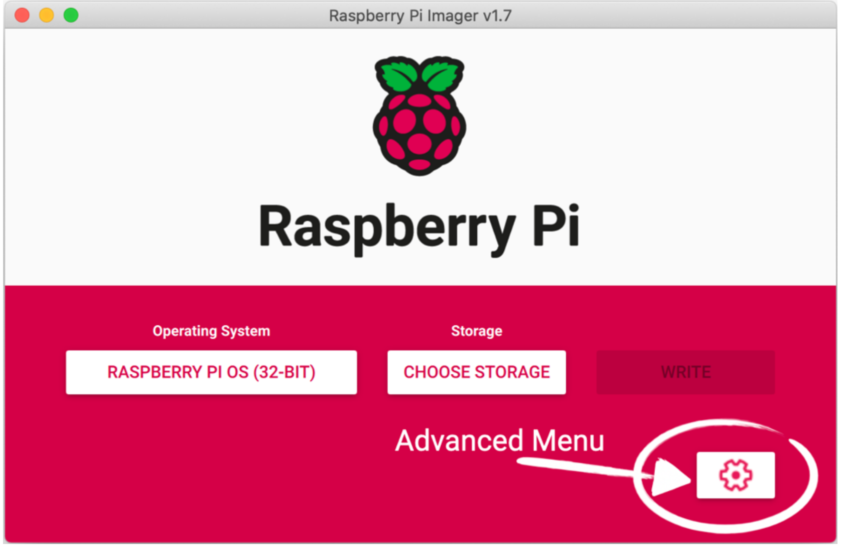
NOTE You can see which operating systems are most often downloaded, on our statistics page.

You can now insert the SD card into the Raspberry Pi and power it up. For Raspberry Pi OS, if you need to manually log in, the default user name is pi, with password raspberry, and the default keyboard layout is set to United Kingdom (UK).

You should change the default password straight away to ensure your Raspberry Pi is secure.

Advanced Options

When you have the Raspberry Pi Imager open, and after you have selected the operating system to install, a cog wheel will appear allowing you to open an "Advanced Options" menu if it is supported by the operating system. This menu lets you carry out tasks like enabling SSH, or setting your Raspberry Pi's hostname, before first boot.



Advanced Menu

Amongst other things the Advanced Options menu is useful for when you want to configure a headless Raspberry Pi.

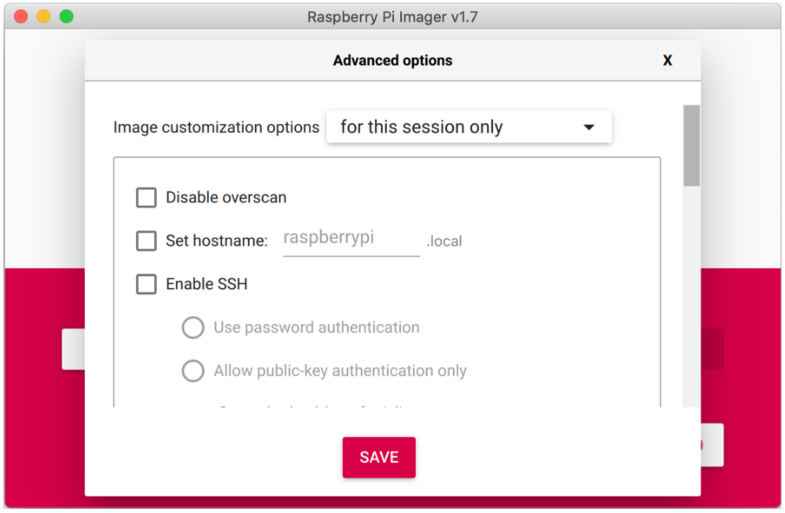


Image customization options: for this session only

□ Disable overscan

□ Set hostname: raspberrypi .local

□ Enable SSH

○ Use password authentication

○ Allow public-key authentication only

NOTE In older versions of Imager you should push Ctrl-Shift-X to open the "Advanced" menu.

Downloading an Image

If you are using a different tool than Raspberry Pi Imager to write to your SD Card, most require you to download the image first, then use the tool to write it to the card. Official images for recommended operating systems are available to download from the Raspberry Pi website downloads page. Alternative operating systems for Raspberry Pi computers are also available from some third-party vendors.

You may need to unzip the downloaded file (.zip) to get the image file (.img) you need to write to the card.

NOTE The Raspberry Pi OS with desktop image contained in the ZIP archive is over 4GB in size and uses the ZIP64 format. To uncompress the archive, an unzip tool that supports ZIP64 is required. The following zip tools support ZIP64: 7-Zip for Windows, The Unarchiver for macOS, and unzip on Linux.

Installing over the Network

Edit this on GitHub

WARNING Network installation is only enabled by default on Raspberry Pi 4 and Raspberry Pi 400, and only when using the latest supported bootloader. See Updating to the LATEST / STABLE bootloader for details of how to update your bootloader.

When you get a new Raspberry Pi, you also need an SD Card with an operating system installed on it. The easiest way to get the operating system onto the SD Card is to use the Raspberry Pi Imager application running on another computer to copy it to your SD Card.

But how do you get the operating system onto your SD card if you don't have another computer?

If you do not have another computer to run Raspberry Pi Imager you can start the application directly on a Raspberry Pi 4 or Raspberry Pi 400 by connecting it to the Internet with an Ethernet cable. This will allow you to install the operating system onto a **blank** SD Card directly from the network without using another computer.

Using Network Installation

You will need a keyboard to make use of the network installation feature. While a Raspberry Pi 400 always has a keyboard "attached", if you're using a Raspberry Pi 4 you will need to plug in a USB keyboard.

Either insert a **blank** SD Card into the SD card slot of your Raspberry Pi, or press and hold the SHIFT key.

Power on your Raspberry Pi. As always it will first look for an SD Card, and then a USB drive, to find bootable media. However if you have a keyboard attached, the Raspberry Pi will now show the network installation screen.

In the background the Raspberry Pi is still looking for a bootable image, but you can now start a network installation by holding down the SHIFT key. You will then be prompted to connect your Raspberry Pi to the network with an Ethernet Cable.

NOTE You can also start network install by holding SHIFT when powering on the device.

Plug your Raspberry Pi into the network using an Ethernet cable. When it detects a cable has been inserted it should start downloading the Raspberry Pi installer. If the download fails, you can repeat the process to try again.

Eventually it should start the Raspberry Pi Imager application allowing you to install a full operating system to a blank SD Card or a USB Drive.

NOTE More information can about using the Raspberry Pi Imager can be found in the section on installing your operating system.

After installing the operating system onto your blank SD Card you will no longer see the network installation screen on boot. If you do want to run it, you can hold down the SHIFT key while powering on the device. But take care not to overwrite any working SD cards that you want to keep!
See HTTP boot for details about how to configure network install.

Installing Images on Chrome OS

Edit this on GitHub

The easiest way to write images to an SD card and USB drives with Chrome OS is to use the official Chromebook Recovery Utility. It can be used to create Chromebook Recovery media, and it will also accept .zip files containing images.

- Download the Chromebook Recovery Utility.
- Download the Raspberry Pi OS zip archive.
- Launch the **Recovery Utility**
- Click on the **Settings Gears** icon in the upper right-hand corner, next to the window close icon.
- Select the **Use Local Image** option.
- Choose the .zip file you downloaded.
- Insert the SD card and click **Continue**.
- Read the warning and click the **Create now** button.
- Wait for the progress bar to complete twice (for unpacking and writing). This might take a few minutes. Once the process is complete, a big green checkmark will be shown.
- Close the program and eject the card.

Installing Images on Linux

Edit this on GitHub

Raspberry Pi Imager is typically the easiest option for most users to write images to SD cards, so it is a good place to start. If you're looking for more advanced options on Linux, you can use the standard command line tools below.

NOTE Using the dd tool can overwrite any partition of your machine. If you specify the wrong device when using dd, you could delete your primary Linux partition. Please be careful.

Discovering the SD card mountpoint and unmounting it

- Run lsblk -p to see which devices are currently connected to your machine.
- If your computer has a slot for SD cards, insert the card. If not, insert the card into an SD card reader, then connect the reader to your computer.
- Run lsblk -p again. The new device that has appeared is your SD card (you can also usually tell from the listed device size). The naming of the device will follow the format described in the next paragraph.

- The left column of the results from the lsblk -p command gives the device name of your SD card and the names of any partitions on it (usually only one, but there may be several if the card was previously used). It will be listed as something like /dev/mmcblk0 or /dev/sdX (with partition names /dev/mmcblk0p1 or /dev/sdX1 respectively), where X is a lower-case letter indicating the device (eg. /dev/sdb1). The right column shows where the partitions have been mounted (if they haven't been, it will be blank).
- If any partitions on the SD card have been mounted, unmount them all with umount, for example umount /dev/sdX1 (replace sdX1 with your SD card's device name, and change the number for any other partitions).

Copying the image to the SD card

- In a terminal window, write the image to the card with the command below, making sure you replace the input file if= argument with the path to your .img file, and the /dev/sdX in the output file of= argument with the correct device name. **This is very important, as you will lose all the data on the hard drive if you provide the wrong device name.** Make sure the device name is the name of the whole SD card as described above, not just a partition. For example: sdd, not sdds1 or sddp1; mmcblk0, not mmcblk0p1.

```
sudo dd if=2021-10-30-raspios-bullseye-armhf.img of=/dev/sdX bs=4M conv=fsync
```

- Please note that block size set to 4M will work most of the time. If not, try 1M, although this will take considerably longer.

Copying a zipped image to the SD card

In Linux it is possible to combine the unzip and SD copying process into one command, which avoids any issues that might occur when the unzipped image is larger than 4GB. This can happen on certain filesystems that do not support files larger than 4GB (e.g. FAT), although it should be noted that most Linux installations do not use FAT and therefore do not have this limitation.

The following command unzips the zip file (replace 2021-10-30-raspios-bullseye-armhf.zip with the appropriate zip filename), and pipes the output directly to the dd command. This in turn copies it to the SD card, as described in the previous section.

```
unzip -p 2021-10-30-raspios-bullseye-armhf.zip | sudo dd of=/dev/sdX bs=4M conv=fsync
```

Checking the image copy progress

- By default, the dd command does not give any information about its progress, so it may appear to have frozen. It can take more some time to finish writing to the card. If your card reader has an LED, it may blink during the write process.
- To see the progress of the copy operation, you can run the dd command with the status option.

```
 sudo dd if=2021-10-30-raspios-bullseye-armhf.img of=/dev/sdX bs=4M conv=fsync status=progress
```

- If you are using an older version of dd, the status option may not be available. You may be able to use the dcfldd command instead, which will give a progress report showing how much has been written. Another method is to send a USR1 signal to dd, which will let it print status information. Find out the PID of dd by using pgrep -l dd or ps a | grep dd. Then use kill -USR1 PID to send the USR1 signal to dd.

Optional: checking whether the image was correctly written to the SD card

- After dd has finished copying, you can check what has been written to the SD card by dd-ing from the card back to another image on your hard disk, truncating the new image to the same size as the original, and then running diff (or md5sum) on those two images.
- If the SD card is much larger than the image, you don't want to read back the whole SD card, since it will be mostly empty. So you need to check the number of blocks that were written to the card by the dd command. At the end of its run, dd will have displayed the number of blocks written as follow:
- xxx+0 records in
- yyy+0 records out

yyyyyyyyyy bytes (yyy kB, yyy KiB) copied, 0.00144744 s, 283 MB/s

We need the number xxx, which is the block count. We can ignore the yyy numbers.

- Copy the SD card content to an image on your hard drive using dd again:

```
sudo  dd if=/dev/sdX of=from-sd-card.img bs=4M count=xxx
```

if is the input file (i.e. the SD card device), of is the output file to which the SD card content is to be copied (called from-sd-card.img in this example), and xxx is the number of blocks written by the original dd operation.

- In case the SD card image is still larger than the original image, truncate the new image to the size of the original image using the following command (replace the input file reference argument with the original image name):

```
truncate --reference 2021-10-30-raspios-bullseye-armhf.img from-sd-card.img
```

- Compare the two images: diff should report that the files are identical.

```
diff -s from-sd-card.img 2021-10-30-raspios-bullseye-armhf.img
```

- Run sync. This will ensure that the write cache is flushed and that it is safe to unmount your SD card.
- Remove the SD card from the card reader.

Installing Images on Mac OS

Edit this on GitHub

Raspberry Pi Imager is the recommended option for most users to write images to SD cards. However, if you do not want to use the Imager you can still copy an operating system to the card from the command line.

Finding the SD Card

Insert the SD card in the slot, or connect the SD card reader with the SD card inside, and type diskutil list at the command line. You should see something like this,

```
 /dev/disk0 (internal):
   #:                TYPE NAME                 SIZE      IDENTIFIER
   0:                GUID_partition_scheme       500.3 GB   disk0
   1:                EFI EFI                314.6 MB   disk0s1
   2:                Apple_APFS Container disk1   500.0 GB   disk0s2

 /dev/disk1 (synthesized):
   #:                TYPE NAME                 SIZE      IDENTIFIER
   0:          APFS Container Scheme -     +500.0 GB   disk1
              Physical Store disk0s2
   1:          APFS Volume Macintosh HD    89.6 GB    disk1s1
   2:          APFS Volume Preboot        47.3 MB    disk1s2
   3:          APFS Volume Recovery       510.4 MB   disk1s3
   4:          APFS Volume VM            3.6 GB     disk1s4

 /dev/disk2 (external, physical):
   #:          TYPE NAME               SIZE      IDENTIFIER
   0:          FDisk_partition_scheme      *15.9 GB    disk2
   1:          Windows_FAT_32 boot        268.4 MB   disk2s1
   2:          Linux                 15.7 GB    disk2s2
```

Here the SD Card is /dev/disk2 however your disk and partition list may vary.

Copying the Image

WARNING Using the dd command line tool can overwrite your Mac's operating system if you specify the wrong disk device. If you're not sure about what to do, we recommend you use the Raspberry Pi Imager tool.

Before copying the image you should unmount the SD Card.

```
diskutil unmountDisk /dev/diskN
```

You can then copy the image,

```
sudo dd bs=1m if=path_of_your_image.img of=/dev/rdiskN; sync
```

replacing N with the disk number.

NOTE You should use rdisk (which stands for 'raw disk') instead of disk, this speeds up the copying.

This can take a few minutes, depending on the image file size. You can check the progress by pressing Ctrl+T. After the dd command finishes, you can eject the card:

```
sudo diskutil eject /dev/rdiskN
```

Troubleshooting

- If the command reports dd: /dev/rdiskN: Resource busy, you need to unmount the volume first sudo diskutil unmountDisk /dev/diskN.
- If the command reports dd: bs: illegal numeric value, change the block size bs=1m to bs=1M.
- If the command reports dd: /dev/rdiskN: Operation not permitted, go to System Preferences -> Security & Privacy -> Privacy -> Files and Folders -> Give Removable Volumes access to Terminal.
- If the command reports dd: /dev/rdiskN: Permission denied, the partition table of the SD card is being protected against being overwritten by mac OS.

Erase the SD card's partition table using this command:

```
sudo diskutil partitionDisk /dev/diskN 1 MBR "Free Space" "%noformat%" 100%
```

That command will also set the permissions on the device to allow writing.

Installing Images on Windows

Edit this on GitHub

Raspberry Pi Imager is our recommended option for most users to write images to SD cards, so it is a good place to start. If you're looking for an alternative on Windows, you can use balenaEtcher, Win32DiskImager or imgFlasher.

balenaEtcher

- Download the Windows installer from balena.io
- Run balenaEtcher and select the unzipped Raspberry Pi OS image file
- Select the SD card drive
- Finally, click **Burn** to write the Raspberry Pi OS image to the SD card
- You'll see a progress bar. Once complete, the utility will automatically unmount the SD card so it's safe to remove it from your computer.

Win32DiskImager

- Insert the SD card into your SD card reader. You can use the SD card slot if you have one, or an SD adapter in a USB port. Note the drive letter assigned to the SD card. You can see the drive letter in the left hand column of Windows Explorer, for example **G:**
- Download the Win32DiskImager utility from the Sourceforge Project page as an installer file, and run it to install the software.
- Run the Win32DiskImager utility from your desktop or menu.
- Select the image file you extracted earlier.
- In the device box, select the drive letter of the SD card. Be careful to select the correct drive: if you choose the wrong drive you could destroy the data on your computer's hard disk! If you are using an SD card slot in your computer, and can't see the drive in the Win32DiskImager window, try using an external SD adapter.
- Click 'Write' and wait for the write to complete.
- Exit the imager and eject the SD card.

Upswift imgFlasher

- Download portable Windows version from upswift.io
- Run imgFlasher and choose an image or zip file
- Choose SD card or USB drive
- Click on 'Flash'
- Wait until the flash is completed.

SD Card for Raspberry Pi

Edit this on GitHub

Raspberry Pi computers use a micro SD card, except for very early models which use a full-sized SD card.

WARNING Because of a hardware limitation in the Raspberry Pi Zero, 1 and 2, the boot partition on the SD card must be 256GB or less otherwise the device will not boot up. Later models of Raspberry Pi 2 — with a BCM2837 SoC — along with the Raspberry Pi 3, 4, Zero 2 W, and the Raspberry Pi 400 do not have this limitation. This does not affect Raspberry Pi OS, which always uses a small boot partition.

Recommended Capacity

We recommend using an SD card of 8GB or greater capacity with Raspberry Pi OS. If you are using the lite version of Raspberry Pi OS, you can use a 4GB card. Other operating systems have different requirements: for example, LibreELEC can run from a smaller card. Please check with the supplier of the operating system to find out what capacity of card they recommend.

Troubleshooting

If you are having problems with your SD card:

- Make sure you are using a genuine SD card. The best way to avoid fake SD cards is to always buy from a reputable supplier.
- Make sure you are using a good quality power supply: we recommend using an official Raspberry Pi power supply.
- The cable from the power supply unit to the Raspberry Pi can also cause problems. This is usually due to the resistance of the wires in the USB power cable; to save money, USB cables have as little copper in them as possible, which causes a voltage drop across the cable.
- Make sure you shut down the operating system correctly before you power down the Raspberry Pi.

Connecting a Display

Edit this on GitHub

Unless you're setting up your Raspberry Pi to operate headless, for regular use you'll want to plug the Raspberry Pi in to a display: either a computer monitor, or a televison.

Connecting using HDMI

The Raspberry Pi has an HDMI port which you can connect directly to a monitor or TV with an HDMI cable. This is the easiest solution; some modern monitors and TVs have HDMI ports, some do not, but there are other options.

NOTE The Raspberry Pi 4 has two micro HDMI connectors, which require a good-quality micro HDMI cable, especially when using 4K monitors or television. Raspberry Pi sells a suitable cable.

If you're using your Raspberry Pi with a monitor with built-in speakers and are connecting to it using an HDMI cable you can also use it to output sound.

Connecting using DVI

For monitors with a DVI port, you can use an HDMI-to-DVI cable, or an HDMI cable with a DVI adapter.

NOTE Unlike HDMI the DVI standard does not support audio

Connecting using VGA

For monitors with VGA only, you can use an HDMI-to-VGA adapter. Note that VGA does not support audio.

Using the Composite Port

Most models of Raspberry Pi have a composite out port for connecting to analog devices, but the type of connector varies depending on the model. The original Raspberry Pi used an RCA connector, and a standard RCA composite video lead will work. Others models (Raspberry Pi B+ and later) combine the audio out and composite out on to the same 3.5mm jackplug. This requires a particular type of lead, with audio left on the tip, audio right on ring 1, ground on ring 2, and video on the sleeve. This is the same as leads used on the Zune, and on Apple devices.

WiringPi: sudo apt-get install wiringpi