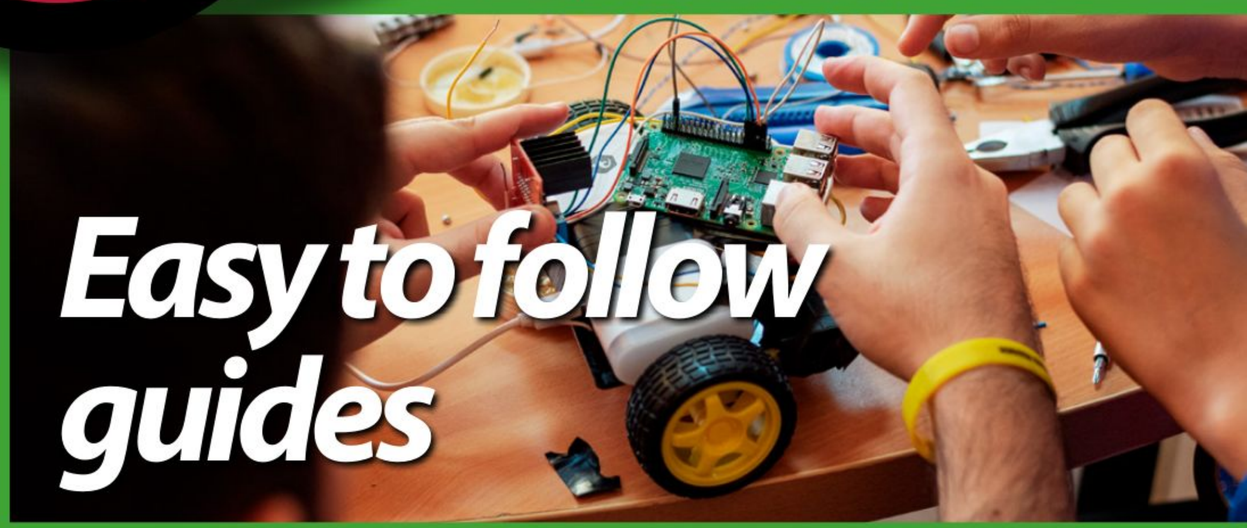
NEW EASY TO FOLLOW USER GUIDES

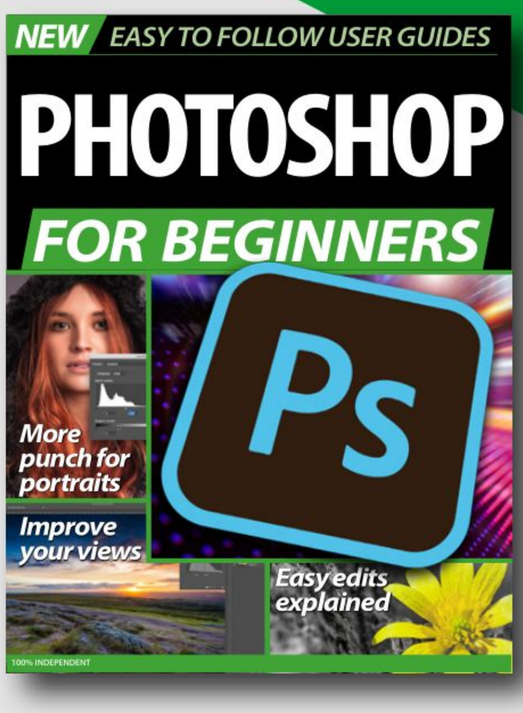
RASPBERRY PI

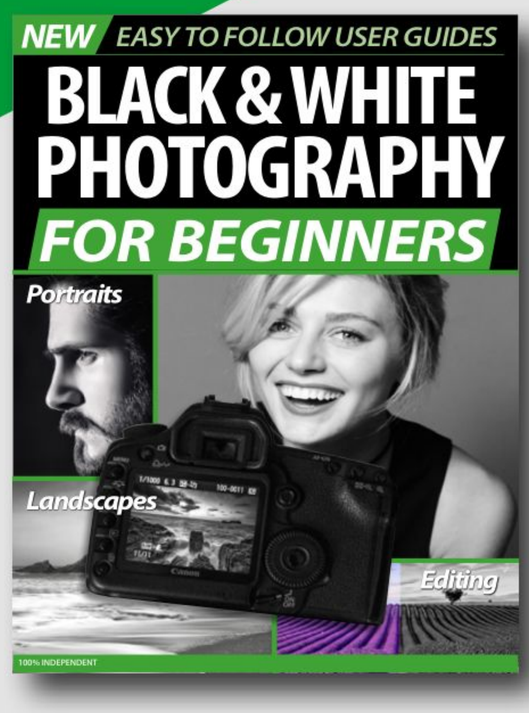
FOR BEGINNERS

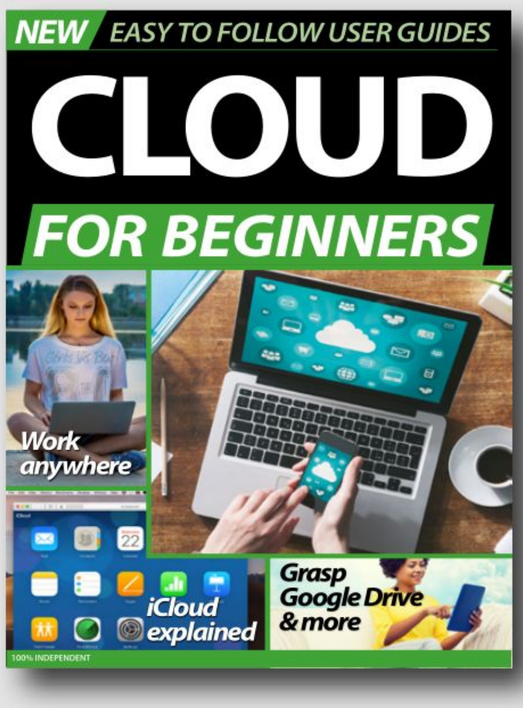


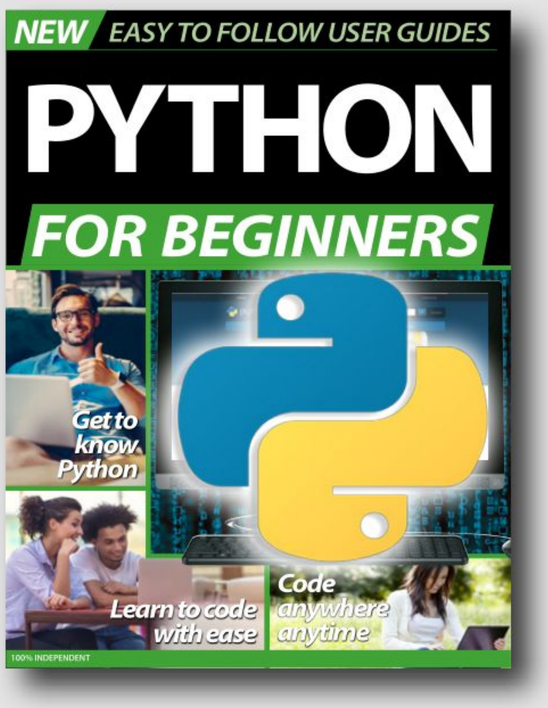
100% INDEPENDENT

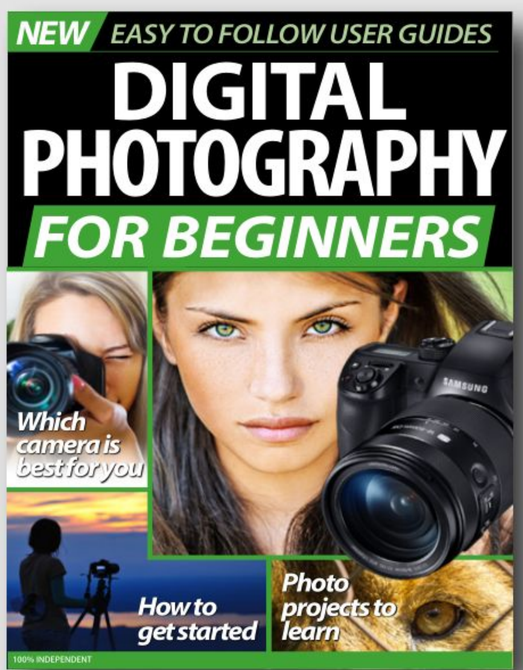
Discover more of our guides...











RASPBERRY PI

FOR BEGINNERS

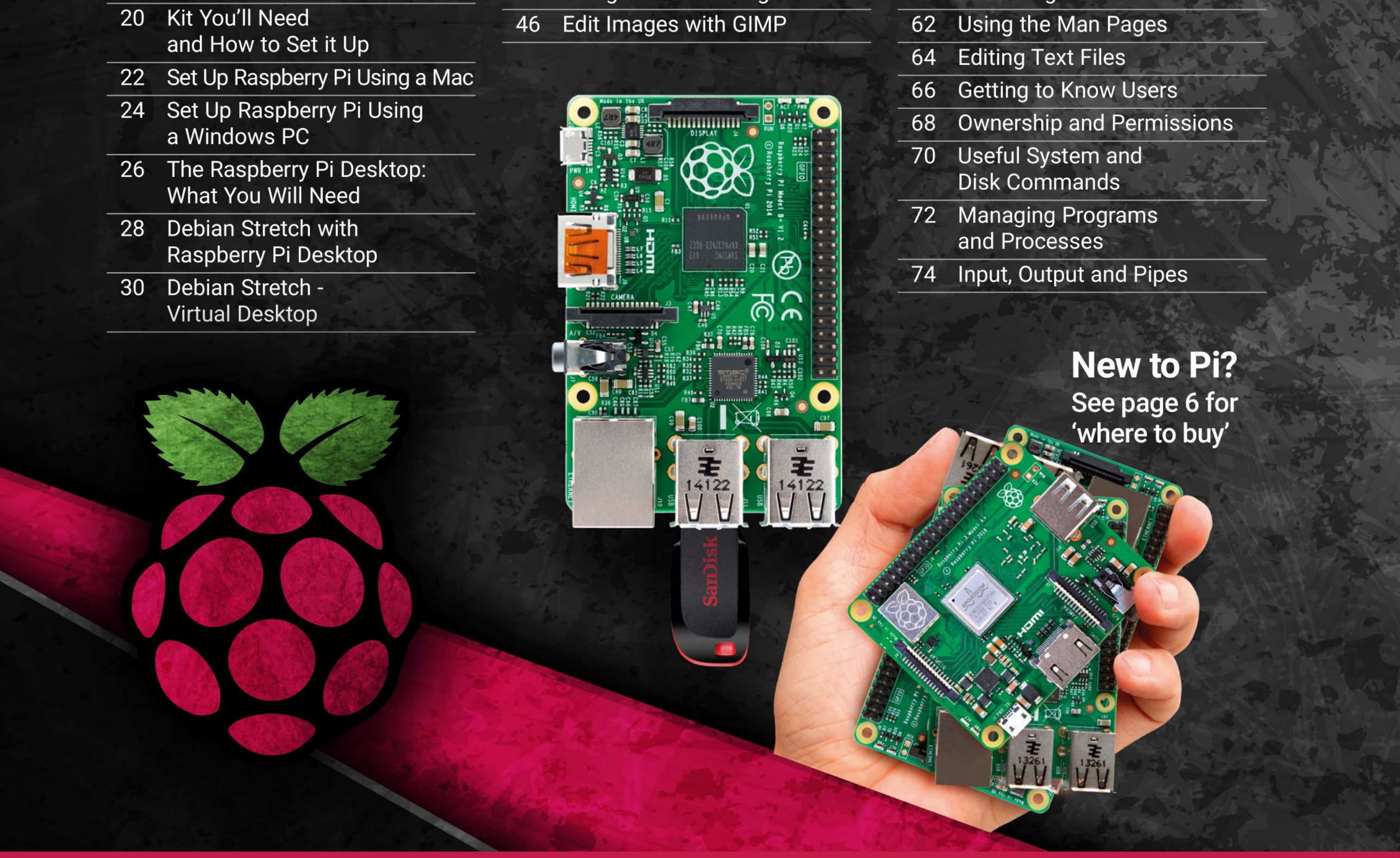
Starting something new can be daunting. Learning a skill or mastering a new piece of hardware is tough. Even tougher if you have no-one at hand to help. Conversely as the complexity of our consumer technology increases, the size of the requisite instruction manual decreases or in some cases it simply disappears. At numerous times in our lives we have all been "beginners", there is no shame in that fact and rightly so. How many times have you asked aloud, "What does this button do?". "Why doesn't that work?". "What do you mean it doesn't do that?". "HELP!". At the start of any new journey or adventure we are all beginners but fortunately for you we are here to stand beside you at every stage.

Over this extensive series of titles we will be looking in great depth at the latest consumer electronics, software, hobbies and trends out of the box! We will guide you step-by-step through using all aspects of the technology that you may have been previously apprehensive at attempting. Let our expert guide help you build your technology understanding and skills, taking you from a novice to a confident and experienced user.

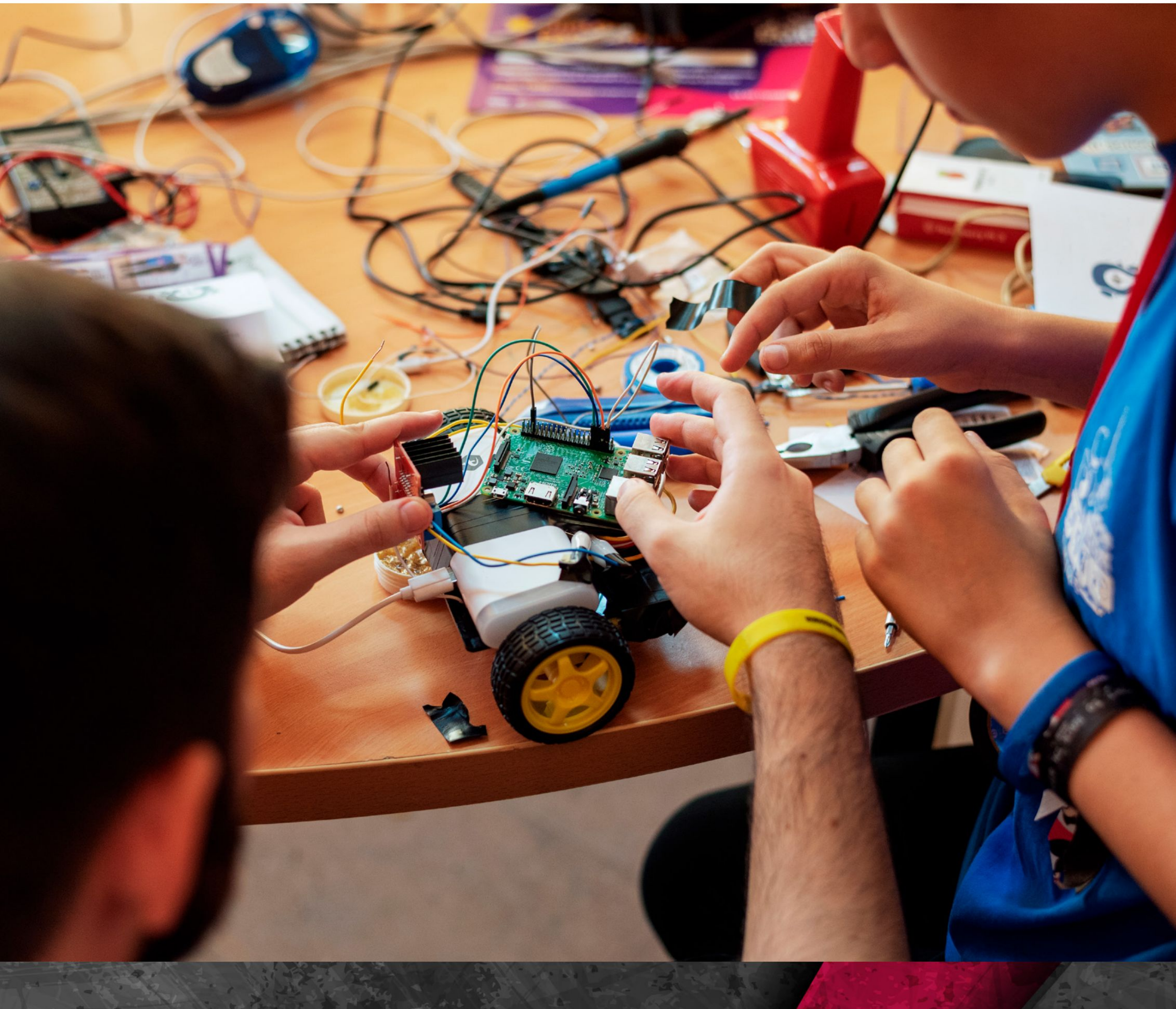
Over the page our journey begins. We would wish you luck but we're sure with our support you won't need it.

Contents

Raspberry Pi - The Complete Guide



New to Pi?
See page 6 for 'where to buy'



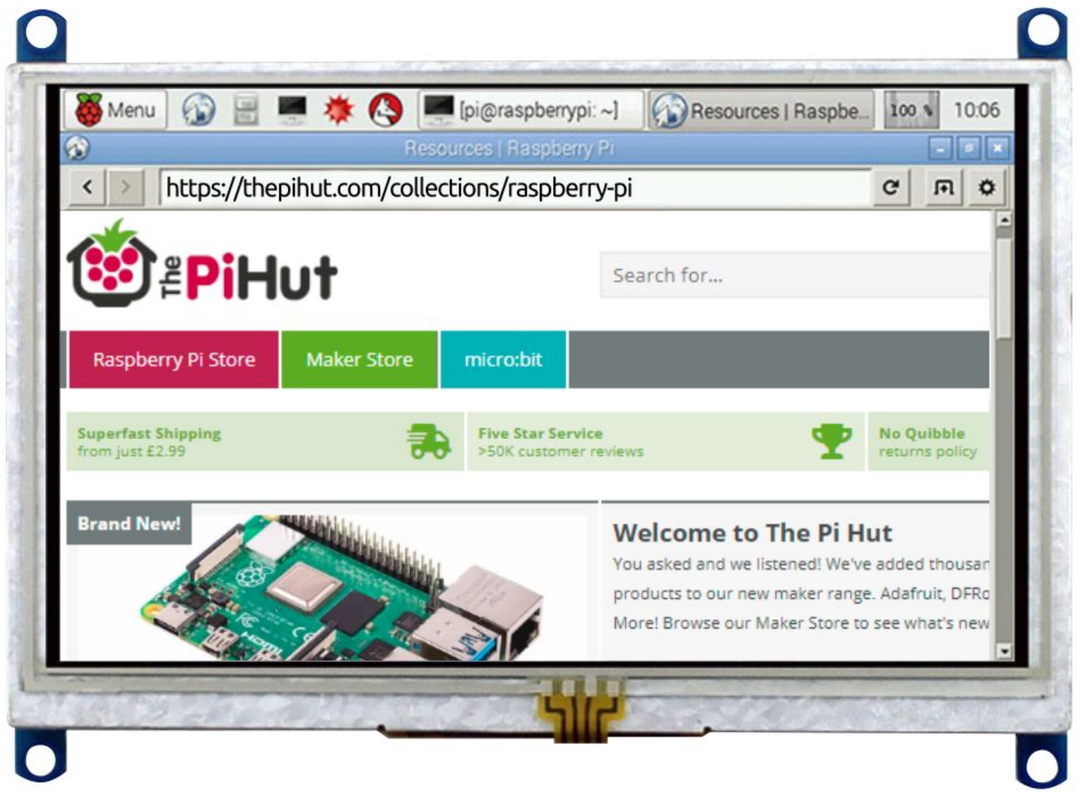
"...unleash your imagination with the power of the Raspberry Pi..."

New to Pi? Where to Buy?

If you're starting out with the Raspberry Pi, it helps to know where to get hold of one. There are four main Raspberry Pi approved resellers throughout Europe, and hundreds more internationally. Let's look at the top four Pi sellers.

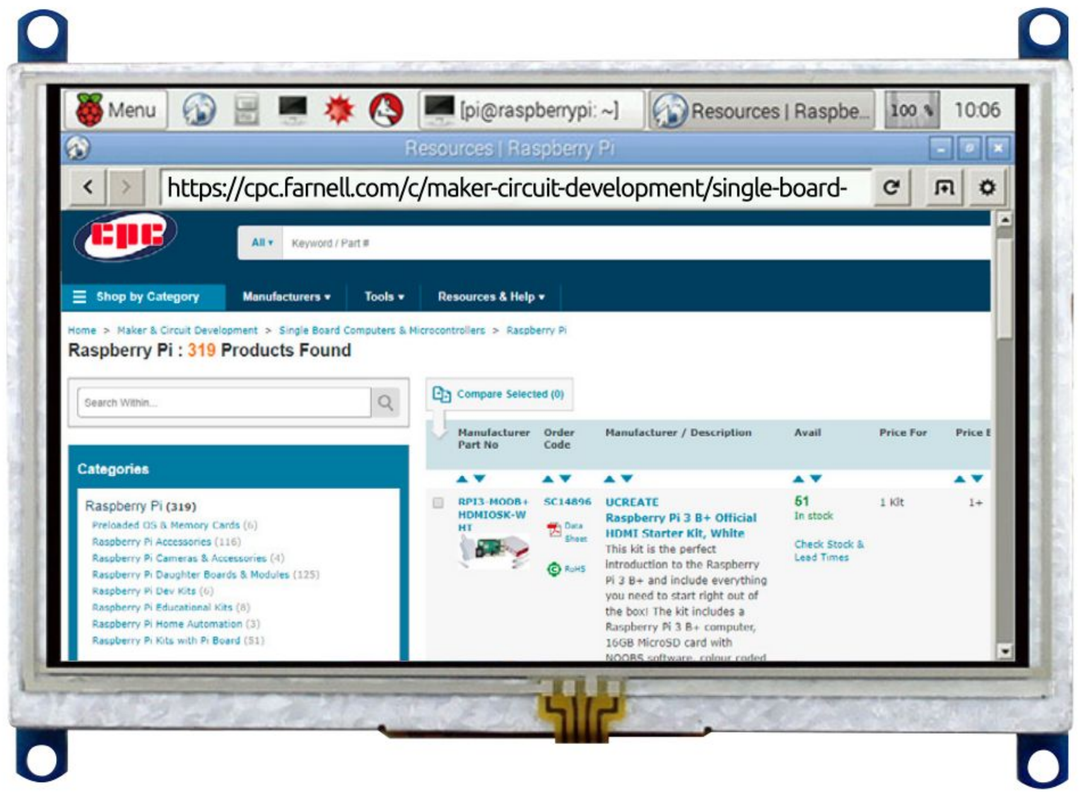
THE PI HUT

Open a browser and navigate to **https://thepihut.com/collections/raspberry-pi**. The Pi Hut is one of the most popular online shops for all things Raspberry Pi, including the Pi itself, cases, HATs, kits and bundles, cables, addons and much more. Prices vary depending on what you're looking for, but they're some of the best around.



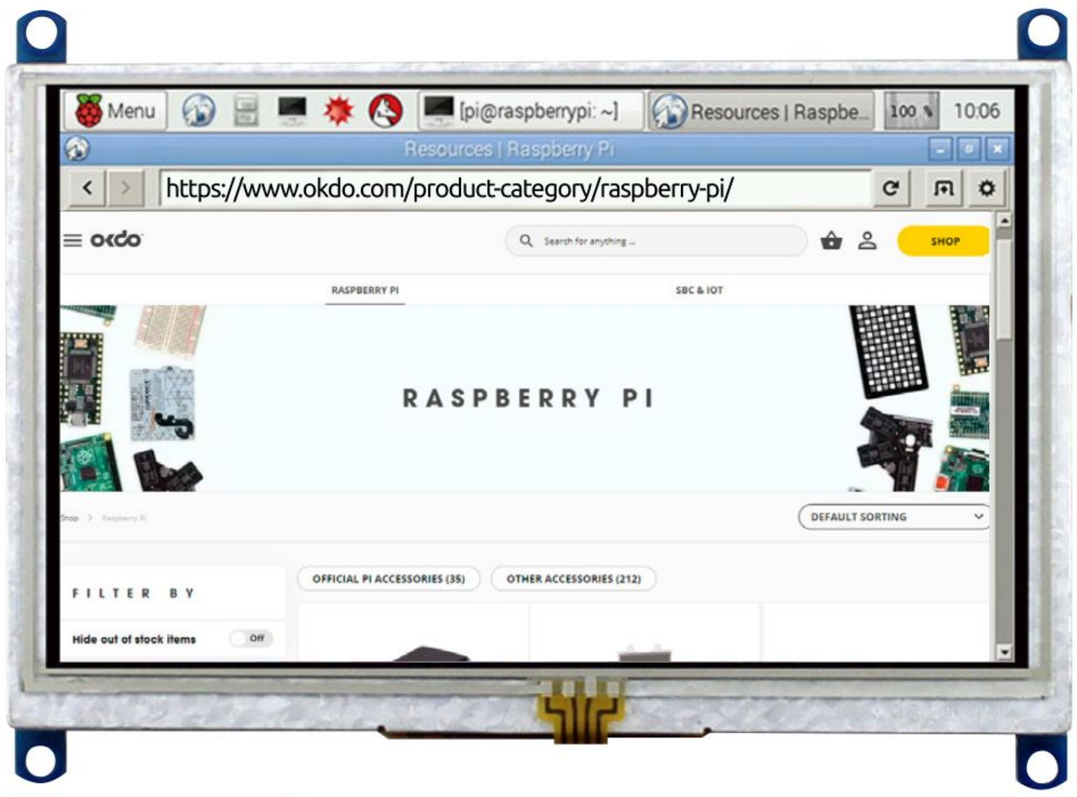
CPC

On its release in 2012 CPC Farnell was one of the original resellers of the Raspberry Pi. Therefore, not only does CPC stock a huge amount of Pi-related goodies, but it's also one of the premier global online electronics stores. Browse the Pi section at: **https://cpc.farnell.com/c/maker-circuit-development/single-board-computers-microcontrollers/raspberry-pi**.



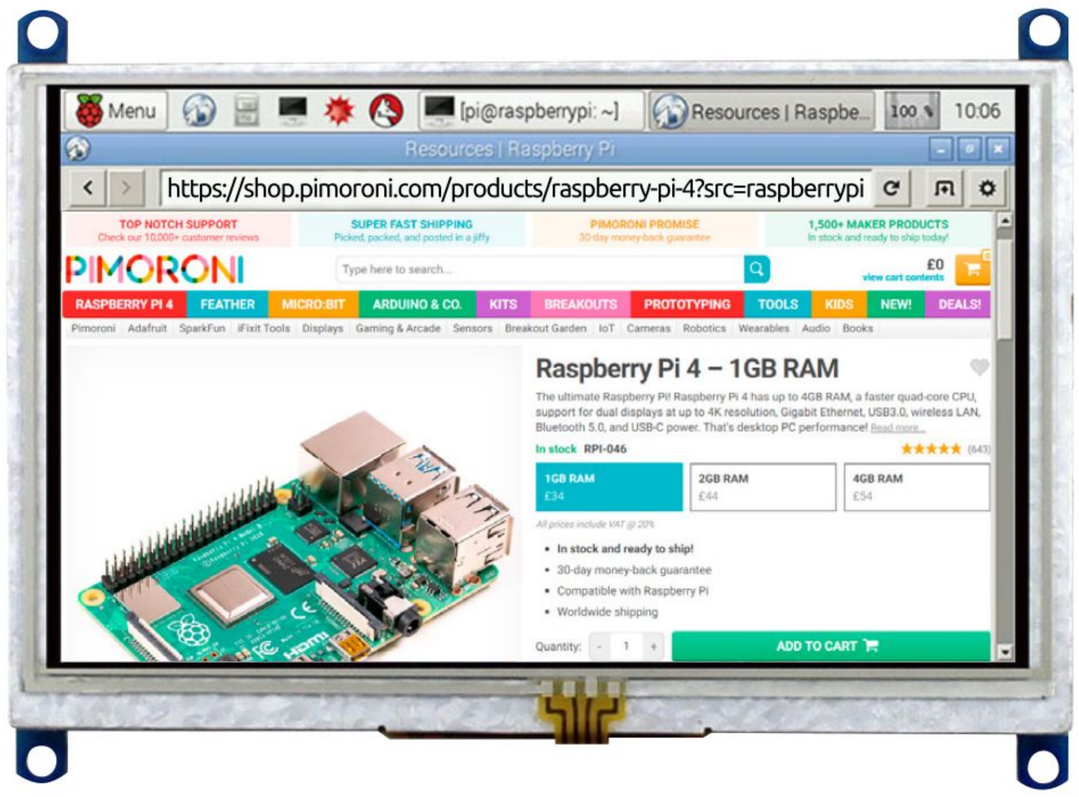
OKDO

With its huge range of Raspberry Pi products, including all manner of maker accessories, kits, parts and addons, OKdo is one of the top go-to stores for all the tinkerers out there. You'll find the Pi section of the shop at **https://www.okdo.com/product-category/raspberry-pi/**, so hunt around and see what's available.



PIMORONI

Founded in 2012, and the makers of the excellent Pibow case for the original Raspberry Pi, Pimoroni is a superb resource for tinkerers, makers and Pi fans everywhere. Found at **https://shop.pimoroni.com/products/raspberry-pi-4?src=raspberrypi**, Pimoroni stocks everything you'll need for your Pi, among other great technologies.

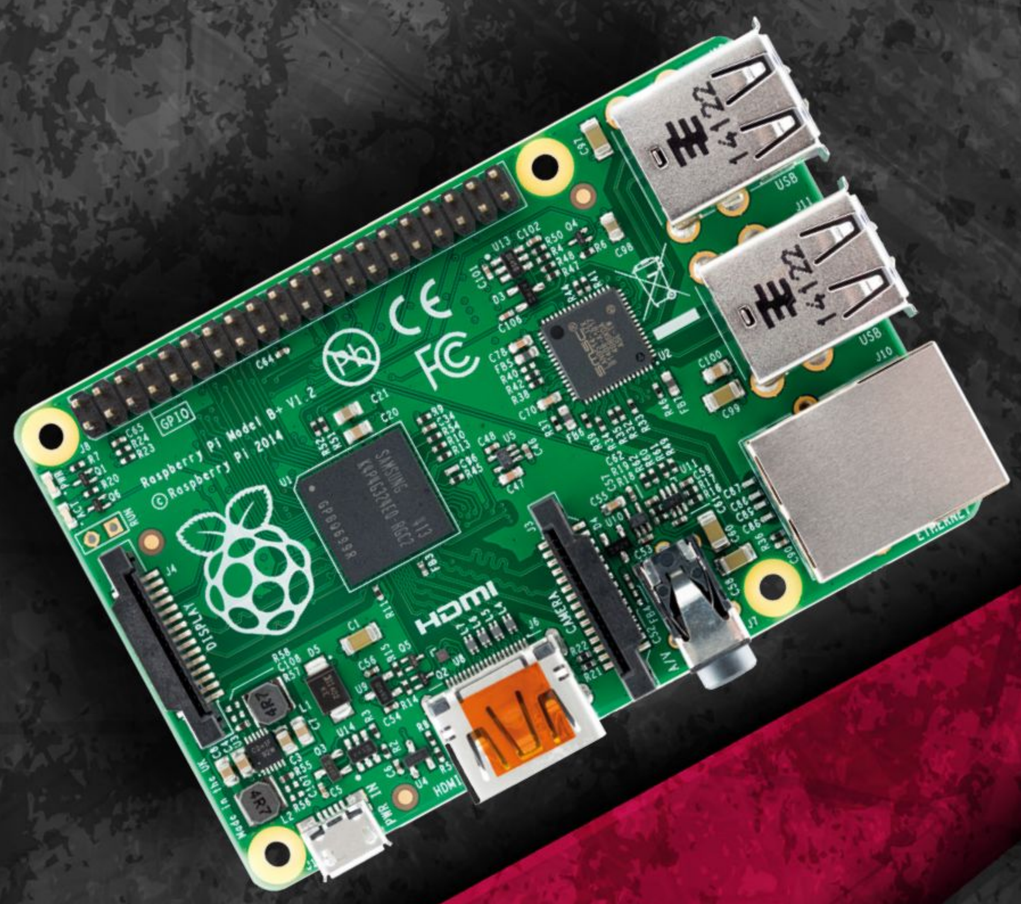


MORE PI

Naturally, you don't have to go through any of these recommended stores. Look around the Internet and you'll find plenty of other places to get hold of a Pi. Both Amazon and eBay are firm favourites for international customers.

A Slice of Raspberry Pi

Do you already own a Raspberry Pi but don't know what to do with it, or are you on the verge of purchasing your first Pi? This section of the book will help you build the foundations required to get the most from this amazing, tiny computer.



Say Hello to Raspberry Pi

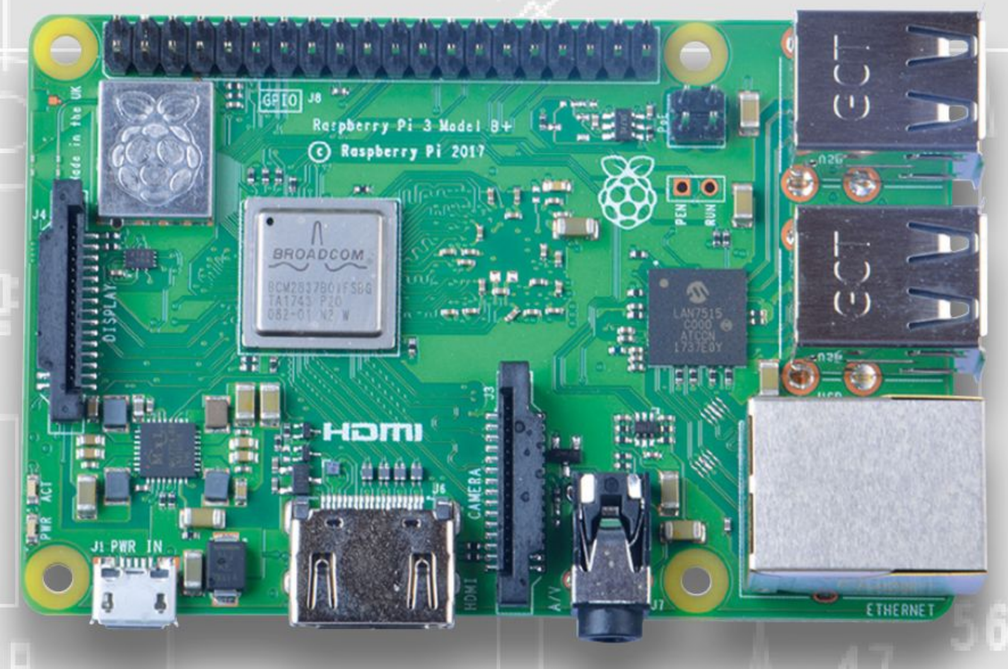
Let's take a look at the best-selling British computer ever: the Raspberry Pi! This bare-bones machine just looks like any other piece of circuitry, until you realise it's a full-blown credit-card sized computer ideal for learning programming and home hacking.

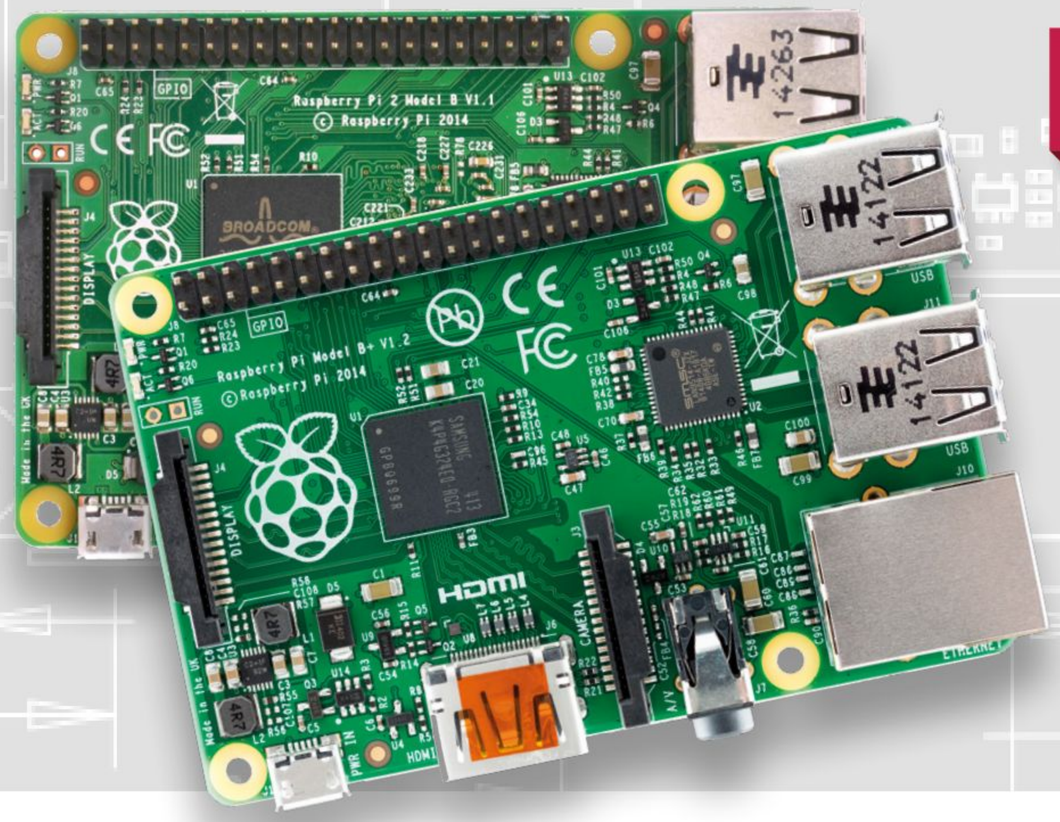
Raspberry Pi 2 Model B
Raspberry Pi 1 Model B+

The older Raspberry Pi 2 model can still be found on sale. It has the same form factor as the Raspberry Pi 3 with 4 USB Ports, Ethernet connection and Micro USB.

It may look the same but it has a slower 900 MHZ quad-core ARM Cortex-A7 CPU and 1GB RAM. This brings it close in line to many low-end desktop computers.

Complicating matters slightly is the existence of a Raspberry Pi Model 1 B+, which is identical in layout to the Raspberry Pi Model 2 but with an even slower ARMv6 700MHz processor. If you are confused which type you own then type cat /proc/cpuinfo at the command line. If it lists four ARMv7 processors you are using a Raspberry Pi 2, if you see just one with ARMv6 then you have the older Raspberry Pi Model 1 B+.





Raspberry Pi 3 Model B+

The Raspberry Pi 3 Model B+ was launched on 14 March - International Pi day 2018. This is an improved version with a faster 1.4GHz, 64-bit quad-core ARM Coretx-A53 processor, a dual-band 802.11ac wireless LAN and Bluetooth 4.2 controller, faster Gigabit Ethernet, improved mass storage, USB booting, improved thermal management and Power-Over-Ethernet support. All other aspects of the new Raspberry Pi are the same as the previous Pi 3 and as such, all the previous content you come across is compatible. if you're new to the wonderful world of the Raspberry Pi, this is the board we recommend you buy.

The Raspberry Pi is a British built low cost computer that enables everybody to learn computing, start programming and explore basic electronics. It's the size of a credit card but capable of running a full operating system and doing everything a desktop does.

More importantly, with the Raspberry Pi you install your own operating system, connect all your devices and create your own programs using languages like Scratch and Python. There's no case so you can hook up electronic circuits to the pins and control them to get input and output, so you can hack together electronic projects at home.

Setting up a Raspberry Pi is pretty simple, and throughout this book we'll take you step-by-step from unboxing your Raspberry Pi to setting it up and getting started.

There are many different models of Raspberry Pi available and each has slightly different features (see opposite).

The fundamentals of each Raspberry Pi are similar though. Each model is a lightweight computer on a single board that's roughly the size of a credit card. Each Raspberry Pi board features a processor made by ARM, which is similar to the models you find in a mobile phone. The ARM processor is fast and lightweight but it runs a different set of software than you might be used to. There are many different operating systems (OS) available, but for most of this book we'll focus on one called Raspbian JESSIE with PIXEL, which is the OS recommended by the Raspberry Pi Foundation.

There's no hard drive on the Raspberry Pi, instead the operating system is installed on an SD Card (the cards typically used in cameras). The operating system is installed onto the SD Card by copying the files to the SD Card using a computer. We'll show you how to set up Raspbian JESSIE with PIXEL (and other OS's) using a Mac, Windows or Linux computer.

Once you have an OS installed on the SD Card you can set up your Raspberry Pi. The Raspberry Pi is connected to a monitor via a HDMI socket whilst a keyboard and mouse is attached to the USB socket. A smaller Micro USB connection is used to provide power to the device. Most Raspberry Pi models feature an Ethernet connection, and an Ethernet cable is connected from your Raspberry Pi to a socket on your modem router.

The Raspberry Pi 3 has built in wireless networking and Bluetooth, making it easy to get online. You can attach a USB Wi-Fi dongle or Ethernet adaptor to older models.

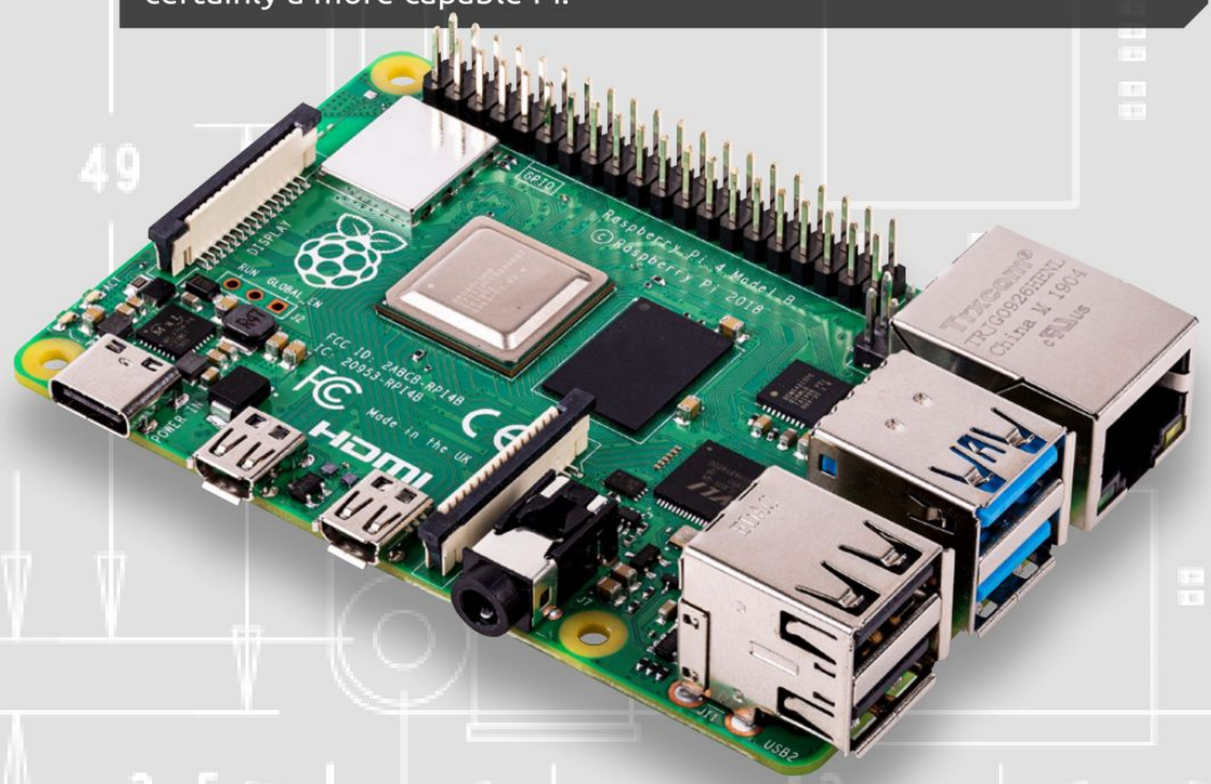
First it will help to know which Raspberry Pi you are using. This can be tricky as there have been several distinct models so far. The layouts opposite will help you determine which Raspberry Pi you have.

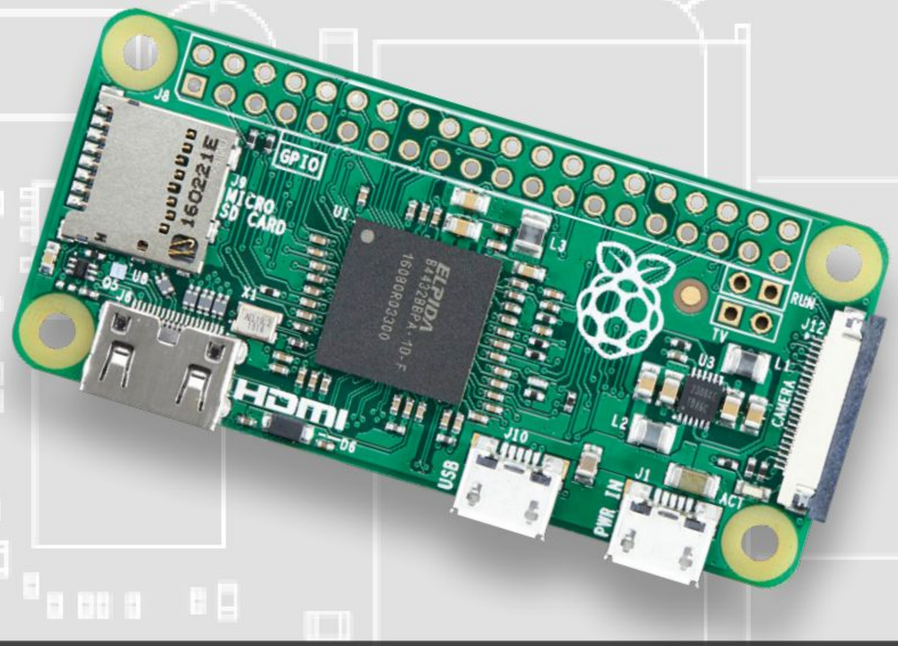
This book covers all the different models, and they offer largely similar features, although newer models offer additional extras. Once you know which Raspberry Pi you own, you can get it up and running.

Raspberry Pi 4 Model B

Released on June 24th 2019, the Raspberry Pi 4 Model B introduced a new direction in the layout and hardware specification of the Pi. The Pi 4 Model B now comes in three different memory size versions: 1GB, 2Gb, and 4GB. Obviously, the 4GB version is the most powerful of the three, and the Pi with the highest amount of on-board memory. Subsequently, it was sold out on the day of release and as such, has taken some time to get into the hand of the mainstream Pi users.

There's also an improved CPU, dual-monitor support in the form of a pair of micro-HDMI ports, and improved connectivity. Overall, it's certainly a more capable Pi.



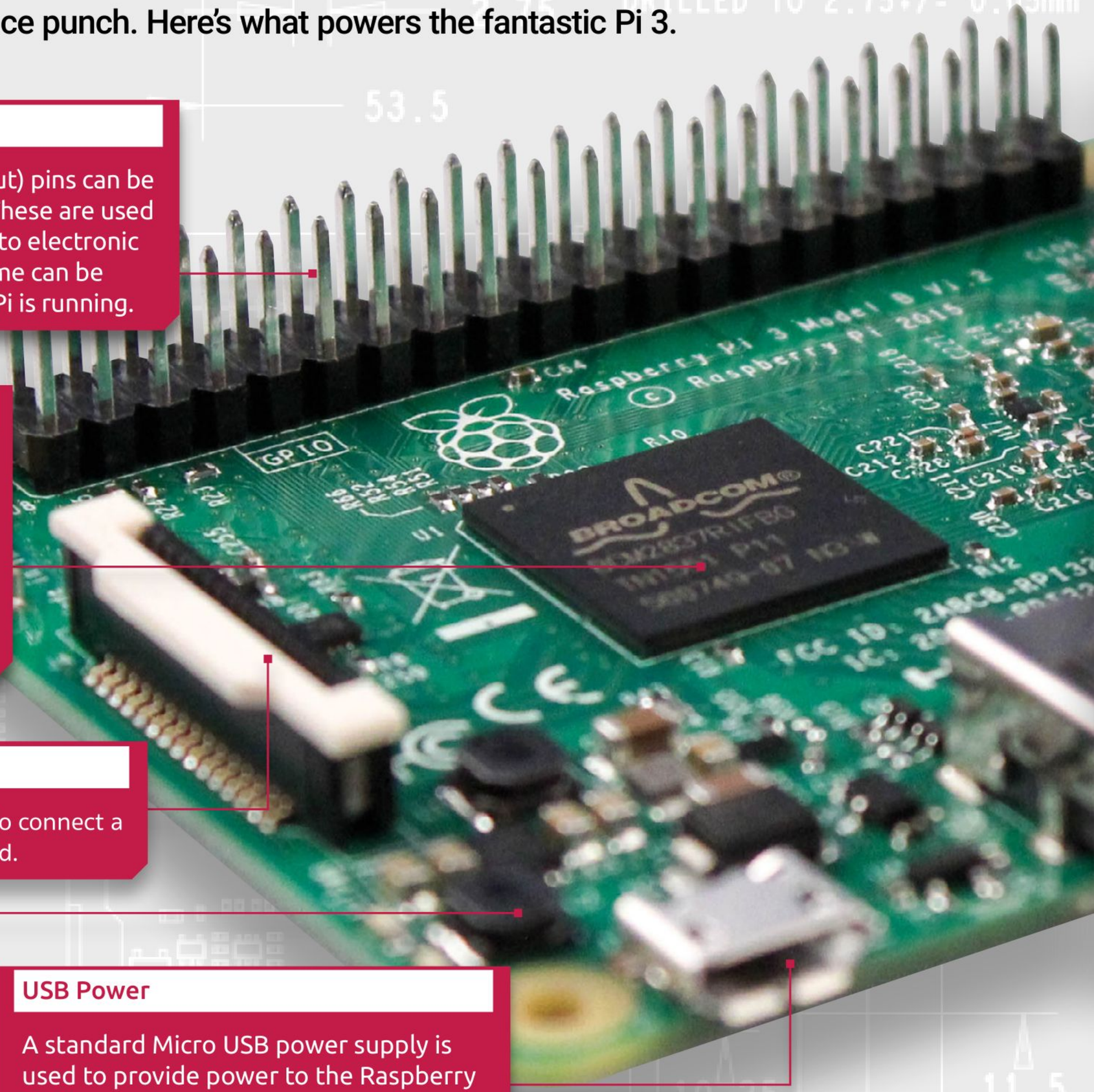


Raspberry Pi Zero

The new Raspberry Pi Zero is a super small, super cheap computer that costs just £4. It's half the size of the model A+ but has a surprising amount of power, including a 1Ghz single-core CPU and 512MB RAM. It has the full 40-pin GPIO header but you'll need to buy the pins separately and solder them in. It uses the same Micro-SD card as the other Raspberry Pi devices, so you can swap SD cards between them. Last but not least, it draws an incredibly low amount of power (Micro USB) making it ideal for small power devices. The ports are smaller than you'd find on the larger Raspberry Pi though, so you'll need a mini-HDMI adaptor as well as a micro-USB adaptor to connect devices. A four-port USB Hub and Ethernet adaptor will come in handy too.

Get to Know the Raspberry Pi

While the Raspberry Pi 4 is the newest model, the Pi 3 models are the most popular among the community. They're cheaper, more compatible with current hardware and software, and still pack a significant performance punch. Here's what powers the fantastic Pi 3.



40 GPIO Pins

The GPIO (General Purpose Input Output) pins can be accessed directly on the Raspberry Pi. These are used in projects to connect the Raspberry Pi to electronic circuits and control electric devices. Some can be turned on and off while the Raspberry Pi is running.

ARM-Powered

At the heart of the Raspberry Pi 3 is a Broadcom BCM2837 System on Chip (SoC). It contains a powerful 1.2GHz 64-bit quad-core ARM Cortex-A53 CPU. This is 50-60 percent faster than the Raspberry Pi 2 and ten times as fast as the original Raspberry Pi.

DSI

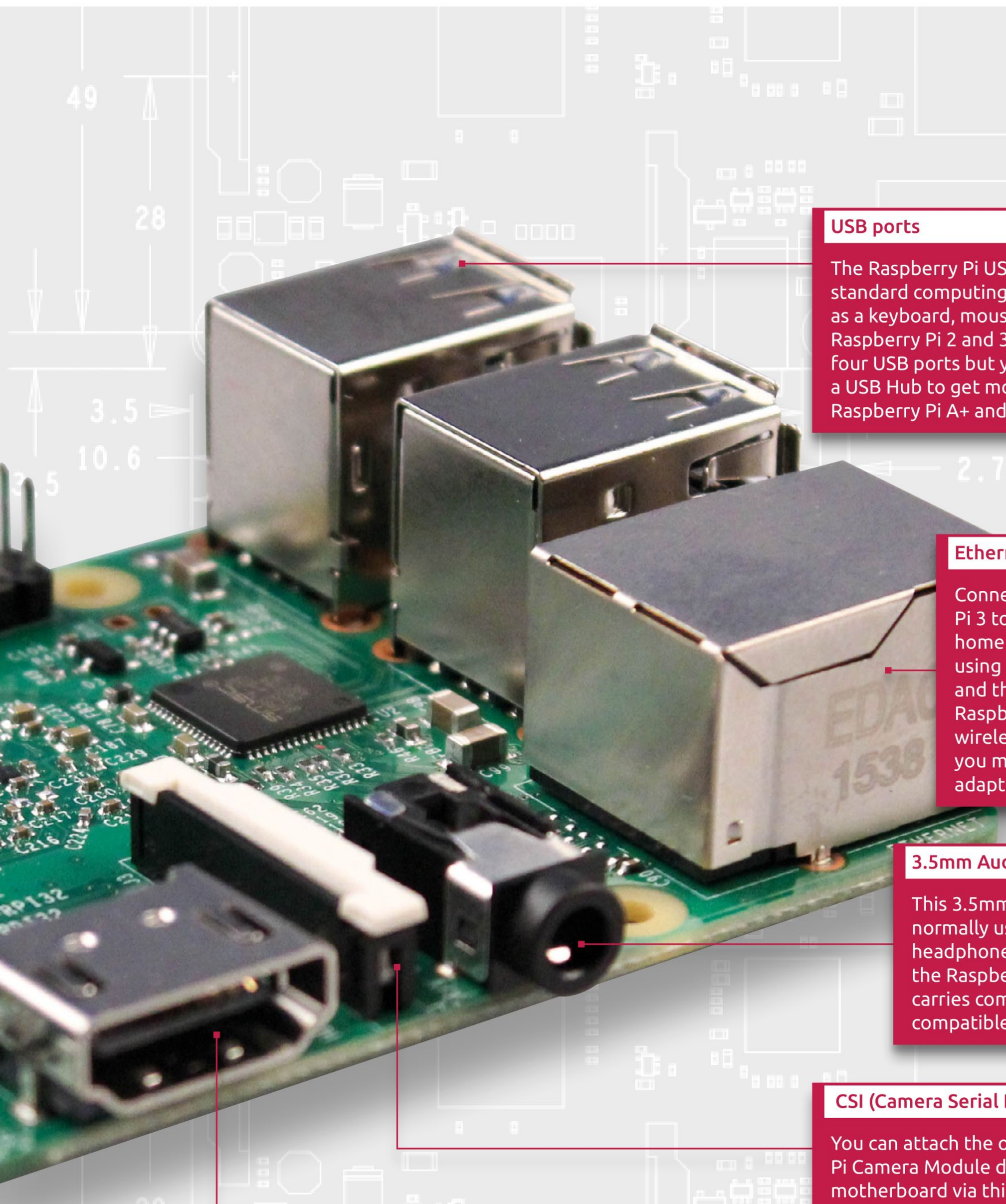
The Display Interface (DSI) can be used to connect a display directly to the Raspberry Pi board.

SD Card Slot

Instead of a hard drive you use an SD Card with the Raspberry Pi. The Raspberry Pi 3 (pictured here) uses a Micro SD Card. Some older Raspberry Pi models use a regular SD Card.

USB Power

A standard Micro USB power supply is used to provide power to the Raspberry Pi. You don't get a power supply but can use any 5V 2000mA Micro USB power supply and compatible cable, or buy an official power supply separately.



USB ports

The Raspberry Pi USB Ports to connect standard computing accessories such as a keyboard, mouse and webcam. Raspberry Pi 2 and 3 boards contain four USB ports but you might need a USB Hub to get more ports on the Raspberry Pi A+ and Pi Zero.

Ethernet and Wi-Fi

Connect the Raspberry Pi 3 to a network or home modem router using an Ethernet cable and this connection. The Raspberry Pi 3 has built in wireless networking, but you must add a USB Wi-Fi adaptor to older models.

3.5mm Audio/Video jack

This 3.5mm audio jack is normally used to attach headphones or speakers. On the Raspberry Pi 2 and 3 it also carries composite video using a compatible cable.

CSI (Camera Serial Interface)

You can attach the official Raspberry Pi Camera Module directly to the motherboard via this CSI connection.

HDMI

The HDMI connection enables you to hook up the Raspberry Pi to most modern monitors and digital televisions. It carries both video and audio.

18+

WHAT IS AVAXHOME?

AVAXHOME-

the biggest Internet portal, providing you various content: brand new books, trending movies, fresh magazines, hot games, recent software, latest music releases.

Unlimited satisfaction one low price
Cheap constant access to piping hot media
Protect your downloadings from Big brother
Safer, than torrent-trackers

18 years of seamless operation and our users' satisfaction

All languages
Brand new content
One site

AVXLIVE.ICU

AvaxHome - Your End Place

We have everything for all of your needs. Just open https://avxlive.icu

Up Close with the Raspberry Pi 4 Model B

Released at the end of June 2019, the Raspberry Pi 4 Model B is a significant improvement in terms of hardware. Labelled the 'Ultimate' Raspberry Pi, this new generation offers true desktop computing power.

FASTER AND CONNECTED

There's a lot to like about the Raspberry Pi 4: Up to 4GB of on-board memory, a faster quad-core CPU, support for dual 4K displays via a pair of micro-HDMI ports, and more. Let's take a look at the Pi 4 and see what its got under-the-hood.

PRICING

Since the Raspberry Pi 4 now offers three different versions within the new model, it stands to reason that the pricing for it has changed slightly. The 1GB memory version of the Pi 4 is priced at around £34, depending on where you shop. The 2GB memory version is on sale for around £44, while the top-end, 4GB memory version will set you back in the region of £54. Together with the cost of the Pi itself, you will also need to factor in one or two micro-HDMI cables (depending on whether you want to connect one or two monitors), each costing roughly £5-plus.

Improved GPU

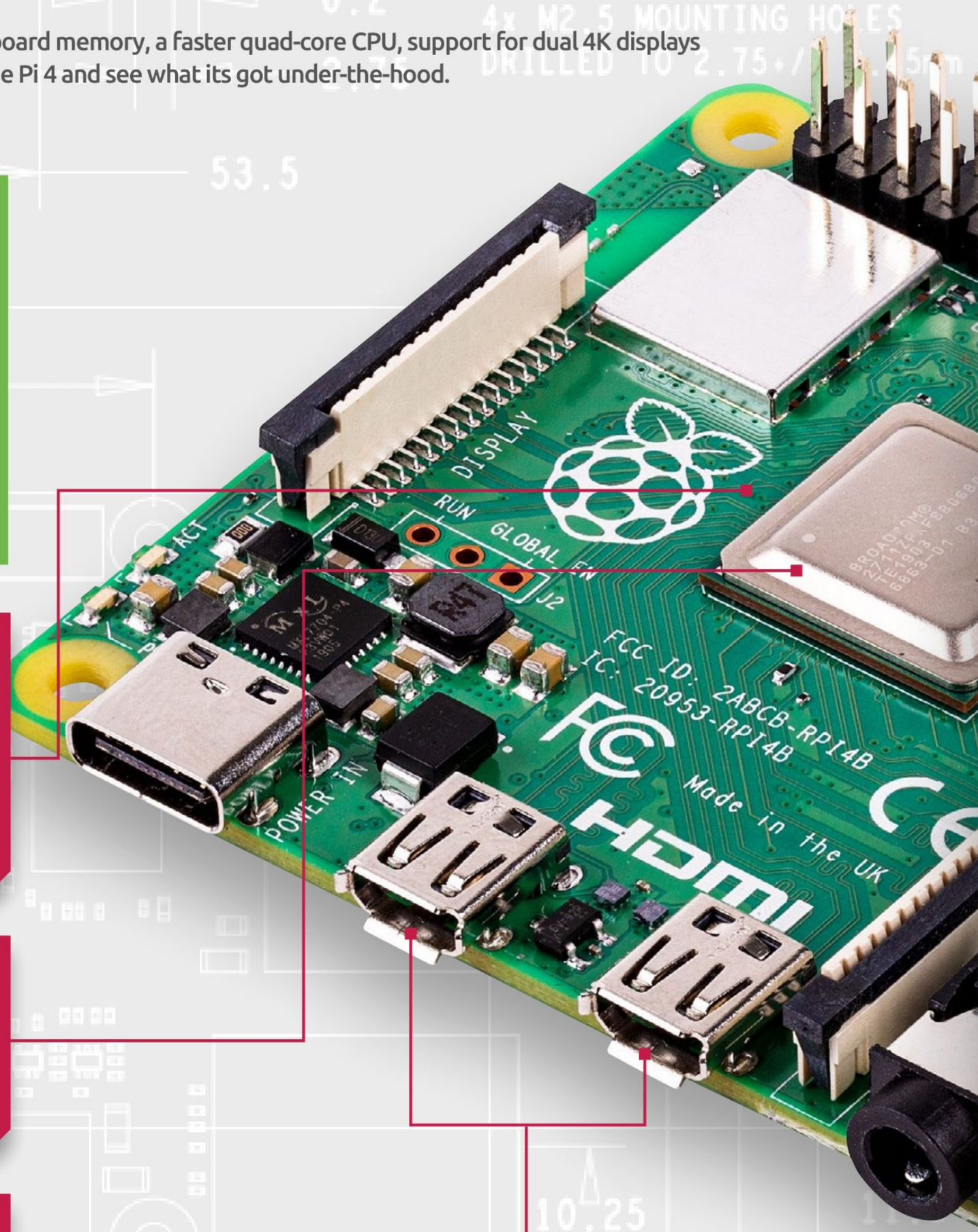
With a VideoCore 6 GPU, the Pi 4 is capable of 4K resolutions at 60FPS (Frames per Second), and thanks to some tweaks to the video codes in both Raspbian and the version of VLC packaged with Raspbian, the Pi 4 is now a pretty decent media device.

Improved Processing

The 1.5GHz, 64-bit quad-core ARM Cortex-A72 CPU, is the driving force behind the Pi 4's improved performance. Thanks to this CPU, you're able to enjoy faster apps and content.

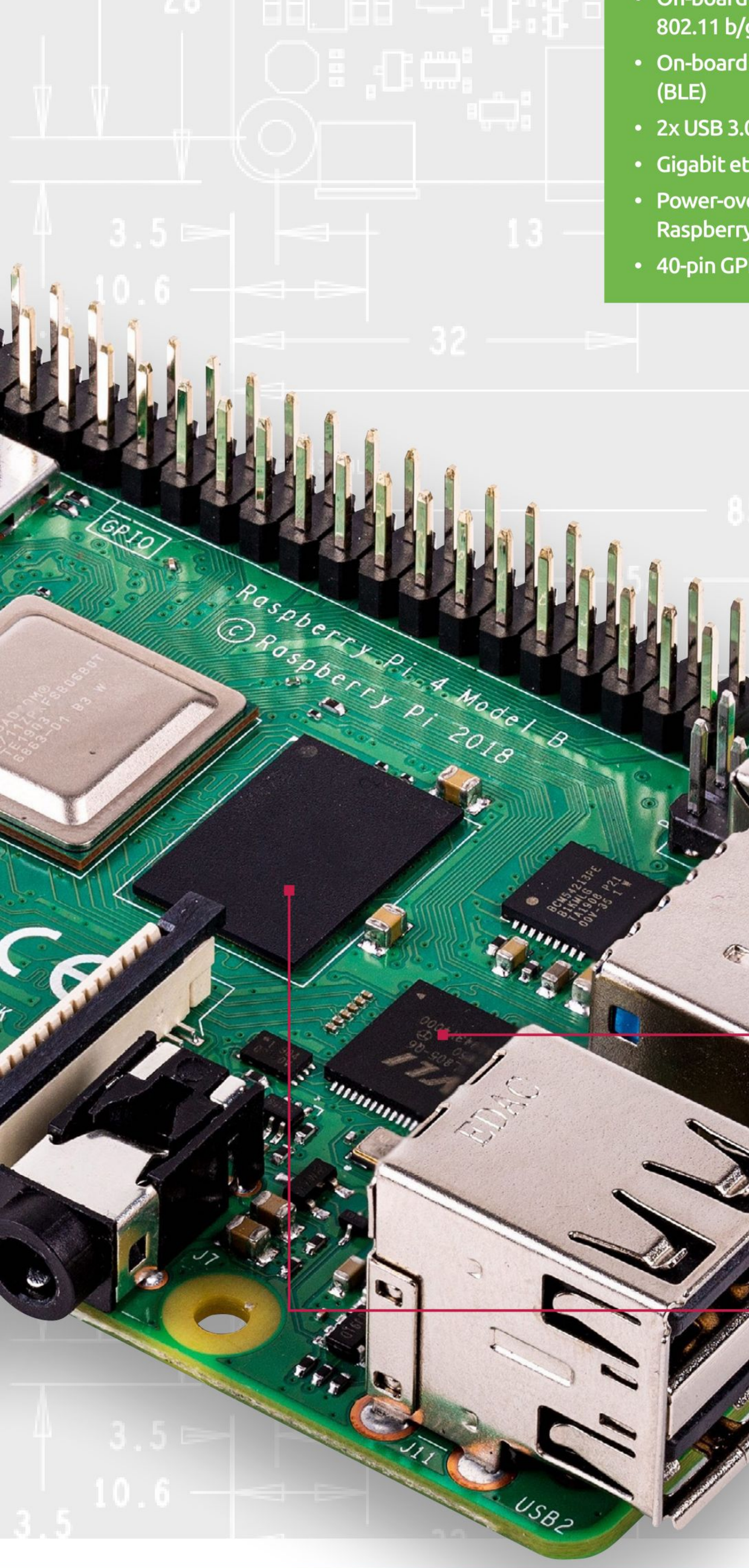
Dual Display

In a shock move the team behind the Pi 4 decided to include a pair of micro-HDMI ports. This means you're able to connect two monitors for dual-screen displays.



PI 4 SPEC SHEET

- 1.5GHz 64-bit quad-core ARM Cortex-A72 CPU (ARM v8, BCM2837)
- 1GB, 2GB or 4GB RAM (LPDDR4)
- On-board wireless LAN - dual-band 802.11 b/g/n/ac
- On-board Bluetooth 5.0, low-energy (BLE)
- 2x USB 3.0 ports, 2x USB 2.0 ports
- Gigabit ethernet
- Power-over-Ethernet (requires the Raspberry Pi POE HAT)
- 40-pin GPIO header
- 2× micro-HDMI ports (up to 4Kp60 supported)
- H.265 (4Kp60 decode)
- H.264 (1080p60 decode, 1080p30 encode)
- OpenGL ES, 3.0 graphics
- DSI display port, CSI camera port
- Combined 3.5mm analogue audio and composite video jack
- Micro-SD card slot
- USB-C power



USB 3.0

The inclusion of a couple of USB 3.0 ports is an excellent choice. The faster data bandwidth of USB 3.0 will, for example, enable the Pi to read and write that data, both to and from an external drive, far quicker than ever before.

Connectivity

Now with Bluetooth 5.0, alongside 802.11 b/g/n/ac wireless, and a gigabit Ethernet port, the Pi 4 is certainly well connected.

Memory

The RPi4 now comes with a choice of memory: 1GB, 2GB, or a 4GB model. The more memory means you're able to have more applications open, without degrading the overall performance of the Pi.

Raspbian: The Complete Operating System

The main Raspberry Pi operating system is Raspbian, which is a Linux-based OS. While the Pi is primarily an educational and project board, with Raspbian, it does become a fully-functional desktop computer.

POWERING THE PI

The Raspbian OS has been powering the Raspberry Pi since its release back in 2013. At its core, Raspbian is a Linux operating system, based on the hugely popular Debian flavour of Linux.

Mike Thompson and Peter Green created Raspbian in 2012, to work on the low-performance ARM CPUs found in the early Pi models. Toward the end of 2014, Raspbian was further improved to work on the then new model Pi, the Raspberry Pi 2; utilising the 3.18 Linux kernel while still having Debian 7 (codenamed Wheezy) as the base.

Mid-way through 2015, Raspbian started using Debian 8 (Jessie) as the base, with kernels 4.1, 4.4, and 4.9. Kernel 4.9 was also used from 2017's Raspbian release, which has since used Debian 9 (Stretch) as the foundation distribution.

More recently, from November 2018, the Raspberry Pi Foundation has split the Raspbian OS project into three separate distributions. Although these are essentially the same core operating system, those available now are a Lite, Minimal and Full desktop set of images.

One of the primary reasons for the three-way split is due to the overall size of the continually improving Raspbian. In its early years, the Raspbian desktop image size was a little under 1GB. This image contained everything the Pi user of the time needed to get their projects up and running, as well as enjoying third-party applications such as LibreOffice. Slowly, as the Pi itself evolved, the Raspbian image increased. These days the image weighs in at around 1.8GB and it's likely to continue to expand over the coming years as new software is improved on and added, alongside changing hardware. However, the three-way split addresses the issue of an ever-expanding OS image, while still enabling the user to get the most from their Raspberry Pi.

The Lite version image comes in at roughly 350MB in size. While it may seem like a logical choice to opt for the Lite version, it's not recommended for every user. The Lite version basically installs without any of the software you expect from the previous versions of Raspbian; it doesn't even come with the graphical desktop. This makes it an incredibly streamlined and efficient version of the OS that's ideal for those who run headless Pi servers (a Pi without a monitor, keyboard or mouse attached, that the user remotely connects to in order to input commands). You will need to know your Linux commands in order to get the most from the Lite version, so for most beginners it's not a good place to start. The more advanced users will, however, be able to setup their Pi projects without the unnecessary baggage of all the other applications clogging up the Pi's precious system resources. It's worth noting that the Lite edition can be made up to a fully functional desktop version by installing all the relevant software manually.

The Minimal edition is now the default version recommended for most new Raspberry Pi users. This version is around 1GB in size and includes the PIXEL desktop, Chromium web browser, new hardware accelerated VLC player and Python. It does not include LibreOffice, Scratch, Sonic Pi and many of the other tools, programs and applications that previous versions of Raspbian included. It's a perfect start for most users and those who are more familiar with the workings of Raspbian and Linux can easily add more apps later, as they require.

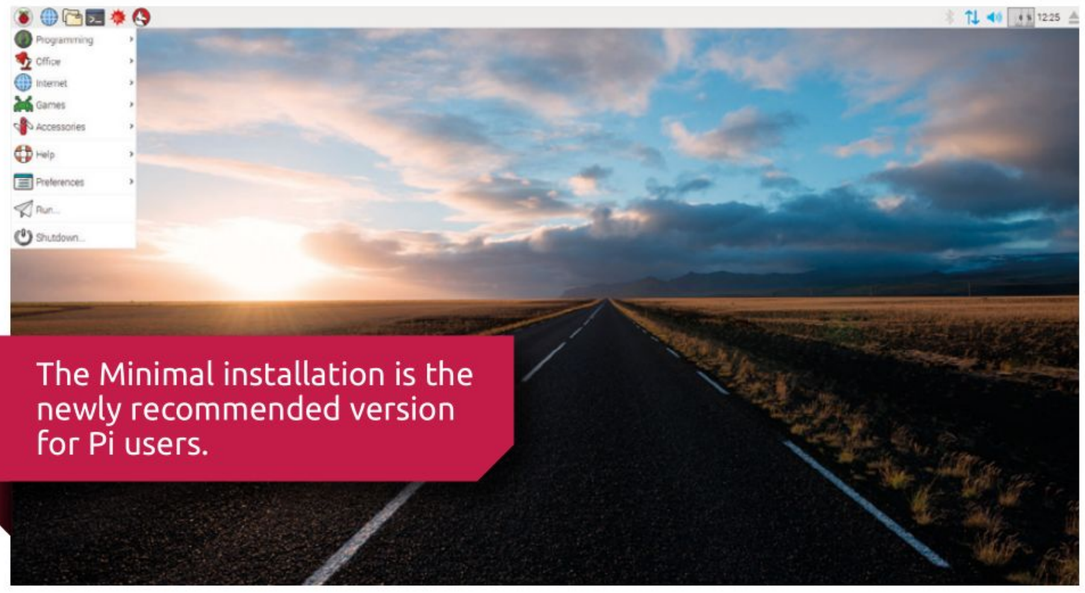
The Full desktop version is 1.8GB in size (at the time of writing) and as you imagine, contains pretty much everything you can squeeze into Raspbian. You get the PIXEL desktop, LibreOffice, VLC, browser, games, programming resources and much more.

So, which version should you download and use on your Pi? Normally, as recommended by the Raspberry Pi Foundation, most new Pi users are best opting for the Minimal desktop; but that depends greatly on what your plans are for using your Raspberry Pi. We recommend new users begin with the Full desktop version, as this is the version that gives you the complete Pi experience along with all the associated software; even if you only use a small percentage of it. In fact, this entire book was written using the Full desktop version on a Raspberry Pi 3 Model B+.

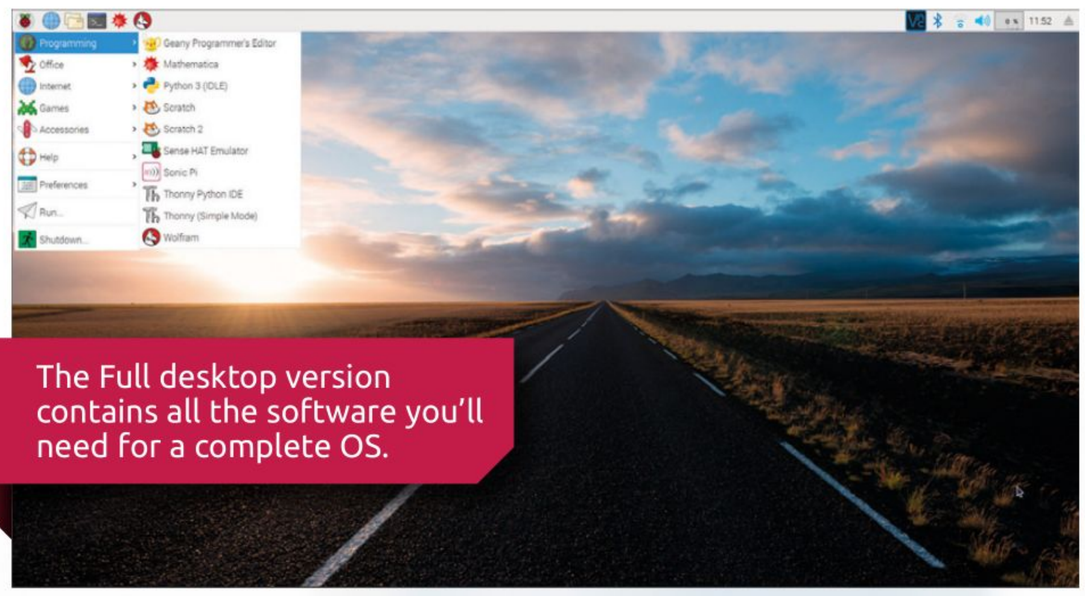
As you learn more about the Raspberry Pi, Raspbian and Linux and begin to form your Pi into a working project, you can easily downgrade the version of Raspbian to reflect your use. One day, you may even opt for the Lite, command line only version, occasionally connecting to it remotely and issuing a few commands. As with most things Pi-related, it's purely up to you.

Raspbian: The Complete Operating System

The Lite version doesn't even have a graphical desktop; it's command line only.



The Minimal installation is the newly recommended version for Pi users.



The Full desktop version contains all the software you'll need for a complete OS.

We recommend beginners try the Full desktop version first, then downgrade when they know more about the Pi and its OS.

Which Pi is Right for Me?

With several models of Raspberry Pi available to purchase, you can be forgiven for any confusion over which model you should buy. So, to help save you time and money, let's see which Pi is best for your needs.

THE POWER OF PI

With many different versions, across four generations of models, the Raspberry Pi is certainly a busy little board. However, while each Pi offers something slightly different, for the newcomer it's a confusing medley of hardware specifications and model numbers. Which Pi, then, is best for you?

Naturally, that question depends greatly on what it is you want to do with your Raspberry Pi. If you have a particular project in mind, such as a home media centre server, then you'd probably opt for the more powerful and hardware-capable model of Raspberry Pi. If you want to set up a wireless security camera project, using the Raspberry Pi as the core hardware, then perhaps the Pi Zero W would be a better fit. To begin with, let's have a brief look at the models available.

FIRST GENERATION:

Although now quite old, in computing terms, the first generation Raspberry Pi models are still available to purchase. These are the Raspberry Pi 1 Model A+ and Pi 1 Model B+.

The Pi 1 Model A+ was released in November 2014 and replaced the original Model A. It features the now standard 40-pin GPIO, Micro SD Card, lower power consumption, and better audio circuitry. It's also a smaller package than the original Pi, while having a 700MHz processor and 512MB of memory.

While a worthy Pi, it does lack the built-in networking of its newer model counterparts; both Ethernet and Wi-Fi. The processor is somewhat lacking in the performance department, which can be an important factor if you are thinking of using this model for any serious CPU-related projects. In short, it's probably worth avoiding the Pi 1 Model A+.

The Pi 1 Model B+ is a far more capable design than its sibling Model A+. Released in July 2014, the Pi 1 Model B+ boasts built-in 100Mb Ethernet and four USB 2.0 ports. It does, however, still feature the same lacklustre processor as the Pi 1 Model A+, but its extra USB and Ethernet ports help make up for any lack in available hardware.

The first generation models are decent enough, providing you're undertaking a relatively low-level project. Fundamentally, although you can pick them up from a variety of online stores slightly cheaper than the current models, they probably aren't worth the savings.

SECOND GENERATION:

There are two, Pi 2, second generation models available: the Model B and the Model B version 1.2. Although you may be hard pressed to find an original Pi 2 Model B (without the 1.2 version), so we will be referring solely to the Pi 2 Model B as the 1.2 version that was released mid-2016.

The Pi 2 Model B offered the user a more powerful Pi experience. With 1GB of memory, an upgraded 900MHz quad-core ARM Cortex-A7 processor, and an improved HDMI port. As with the previous first generation Model B, this version also comes with a built-in 100Mb Ethernet port and four 2.0 USB port hub.

Overall, this is a superior choice of Raspberry Pi over the first generation models. The extra processing power, even though it's only 200MHz, does make a difference, alongside the extra 512MB of memory. If you find one that's significantly cheaper than a more recent Pi, then it's worth considering.

THIRD GENERATION:

The third generation Pi models are significantly more capable than the previous, but they may cost slightly more than the second generation model. However, you do get more Pi for your money.

The first third generation model to be released was the Pi 3 Model B, in February 2016. A newer quad core 1.2GHz Broadcom BCM2837 64-Bit processor, 1GB of memory, and a 4-pole stereo output and composite video port meant that this was the power-Pi to have. It's still a very good model to use, and can accomplish most of the tasks its younger sibling, the Pi 3 Model B+, can do.

The second of the third generation models released was the upgrade to the Model B, the Pi 3 Model B+, released in March 2018. With an impressive 1.4GHz quad core processor, 1GB of faster LPDDR2 memory, dual frequency built-in Wi-Fi and a gigabit Ethernet port, the Pi 3 Model B+ is the most powerful Raspberry Pi to date, and is therefore capable of running all your projects without any problems.

The newest member of the third generation Pi models is the replacement for the second generation A+. Released in November 2018, the Pi 3 Model A+ enjoys the same processor as the more powerful Pi 3 Model B+, but has half the available memory, at 512MB. It also loses the Ethernet port and three of the USB ports, but it does boast dual band Wi-Fi and a far smaller footprint than previous models.

Out of the third generation models, the ones to look out for are the Model B+ and Model A+. The B+ will give you a more powerful Pi experience, but it costs slightly more, whereas the Model A+ is much smaller, but lacks the extra memory and additional USB ports.

FOURTH GENERATION:

Introduced on 24th June 2019, the Raspberry Pi 4 Model B is a significant leap in terms of Pi performance and hardware specifications. It was also one of the quickest models, aside from the original Pi, to sell out.

With a new 1.5GHz, 64-bit, quad-core ARM Cortex-A72 processor, and a choice of 1GB, 2GB, or 4GB memory versions, the Pi 4 is one-step closer to becoming a true desktop computer. In addition, the Pi 4 was launched with the startling decision to include dual-monitor support, in the form of two micro-HDMI ports. You'll also find a pair of USB 3.0 ports, Bluetooth 5.0, and a GPU that's capable of handing 4K resolutions and OpenGL ES 3.0 graphics.

In short, the Pi 4 is the most powerful of the current Raspberry Pi models. However, the different memory versions have an increased cost. The 1GB version costs £34, 2GB is £44, and the 4GB version will set you back £54. Remember to also factor in one or two micro-HDMI cables with your order.

ZEROS:

In between the first and second generation Raspberry Pi models, the foundation launched the Pi Zero and Zero W.

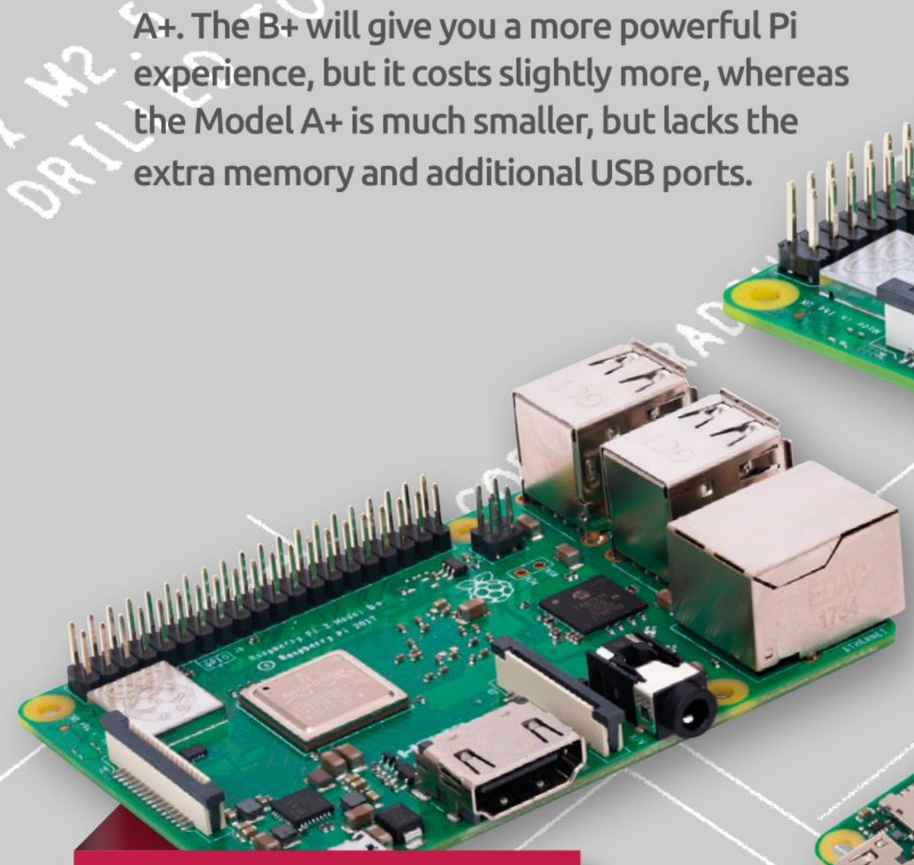
The Raspberry Pi Zero was a significant release, as the extremely popular Pi was now even smaller. Measuring at just 65 x 30 x 5mm, the Zero still managed to pack in a single core 1GHz processor, 512MB of memory, a mini-HDMI port, micro USB port, 40-pin GPIO and a micro-SD card slot. However, it lacked wireless and other networking capabilities, so you would need to factor in a USB hub and network hardware.

The Raspberry Pi Zero W, on the other hand, is a far better choice. The processor and memory are the same, as are the other hardware items, but, as the W indicates, this model comes with wireless networking built-in. With a 2.4GHz single-band Wi-Fi module, as well as Bluetooth 4.1, the Pi Zero W is an impressive, slim bit of hardware.

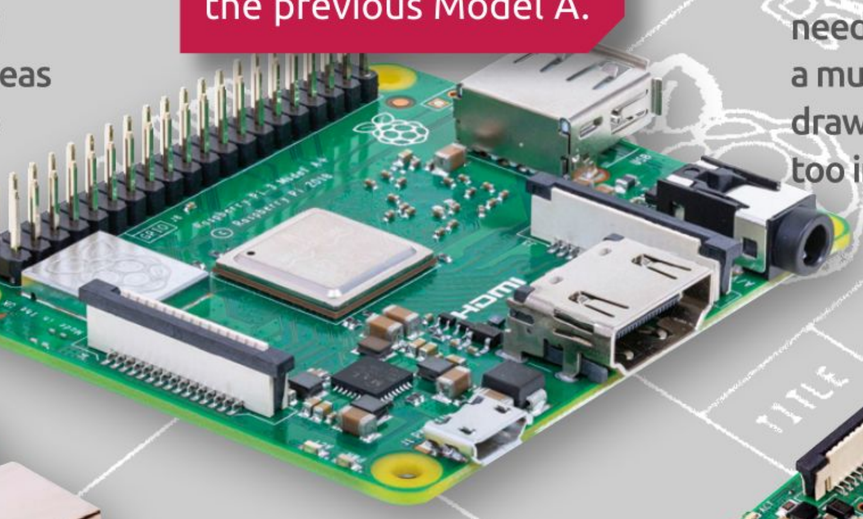
While the Pi Zeros may sound like a logical choice, considering their far smaller footprint, they do lack the performance power of the newer third generation models. We'd recommend you opt for a Pi Zero W over the older Pi Zero, as networking is available out of the box.

In conclusion, the Pi 4 Model B is the main Raspberry Pi worth considering if you want the full Pi experience; use it for programming, gaming, projects, connectivity and so on. The Pi 3 Model A+ can be used for projects that require more power, but where a smaller size is needed, and the Pi Zero W for projects where a much smaller footprint and lower power draw are needed, and CPU performance isn't too important.

The Pi 3 Model A+ was released in November 2018 and greatly improves over the previous Model A.



The Raspberry Pi 3 Model B+ is one of the best Pi models available.



The Raspberry Pi Zero W, with built-in Wi-Fi and Bluetooth, is a great project Pi.

Of the Pi models available, we'd recommend the Pi 4 Model B, Pi 3 Model B+ and Pi Zero W.

Raspberry Pi in Numbers

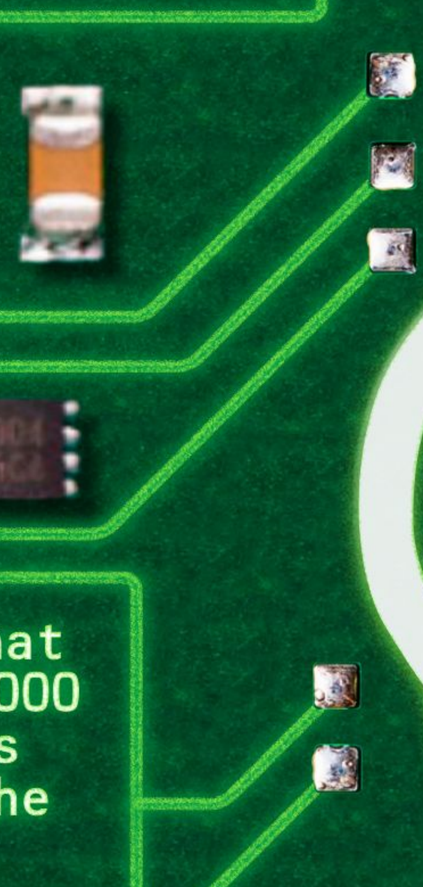
The Raspberry Pi is one of the most successfully launched computers in decades. With the perfect mix of hardware, cost, connectivity, and development, plus a good, stable Linux OS to back it up, the Pi has proved itself to be a near-perfect educational platform. Here's some facts and figures for our favourite flavour of Pi.

3.14159265358979...

There's a Raspberry in Antarctica, where it's as cold as -42C (-45F)

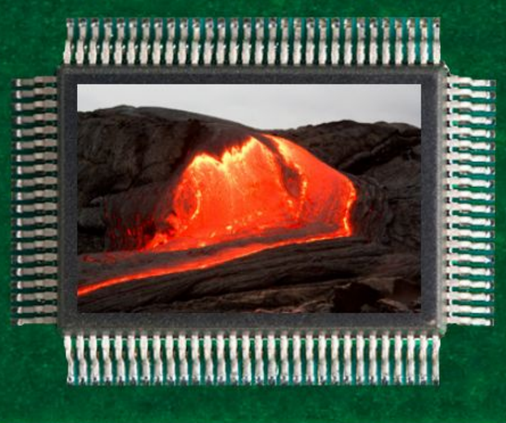
It's estimated that over **250,000** young people every week are learning how to code with a Pi

There are over **2,500** Raspberry Pi Certified Educators



The Pi 4 is between three to four times more powerful than the Pi 3 Model B+

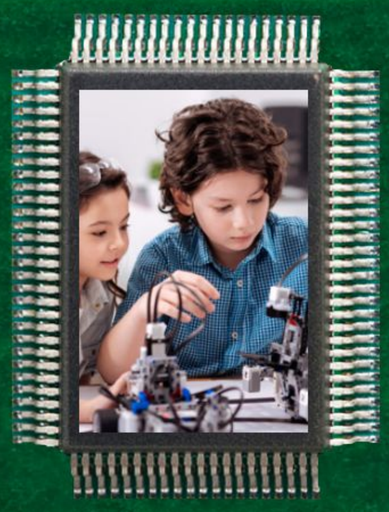
It is estimated that there are over 50,000 code clubs across the world using the Raspberry Pi



During the eruption of the Kilauea volcano in Hawaii in 2018, a Raspberry Pi was used as a seismometer

It was originally going to be called Raspberry Py, as it was only designed to run Python

Sources: The Raspberry Pi Foundation, The Guardian, readwrite.com, Popular Mechanics, CNN, BBC.

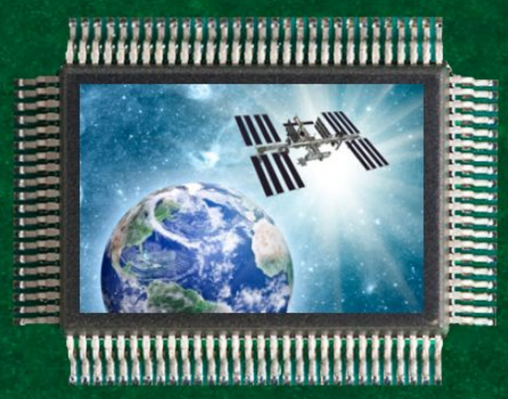


It was originally designed for school children to learn electronics and coding

Pi Zeros are used as small cameras, placed on the backs of Green Sea Turtles, to monitor habitation

February 2019 marked the point where the **25 millionth** Raspberry Pi was sold

There are several Raspberry Pi's in Borneo's Rainforests, monitoring biodiversity



In 2017 two Pi's went onboard the ISS and ran code developed by school children

The Pi 4 is powerful enough to run Windows 10 as a Thin Client

The two most used Pi projects are retro emulation and as a media centre

Stacked end to end, all the Raspberry Pi's would be taller than the Empire State Building

It is estimated that there are over 50,000 code clubs across the world using the Raspberry Pi.

Kit You'll Need and How to Set it Up

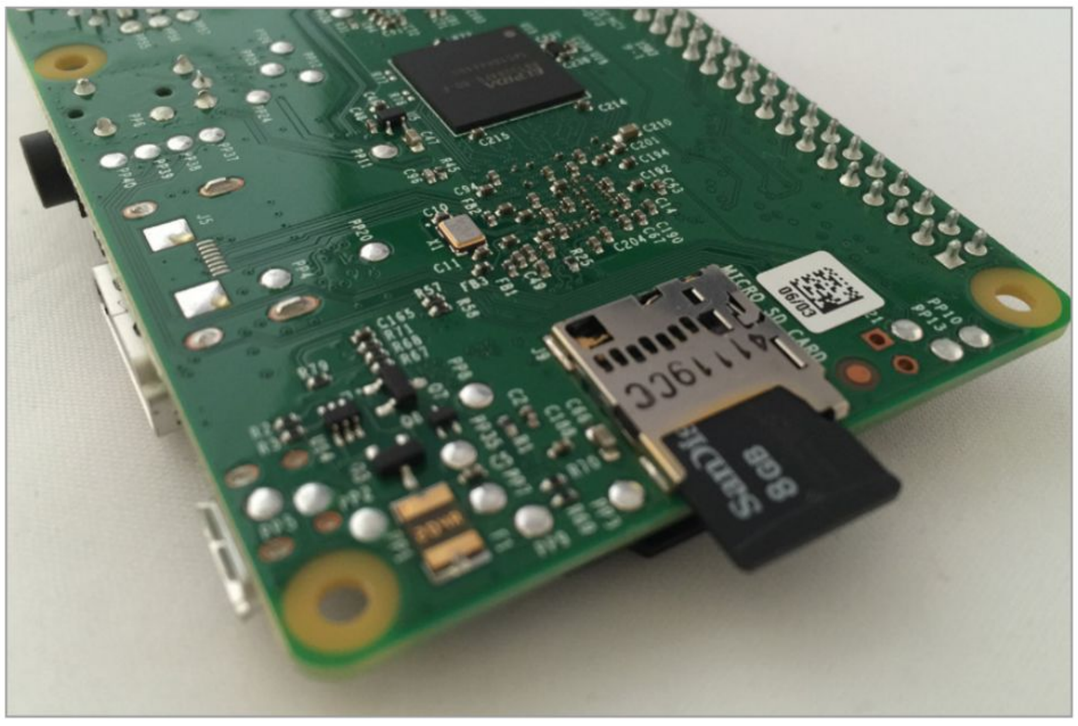
The Raspberry Pi is as bare bones as a computer can get. You get a motherboard, but not much else. So you'll need a few extras to get it up and running but they're things you're likely to have or at least find it easy to get your hands on.

ASK AROUND

The kit list required to set up a Raspberry Pi is pretty basic: keyboard, mouse, HDMI monitor, SD Card and an optional case. Many of these items you'll already have, but don't rush out and buy those you don't. Ask around to see if anybody you know has spares.

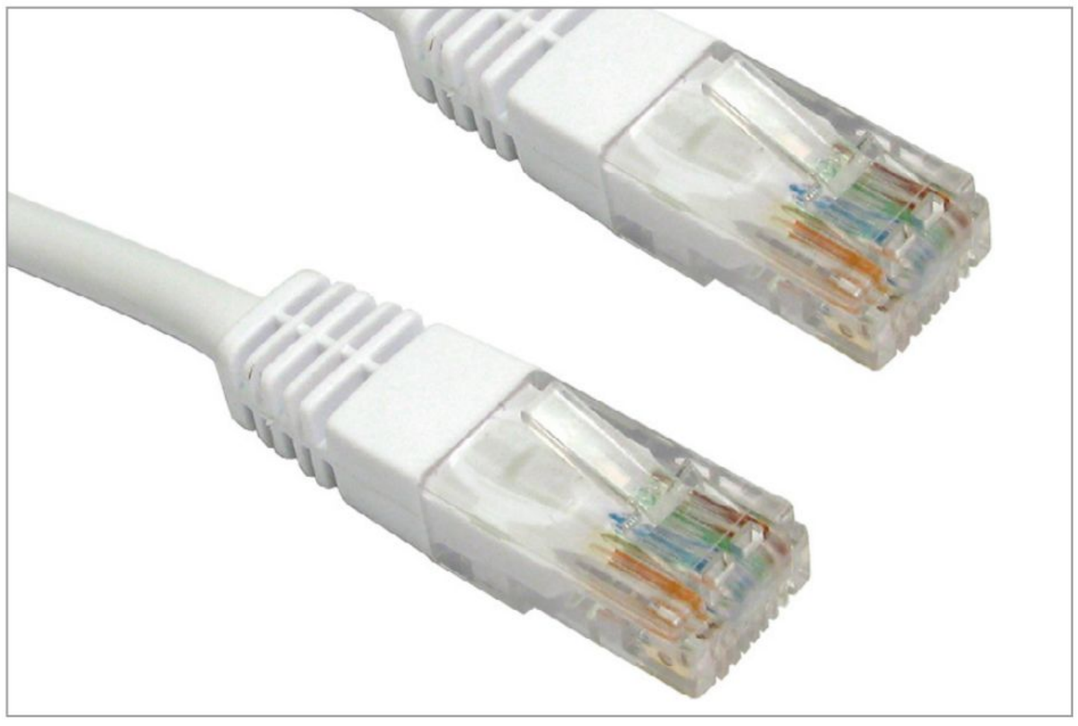
STEP 1 The most important thing you need is an SD Card. These are the storage cards commonly found in digital cameras. All modern Raspberry Pi boards, including the Raspberry Pi 3 and Pi Zero, use Micro SD cards but the older boards may use a larger SD Card.

STEP 2 The SD Card plugs into the SD Card socket. On the Raspberry Pi 4, 3, and Pi Zero, you push the card in and pull it out. On some older models, you push the SD Card in again to eject it. All SD Cards have a notch on one corner that ensures it only goes in one way round. Unless you bought a card with NOOBS preinstalled, you'll need to install the software on it first.



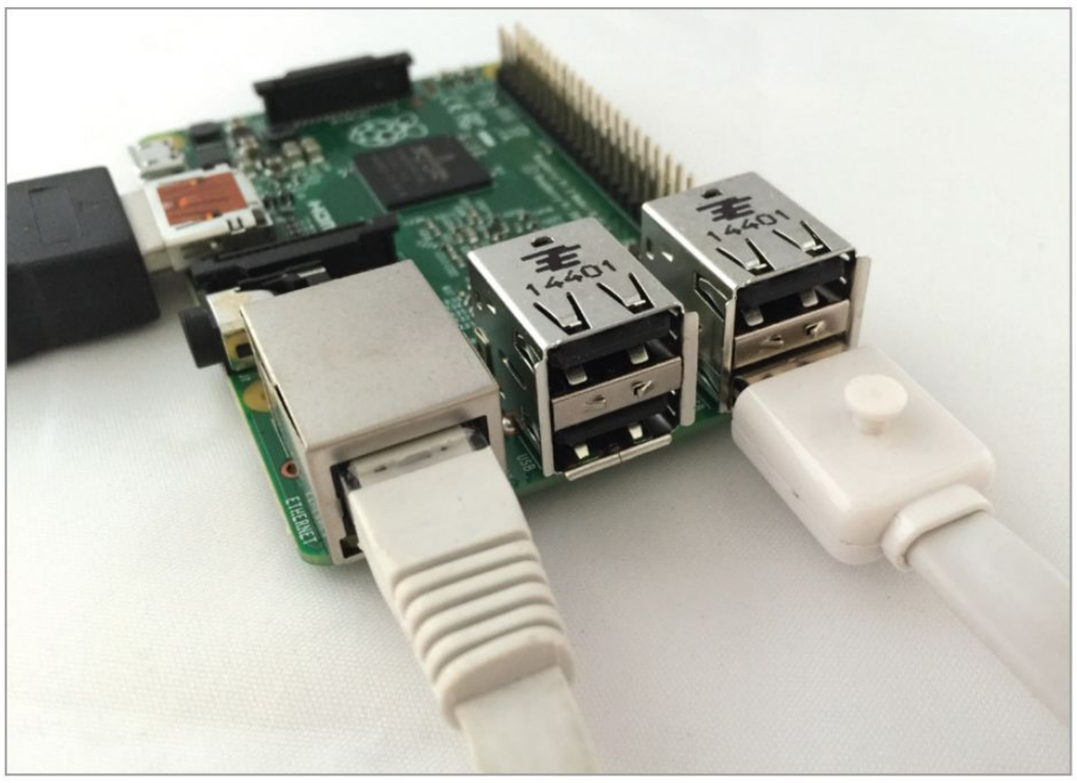
STEP 3 You'll need a HDMI cable to connect your Raspberry Pi to a monitor or television set; these cables are used with modern televisions and video game consoles, so you shouldn't find it too hard to pick up a spare. If you own a Pi Zero, or you've got a Pi 4, you'll need a micro-HDMI cable in order to connect to a monitor or TV.

STEP 4 You'll want to connect your Raspberry Pi to the Internet. The Raspberry Pi 4 and Pi 3 Model B+, have built in wireless networking and Ethernet. The Model A+ and Raspberry Pi 2 include an Ethernet socket, but the Pi Zero and older boards will require either an Ethernet adaptor or a Wi-Fi dongle to get online.



STEP 5 Both the Pi 4 and 3 come with four USB sockets but if you own a Pi Zero, you may need to purchase a USB Hub. These connect into the USB socket, and provide four (or more) USB connections. Typically, you need to connect a keyboard and a mouse, so at least two USB ports are required.

STEP 6 Two things that should be easy to find are a USB Keyboard and USB Mouse. We routinely prefer USB devices that plug directly into the Raspberry Pi, but devices with wireless dongles generally work just as well. Only the Raspberry Pi 4 and 3 Model B+ come with Bluetooth as standard, so you're best bet is to opt for USB, if possible.




You can also choose to house your Raspberry Pi in one of many enclosures like the official one pictured here.

GETTING EVERYTHING CONNECTED

Now that you've got all the basics you'll need to get everything set up. If you haven't installed the NOOBS files on your SD Card, then it's best to do that first (see the next few pages).

STEP 1 Connect your Raspberry Pi to the monitor using the HDMI cable. Now attach the keyboard and mouse to separate USB ports. If your Raspberry Pi only has one USB port, connect the mouse and keyboard to the USB hub and attach it.

STEP 2 Now, if required, connect the Ethernet cable to the Raspberry Pi. Connect the other end of the cable directly into your modem/router or into a network socket.

STEP 3 Finally connect a Micro USB cable to the power supply and attach the other end to a 5V USB power adapter. This is the type used to power most modern smartphones, so you should be able to source a spare. Press the On button to power up your Raspberry Pi.

STEP 4 Consider a case to hold your Raspberry Pi together. Raspberry now makes an official case that you can use to hold all the components and protect it from knocks and drops.

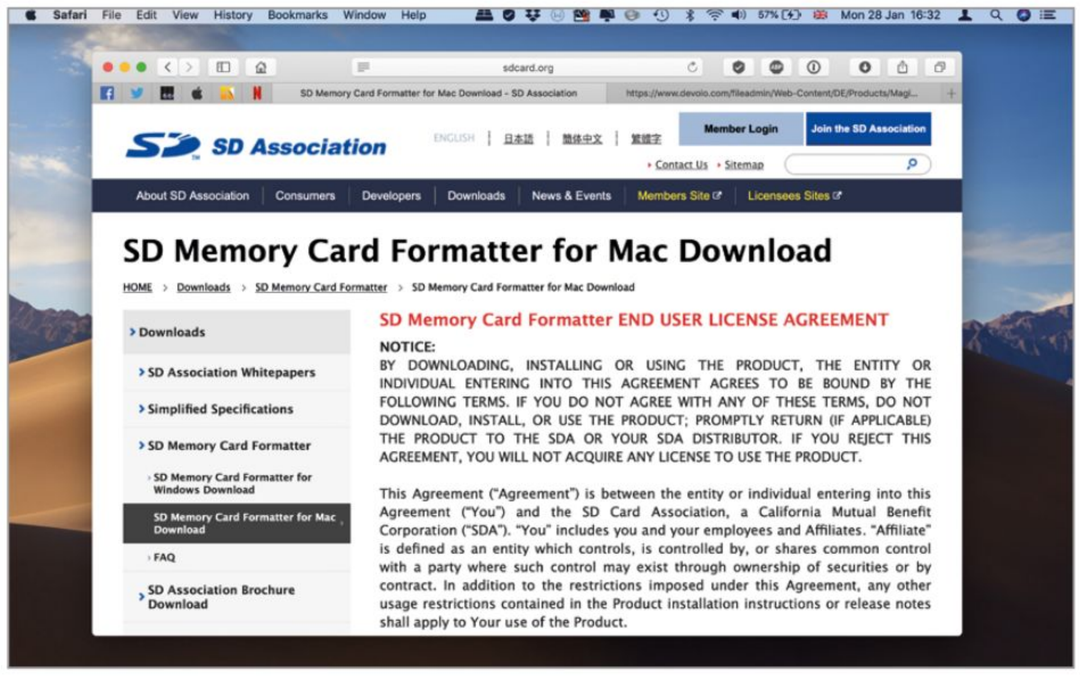
Set Up Raspberry Pi Using a Mac

You should install NOOBS (New Out Of Box Software) on your SD Card before you go any further. This feature looks at how to format an SD Card and copy the software using an Apple Mac computer.

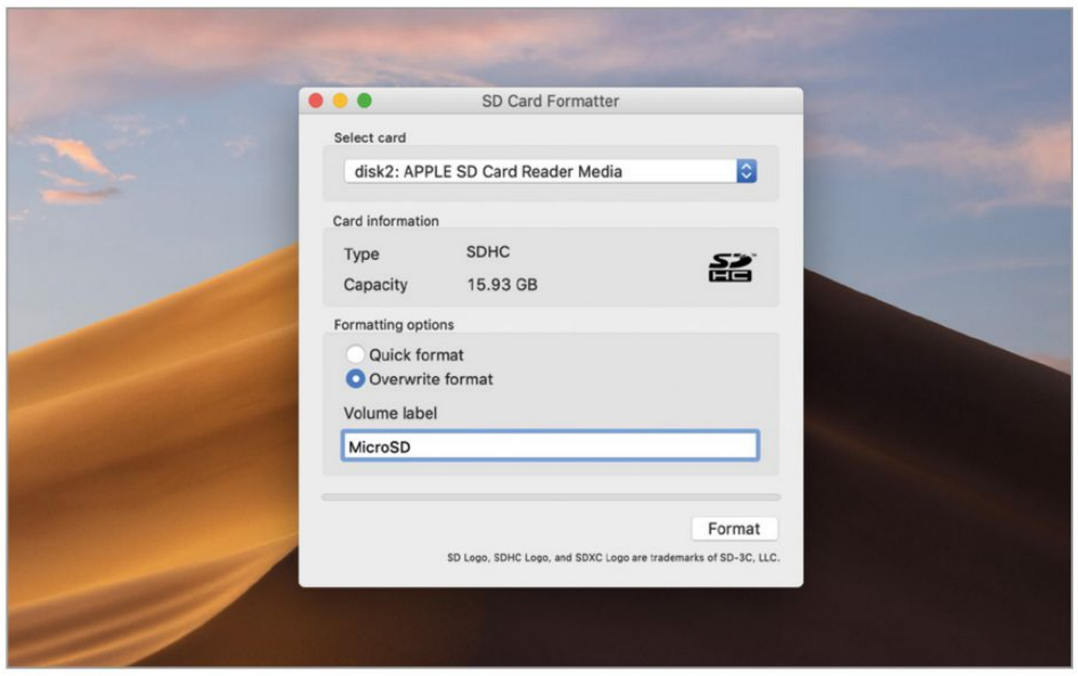
GETTING TO KNOW NOOBS

The easiest way to get up and running is to use NOOBS, a software program created by the Raspberry Pi Foundation. You can buy a NOOBS SD Card from Raspberry Pi but it's easy to make your own with an old unused SD Card (8GB recommended).

STEP 1 We're going to use a utility called SD Card Formatter 5 to erase the SD Card with correct formatting. You can download this app from **https://www.sdcard.org/downloads/formatter_4/**. Click Download SD Formatter for Mac and Accept. Click the SDFormatter.pkg file in your Downloads folder and follow the instructions to install the app.

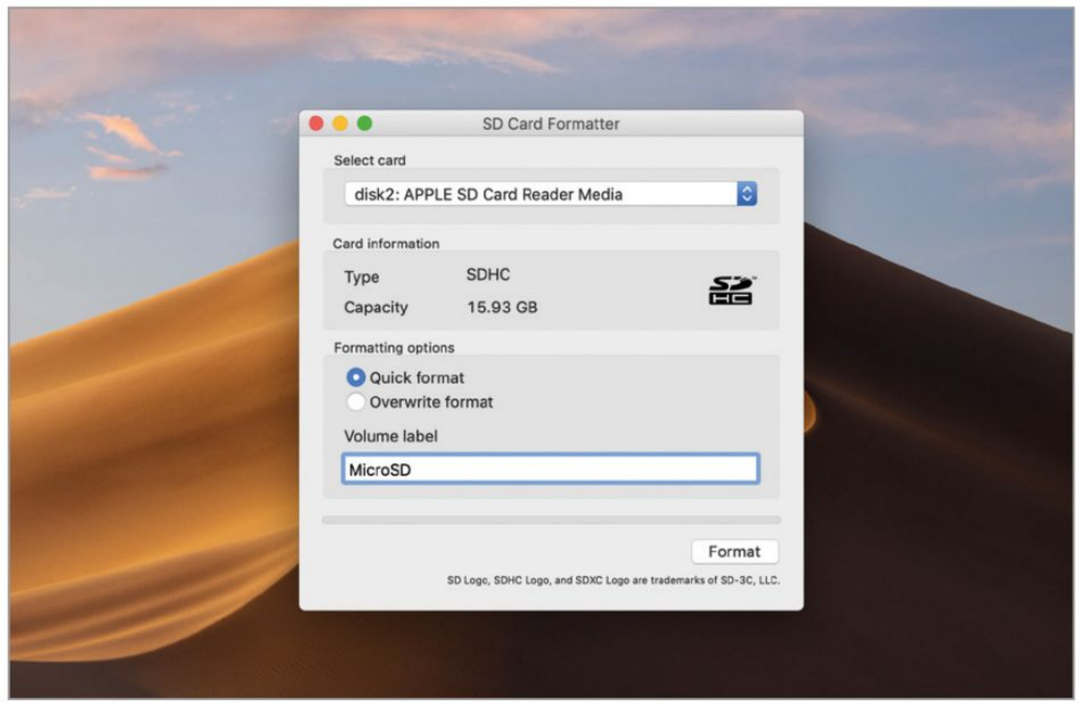


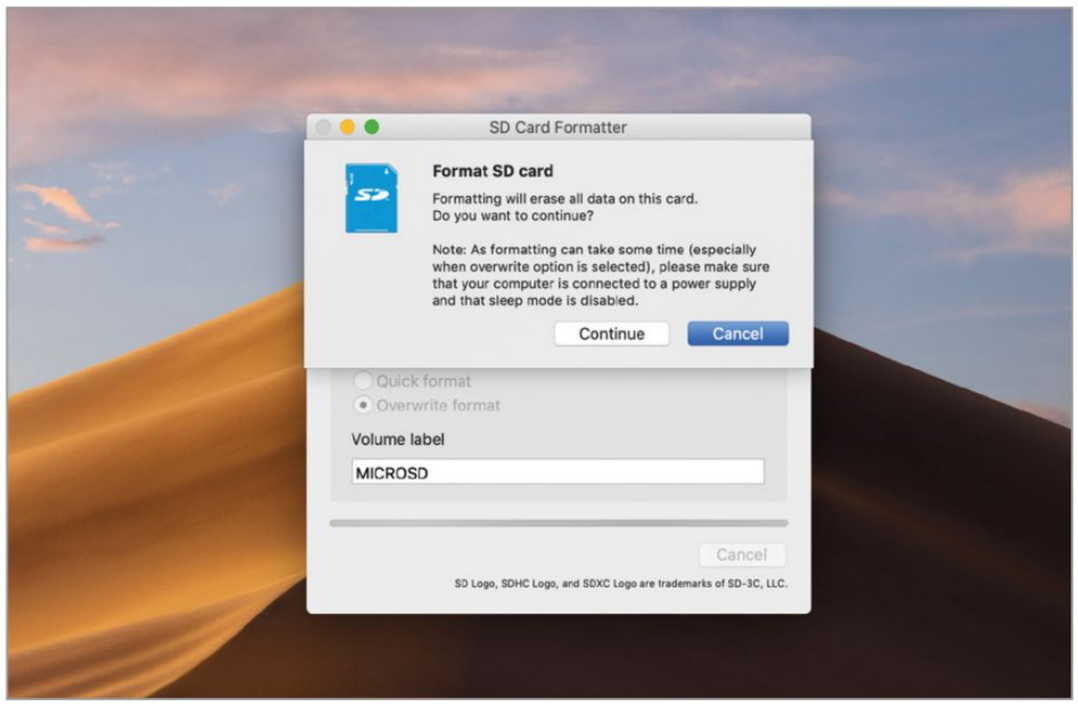
STEP 2 Use an SD Card of at least 8GB capacity to install the operating system on your Raspberry Pi. The card needs to be formatted, which will erase all the data on it, so make sure you have copied any files from it you want to save. Insert the SD Card into your Mac, either directly into the SD Card slot or with an SD Card adaptor.



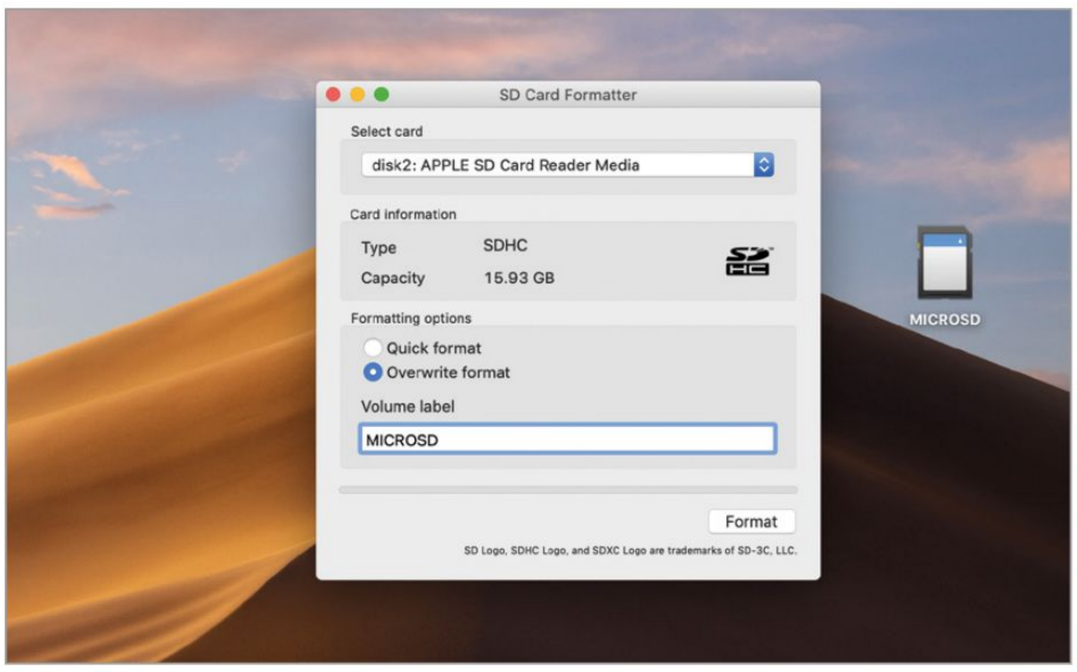
STEP 3 Make sure the card is present in the Select Card area; you should only have one SD Card inserted into your Mac. Open SDFormatter and choose the Overwrite Format option to ensure that all the old data is removed from the card. Enter a name for the card in the Name field to help you identify it; although this isn't necessary for the installation process.

STEP 4 Click the Format button when you are ready to wipe the card. There are other ways to format SD Cards in macOS (in particular Disk Utility) but the advantage of SD Card formatter is that it wipes SD Cards with the correct FAT32 format and doesn't affect the protected partition. It can take a while to format the SD Card, but the progress bar will show you how the process is going.

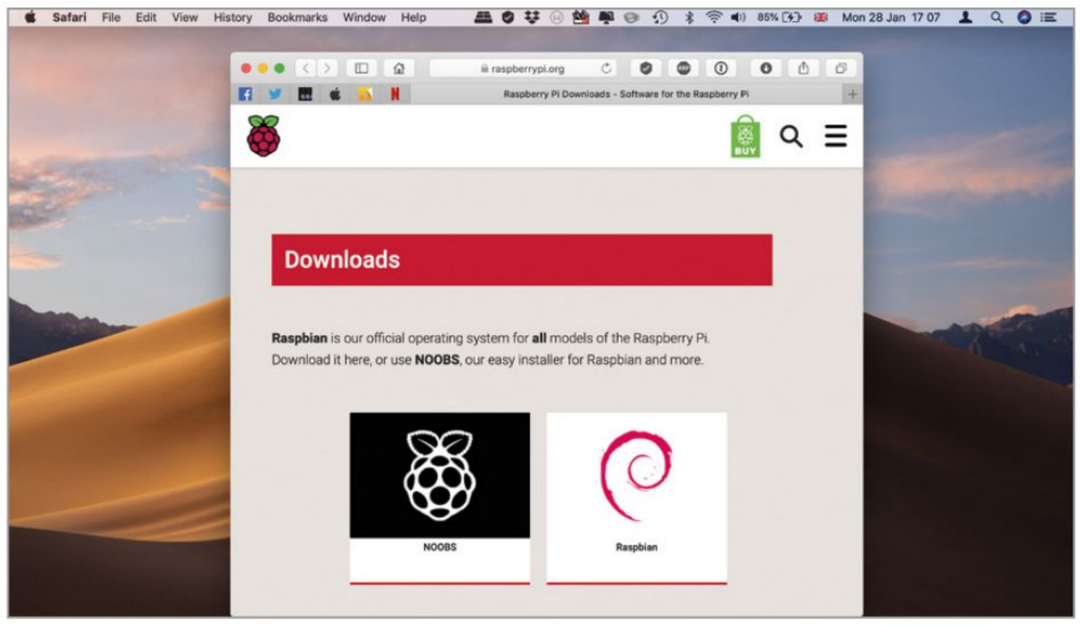




STEP 5 When SD Card formatter has finished formatting the SD Card it will be mounted so you can access it. By default mounted volumes appear as an icon on the desktop. If not open a new Finder window and check for it under Devices. Check that the SD Card is accessible and click Close in SDFormatter.



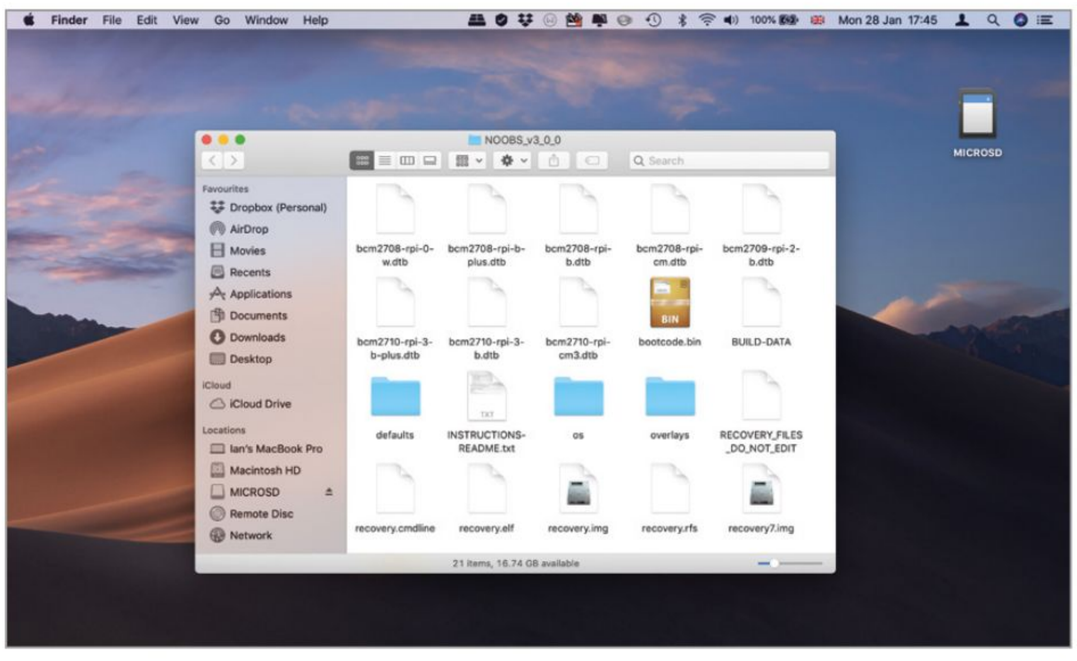
STEP 6 Now it's time to download the NOOBS software from the Raspberry Pi website. Open Safari and enter **www.raspberrypi.org/downloads/** into the Smart Search Field. Scroll down to find the NOOBS section (not NOOBS LITE) and click Download Zip. A zip file containing the NOOBS files will be placed in your Downloads folder.



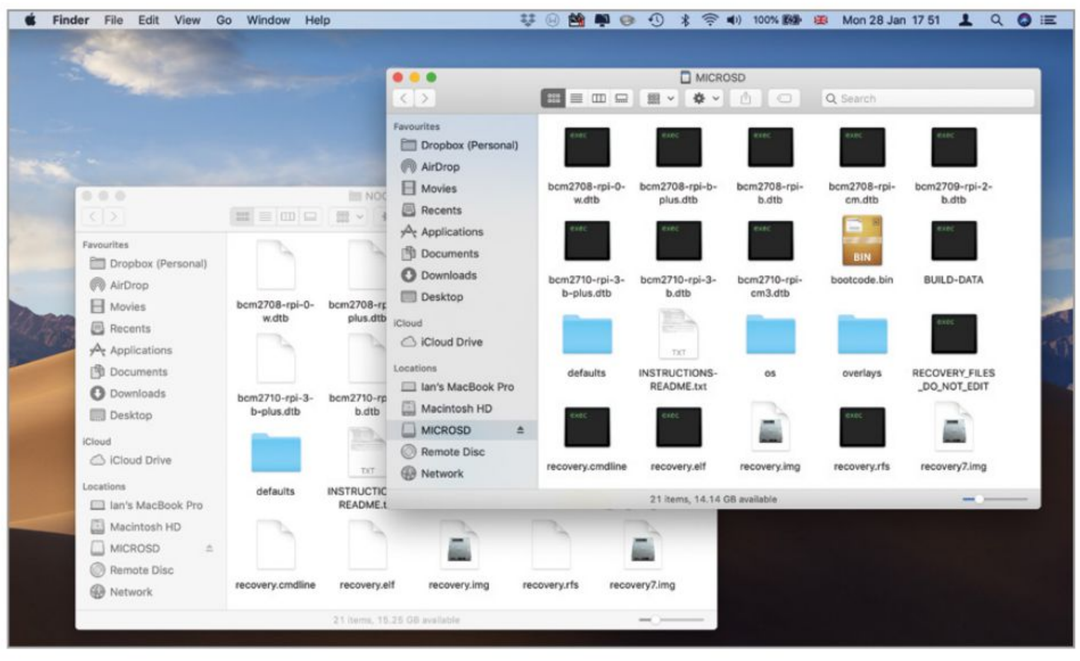
COPY THE SOFTWARE

Transfer the NOOBS files to your SD Card to continue the setup process.

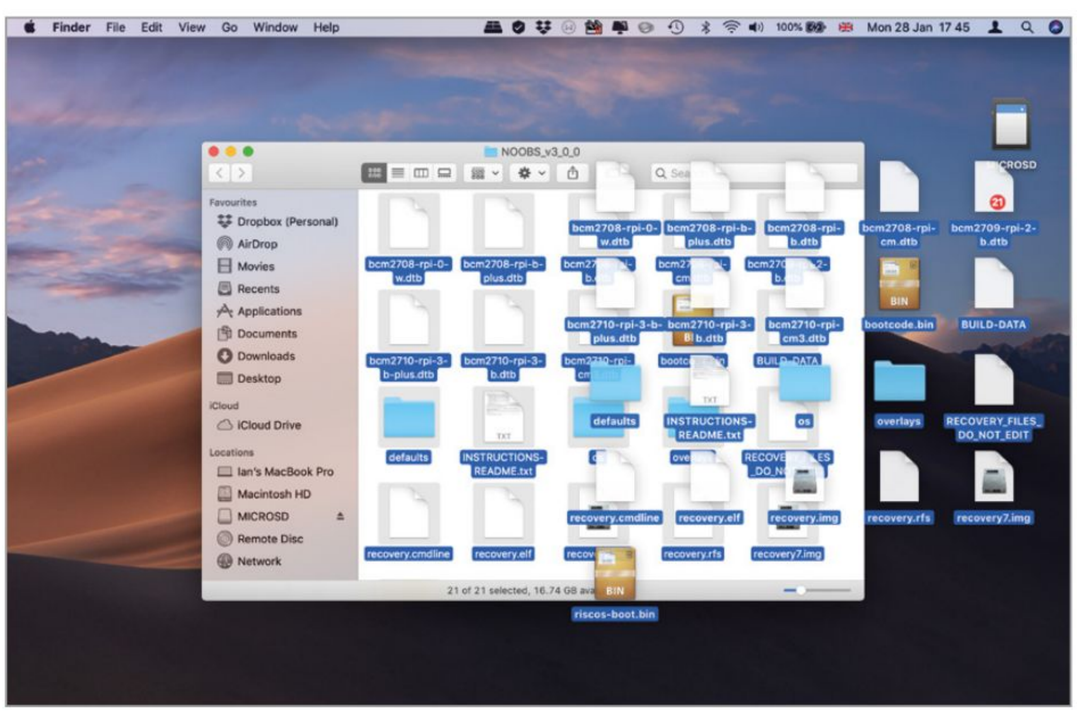
STEP 1 Open the Downloads folder and click on the NOOBS zip file to unzip it. A NOOBS folder should appear in your downloads; it will be marked with the version number you downloaded from the Raspberry Pi Downloads page. Click on the folder to open it and view all the files contained inside.



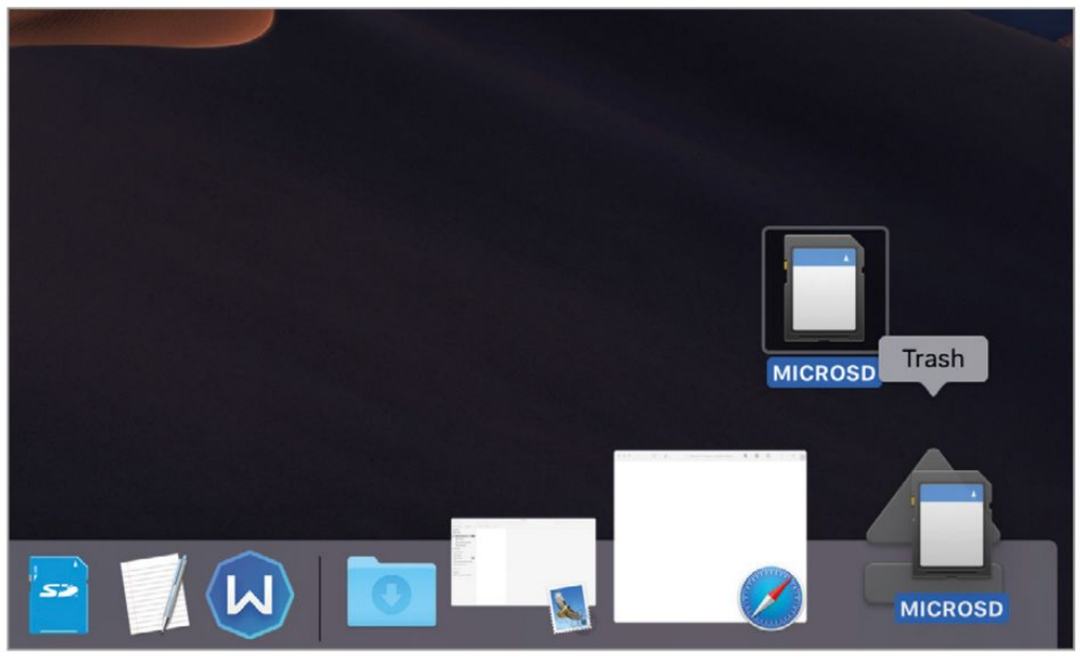
STEP 3 Wait for all of the files to be copied from your Downloads folder to the SD Card. After the files have finished copying open the SD Card and check that all of the files are in the root. You should see "bootcode.bin" and "BUILD-DATA" files, and a "defaults" folder among other files.



STEP 2 Press Command-A to select all the files inside the NOOBS folder. Now drag and drop all the files from the NOOBS folder to the SD Card. This will copy all of these files to the root (the base) of the SD Card. Make sure you copy the files and not the NOOBS folder containing them.



STEP 4 Make sure you eject the SD Card properly. Do not just remove it from the Mac. Instead, drag the SD Card icon to the Trash to eject it from your system. Or open a new Finder window and locate the SD Card under Devices, click the small Eject icon next to it. Now remove the SD Card from your Mac. It is ready to be inserted into your Raspberry Pi.



Set Up Raspberry Pi Using a Windows PC

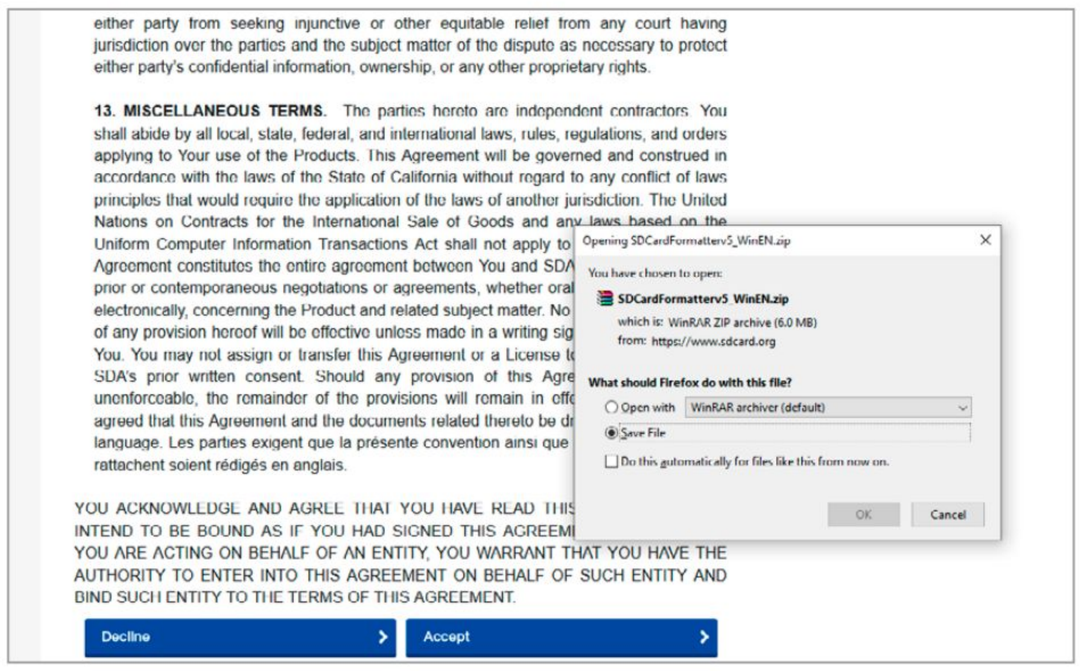
It's easy to set up your Raspberry Pi with a Windows PC by downloading and copying NOOBS. This installs a selection of OS's onto the Raspberry Pi. So let's format your SD Card and install NOOBS using a Windows PC.

SETTING UP WITH NOOBS

NOOBS (New Out Of Box Software) is a program used to make setting up a Raspberry Pi simple. You can buy SD Cards with NOOBS preinstalled, but here we'll look at how to format your SD Card and install NOOBS using a Windows computer.

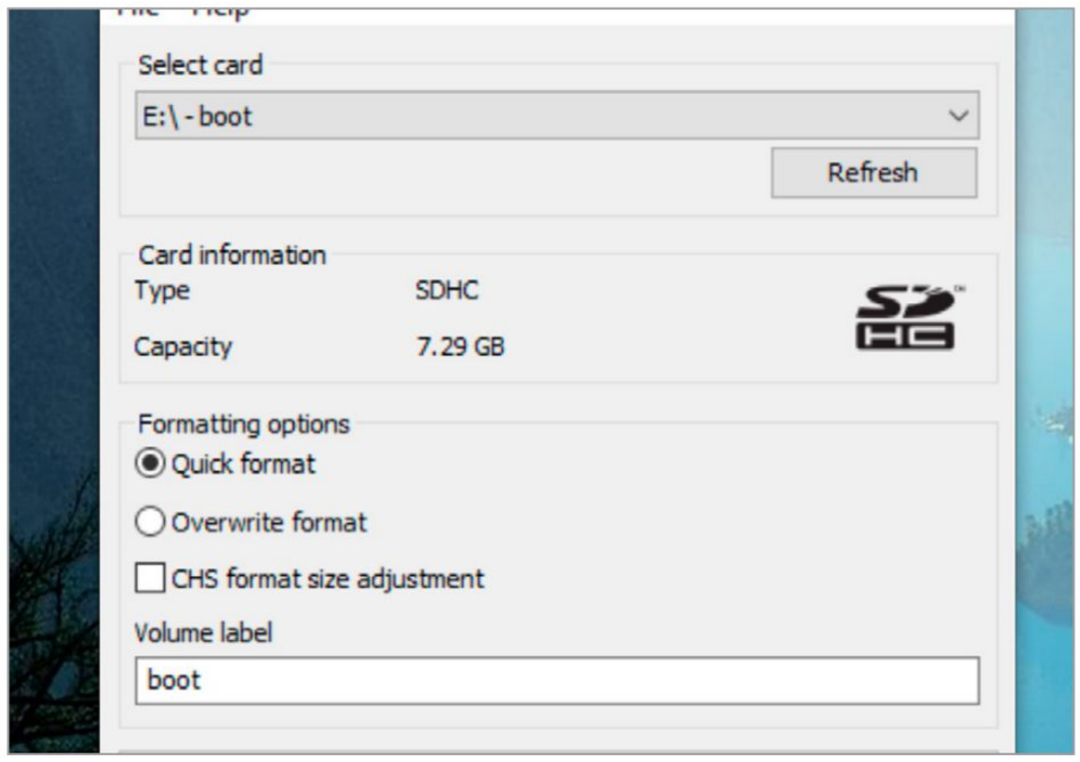
STEP 1

We're going to use a utility called SD Card Formatter to erase the SD Card with correct formatting. You can download this app from **https://www.sdcard.org/downloads/formatter/eula_windows/index.html**. Click the Accept button to start the download of the latest version of the software. Extract the software, and double-click the executable to install and run the app.



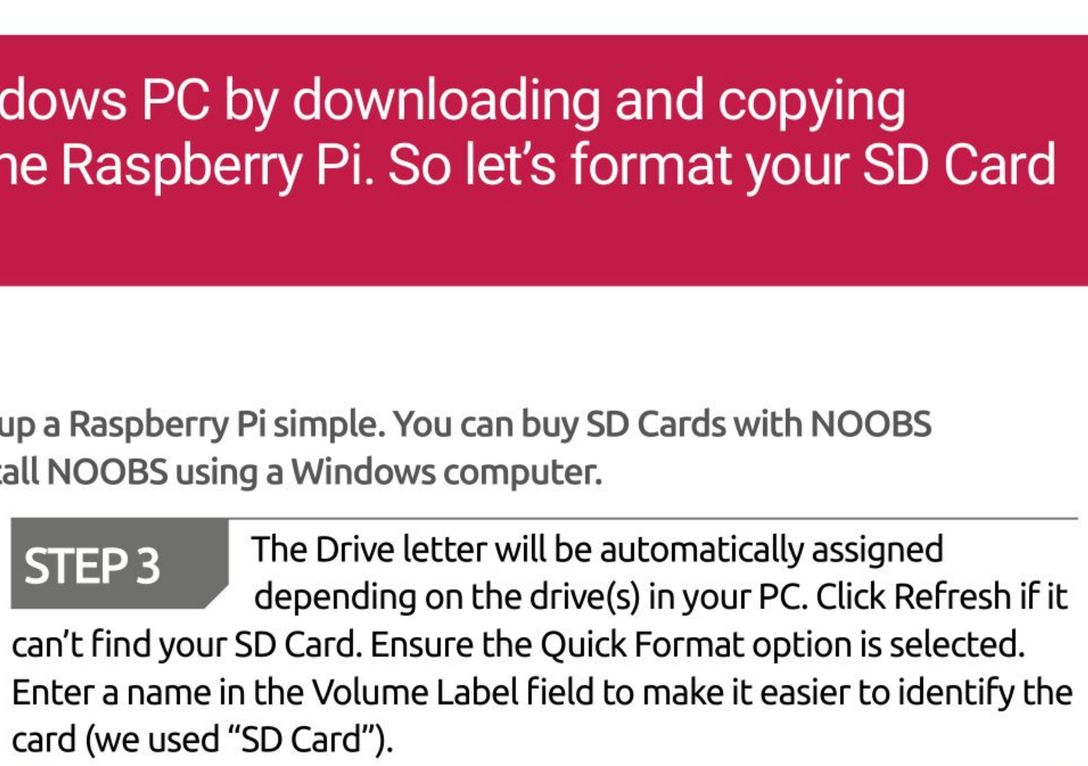
STEP 2

We're going to use an 8GB SD Card to install the operating system on our Raspberry Pi. The card needs to be formatted, which will erase all the data on it, so make sure you have copied any files from it you want to save. Insert the SD Card into your PC, either directly into the SD Card slot or using an SD Card adaptor.



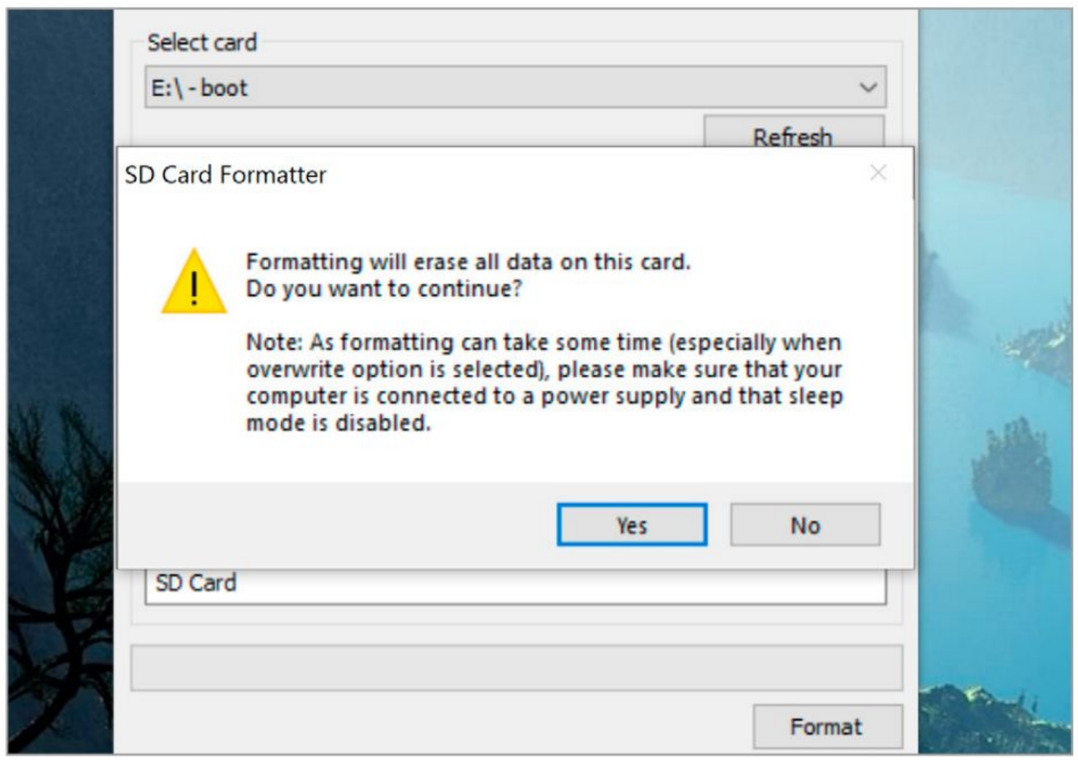
STEP 3

The Drive letter will be automatically assigned depending on the drive(s) in your PC. Click Refresh if it can't find your SD Card. Ensure the Quick Format option is selected. Enter a name in the Volume Label field to make it easier to identify the card (we used "SD Card").

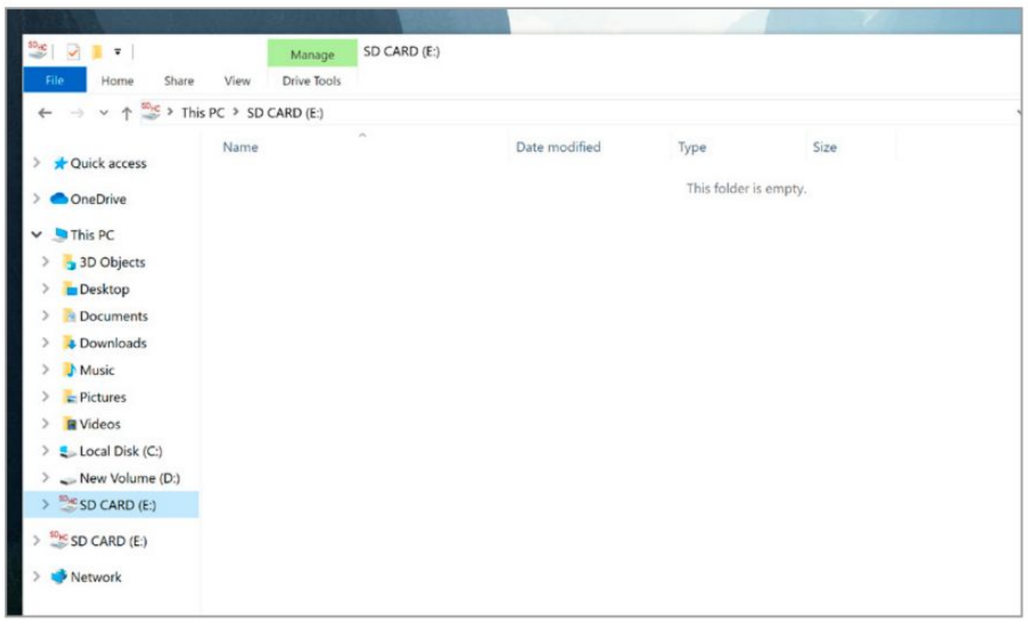


STEP 4

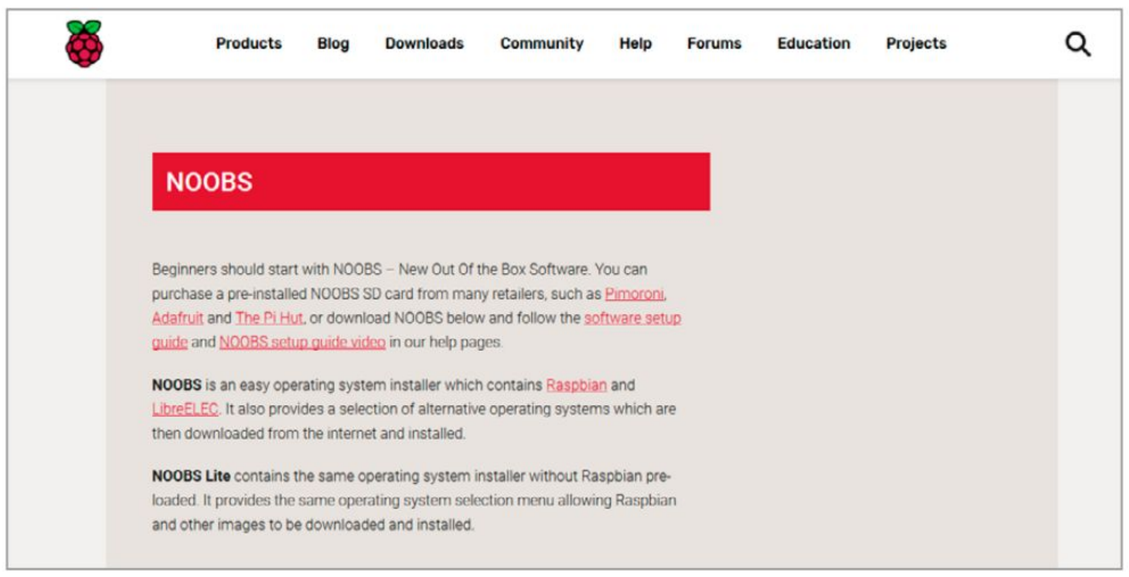
Click the Format button and Yes when you are ready to wipe the card. There are other ways to format SD Cards in Windows, but the advantage of SD Card formatter is that it wipes SD Cards with the correct FAT32 format, and doesn't affect the protected partition. Click OK again, when the format is complete.



STEP 5 Click Exit to close down the SDFormatter program. Click Start > File Explorer and choose the SD Card in the sidebar. The drive should be empty, but we will copy the files to it in the next steps. For now just make sure that you can access the empty root of the SD Card.



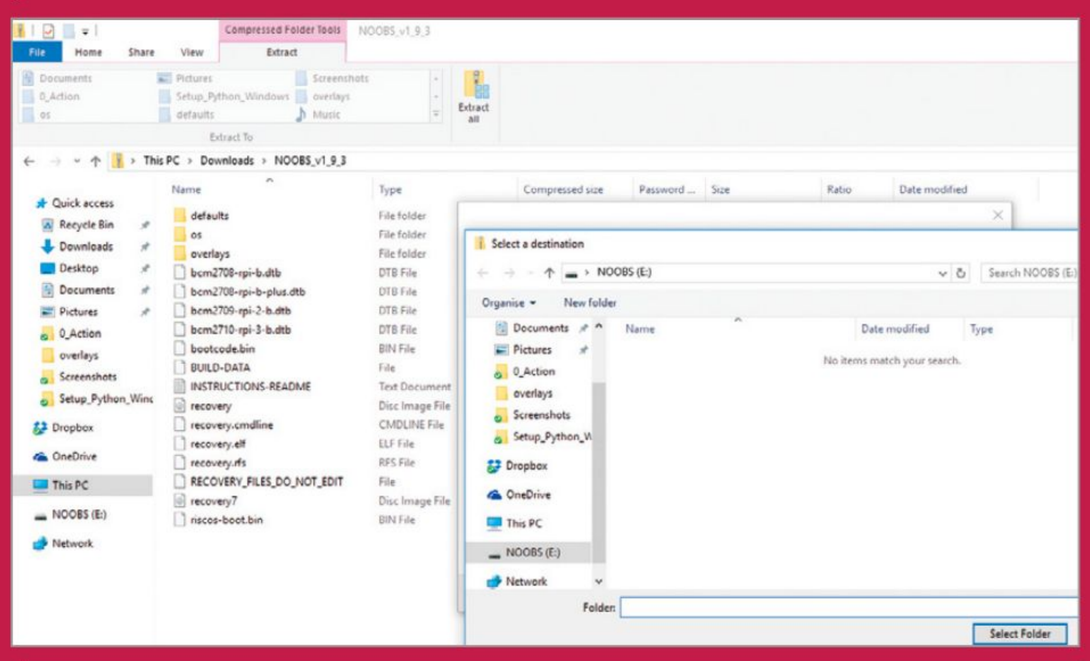
STEP 6 Now it's time to download the NOOBS software from the Raspberry Pi website. Open a web browser and enter **https://www.raspberrypi.org/downloads/noobs/** into the URL field. Scroll down to find the NOOBS section (not NOOBS LITE) and click Download Zip and Save. A zip file containing the NOOBS files is copied to your Downloads folder. Click Open Folder when it is done.



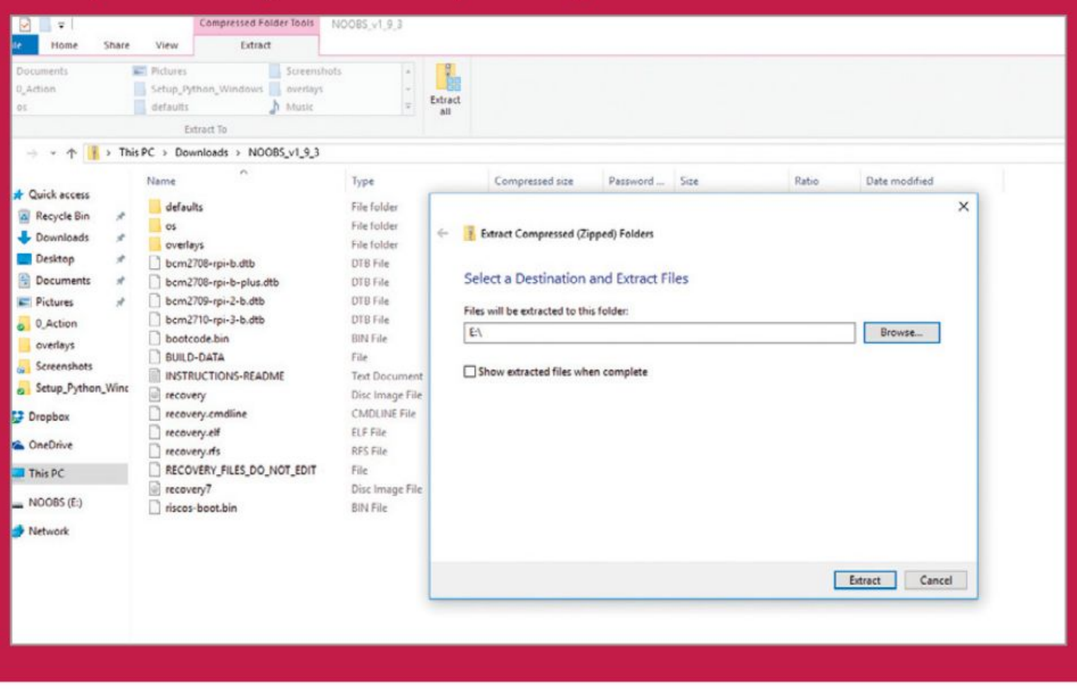
COPY THE SOFTWARE

Continue making your SD card by copying the files across.

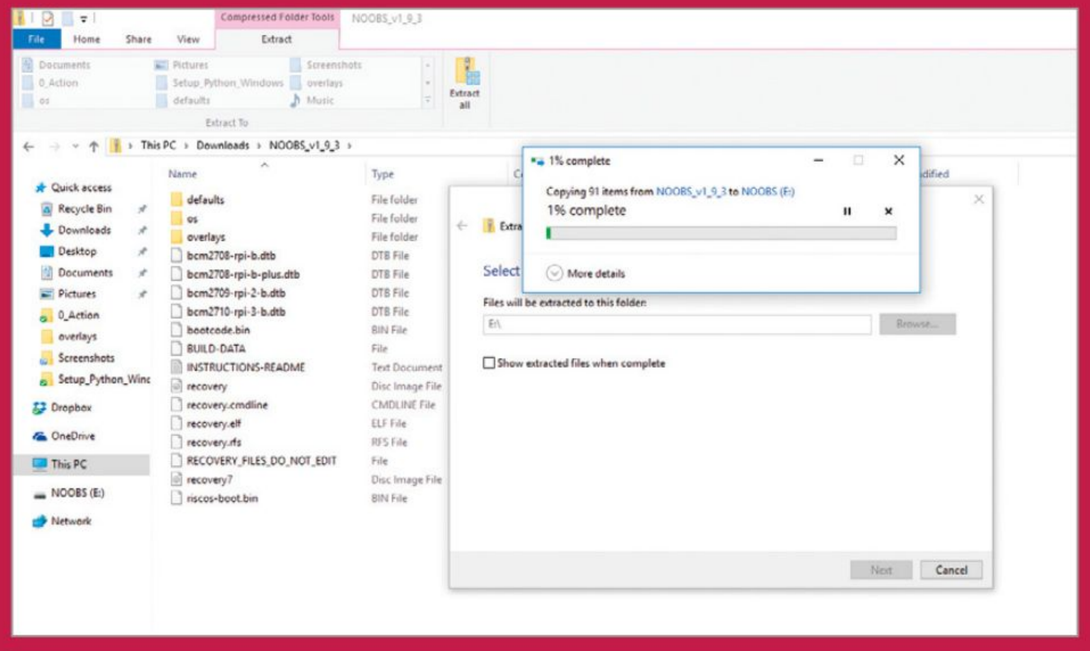
STEP 1 A NOOBS folder should appear in your downloads. It will be marked with the version number that you downloaded from the Raspberry Pi Downloads web site. Double-click the NOOBS zip file in your Downloads folder to view the contents. These are the files you need to copy to the root (base) of your SD Card.



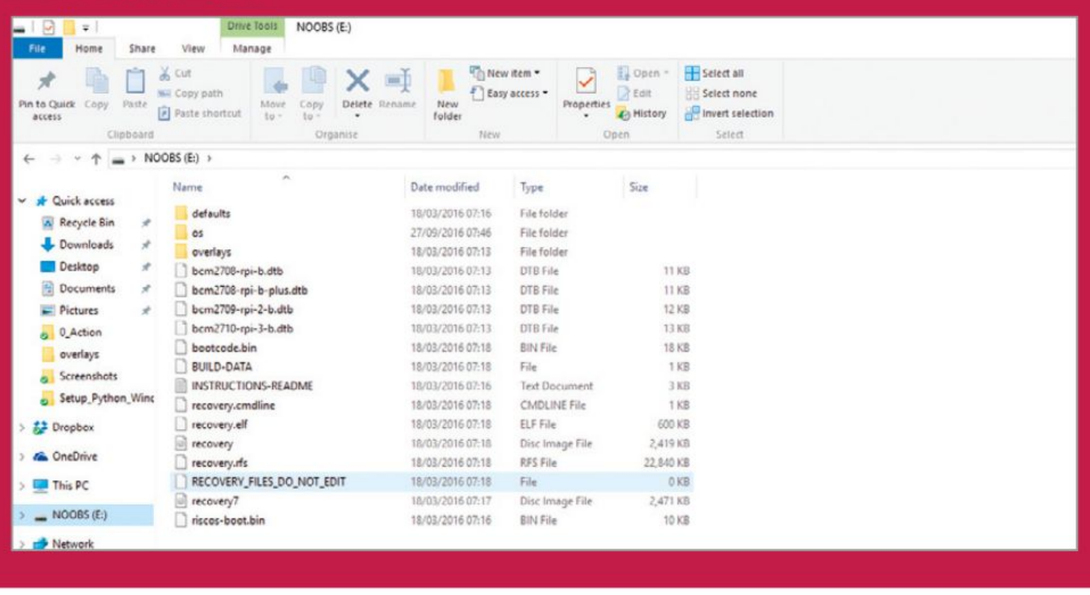
STEP 2 Click Extract All to expand the zipped file and extract its contents. Click Browse in the Select a Destination and Extract Files folder and choose "SD Card(F:)" in the Select a Destination window. Click Select Folder and Extract. The files will now be copied directly from the zip file to your SD Card.



STEP 3 Wait for all of the files to be copied from your Downloads folder to the SD Card. The files that are copied will install an operating system of your choice on the Raspberry Pi when you first boot it up. It is important that the files are copied to the root (base) of the SD Card and are not inside another folder such as the NOOBS folder.



STEP 4 Click Start > File Explorer and choose "SD CARD (F:)" in the sidebar to display the contents. It should now contain all the files NOOBS needs to setup an operating system on your Raspberry Pi. Check that you can see a Defaults folder, bootcode and BUILD-DATA files along with the other files shown here. Remove the SD Card from your Windows PC. It's now ready to be inserted into the Raspberry Pi.

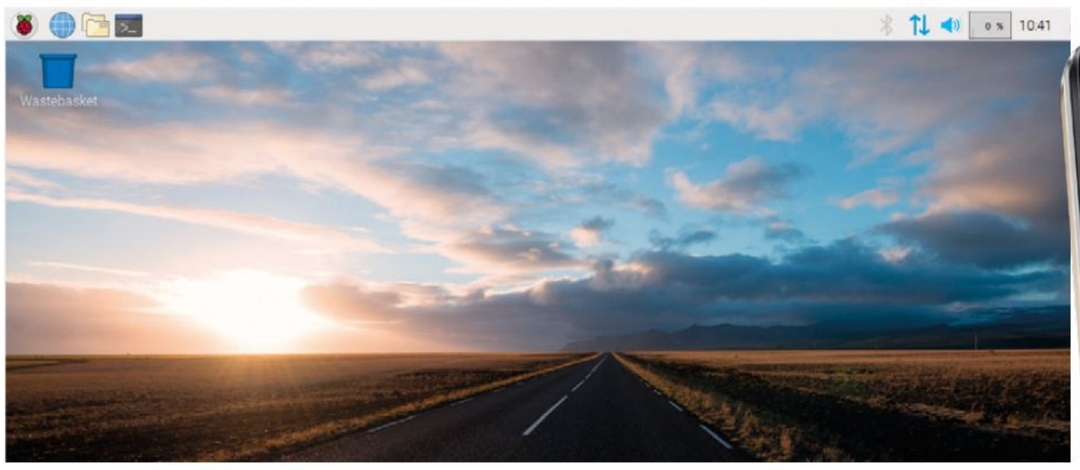


The Raspberry Pi Desktop: What You Will Need

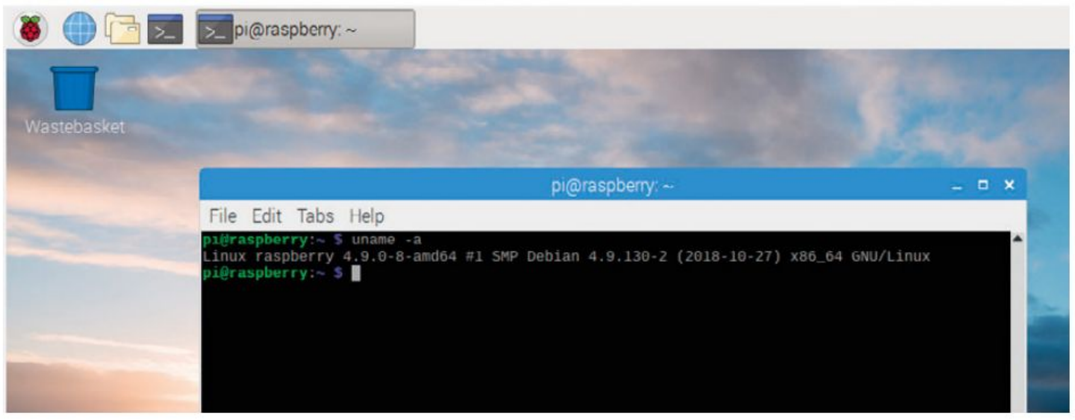
Did you know that there is a way in which you can run the full Raspbian OS desktop without even needing a Raspberry Pi? The Raspberry Pi Desktop edition is a fully working, Debian-based installation of Raspbian that can run on a standard PC.

In 2016, the Raspberry Pi Foundation started work on an x86 version of its popular Raspbian operating system. By the end of the following year, there was a link on the Downloads page and non-Pi owners could now install and use Raspbian in almost the exact same way as if they were in front of a Raspberry Pi.

Since then, the Raspberry Pi Desktop, or as it's correctly called: Debian Stretch with Raspberry Pi Desktop, has followed the same release path as its Pi-based sibling. However, there are some differences you need to be aware of prior to throwing your Pi away and opting for this version instead.



The Raspberry Pi Desktop (x86 version) allows you to view the same desktop and pre-installed software as the Full Desktop version available for the Raspberry Pi. You can install it on any decent PC, or laptop, one that's at least six or seven years old would be perfect and as a PC that age usually struggles with the latest version of Windows 10, you can now install Raspbian instead of boxing that old kit and resigning it to a life of gathering dust in the loft.



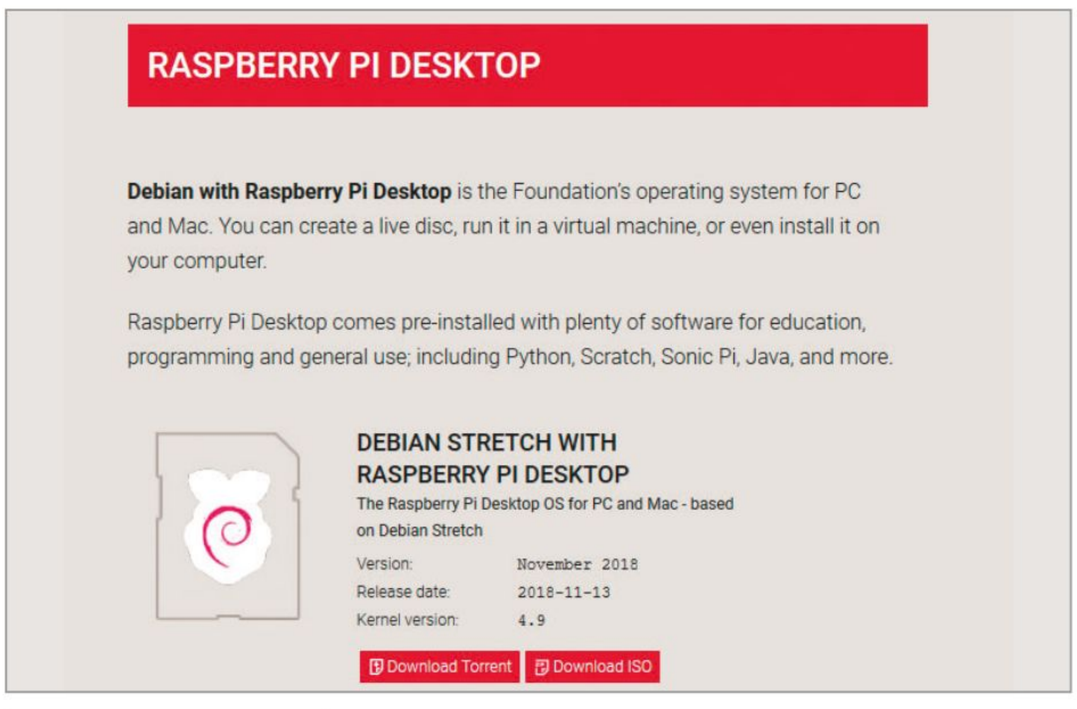
The major difference is that this version has been compiled to run on x86 processors, that's Intel and AMD PC processors. Whereas, the version of Raspbian that you will find on a Raspberry Pi has been compiled and created exclusively for ARM processors; the type of CPU a Raspberry Pi uses. These CPUs use a different architecture to that of an Intel or AMD CPU, so the software required to run on one won't run on the other unless the developer has specifically made either an x86 or ARM version. In short, if you've found a piece of software that you enjoy using on your Raspberry Pi, you won't be able to use it on the Raspberry Pi Desktop x86 version – UNLESS, there's a dedicated x86 version of the software.

Another caveat worth mentioning is that you won't have access to the 40-pin GPIO that the Raspberry Pi features, as this isn't a standard piece of hardware on a normal PC. There is, however, a way around this. If you own a Raspberry Pi Zero (either with or without the W), then you're able to connect the Zero to a spare USB port on the PC, via the Zero's micro USB port and the Raspberry Pi Desktop OS will recognise the Zero as extended hardware and allow you access to the Zero's 40-pin GPIO, which, you have to admit, is pretty clever stuff.

WHAT YOU WILL NEED

Here's a list of items needed if you want to test out the latest version of the Debian Stretch with Raspberry Pi Desktop OS on your PC.

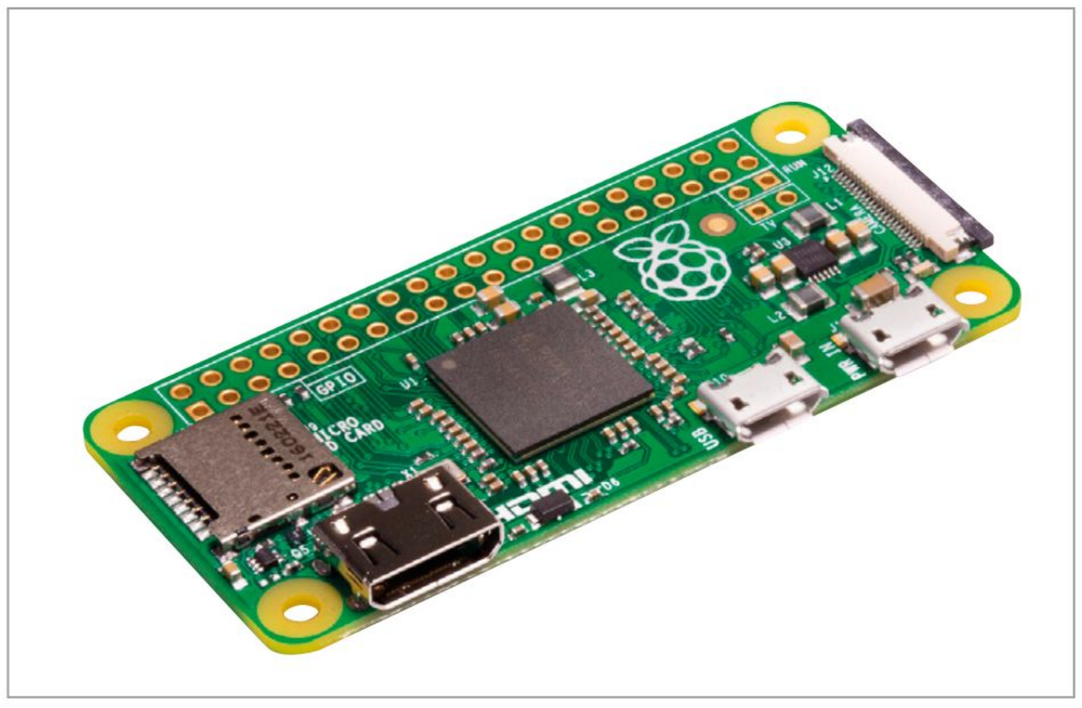
THE OS Naturally you'll need the Raspberry Pi Desktop OS, just as you would with Raspbian or the Pi. Open a browser and download the ISO file that contains the OS from: **https://www.raspberrypi.org/downloads/raspberry-pi-desktop/**.



A COMPUTER An older desktop PC or laptop is an ideal candidate for running the Raspberry Pi Desktop OS. While you can easily use the latest PC/laptop, it's a bit overkill for this particular operating system.

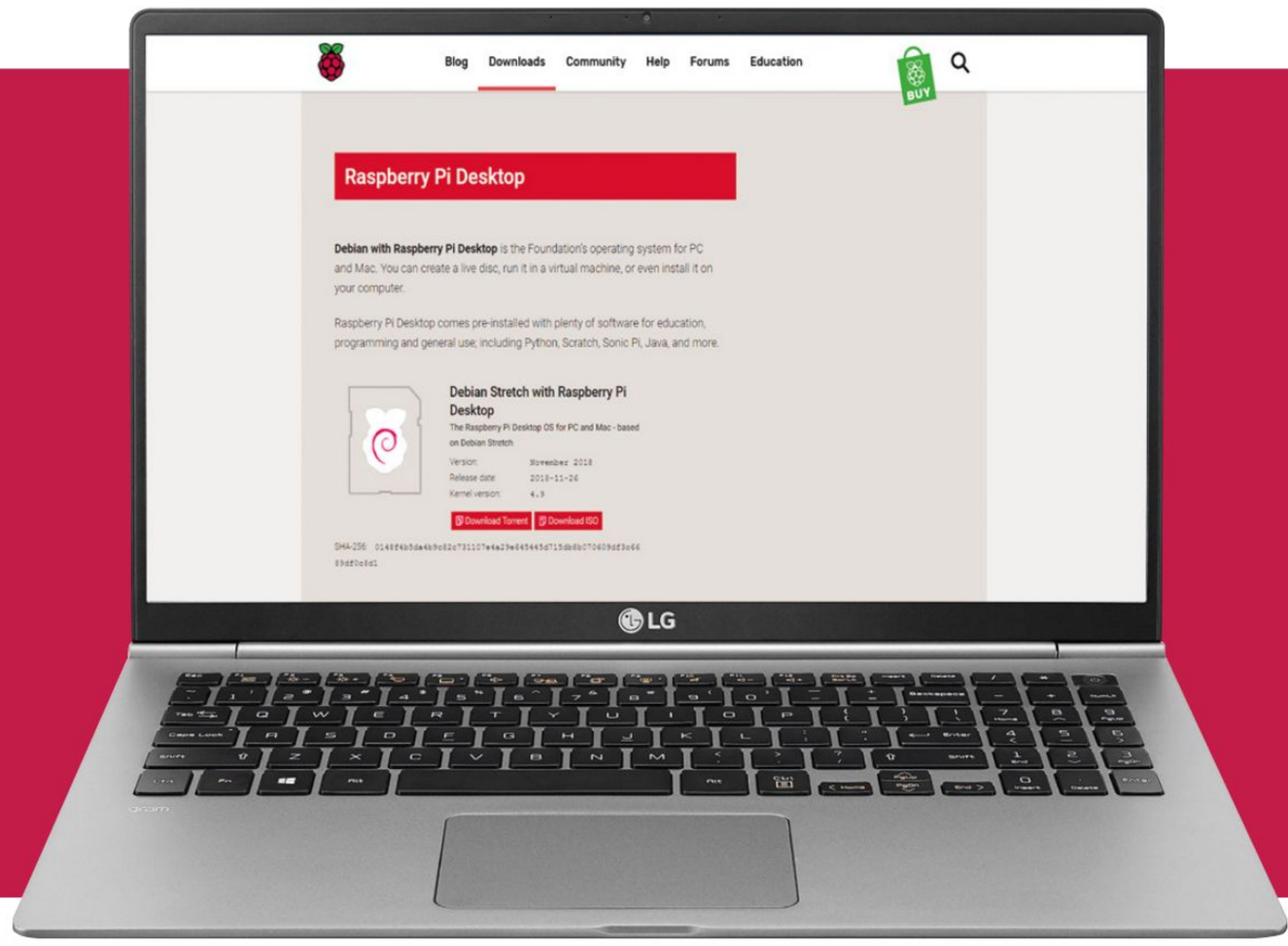
USB STICK The best method of transferring the ISO to a PC for installation is to use a combination of a tool such as Rufus (found at: https://rufus.ie/en_IE.html), which is a piece of software that can create a bootable USB driver from an ISO file and, at least, an 8GB USB stick.

RASPBERRY PI ZERO Although not strictly a necessity, a spare Raspberry Pi Zero will allow you to access the 40-pin GPIO from within the Raspberry Pi Desktop OS. If, however, you don't have a spare Zero, it's not a problem as there's still plenty you can do with Raspbian on a PC.



HOW TO SET UP THE RASPBERRY PI DESKTOP

Setting up the Raspberry Pi Desktop version on a PC or laptop is very simple, and follows roughly the same method as the previous Mac and Windows setups for the Pi. Begin by downloading the ISO image from **https://www.raspberrypi.org/downloads/raspberry-pi-desktop/**, then download Rufus from **https://rufus.ie/en_IE.html**. Insert an 8GB USB stick and run Rufus, then follow the on-screen instructions to locate the downloaded Desktop ISO. Once the image has transferred to the USB stick, remove it from the computer and insert it in the spare laptop/PC. Select **Boot from USB** from your system's BIOS/UEFI then follow the instructions to use and install the Raspberry Pi Desktop OS.



Debian Stretch with Raspberry Pi Desktop

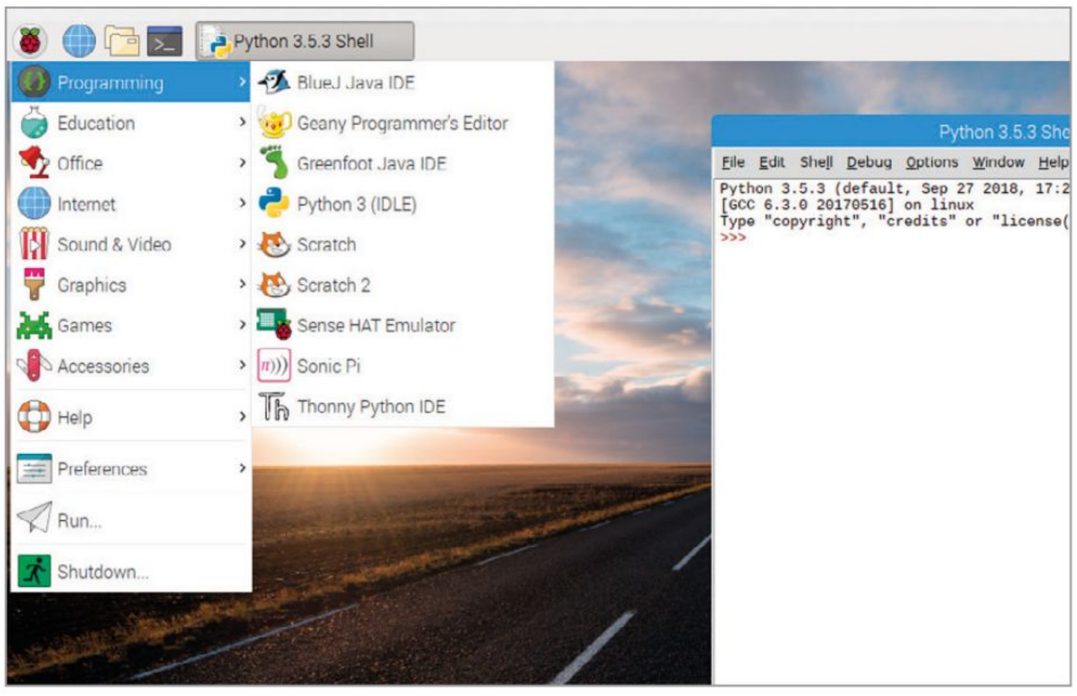
You can be forgiven for thinking that a version of Raspbian that isn't on a Raspberry Pi is somehow lacking and therefore a bit pointless in its use. However, there's a lot that you can do with the Debian Stretch with Raspberry Pi Desktop version.

10 THINGS TO DO WITH X86 RASPBIAN

You can do just as much with this version of Raspbian as with the dedicated Pi version and to demonstrate here's ten great projects to do with Debian Stretch with Raspberry Pi Desktop.

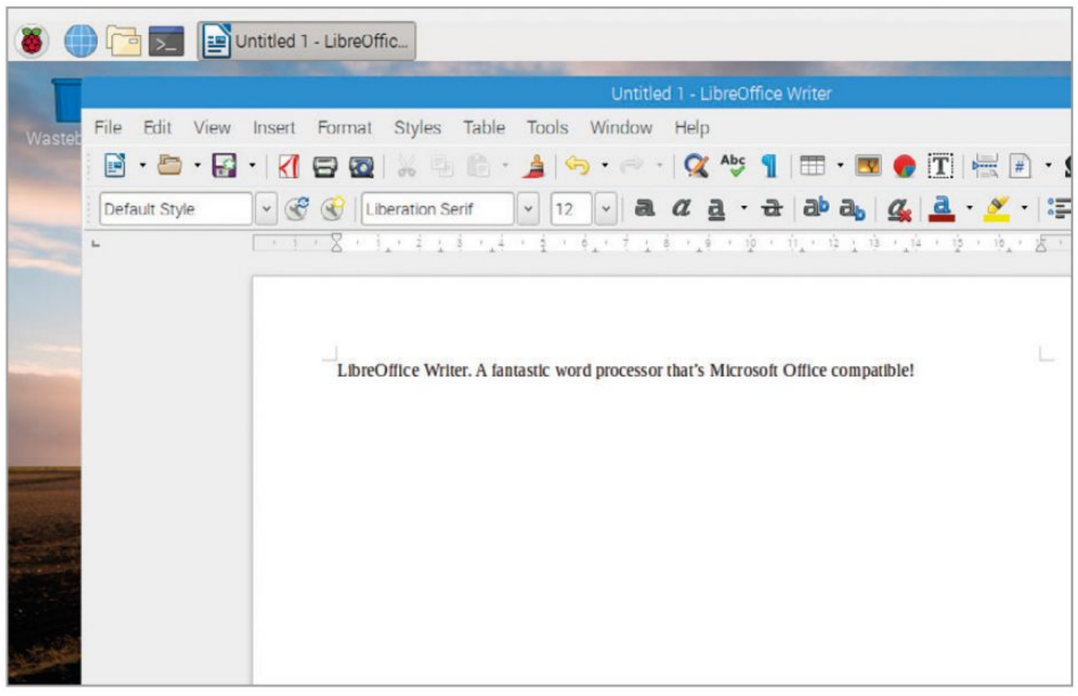
CODING

This version of Raspbian apes the Full Desktop version for the Raspberry Pi, meaning it comes with all the necessary programming languages out of the box. Learn Python, C++, use the HAT emulator and even learn JAVA. It's all there under the Programming menu option.



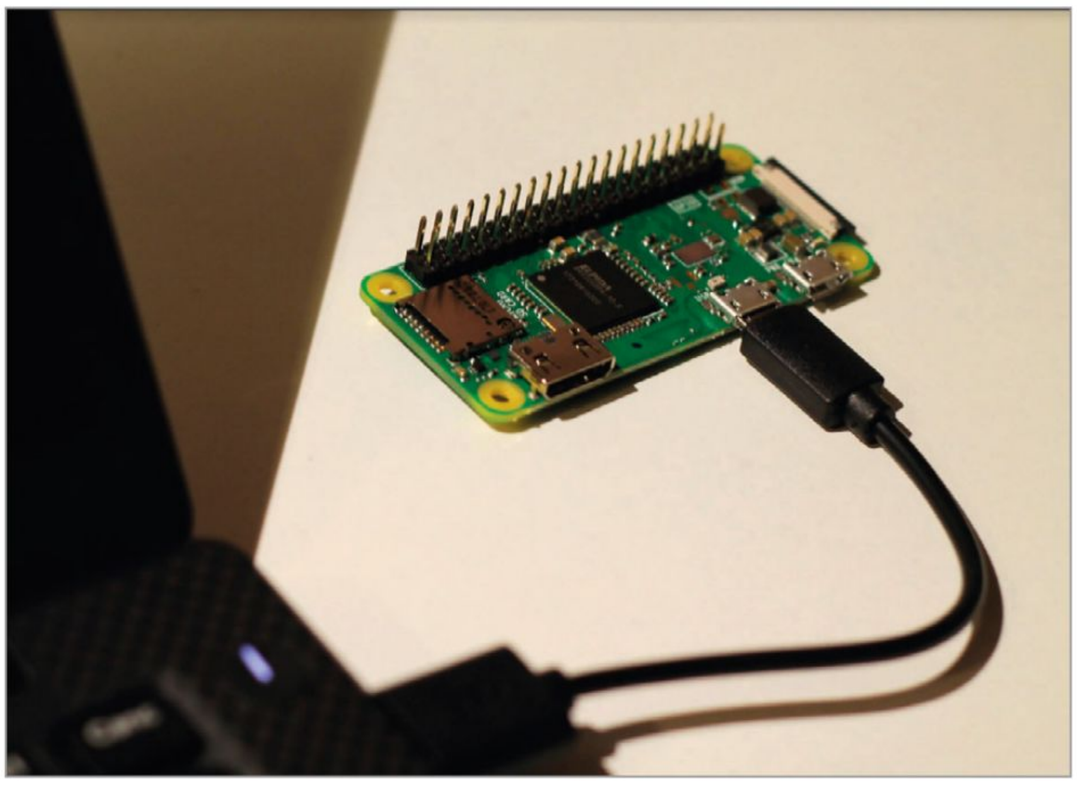
FULL DESKTOP

As the x86 version of Raspbian follows the Full Desktop Pi version, there's an entire office suite pre-installed, allowing you to use that old laptop or PC as a fully functional desktop replacement for Windows or macOS. LibreOffice is also Microsoft Office compatible.



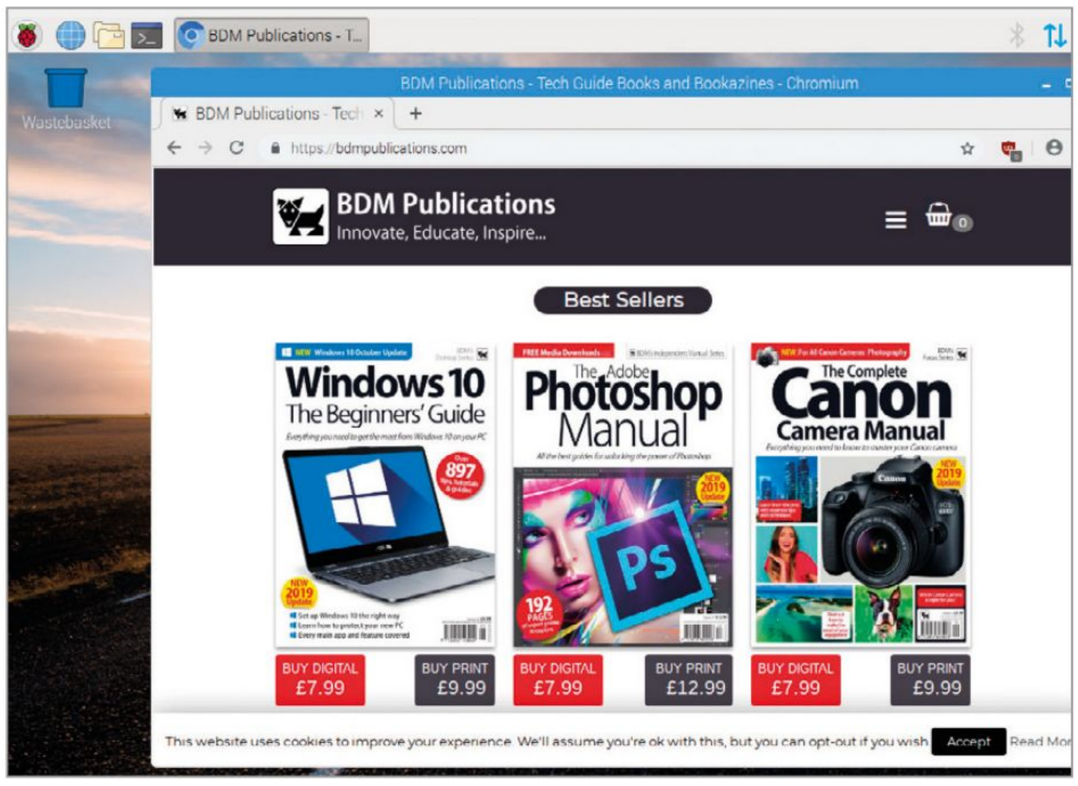
HARDWARE

Hook up a Pi Zero without an SD installed, select GPIO Expansion Board from the options and you will be able to talk to the Zero's 40-pin GPIO via the likes of Python or Scratch. By doing so, you'll have all the benefits of a Raspberry Pi, with the power of your laptop.



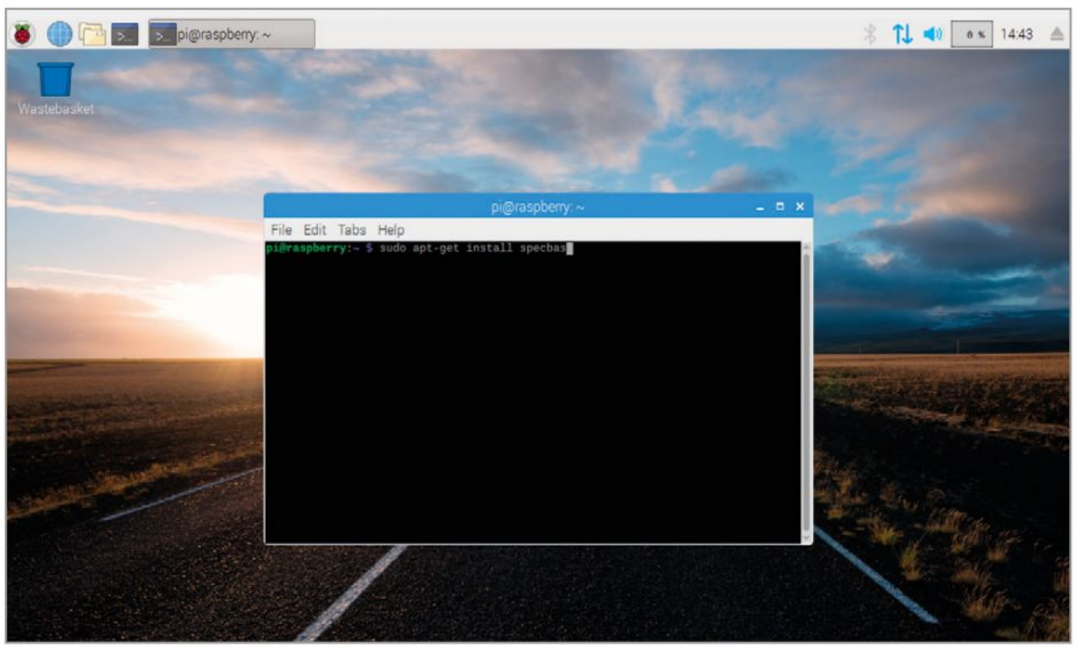
WEB BROWSING

As with most modern operating systems, you can browse the Internet exactly as you would with Windows 10, the latest macOS or similar and because this is a Linux-based OS, there's an added element of security, as it won't be affected by Windows-targeted viruses.



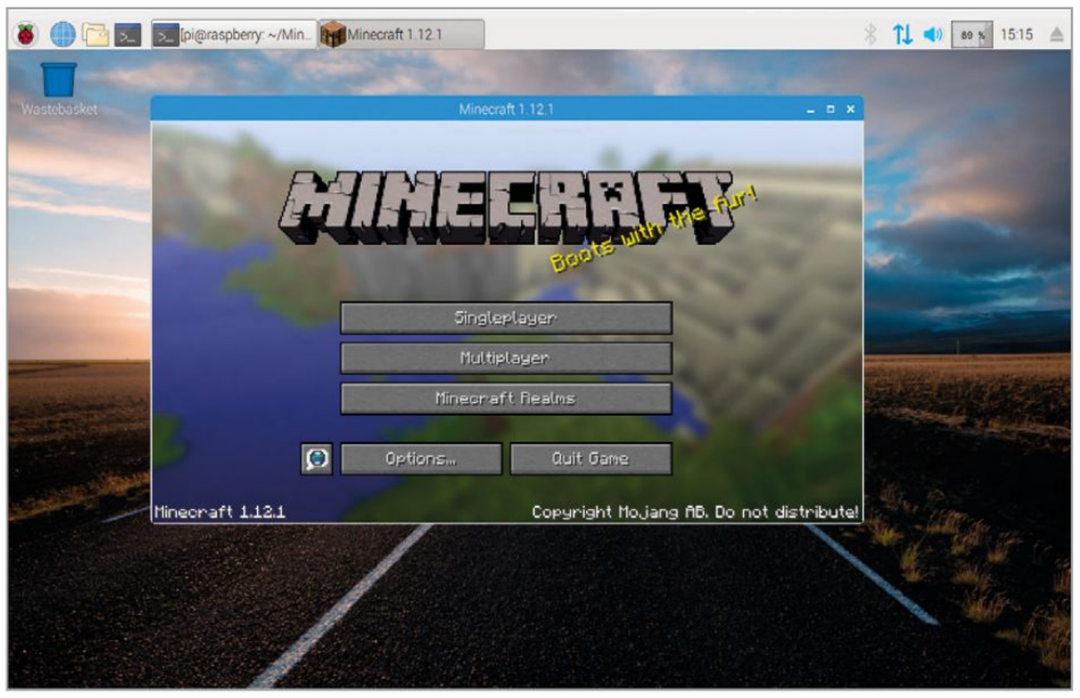
LINUX X86-BASED SOFTWARE

As this is a Debian based version of Linux, you can naturally install any of the tens of millions of Linux x86-based software currently available. Remember though, only software that has both an ARM and an x86 version will work on both the Pi version and this version of Raspbian.



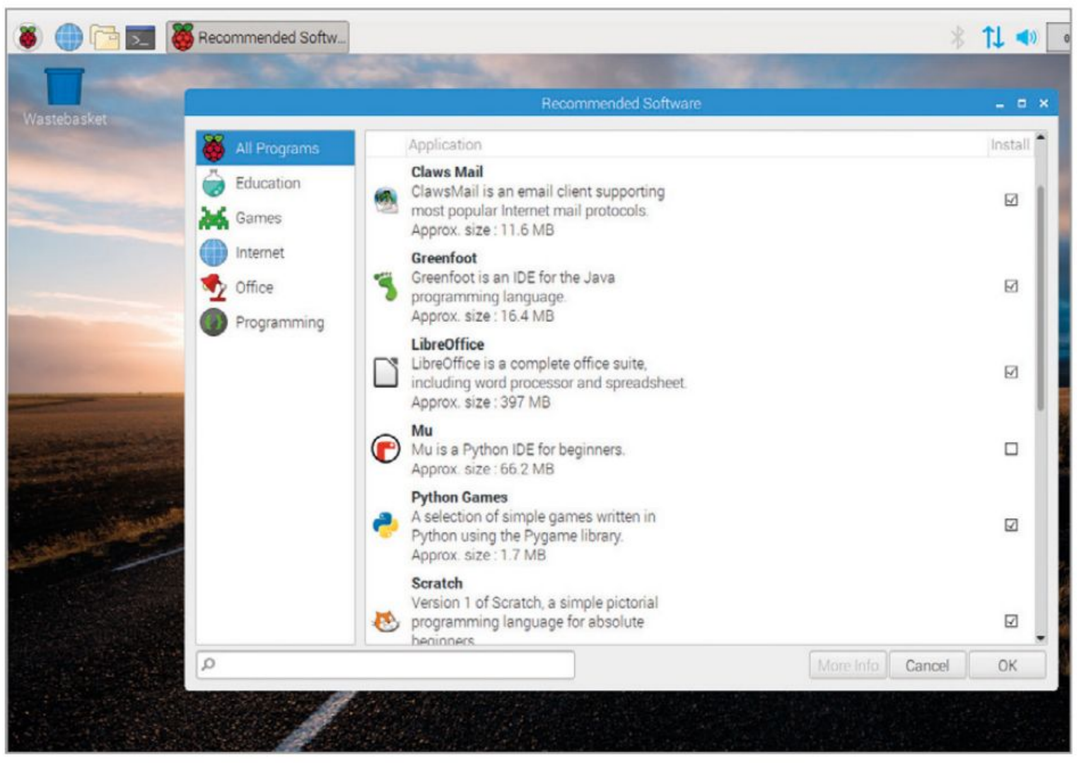
MINECRAFT

Neither the Raspberry Pi 4 or the Pi 3 Model B+ are the most powerful computing devices available. Your older laptop is probably significantly more powerful but even so, you will be able to play games like Minecraft without too much trouble at all.



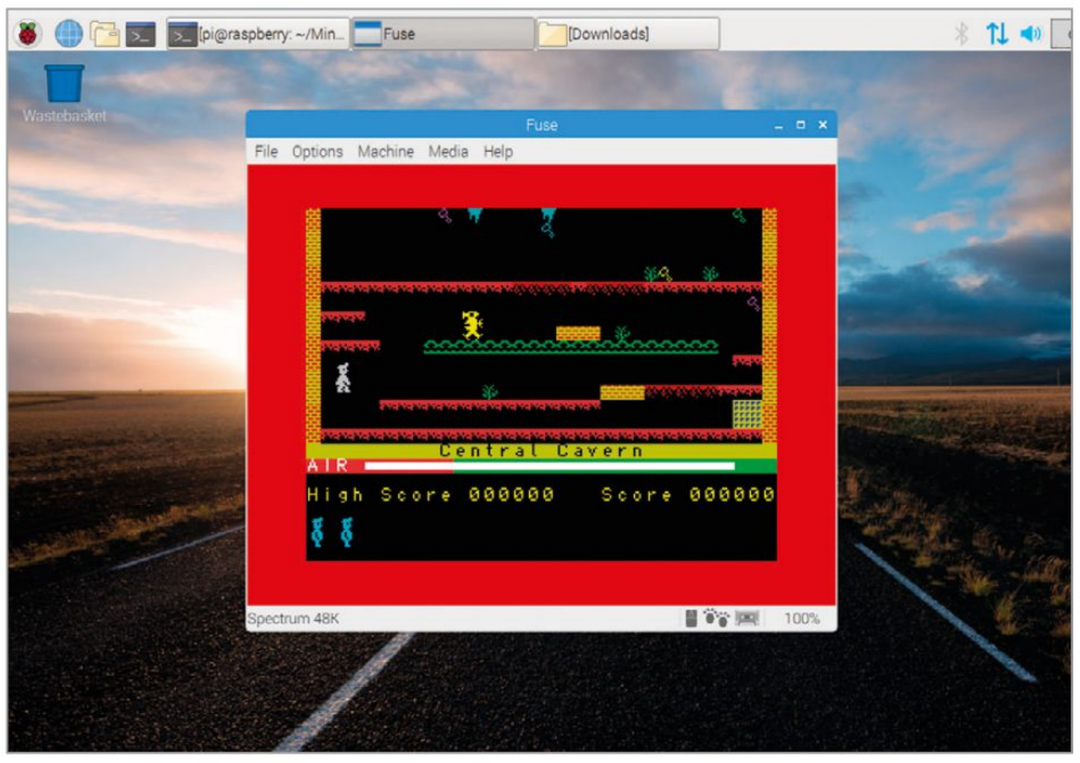
RECOMMENDED SOFTWARE

The Recommended Software option is also available with this version of Raspbian. This tool allows you to pick and choose which of the Pi-favoured software you add or remove. Simply tick the box next to the name of the app.



RETRO GAMING

If retro gaming is more to your taste, then Debian has a huge number of retro emulators available, covering both consoles and home computers. You will need legal ROMs, however, to play the games, but these are widely available from reputable sites.



SSH

It's easy to use SSH from within the Raspberry Pi Configuration tool. Select the Enabled option and then use a client, such as PuTTY, to SSH into the OS. Here we've enabled SSH and connected from another Windows PC, using PuTTY as the SSH client.

GAMING

Of course, there's an equally large selection of modern gaming available for Debian; most of which will run perfectly well under Raspbian in an x86 environment. You will need to Google what's available and try out a few examples as the list is simply too big to mention.

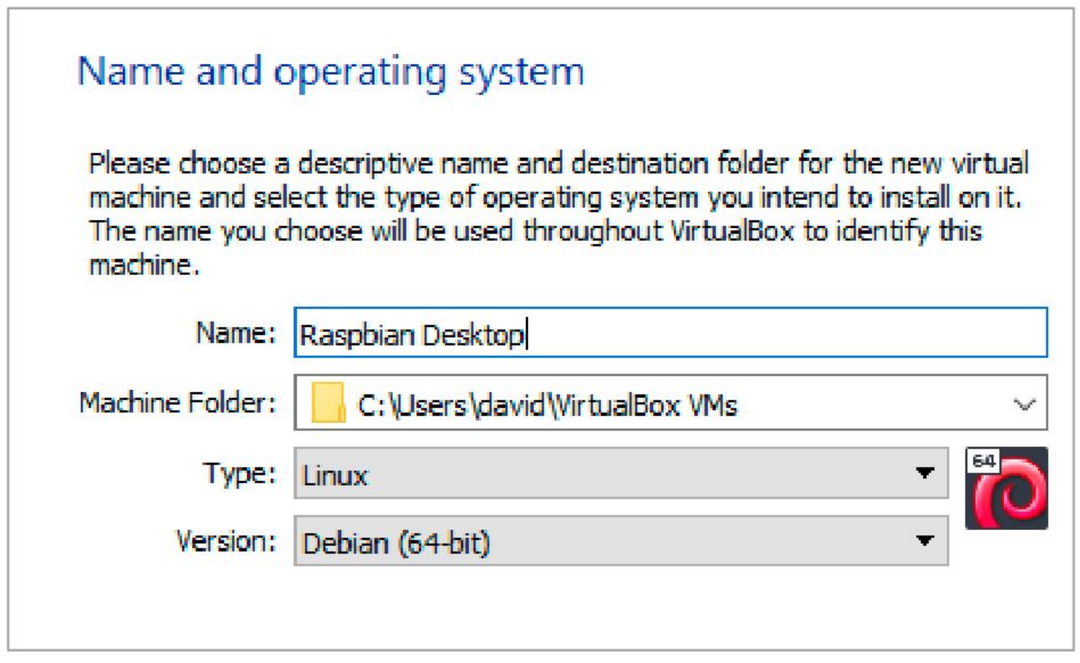
Debian Stretch – Virtual Desktop

Don’t have a spare PC handy for Debian Stretch with the Pi Desktop? Then consider a virtual machine. Within a virtual machine, you can run the Pi desktop without ever affecting your Windows or Mac setup. Here’s a quick how to.

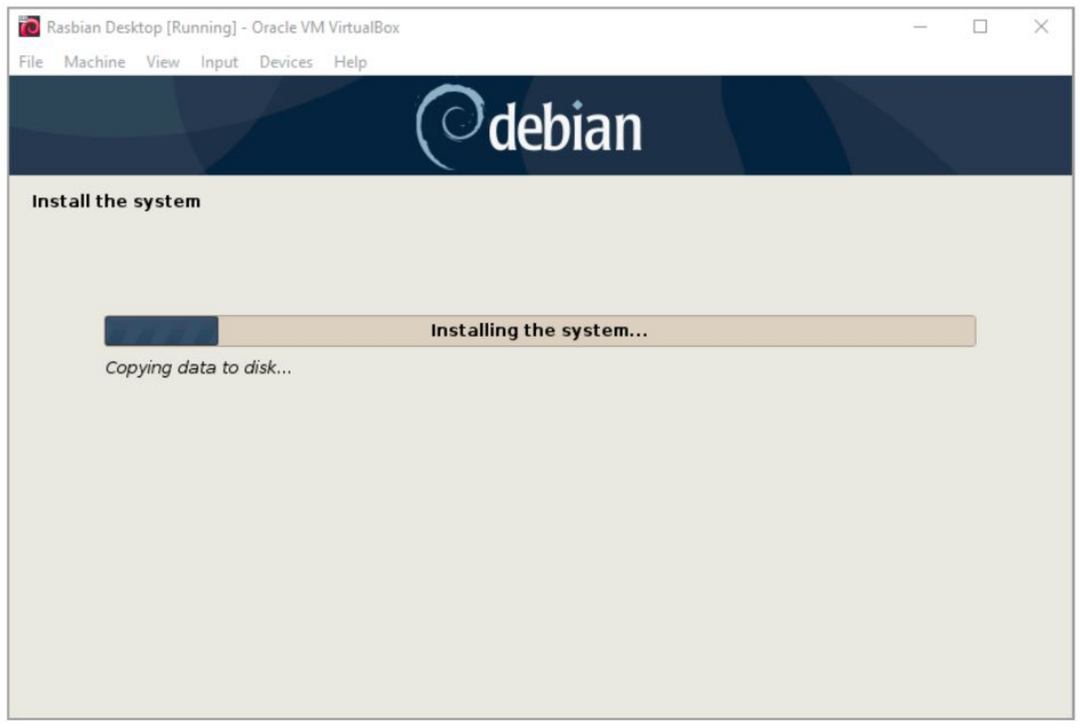
GOING VIRTUAL

You’ll need a copy of VirtualBox to download and install, which you can find at **https://www.virtualbox.org/wiki/Downloads**. Windows users click on Windows Hosts, macOS click OS X Hosts.

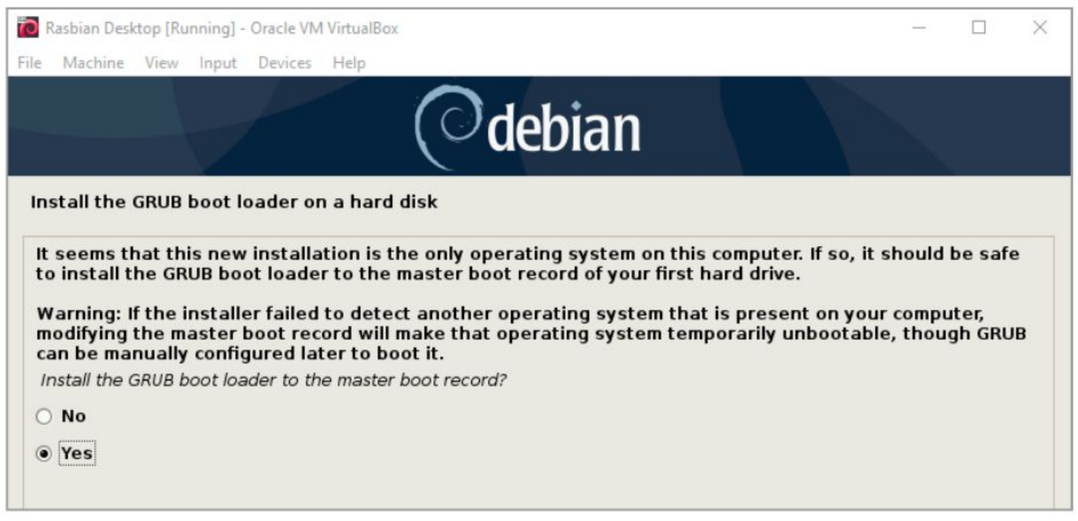
STEP 1 Install the downloaded VirtualBox for your system by following the on-screen instructions. Next, download the Raspberry Pi Desktop ISO from **https://www.raspberrypi.org/downloads/raspberry-pi-desktop/**. Start VirtualBox and, from the menu, click Machine – New. Edit the Type to Linux, version to Debian (64-bit), and name it Raspbian. Click Next and follow the default options.



STEP 2 Start the newly created machine and then locate the downloaded Raspberry Pi Desktop ISO. In the menu titles ‘Debian GNU/Linux Installer Boot Menu’, use the arrow keys and select Graphical Install. Press Enter and follow the on-screen installation instructions. Use the Guided – Entire Disk option when asked, then Write Changes to Disk.



STEP 3 When asked “Install the GRYB boot loader to the master boot record?” Select **Yes**, and choose **/dev/sda**; after a brief time select yes to installing a GRUB boot loader followed by the sda drive identification (not the Manual option). Once installed, click Continue to reboot and you’ll see the Raspbian desktop load up.



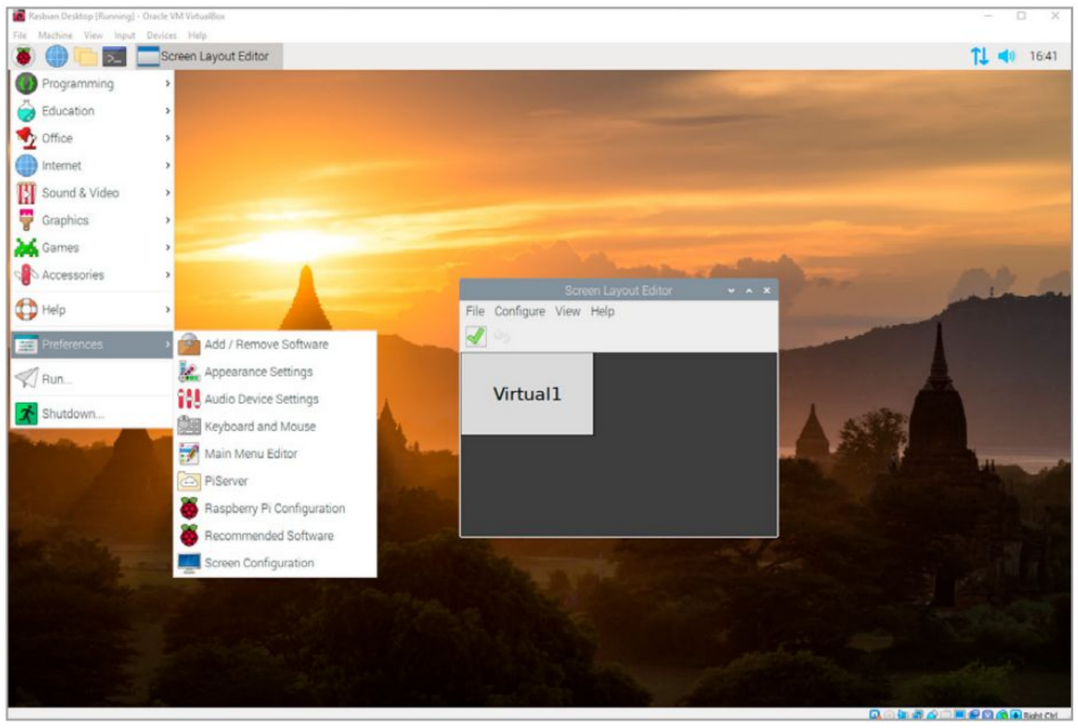
STEP 4 Drop into a Terminal, click the fourth icon along the top menu bar, and enter:

```
sudo apt install build-essential module-assistant dkms
sudo m-a prepare
```

Answer yes to install packages. Click on Devices in VirtualBox, followed by Insert Guest Additions CD. In the Terminal, enter:

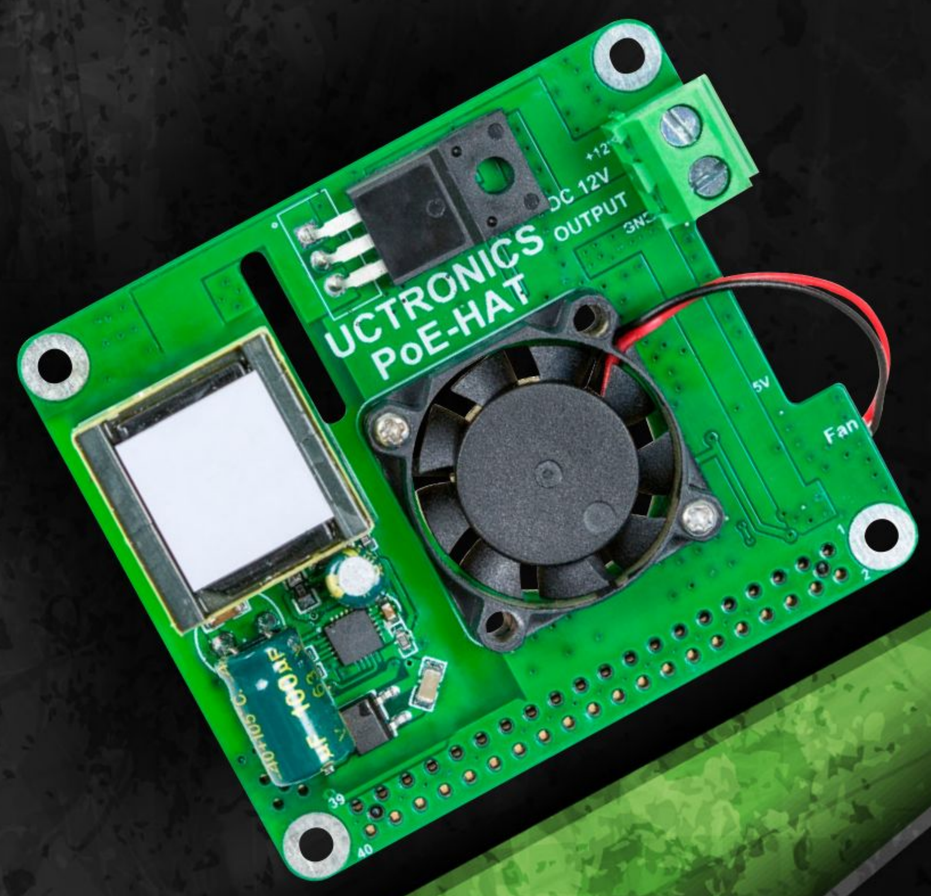
```
sudo sh /media/cdrom/VBoxLinuxAdditions.run
sudo reboot
```

You can now adjust resolution via **Menu > Preferences > Screen Configuration**.



Discovering Raspbian

The Raspbian operating system is a Linux distribution that comes pre-packed with a fantastic collection of programs and apps to help you get the most from your Pi. Raspbian enables the Raspberry Pi to become a fully functioning desktop computer and with it, you can code in the latest version of Python, Scratch and more.



Take a Tour of PIXEL

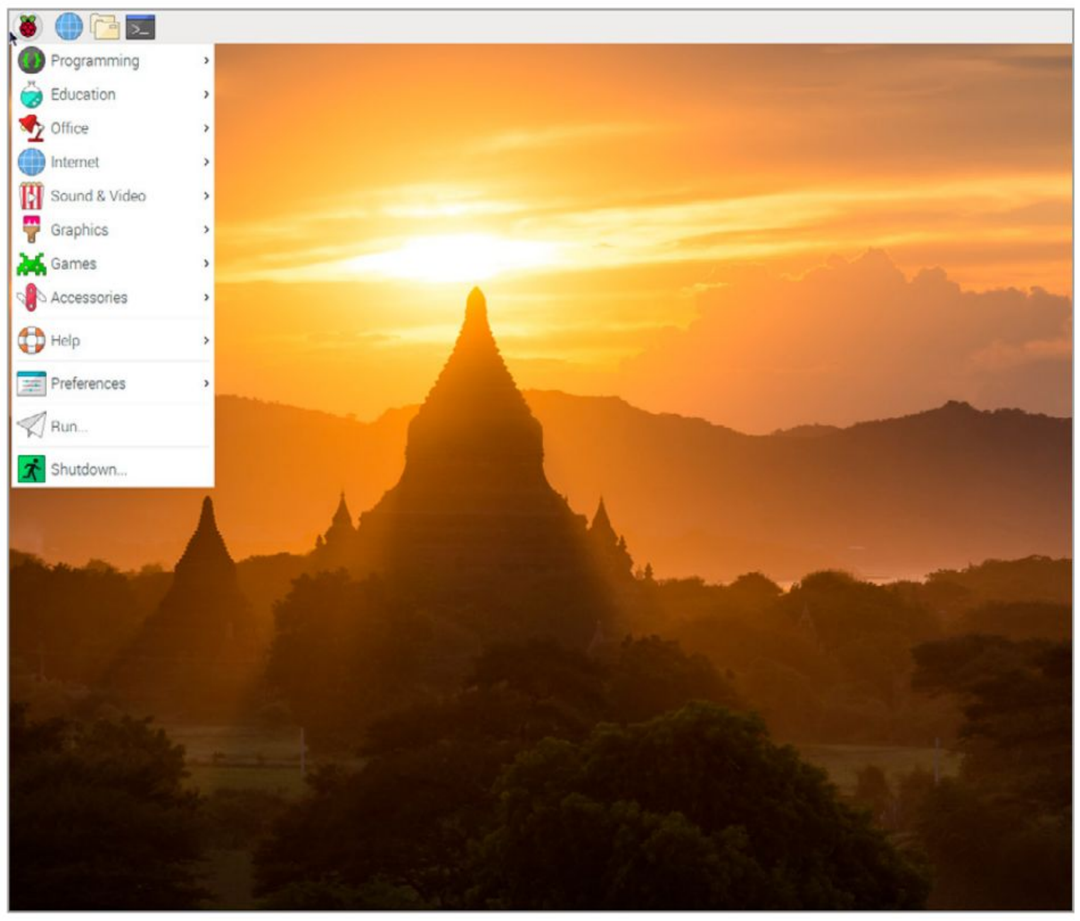
PIXEL is the desktop environment for your Raspberry Pi. It's built on top of Raspbian, the core OS, which is itself designed around the Linux distribution - Debian, specifically, Debian Buster. PIXEL is fine-tuned to run on the Raspberry Pi, taking advantage of its unique hardware.

PIXEL

PIXEL stands for "Pi Improved Xwindow Environment, Lightweight". It's a lightning fast and energy saving desktop that looks good too. It contains almost everything you'll ever need to work with the Pi 4.

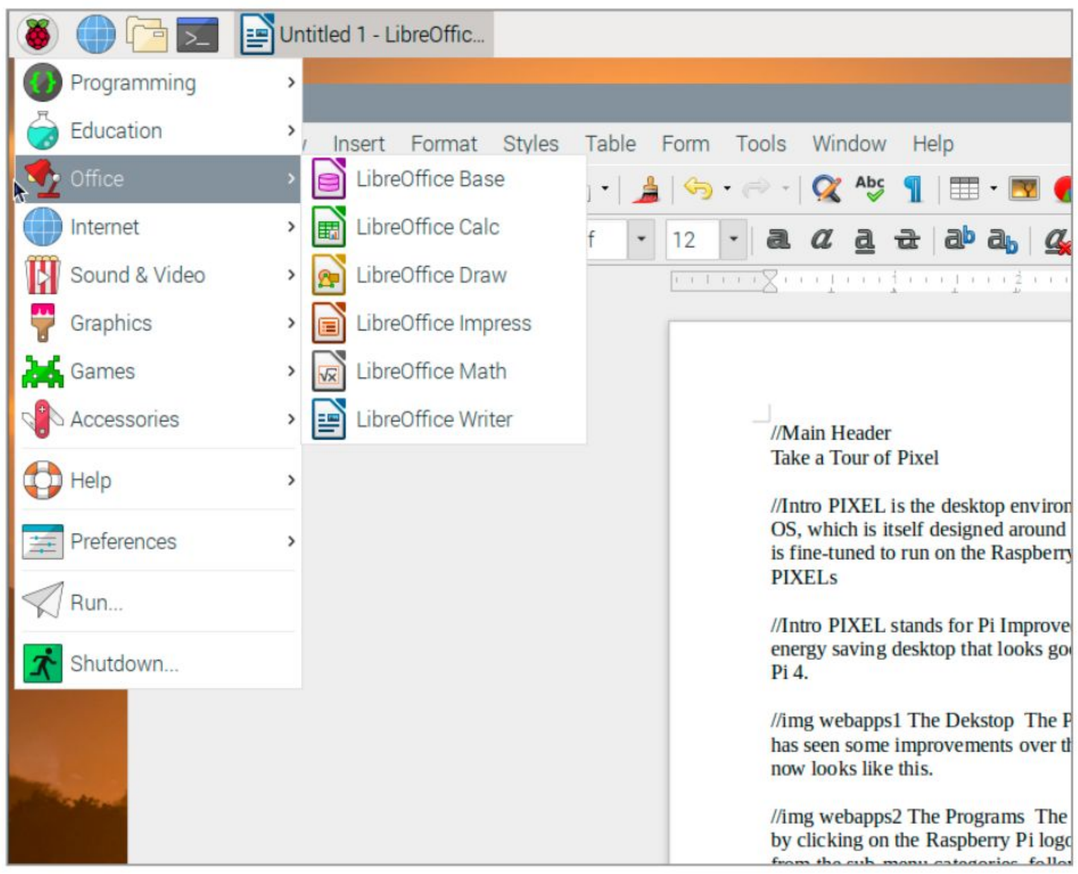
THE DESKTOP

The PIXEL interface is a module that's installed on top of the core OS. It has seen some improvements over the years, and with the release of the Pi 4 and Raspbian Buster, it now looks like this.



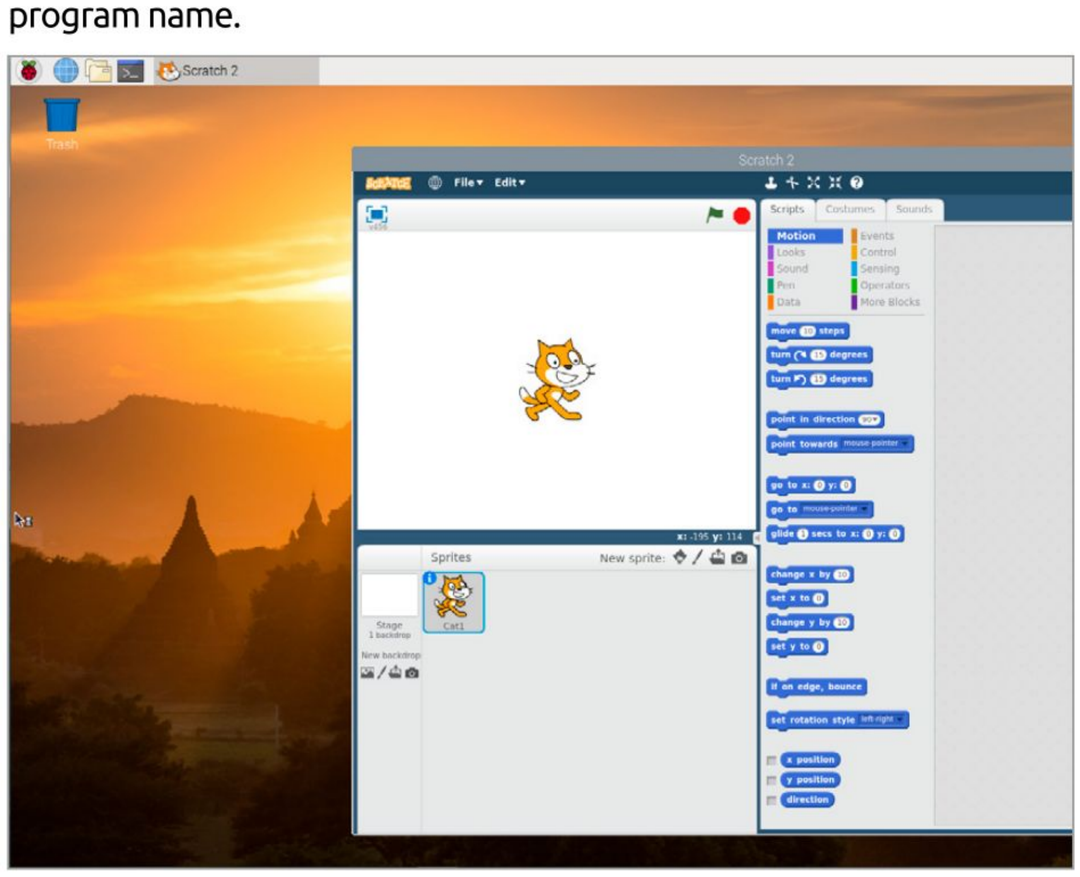
PRODUCTIVITY

You'll notice that within the Main menu, there's a category named Office. In here, you'll find a pre-installed suite of programs called LibreOffice, that are Microsoft Office compatible; there's a word processor, database, drawing, presentation and spreadsheet program.



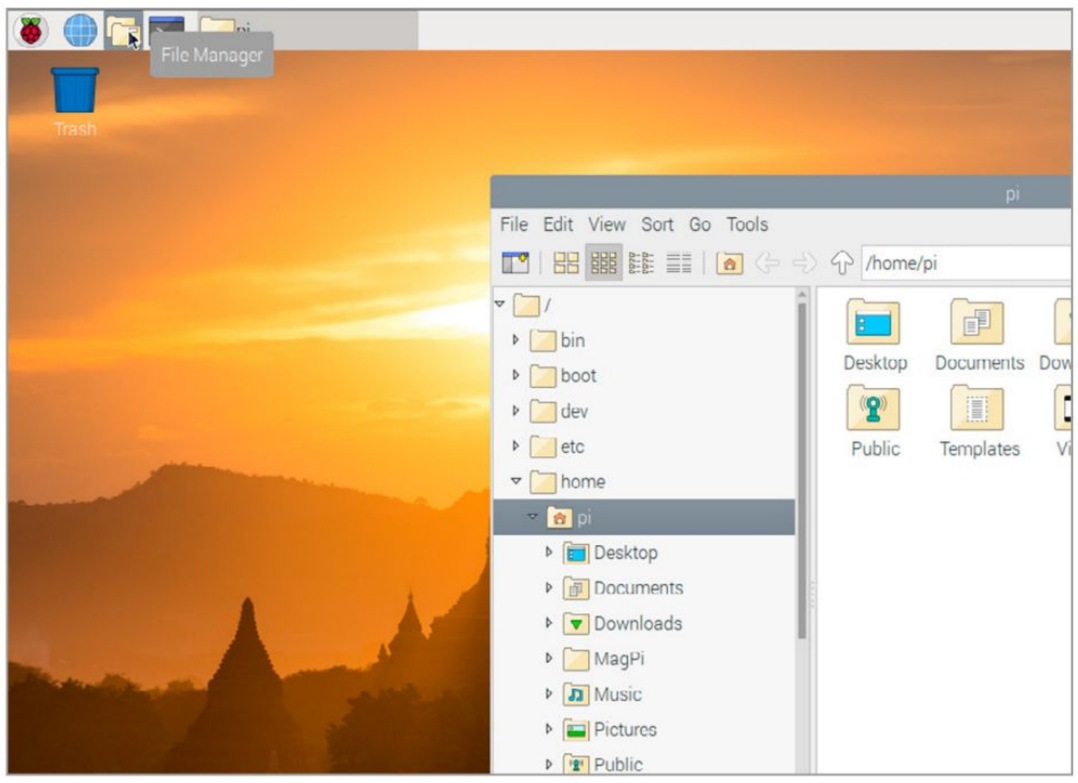
THE PROGRAMS

The programs that come pre-installed with Raspbian are easily located by clicking on the Raspberry Pi logo in the upper left corner of the desktop. Once there, you can select from the sub-menu categories, followed by the program name.



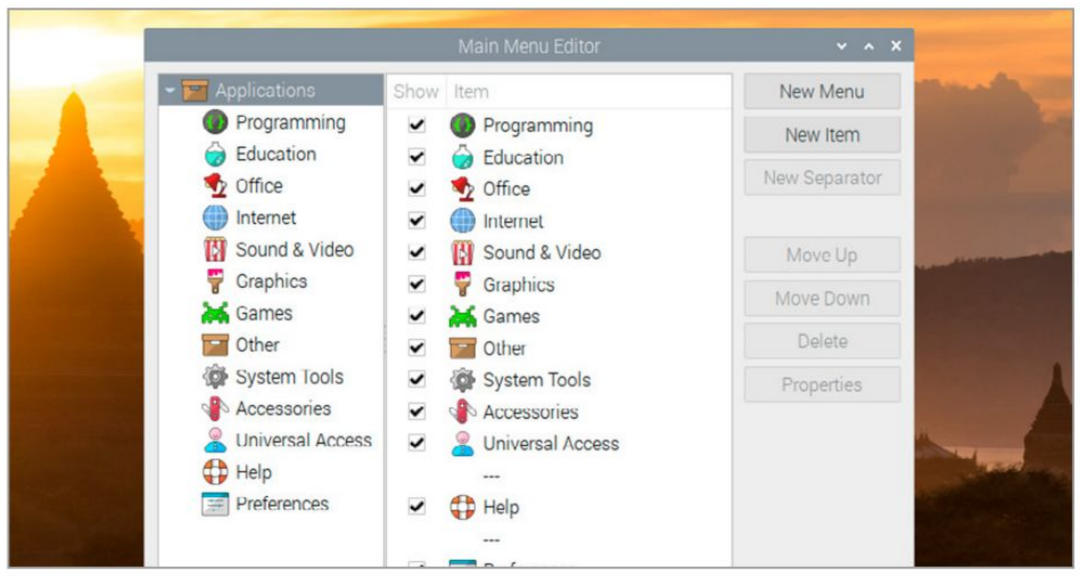
FILE MANAGER

To view the files stored on your Raspbian OS, click on the File Manager icon in the Application Launch Bar (the one that is shaped like a pair of files). This opens a window displaying all the files in your home directory. The Directory Tree gives quick access to common folders, such as Desktop and Documents.



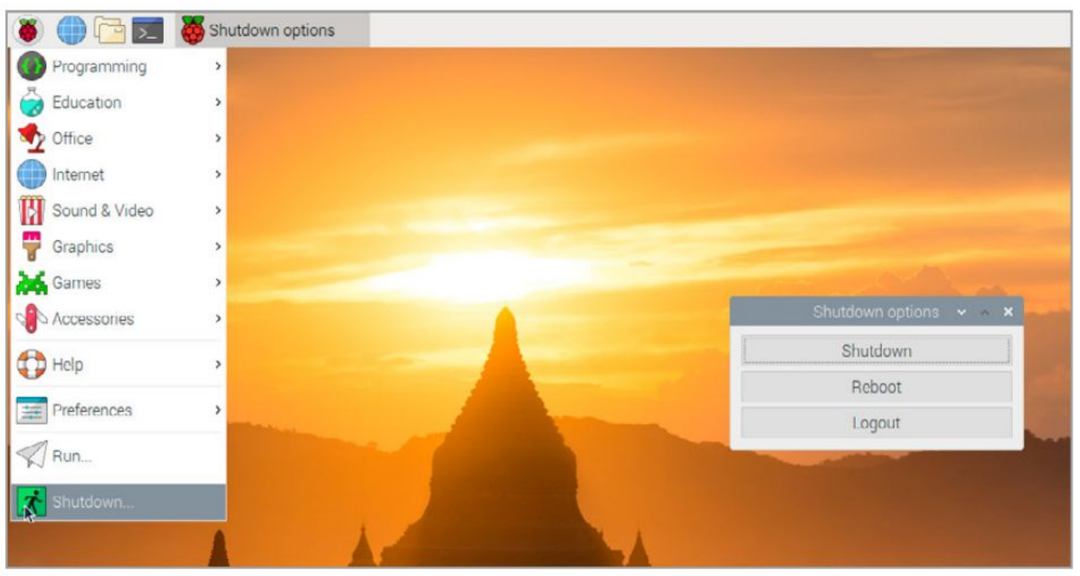
MENU EDITOR

It is possible to remove and add items to the Application Menu using the Main Menu Editor. Choose **Menu > Preferences > Main Menu Editor**, then add or remove ticks next to programs and sections you want to include\exclude. Use Move Up and Move Down to rearrange items in the Menu.



SHUTDOWN

It's important that you switch off your Raspberry Pi safely. When you are finished with your Raspberry Pi session click **Menu > Shutdown**. Three options appear: Shutdown, Reboot, and Logout. Click Shutdown. Always wait for the screen to go blank before removing the USB power.

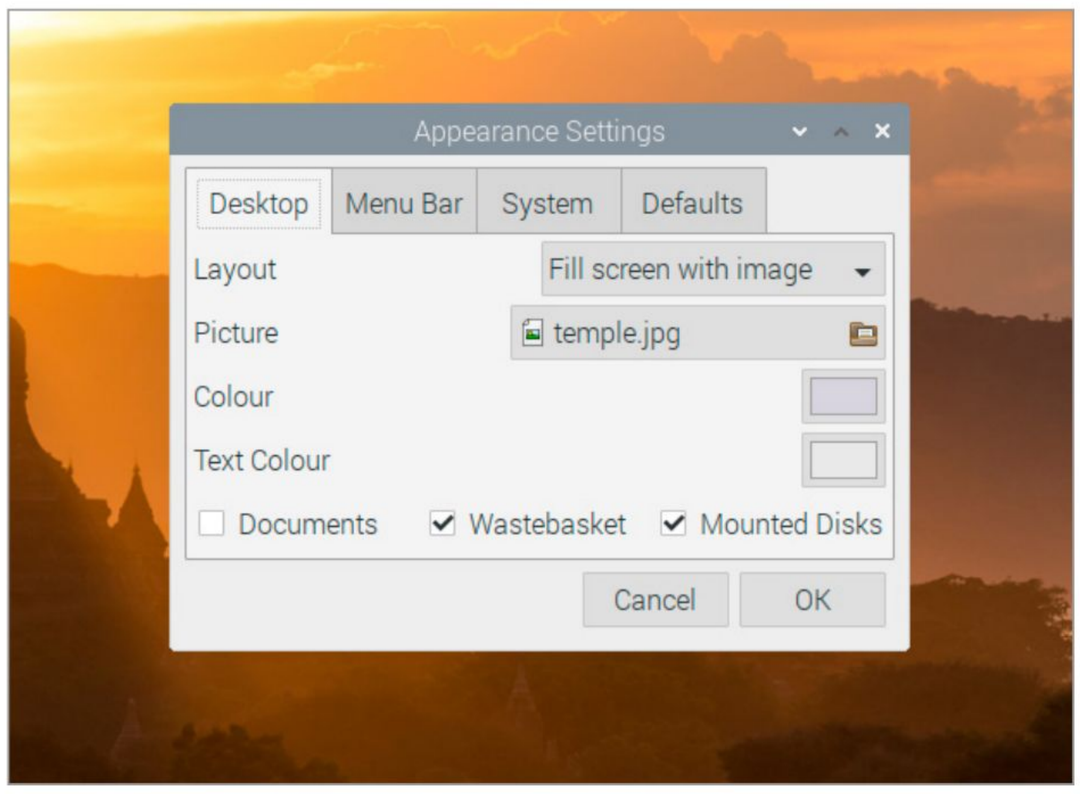


CUSTOMISE THE DESKTOP

As with any other operating system desktop, you're able to customise the Raspberry Pi's PIXEL interface, making it more personal and more you.

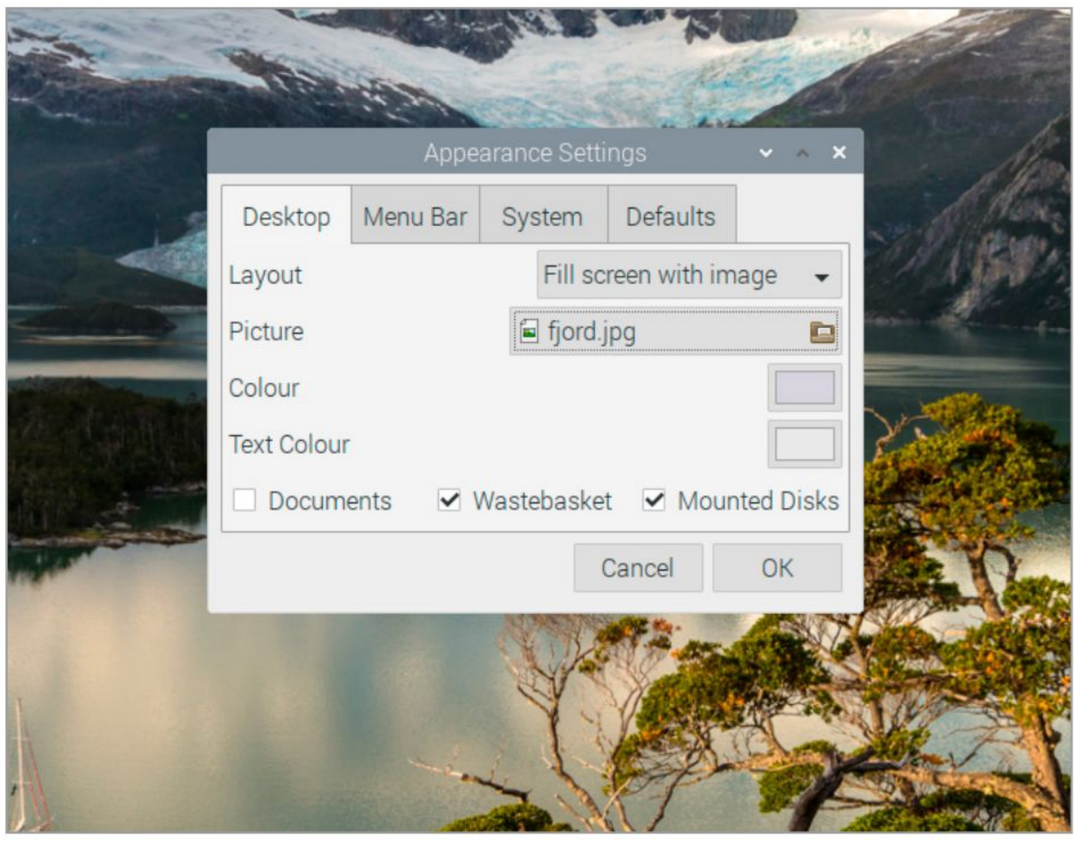
YOUR PIXEL

Choose **Menu > Preferences > Appearance Settings** to open the Appearance Settings window. There are four sections: Desktop, Menu Bar, System, and Defaults. You can use these to adjust the various options for your system, such as colours, layout, and so on.



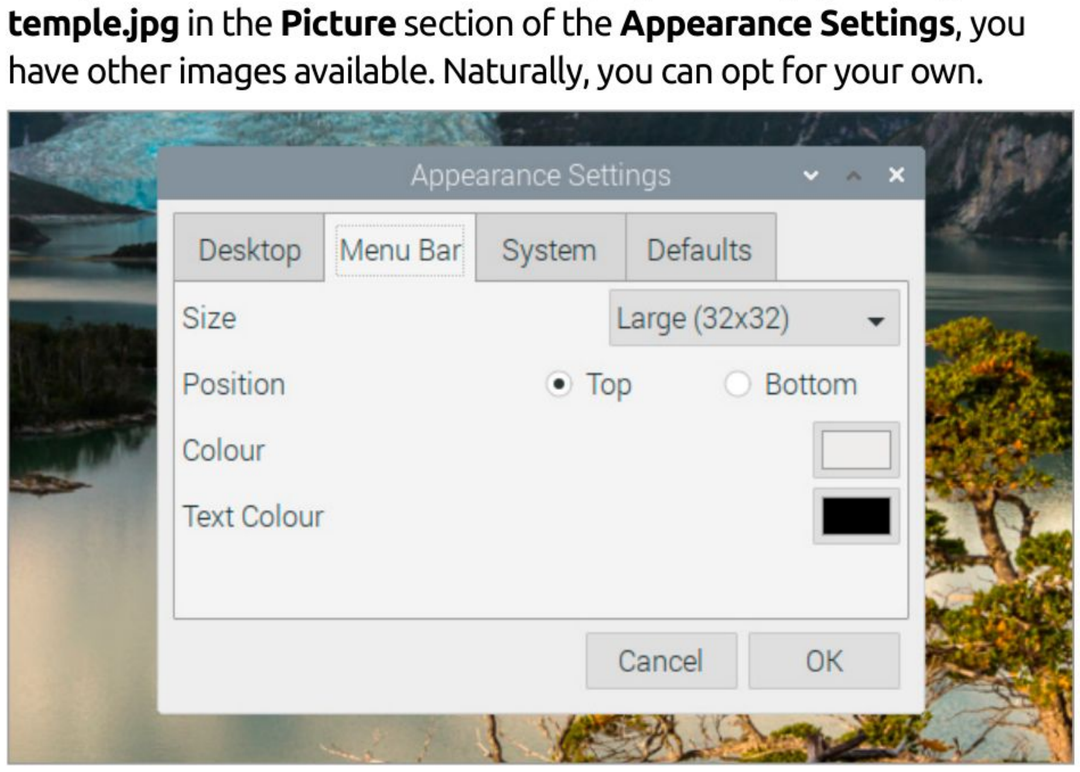
MENU BAR

The second tab in the Appearance Settings window allows you to adjust the menu bar that runs along the top of the desktop. You can change its size, colour, and even its location from the top to the bottom of the screen.



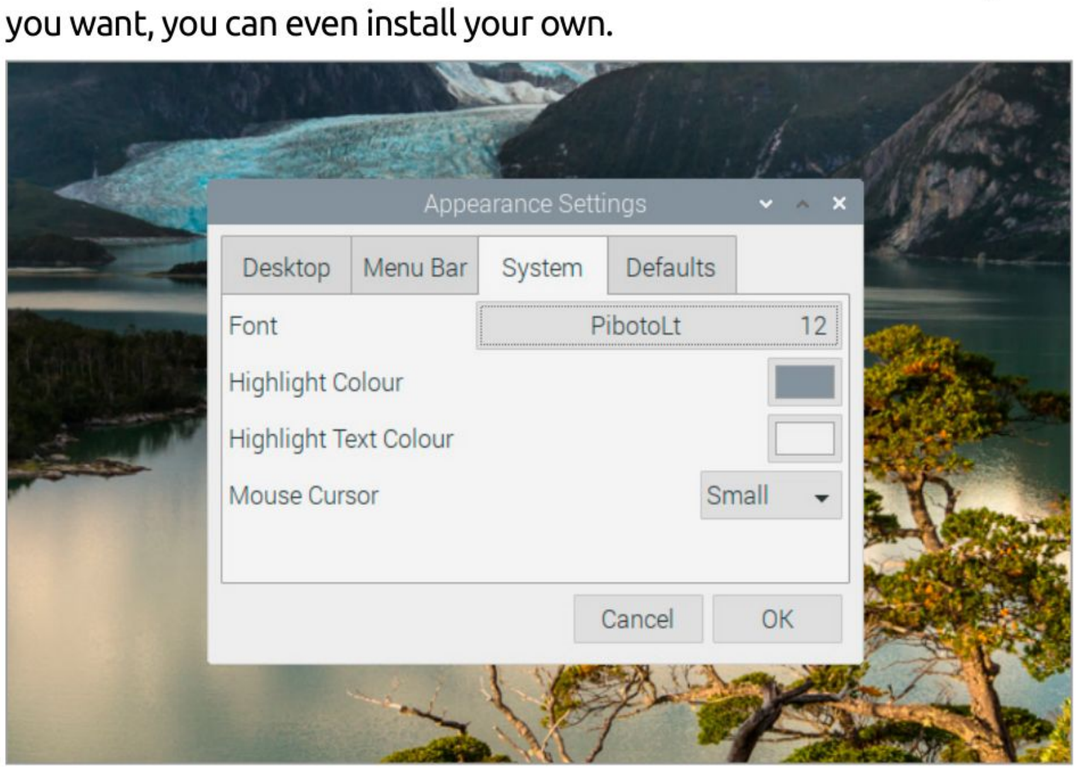
DESKTOP WALLPAPER

The most common change is that of the desktop wallpaper, or desktop background. The default is called 'Temple', however, by clicking on **temple.jpg** in the **Picture** section of the **Appearance Settings**, you have other images available. Naturally, you can opt for your own.



FONTS

The System tab in the Appearance Settings window will enable you to change the core system font, as well as the mouse cursor size. There are numerous fonts to choose from and, if you want, you can even install your own.



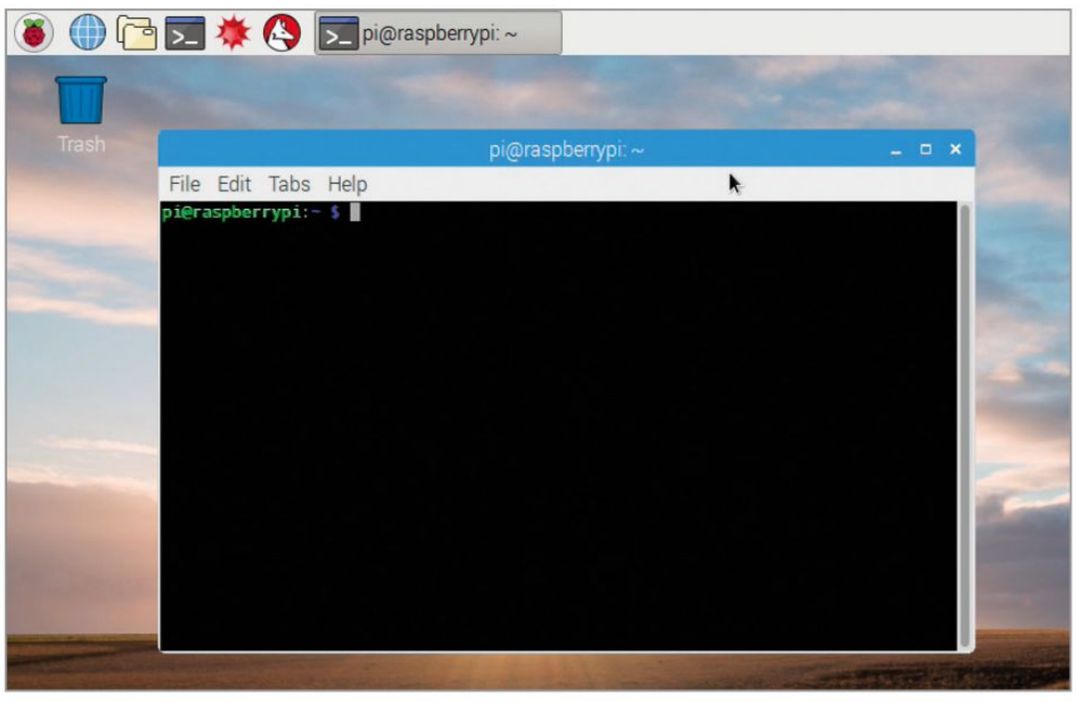
Exploring the Command Line

If you've grown up with Windows or the Mac OS then you might never have encountered the command line. This part of the operating system sits beneath the desktop and is used to control a computer using text commands. You'll need to get familiar with it to use your Raspberry Pi.

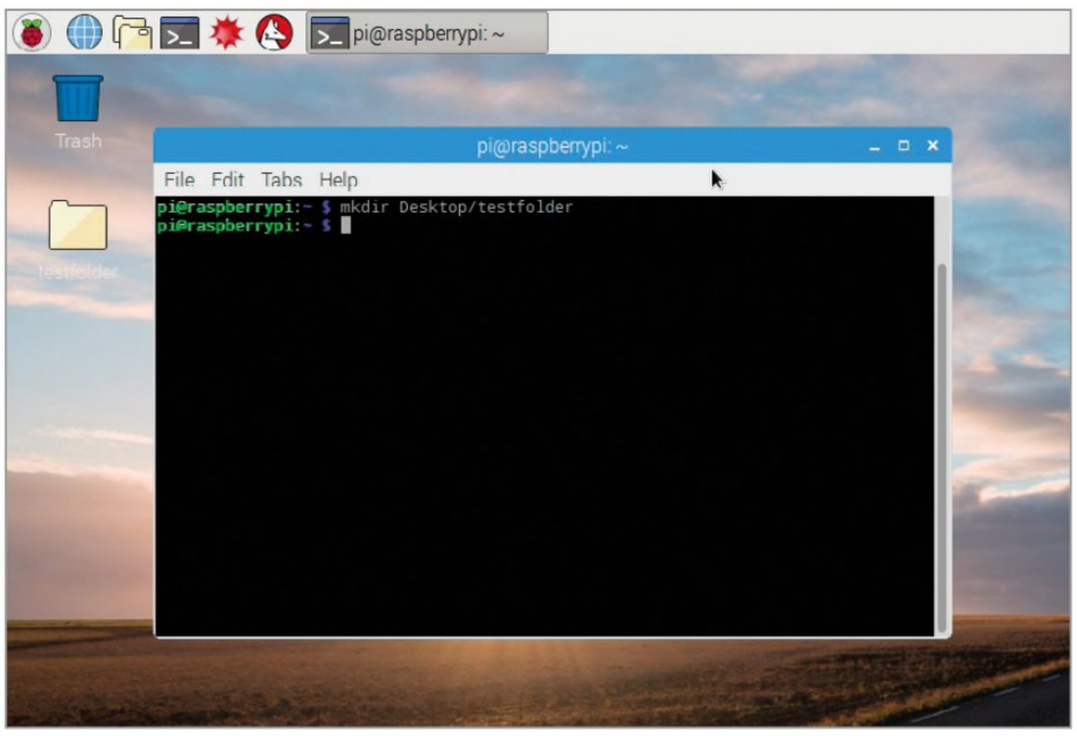
USING TERMINAL

Despite its graphical PIXEL interface, Raspbian is a Linux operating system and this means you'll spend a lot of time working with text commands. Using the command line is an important part of learning to use a Raspberry Pi computer.

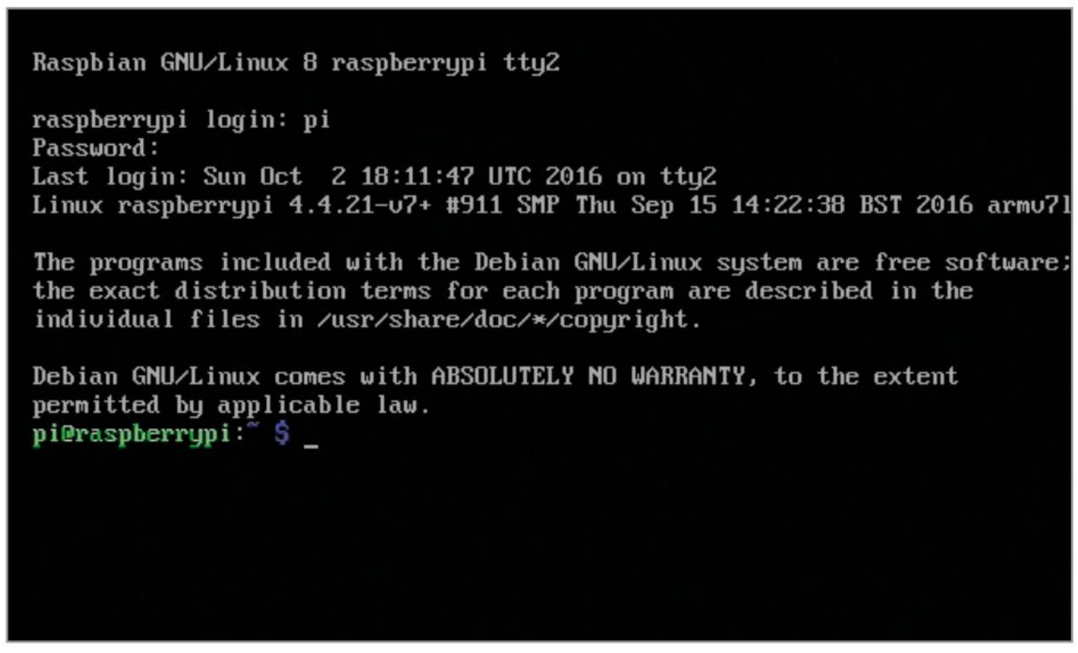
STEP 1 By default the Raspberry Pi boots directly into the PIXEL interface. Here you use a visual metaphor, files, buttons and so on, to make changes to your computer. The command line is where you control a computer by entering text commands. Click Terminal in the Menu bar to view the console.



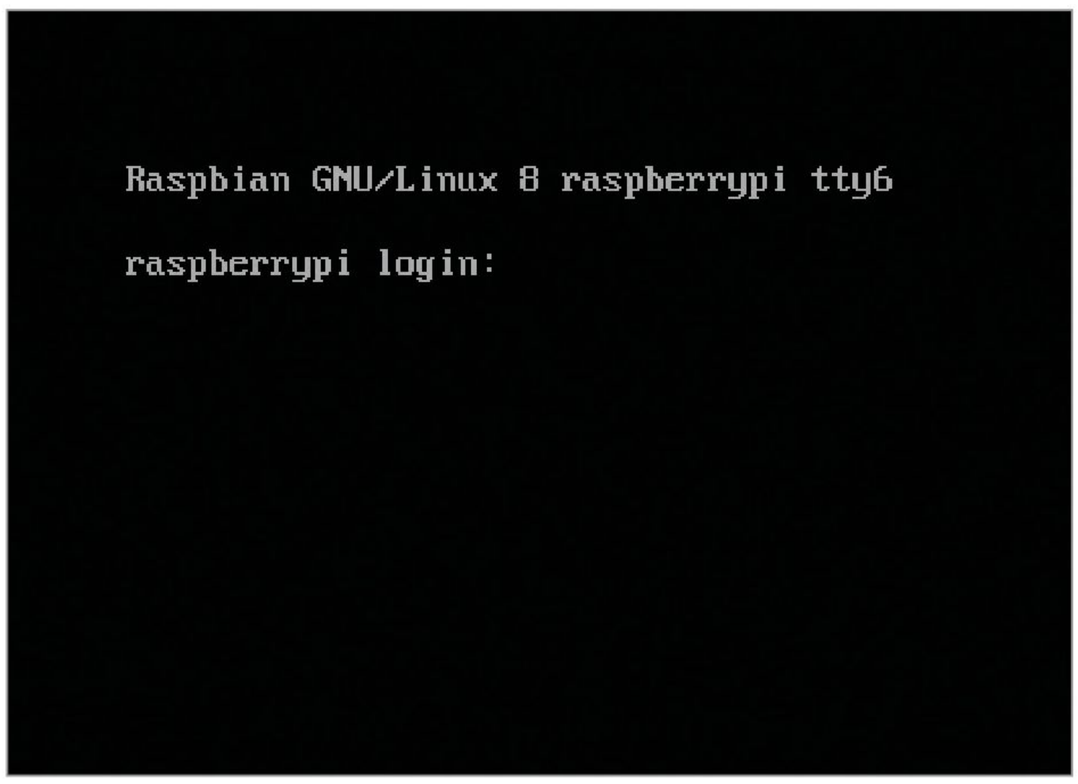
STEP 2 You can enter commands into the Terminal to make changes to the computer. This works alongside the PIXEL interface. Enter: `mkdir Desktop/testfolder` and press return. Notice a new folder appears below the Trash on the desktop. Drag it to trash to get rid of it. You can use the Terminal and PIXEL environments alongside each other.



STEP 3 Another way to switch to a command line environment is to press Control+Alt and F2. This gets rid of the desktop completely and you see just a black screen with text. This is known as a virtual environment and you're now in tty2 ("tty" is a throwback to teletext writers). You'll need to enter your login name ("pi" by default) and password ("raspberry" by default).

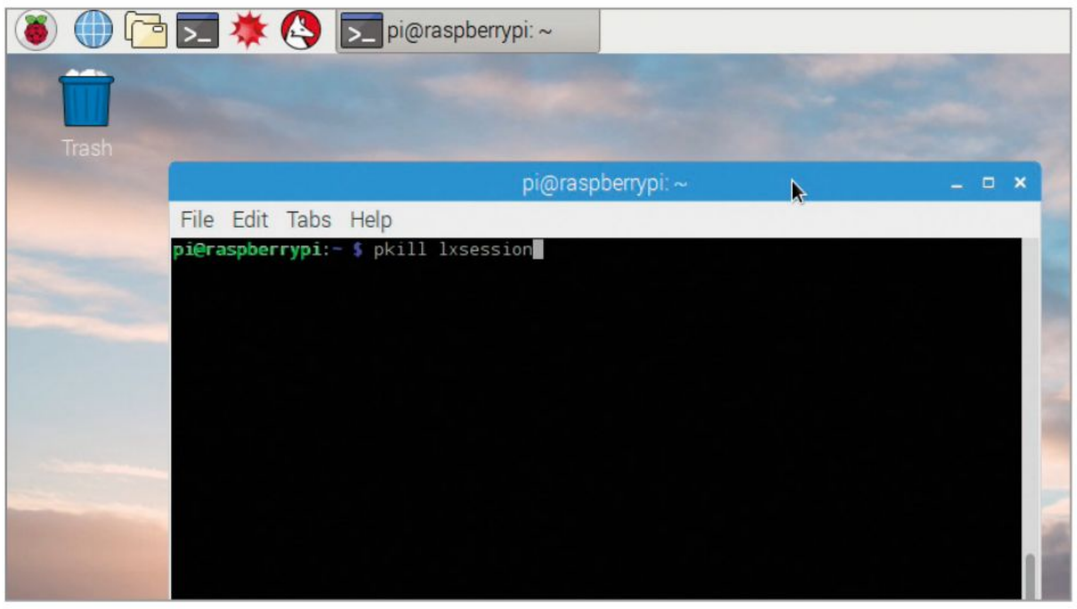


STEP 4 There are six different virtual environments available. Press Control-Alt-F3 to switch to the third, and Control-Alt-F4 to go to the fourth, and so on. You'll need to log on to each one in the first instance but can then jump back and forth between them.



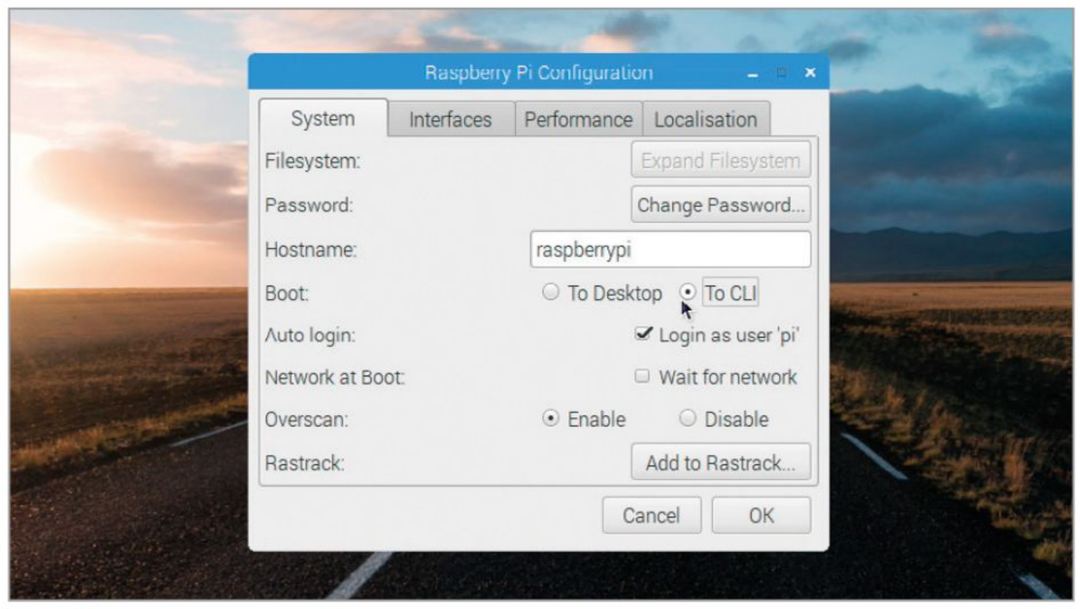
STEP 5

Press Control-Alt-F1 to return to the PIXEL interface. One thing to realise is that PIXEL is just a program running on top of the tty1 session. Open Terminal and enter: `pkill lxsession`. Ouch, there goes your desktop. Don't worry, enter: `startx` to get it back again. Most of the time you just enter commands into the Terminal window on top of the PIXEL environment.



STEP 6

Some users prefer to use just the command line. Open Menu > Preferences > Raspberry Pi configuration and choose To CLI and click OK and Yes. The Raspberry Pi will now boot into the command line interface. Enter: `startx` to get back to PIXEL, open Raspberry Pi Configuration and choose To Desktop to go back to normal.

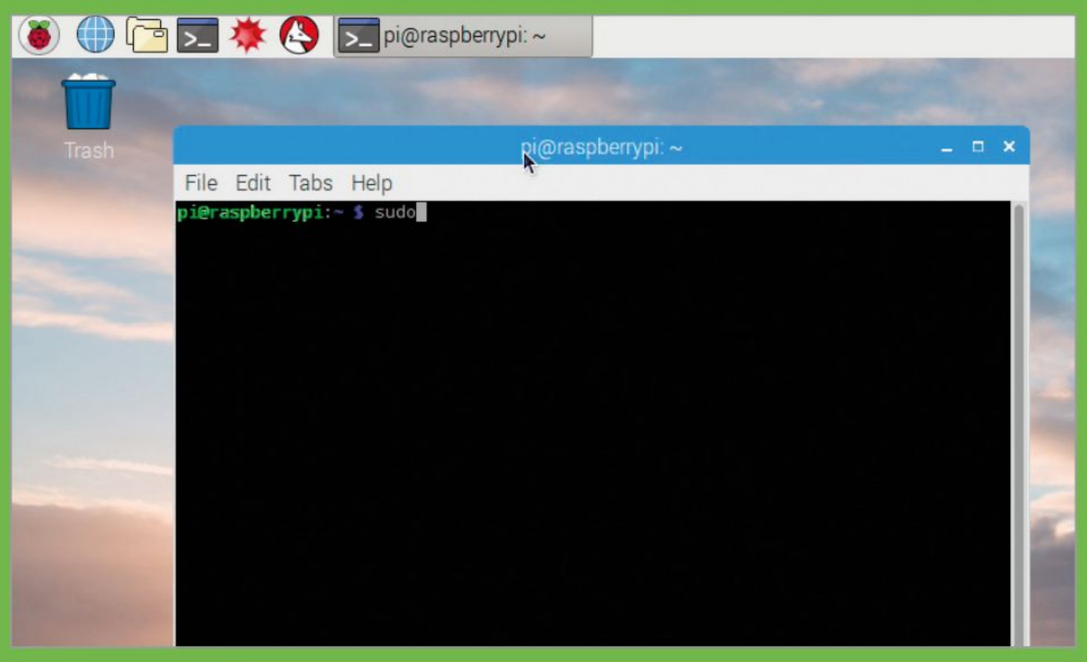


USING SUDO

Discovering the power of the super user.

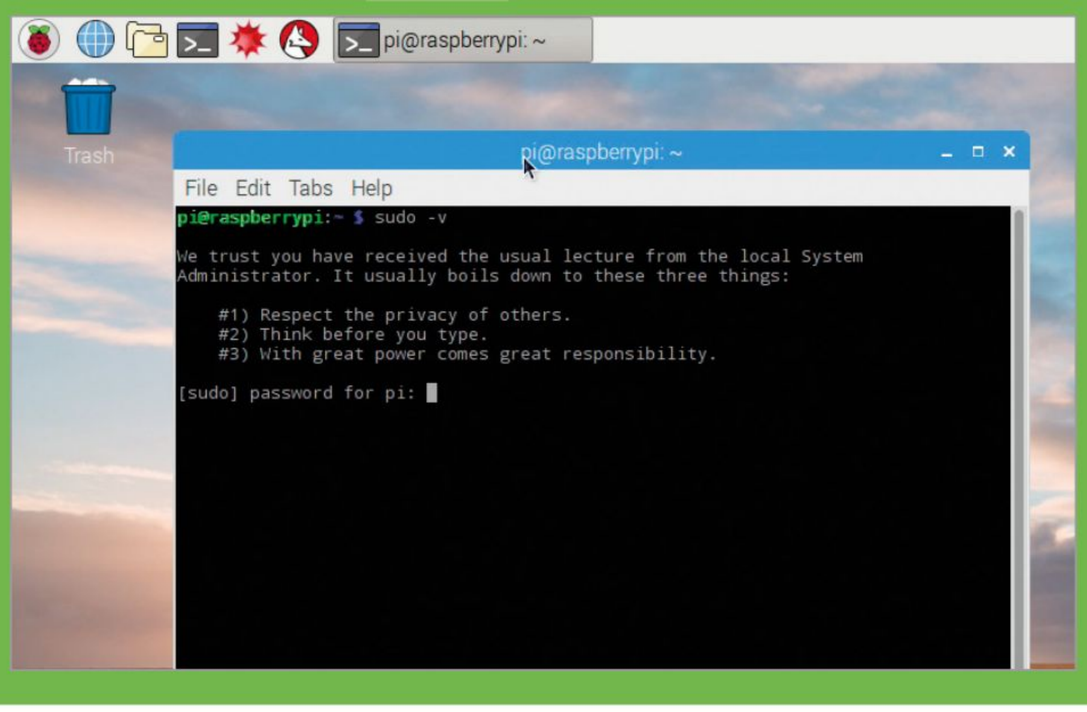
STEP 1

One of the most important command line instructions to know is `sudo`. This command stands for "substitute user do", sometimes incorrectly called "super user do" and allows you to run a command as another user. Typically this is the root user account, which has more access privileges than your user account.



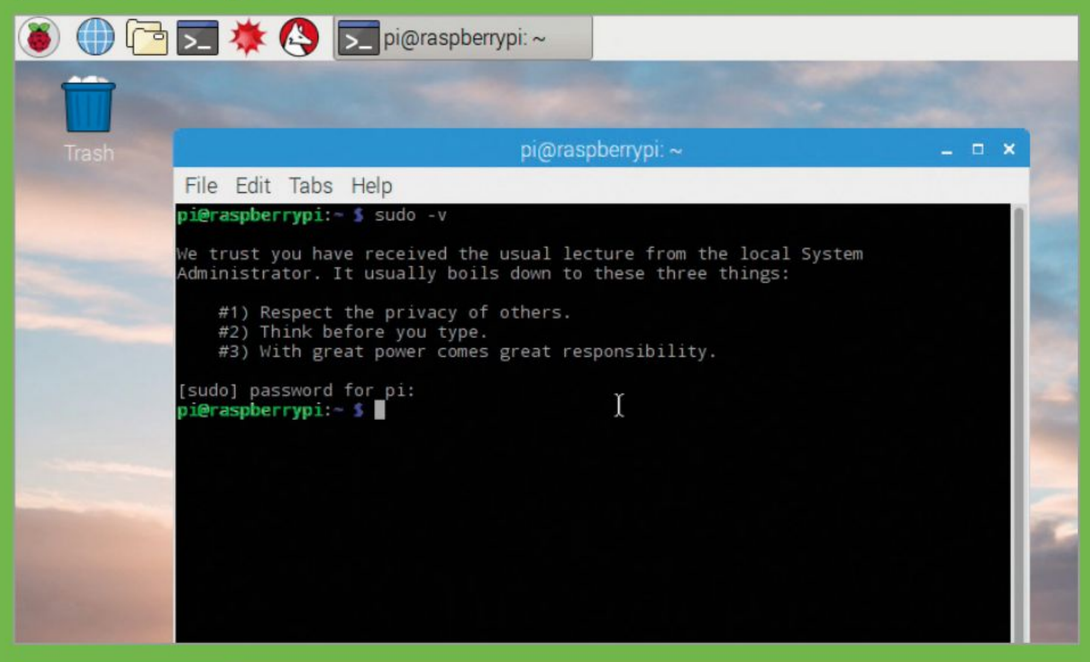
STEP 2

When you start a command with sudo it is run as the root user and is often used when changing files outside of your user account, such as installing new programs. When you enter sudo you will be asked to enter your user password. Enter: `sudo -v` and press enter.



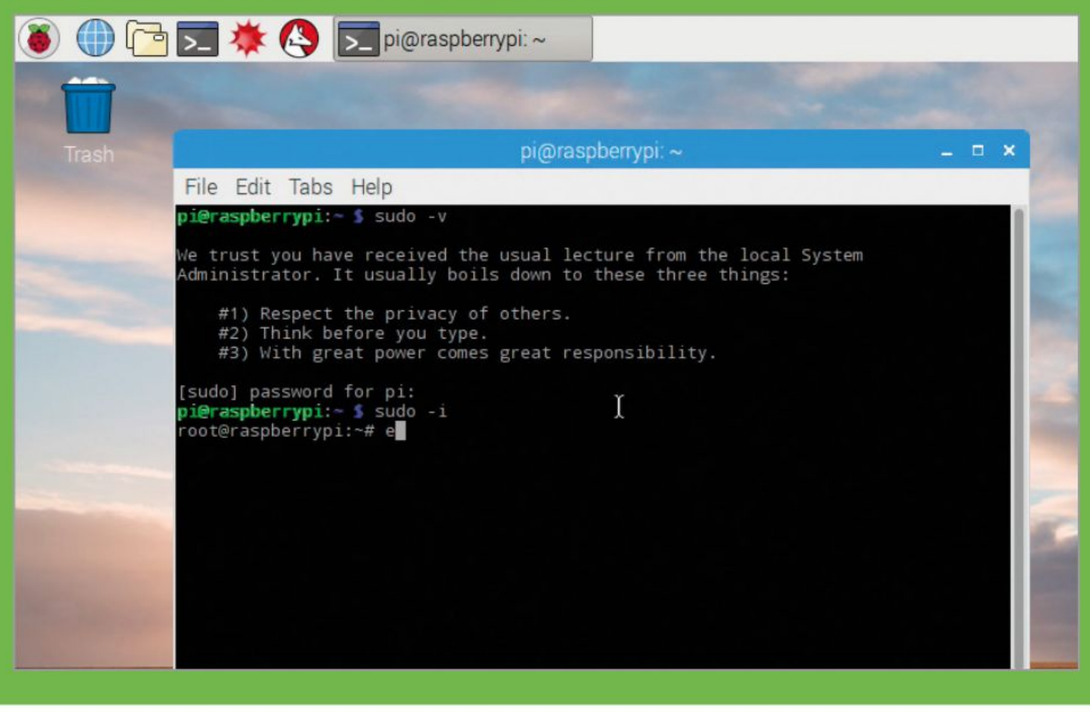
STEP 3

The first time you enter sudo you'll be given a warning message. Commands to make, edit and delete files prefaced with sudo can be used to change any file on the system. Sometimes with powerful, or disastrous, results. Be careful. Enter your password and press return.



STEP 4

Be especially careful of any set of commands that use sudo –i. This takes you to root mode, where every command is run as sudo. Your name (normally "pi") is replaced with ""root". Root mode is generally considered a bad place to be, so enter exit to get out of it as soon as possible.



Setting Up a Static IP Address

While this may sound complex, it's pretty easy if you follow these steps. A static IP ensures other computers can always find your Raspberry Pi. You don't have to do this, but it often comes in handy.

DHCP RESERVATION

Now that you've set up your Raspberry Pi and connected it to the network, you should take time to fix its IP address. This will make it a lot easier for you to connect to the Raspberry Pi from a Mac or Windows computer and share files.

STEP 1

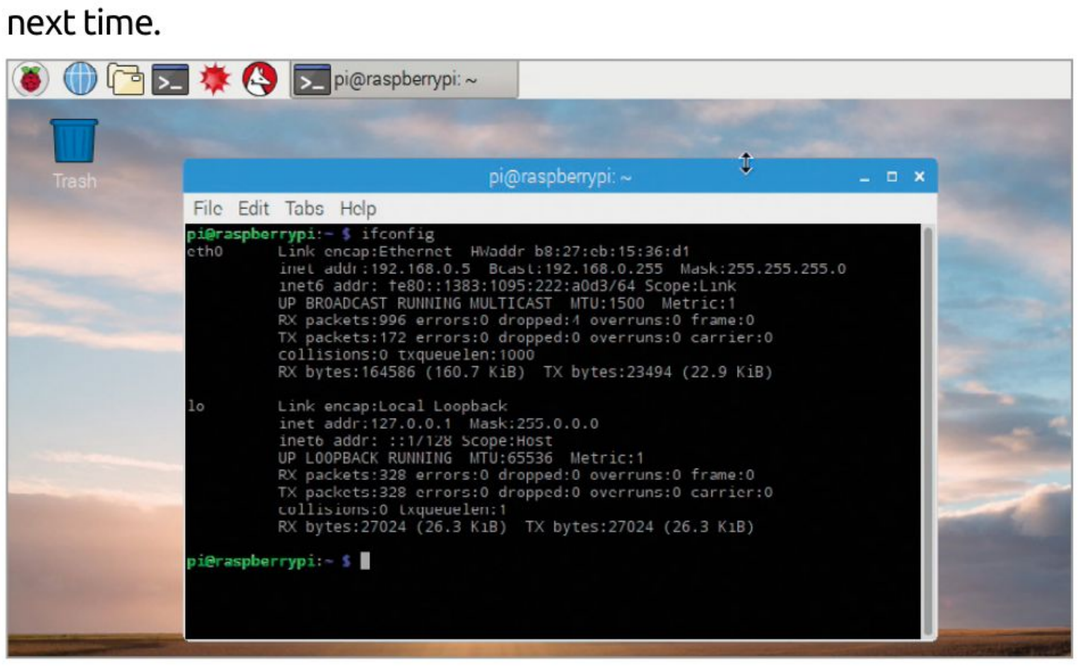
When you connect your Raspberry Pi to a network, the router (or modem/router) assigns it a number, known as its IP (Internet Protocol) address. This is a block of four digits and the first three are usually 192.168.0. After that is a digit for each electronic device.

STEP 2

The router typically takes the first address, so it is usually found at 192.168.0.1. That number is reserved for the router. Often you'll find this number on your router marked "web address". The router then assigns similar IP address numbers to the other devices you own as they are added to the network, so 192.168.0.2 might be your computer, 192.168.0.3 your smartphone and so on.

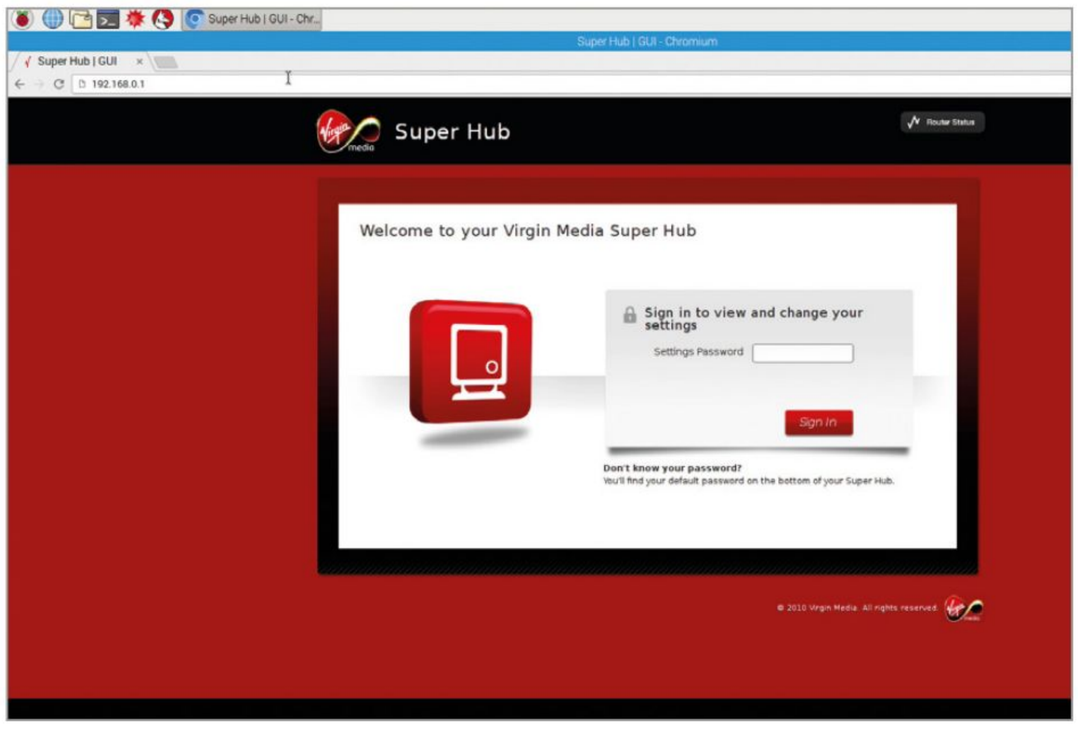
STEP 3

To find out what number your Raspberry Pi is using click on the Terminal icon and enter: `ifconfig` and press Return. You should find it next to "inet addr:". The challenge is that the router assigns this number using DHCP (Dynamic Host Configuration Protocol). When the Raspberry Pi is unplugged it reuses the number and your Raspberry Pi may get a different number next time.

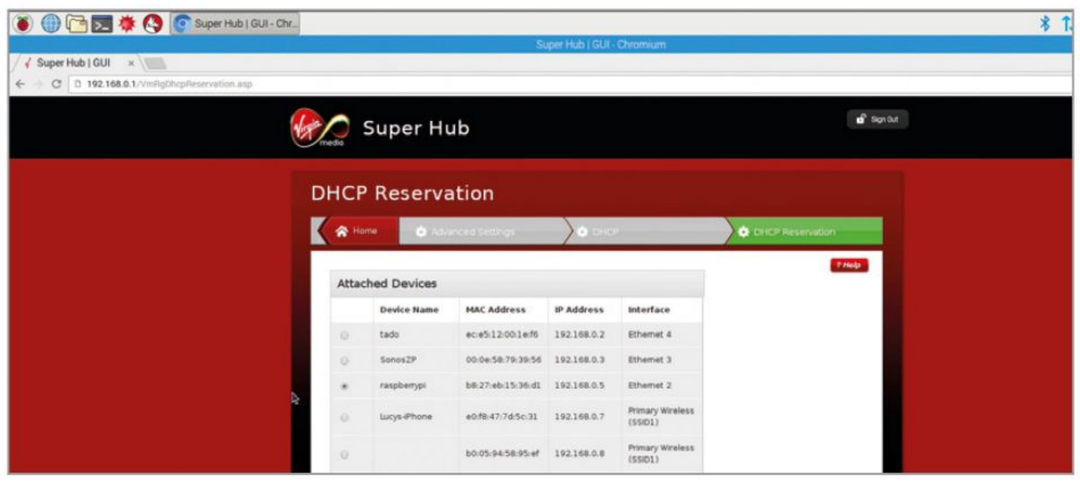


STEP 4

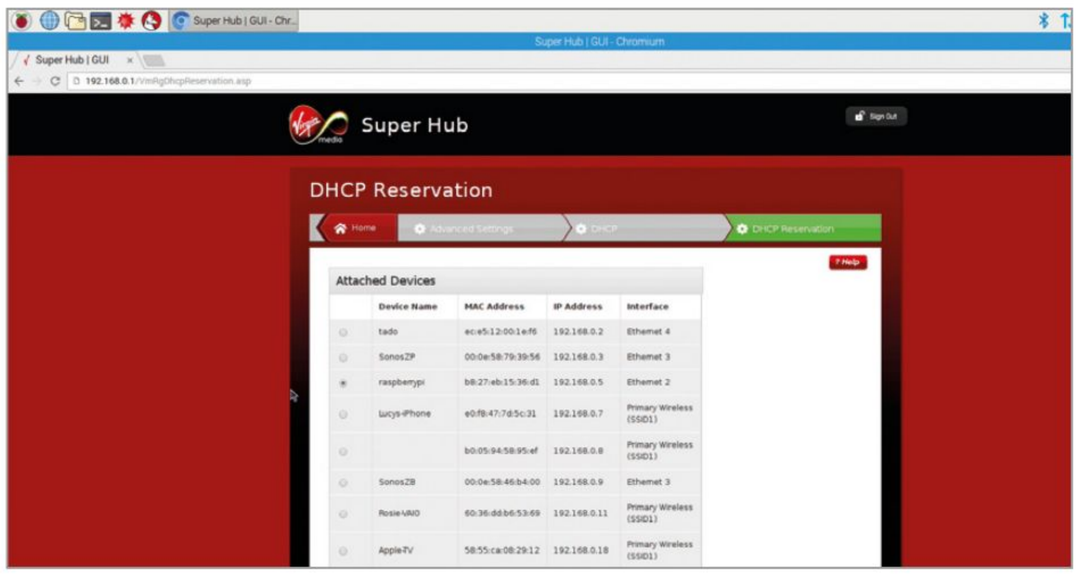
It's better all-round if you can get the Raspberry Pi to use the same IP address every time you connect it to the network. You do this using DHCP Reservation. This is where you tell your router to remember the Raspberry Pi address, and only use that address from now on. Start by opening the Chromium Browser and entering 192.168.0.1 to connect to your router's web interface.



STEP 5 We're using a Virgin Broadband router but the process is similar on most routers. Google the name of your router and "DHCP Reservation" to find the router you're looking for. Enter your admin password (our default password was listed on the back of the router). Click Advanced Settings > DHCP Reservation. In the DHCP Settings section you will find all of the devices connected to your network. Scroll down and find the one with the same IP Address listed in Step 3.



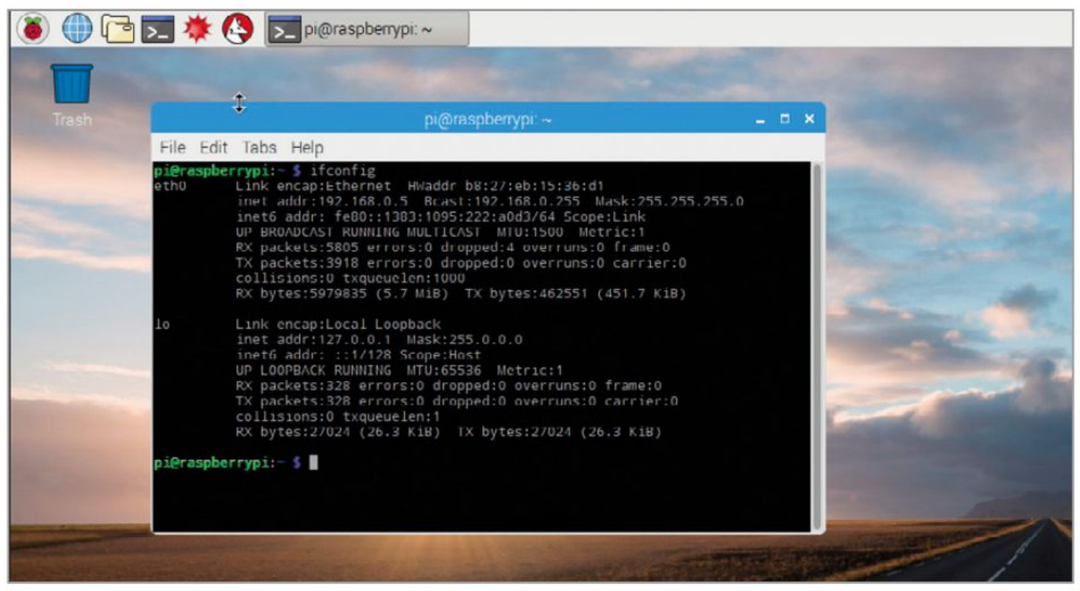
STEP 6 Select the device using the check box on the right and scroll down to the Add Reservation section. It should have filled in the Device Name, MAC Address and IP Address fields. If the Device name is "unknown" change it to "raspberry-pi". Click Add Reservation to ensure it always uses that IP Address. Click Apply to enforce the changes.



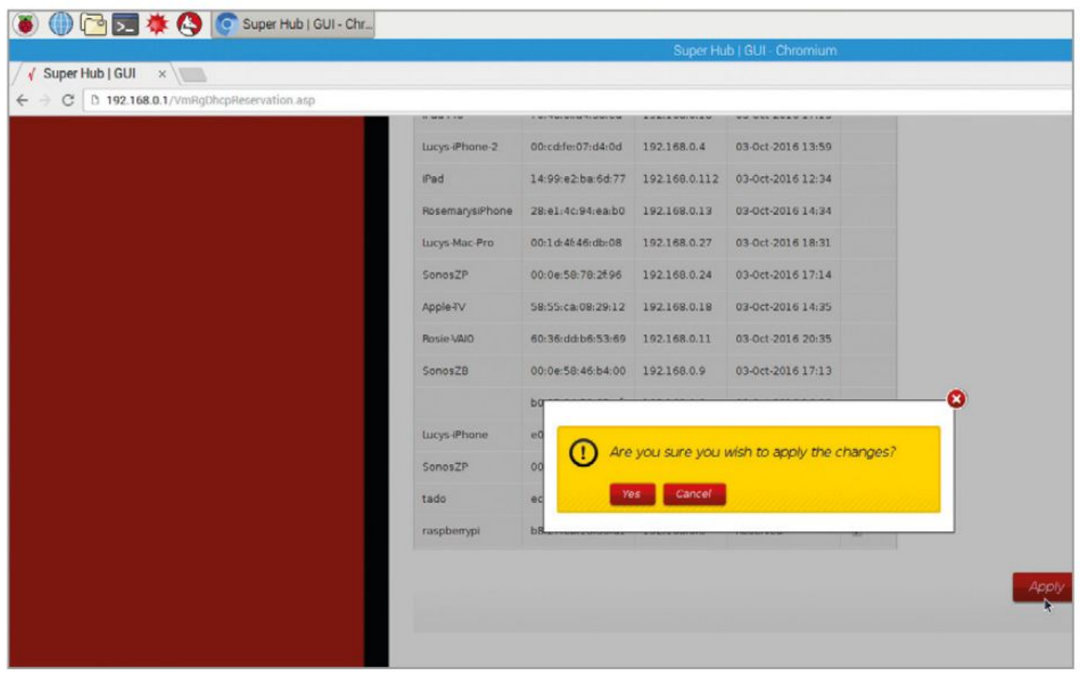
MANAGING YOUR DHCP DEVICES

Now that you have set up a DHCP Reservation, you need to understand how to manage the different devices that may need to use it and connect to the IP.

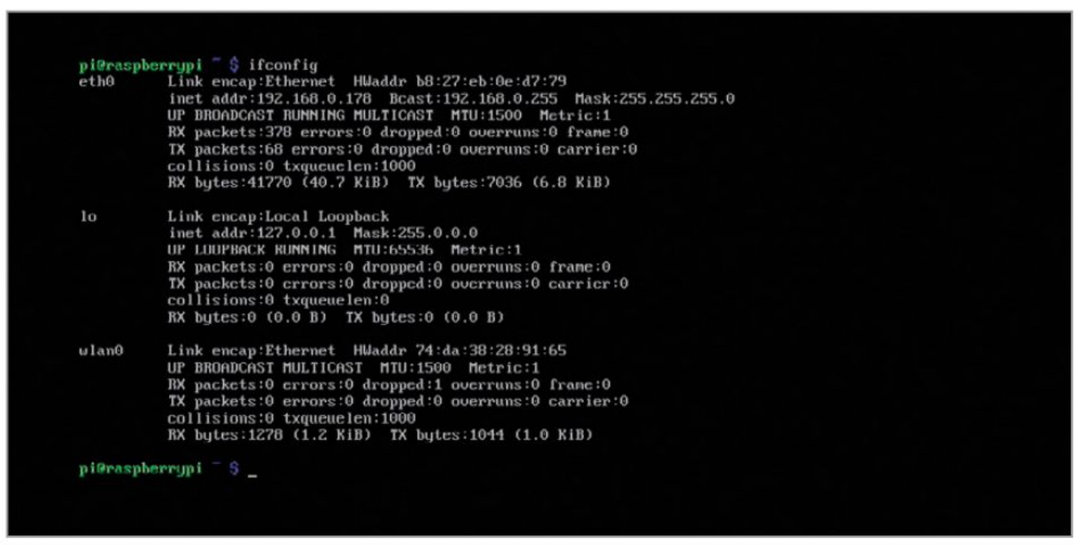
STEP 1 Now that you have set up your Raspberry Pi with a DHCP Reservation it will always connect to the router using that IP Address. You can check this at any time by entering: `ifconfig` into terminal. The router always knows that it is your Raspberry Pi by checking its MAC address. This is the set of six hexadecimal numbers next to "HWaddr" in ifconfig.



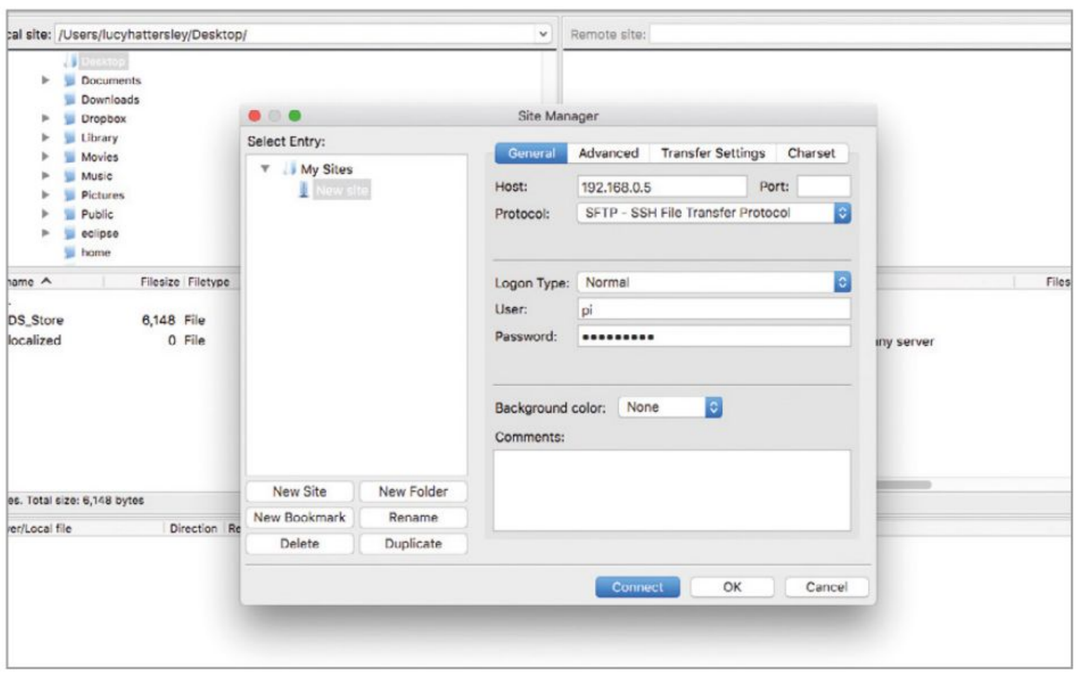
STEP 2 If you decide to free up that IP Address so other devices can use it, you can remove it from the router. Open the router web interface (192.178.0.1) and click Advanced Settings > DHCP Reservation. Scroll down to the IP Lease Table to find the Raspberry Pi. Select the check-box next to it and click Apply and Yes.



STEP 3 Your Raspberry Pi will still have the same IP Address but it isn't guaranteed to keep it. Restart your Raspberry Pi by entering: `sudo shutdown -r` now into a Terminal window. When it has restarted enter: `ifconfig` and you'll find you still use the same IP Address. Over time the router will use other slots for new devices until it reaches its maximum (255 by default). Then it will recycle disconnected devices.



STEP 4 Setting up DHCP Reservation is good practice because it ensures you Raspberry Pi is always going to use the same IP Address. It's also a good idea to write it down or use a label printer to create a label for the Raspberry Pi. You will need this number to connect to the Raspberry Pi via SSH, VPN or FPT (techniques we will use throughout this book).



Connecting to the Pi Remotely

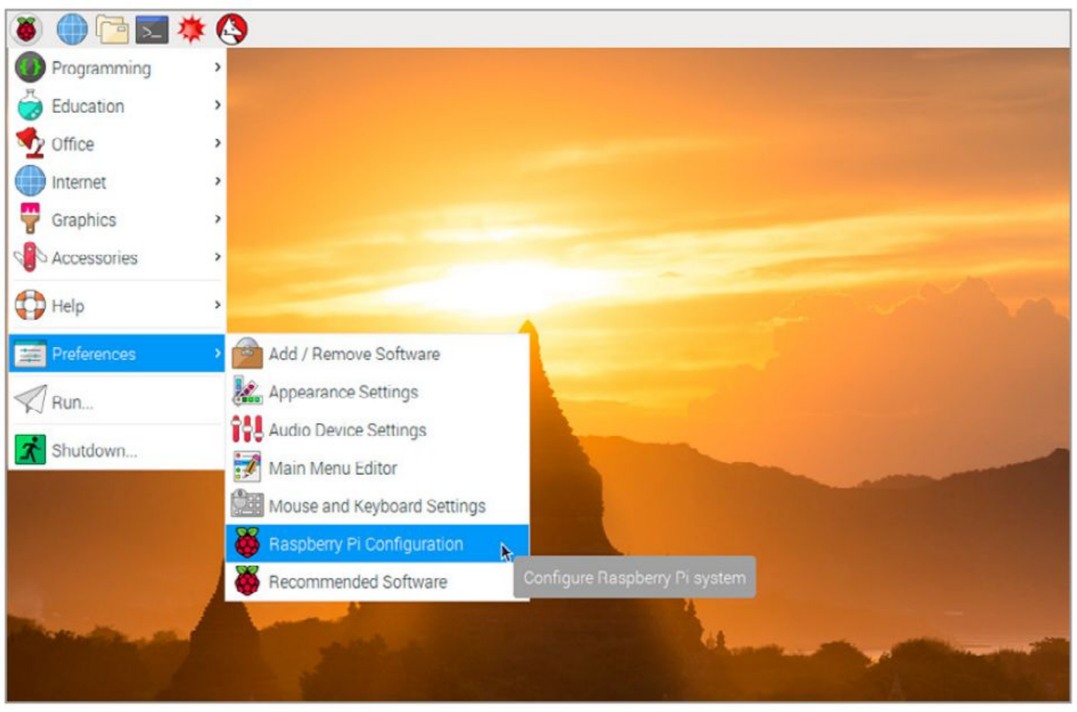
VNC (Virtual Network Computing) is a connection protocol technology that's used to remotely control one computer from another. With VNC set up, you can use your Raspberry Pi from your Mac or PC.

REMOTE CONTROL

The Raspberry Pi is easy to set up and use in projects around the home and, surprisingly, many of these projects don't need a keyboard, mouse or a monitor; these are known as Headless devices. VNC lets you control a Headless Pi from another computer.

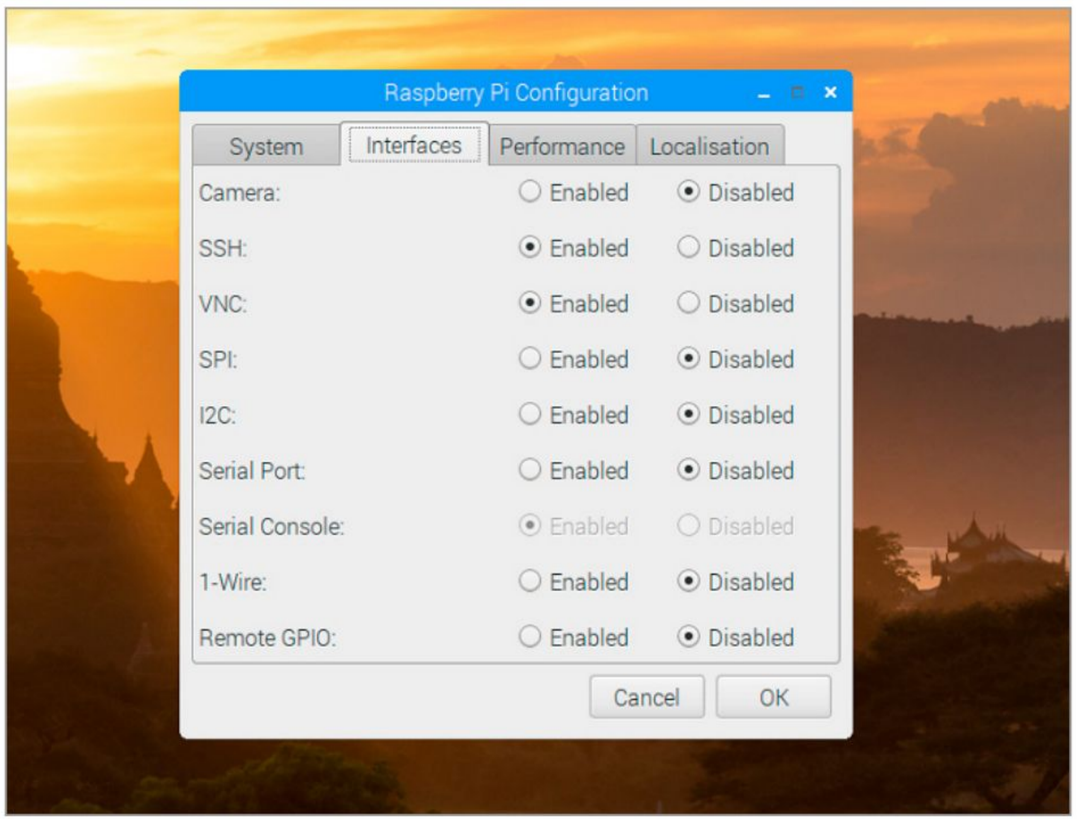
STEP 1

You will first need to ensure that your Pi is connected to a keyboard, mouse and monitor, and is connected to your Wi-Fi router, before you attempt to connect to it remotely. Start by clicking on the Raspberry Pi menu, then scroll down to **Preferences > Raspberry Pi Configuration**.



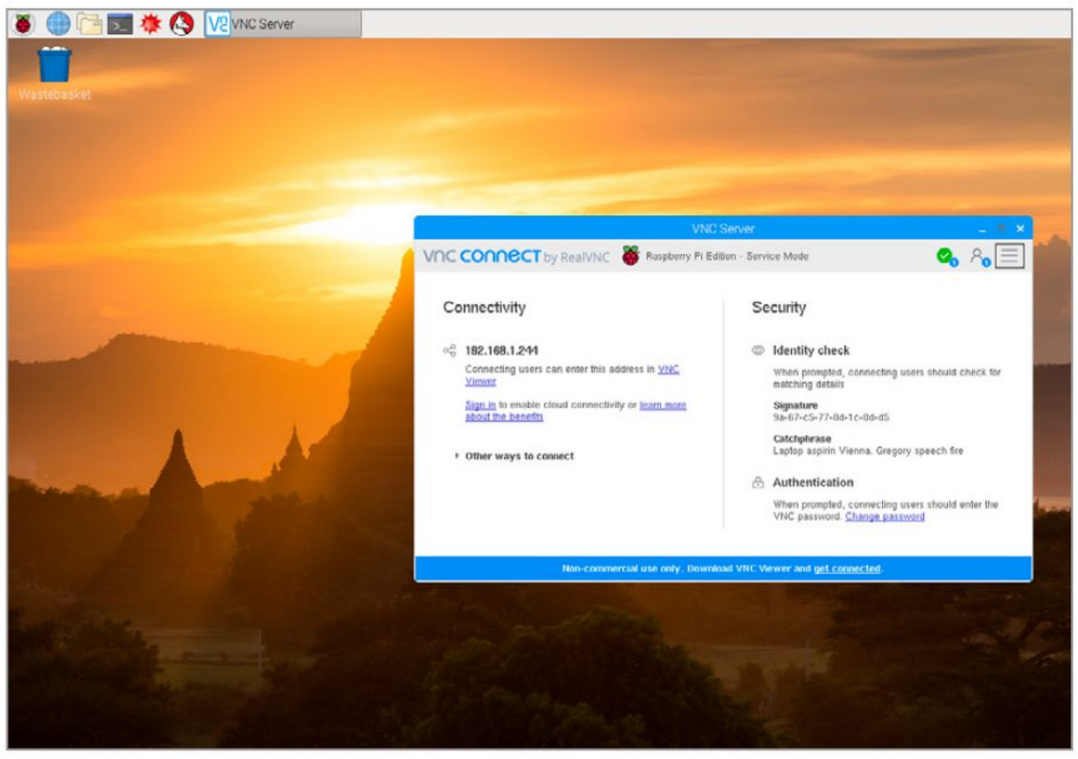
STEP 2

With the Raspberry Pi Configuration window open, click on the Interfaces tab, and ensure that the **VNC** option is **Enabled**. Click on the radio button next to Enabled to activate, then click on the OK button to close the window.



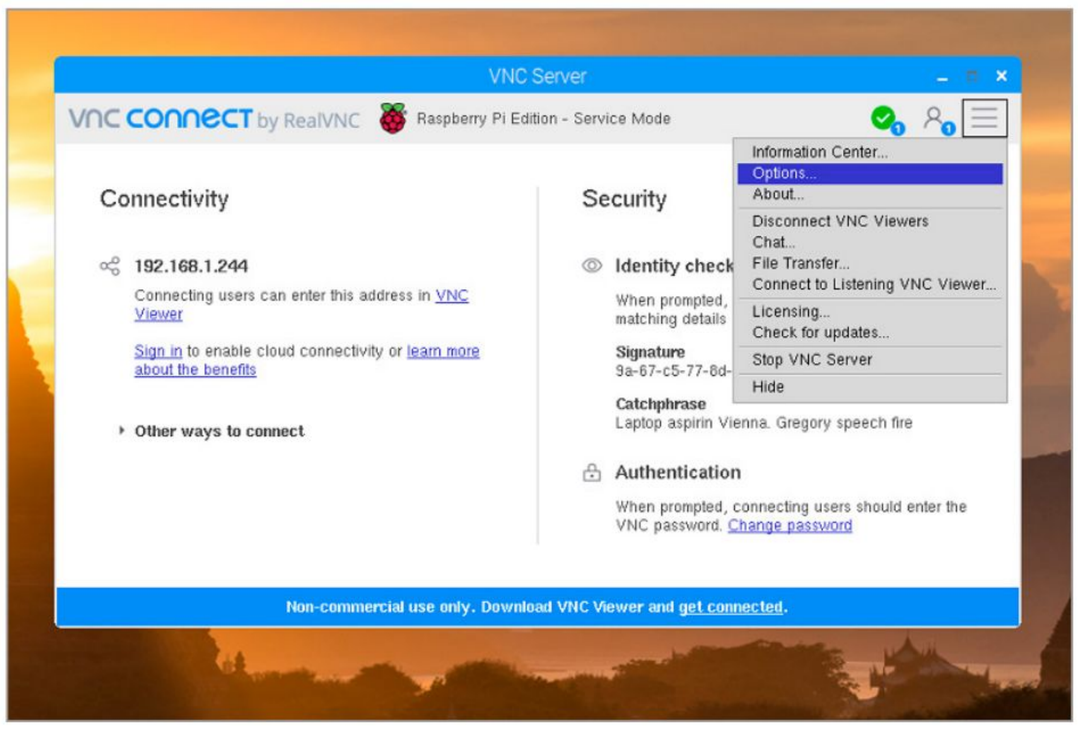
STEP 3

In the upper right corner of the menu bar you'll notice a VNC icon (next to the Bluetooth icon). This is the built-in VNC server, click the icon in the menu bar once to open the VNC Server window.

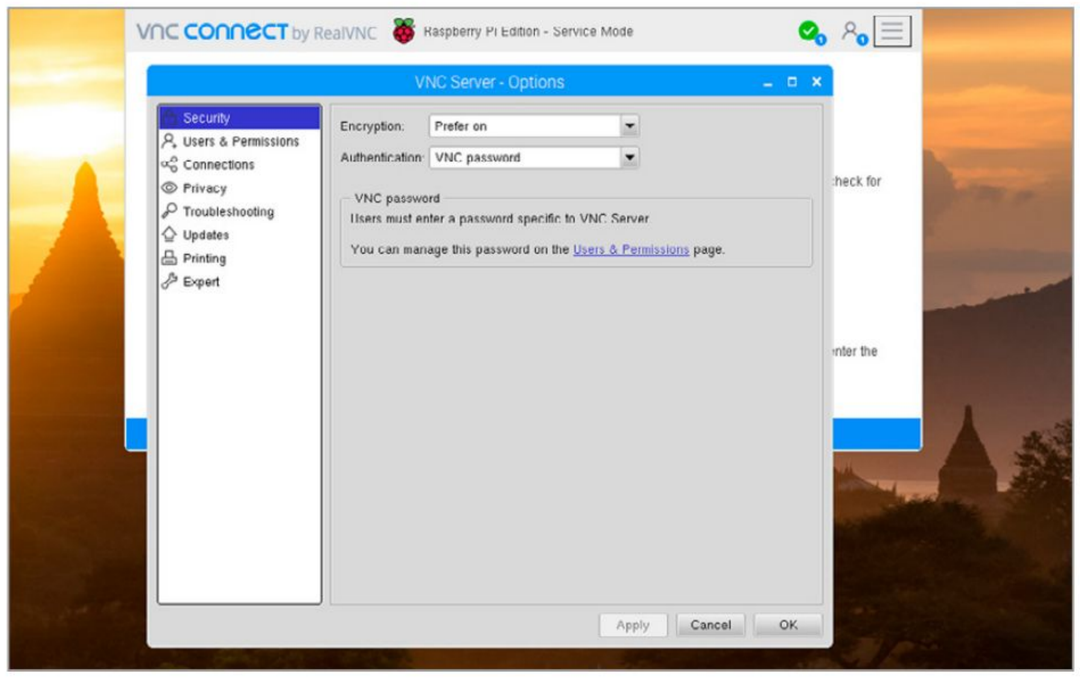


STEP 4

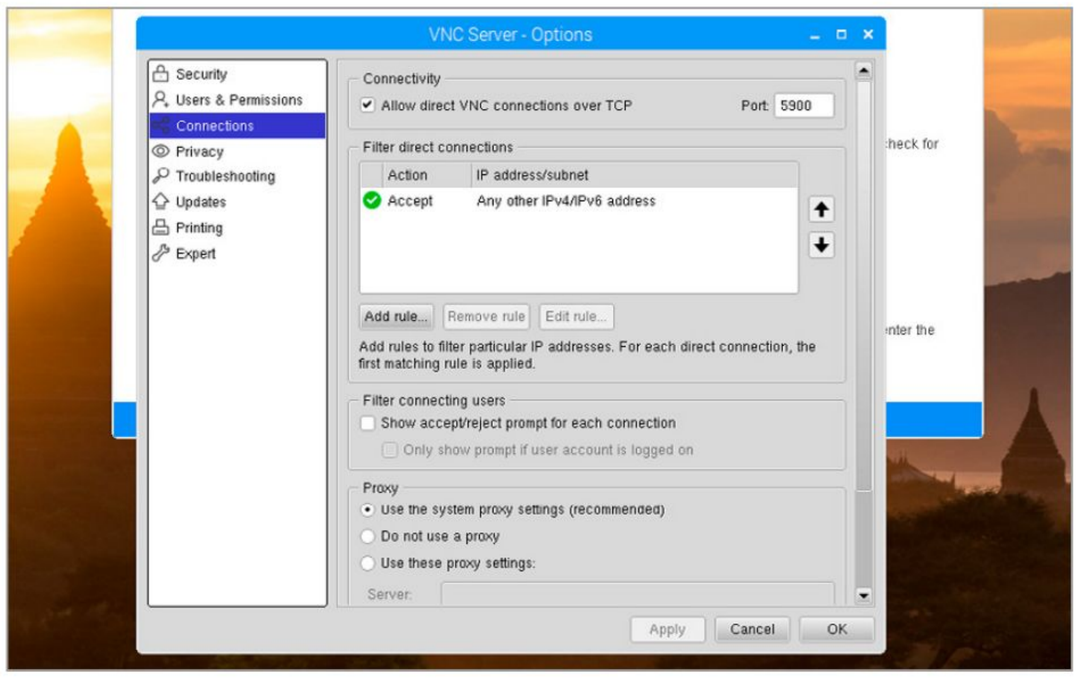
Make a note of the IP Address of the Raspberry Pi as detailed in the left-hand section of the VNC Server window; our example is 192.168.1.244. Now click on the three horizontal lines in the top right of the VNC Server window. From the sub-menu that appears, click on the **Options** entry. This will open a new window pop-up.



STEP 5 With the Security option highlighted on the left, ensure that the Encryption option, to the right, is set to **Prefer On**, and that Authentication is **VNC Password**. This may ask you to enter a password, which you will use when connecting remotely from another computer. Enter a password you'll remember.



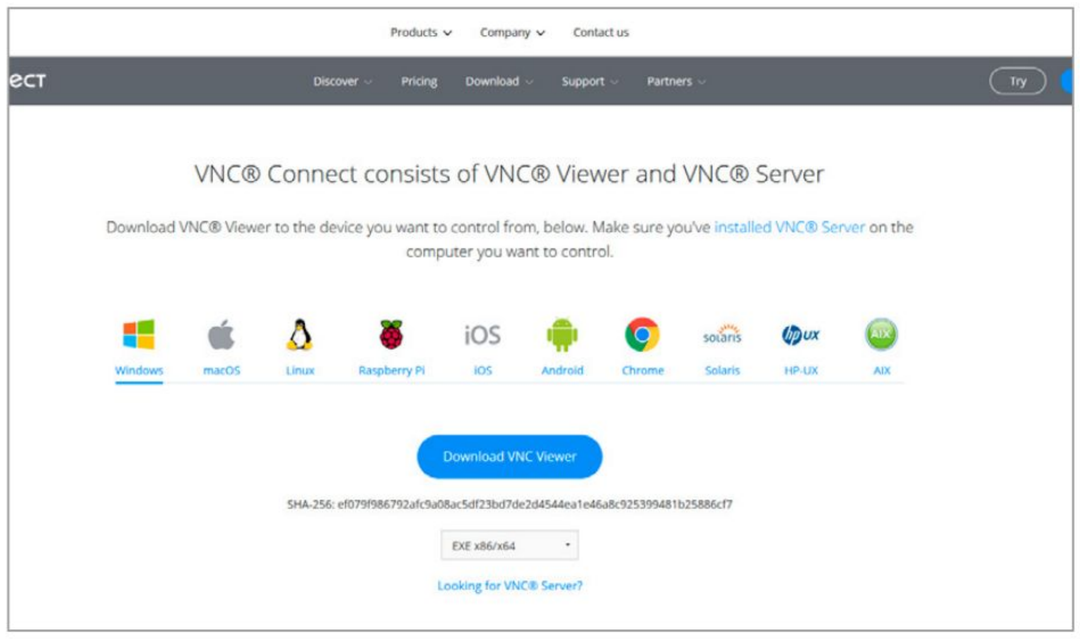
STEP 6 While still in the Options window, click on Connections in the left hand pane. In the right hand pane, ensure that the **Allow Direct VNC Connections Over TCP** is ticked, and that the Port is set to **5900**. It usually is, but sometimes it can be unticked if an update to software has been applied.



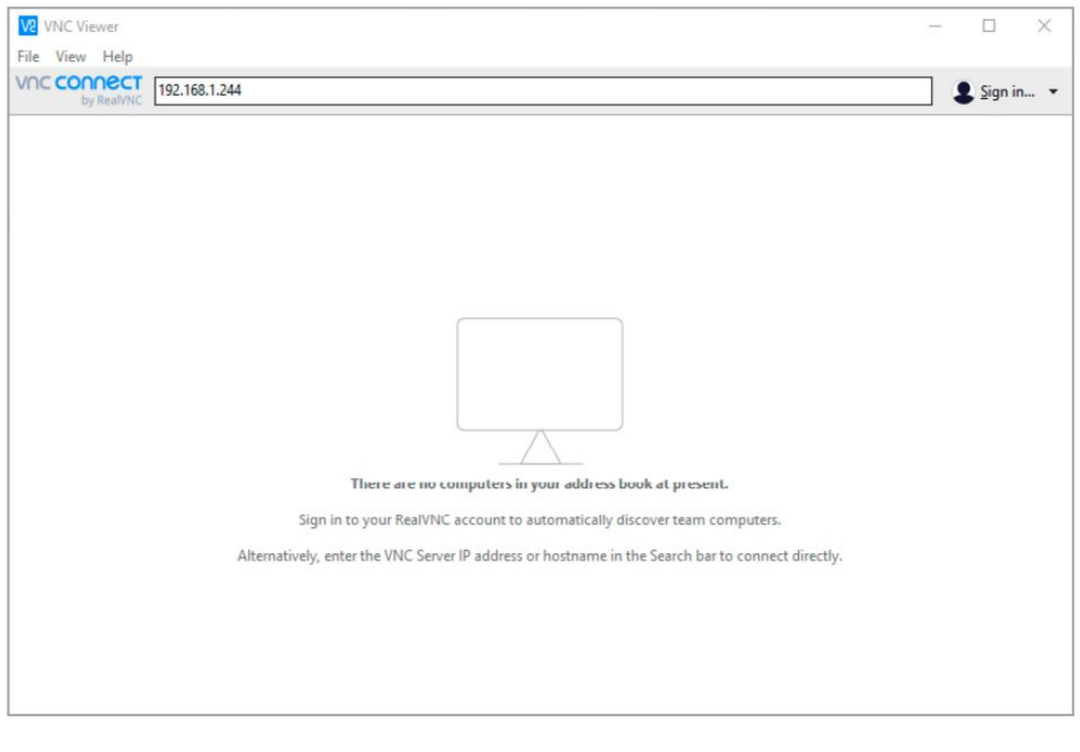
REMOTE VIEWING

The Pi-side of things has now been set up. Just remember the Pi's IP address, and get to your PC or Mac for this next part.

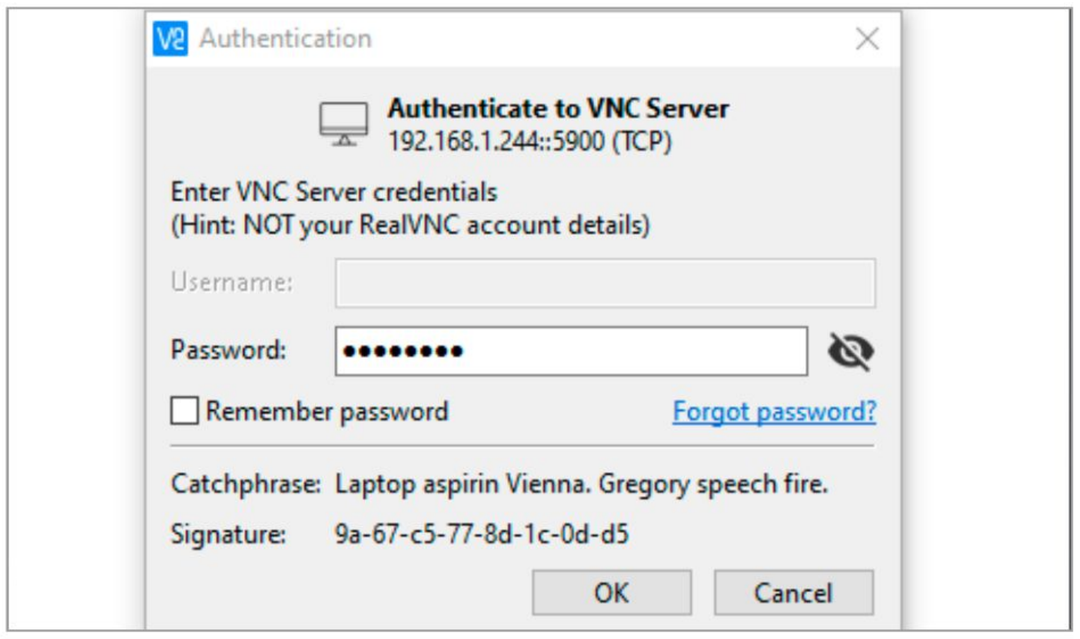
STEP 1 Reboot the Pi, then disconnect the keyboard, mouse and monitor; but keep the power connected. From your PC or Mac, open a browser and navigate to: **https://www.realvnc.com/en/connect/download/viewer/**. Choose your operating system and click the Download VNC Viewer app.



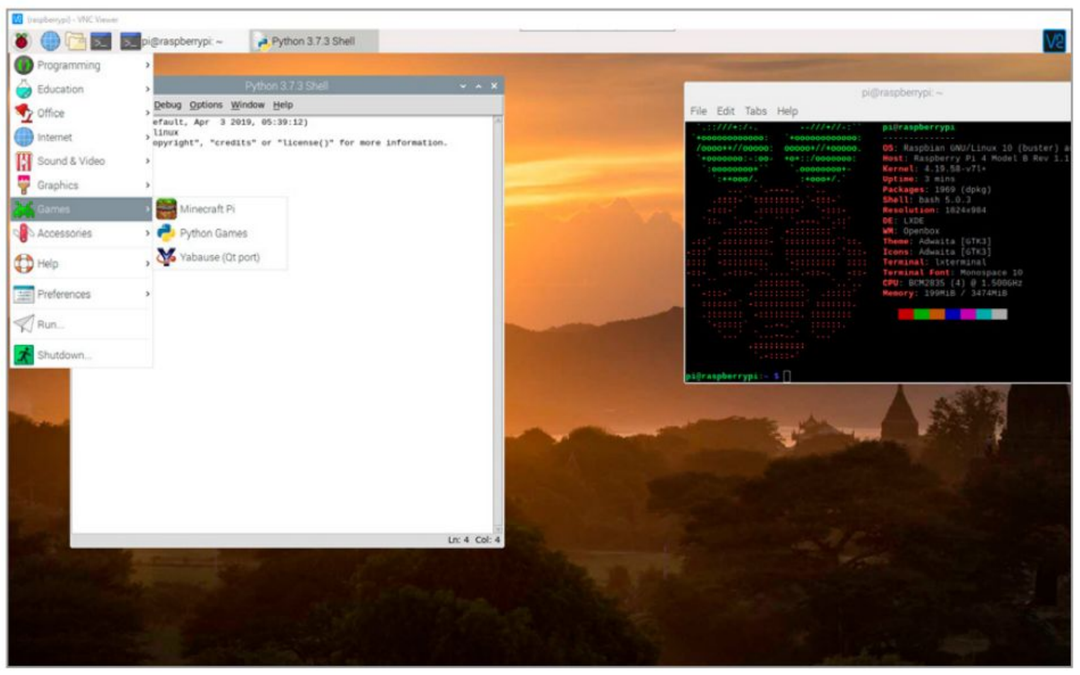
STEP 2 Follow the on-screen instruction to install VNC Viewer on your system, when it's finished, launch the app and choose whether you want to send anonymous data. In the main VNC Viewer window, enter the IP Address of the Raspberry Pi; ours was 192.168.1.244. Hit Enter when you've typed in the IP address.



STEP 3 In the new pop-up window, enter the password you created through the Raspberry Pi's VNC Server options page. You can choose to Remember Password if you want, but if the computer you're using is shared, you can always opt to leave the option unticked. Click OK when ready.



STEP 4 You may be asked to confirm connection to a remote computer, click Continue to make the connection. You are now connected remotely to your Raspberry Pi. As long as the Pi is powered up, and has access to the Wi-Fi signal of the router, then you can place it anywhere and get access without using a mouse, keyboard, or monitor.



Using APT to Install and Remove Programs

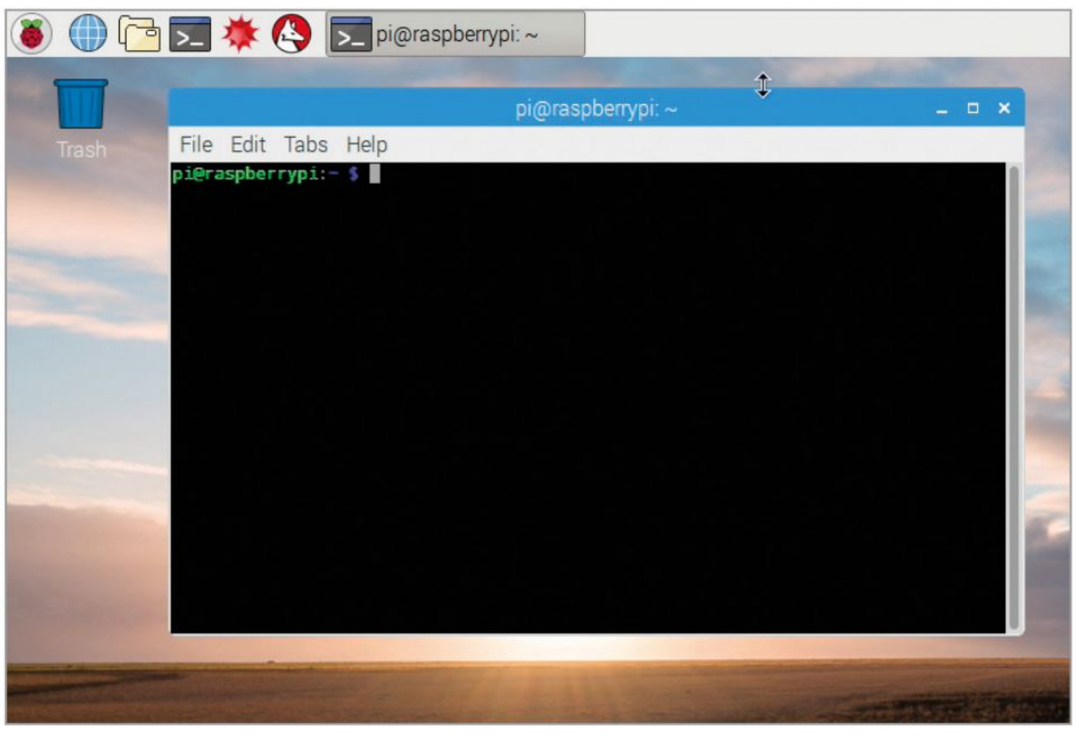
When you start to get serious with your Raspberry Pi, you'll need to install software that is available for Linux, but not part of the Pi Store. These programs are installed from the command line using a service called APT. Learning how to use APT is a vital part of using your Pi.

ADVANCED PACKAGING TOOL

Add / Remove Software is great for finding and installing software but most of the time you manage apps on a Raspberry Pi using a program called APT (Advanced Packaging Tool) using the apt command.

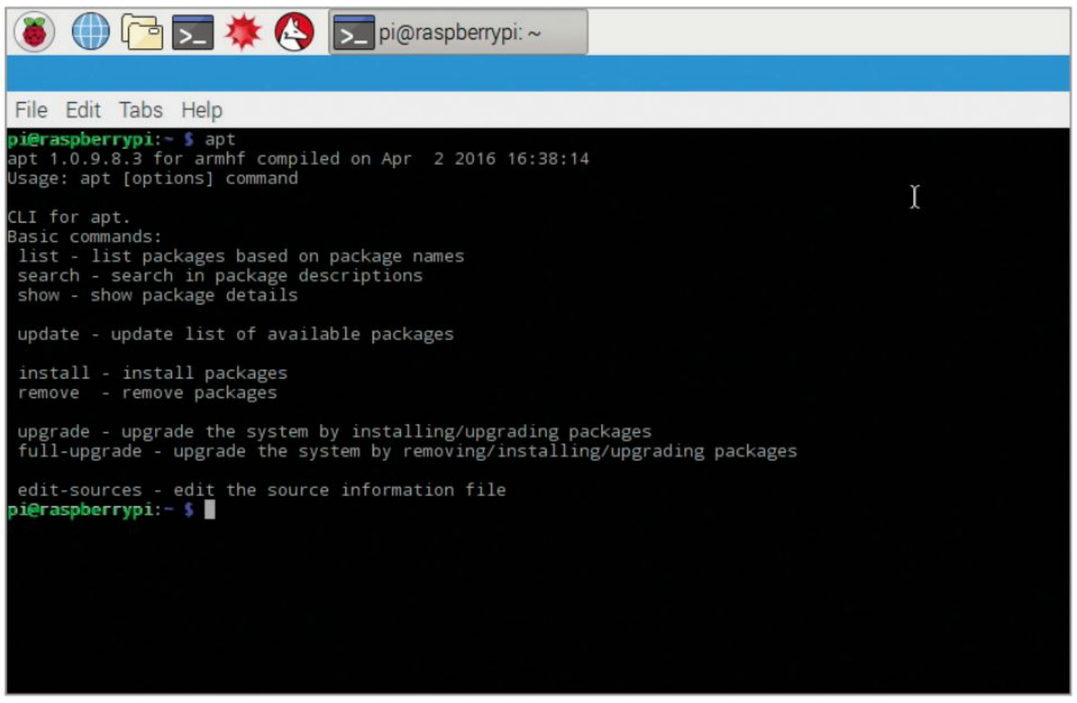
STEP 1

To manage the apps on your Raspberry Pi you'll need to use a tool included with Raspbian called APT (Advanced Packaging Tool). APT is a command line tool so you need to either start your Raspberry Pi in the command line or click on the Terminal icon.



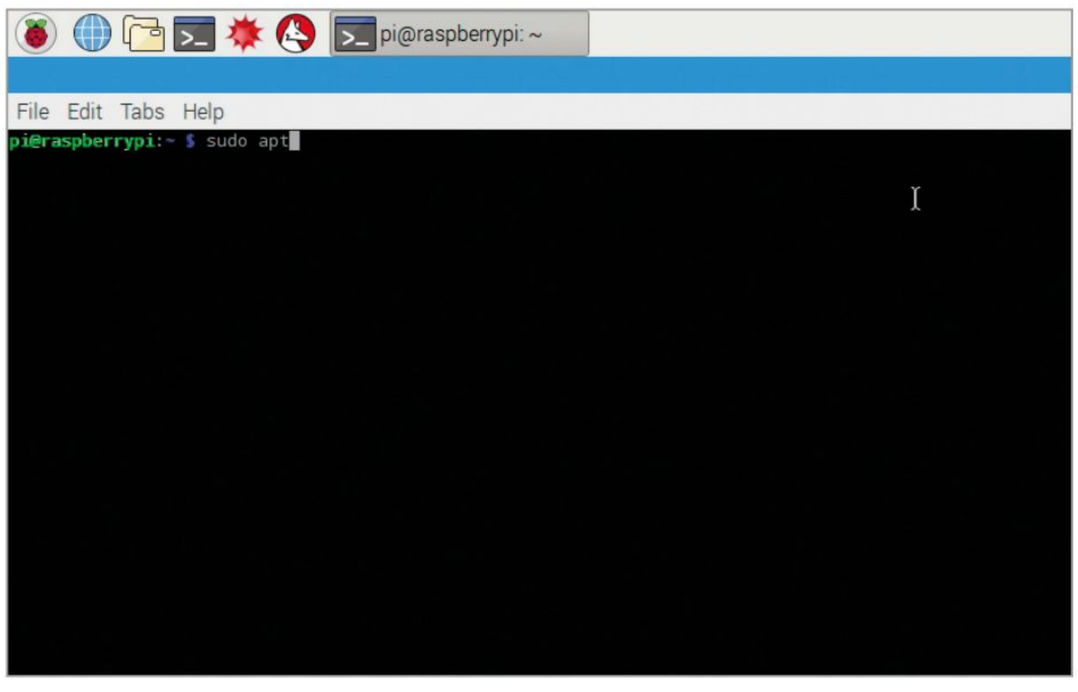
STEP 2

The command used to control packages in Raspbian is apt Type `apt` into the terminal and press Return to get a description of what the app does. The instructions tell us that the "most frequently used commands are update and install". You may also hear about "apt-get" which is an older version of the same tool. It works in much the same way.



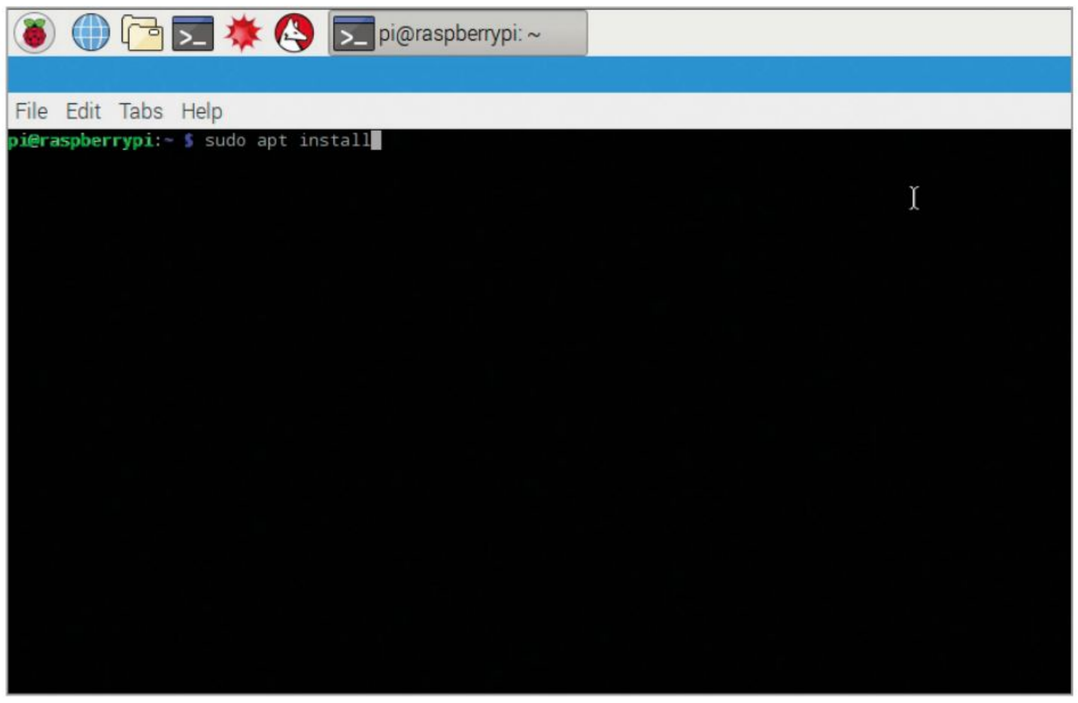
STEP 3

The command used to add new software to your Raspberry Pi is `apt install` followed by the name of the package you want. However, because apt adds (or removes) files outside of your home directory you need to preface apt with the word `sudo`. Almost all of the time you will need to type `sudo apt` when using this tool.

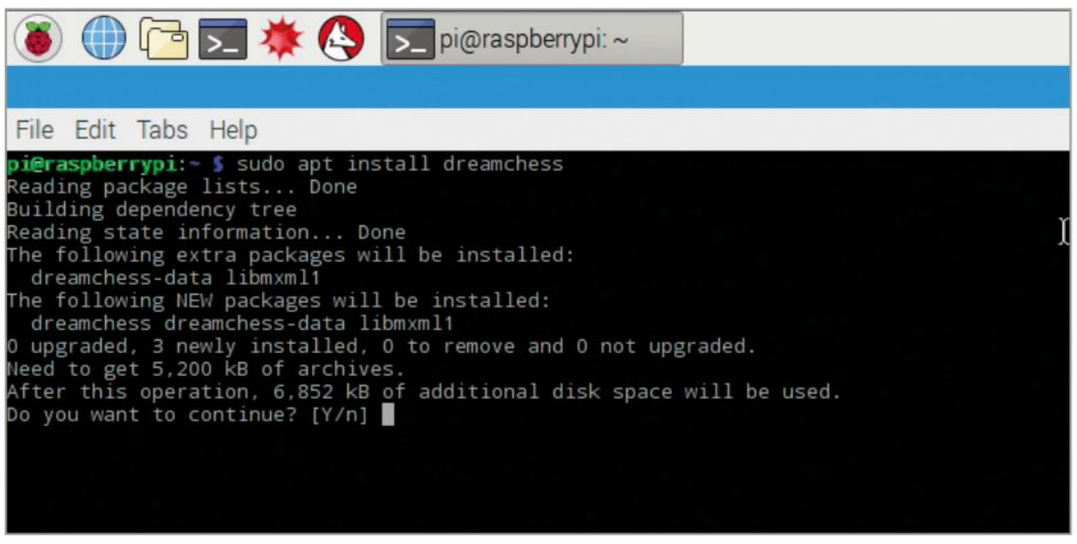


STEP 4

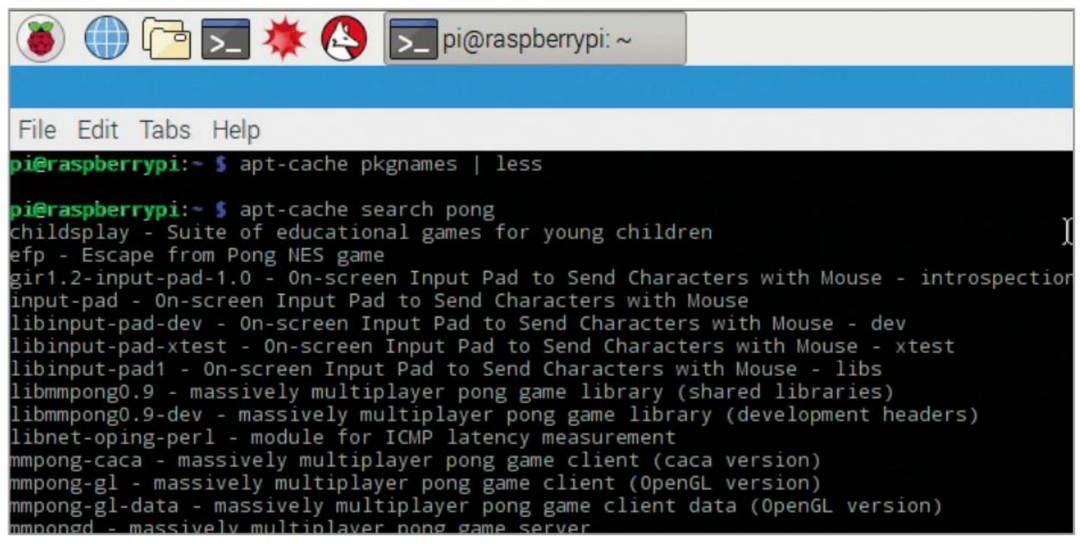
Beginning a command with "sudo" runs the command as a super user, "super" meaning the one above your Pi account, rather than one with super-hero properties. However, if it helps you to think of it that way that's fine. So to install a new program you type: `sudo apt install` followed by the name of the app you want.



STEP 5 You're going to install a chess game called Dreamchess. Enter: `sudo apt install dreamchess` and press Return. Descriptive text will run up the screen. You may see "Do you want to continue [Y/n]?" Enter: `y` and press Return to install the application. You'll find it by choosing: Menu > Games > Dreamchess.



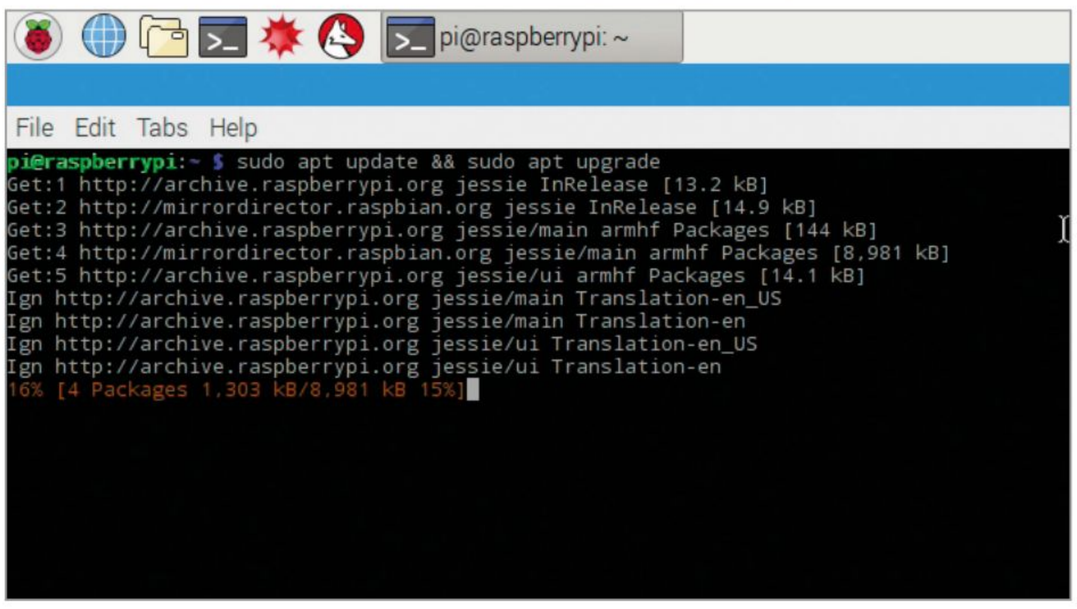
STEP 6 You can use apt-cache to search for apps you can install and remove. Enter: `apt-cache pkgnames` to view all the packages available. Typing `apt-cache pkgnames | less` enables you to view a page at a time (press any key to scroll). You can use apt-cache search to find packages, enter: `apt-cache search pong` to view a list of Pong games you can install.



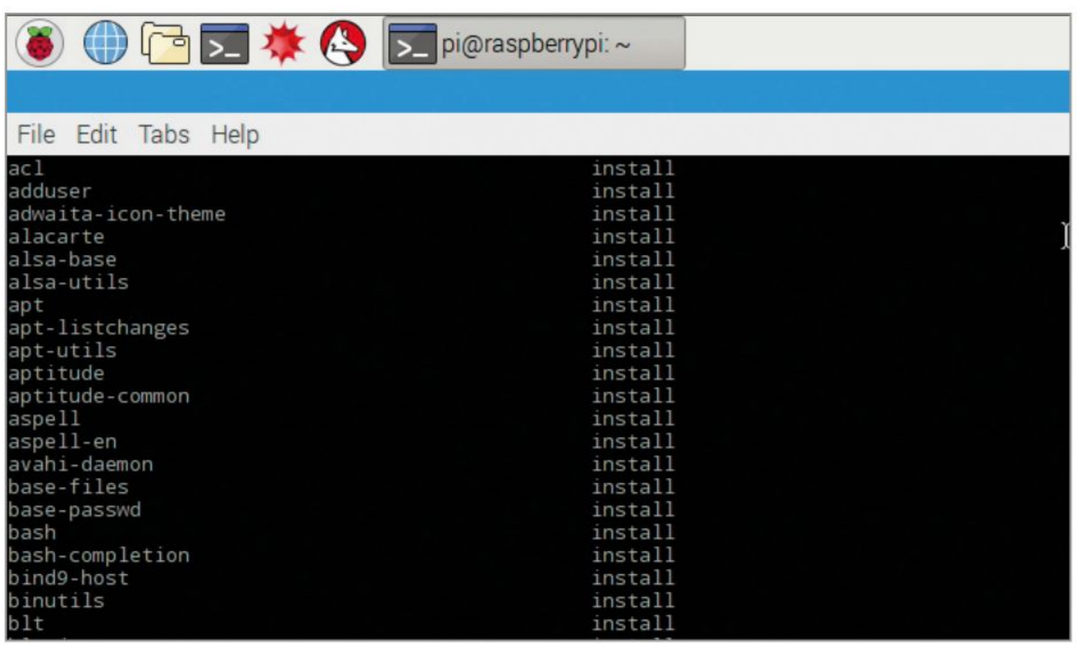
MORE APT COMMANDS

The apt command is also used to list, and delete, any apps you have installed.

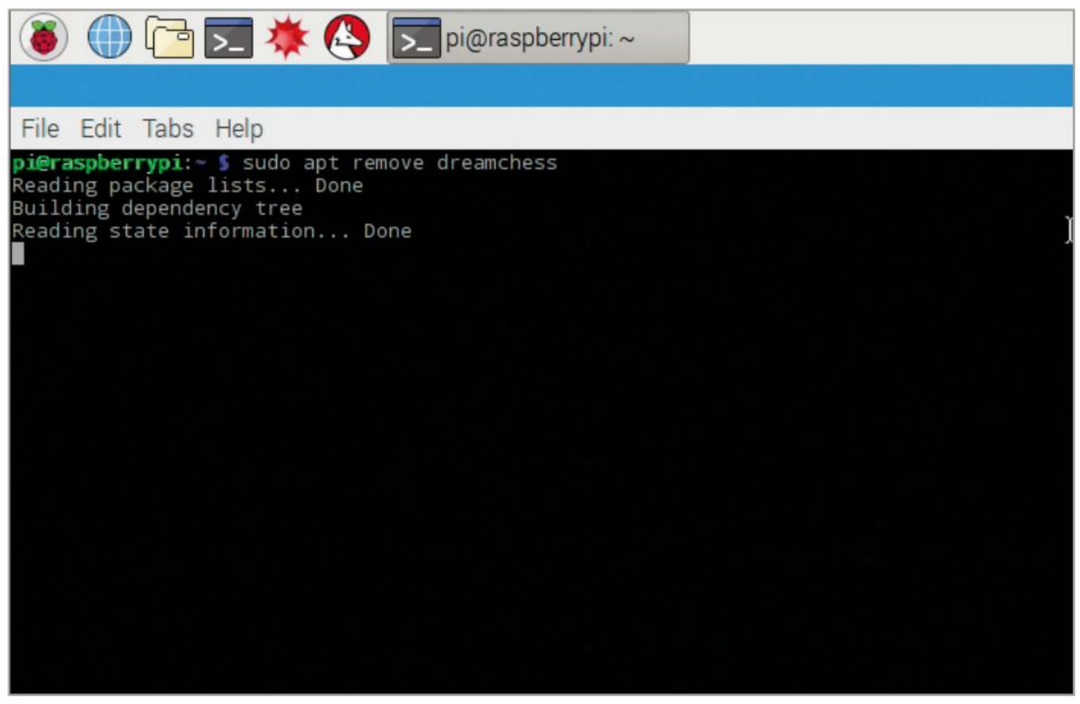
STEP 1 Sometimes you will try to install an app only to get an error that the file cannot be found. In this case you need to update apt so it has the latest links. Enter: `sudo apt update`. This doesn't update the apps you've installed, it just gets a newer listing of apps. To update your apps to the latest versions enter: `sudo apt upgrade`. You'll often find both commands rolled into one: `sudo apt update && sudo apt upgrade`.



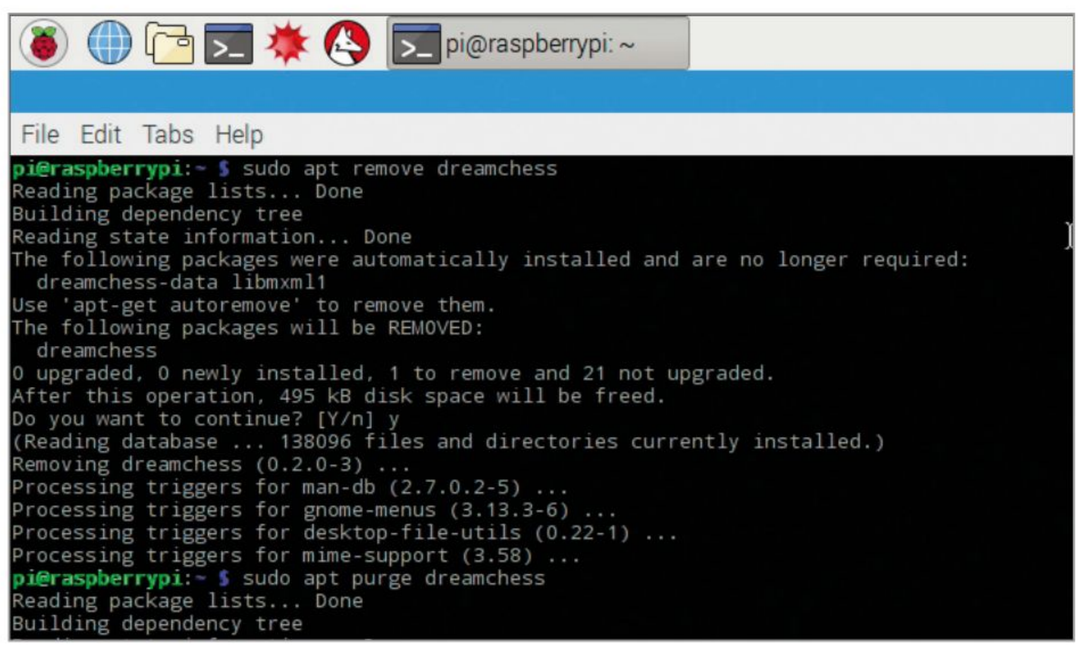
STEP 2 To view the packages you've installed you use a different command called dpkg. Enter: `dpkg --get-selections | grep -v deinstall` to view all of the packages on your Raspberry Pi. Place `| less` after it to view one page at a time. Or `dpkg --get-selections | grep -v deinstall > ~/Desktop/packages.txt` to save the list as a text file on your desktop.



STEP 3 Packages are uninstalled from your Raspberry Pi using the remove command. Enter: `sudo apt remove dreamchess` to start deleting the chess game that you installed. An alert saying "Do you want to continue [Y/n]?| appears, enter: `y` and press Return to delete the program.



STEP 4 Uninstalled packages using apt doesn't remove all of the files. Some are left in case you decide to reinstall the program later. To completely remove a package from your Raspberry Pi use the purge command. Enter: `sudo apt purge dreamchess` to remove all of the supporting files. You can also use `sudo apt clean` to tidy up your packages and free up some drive space.



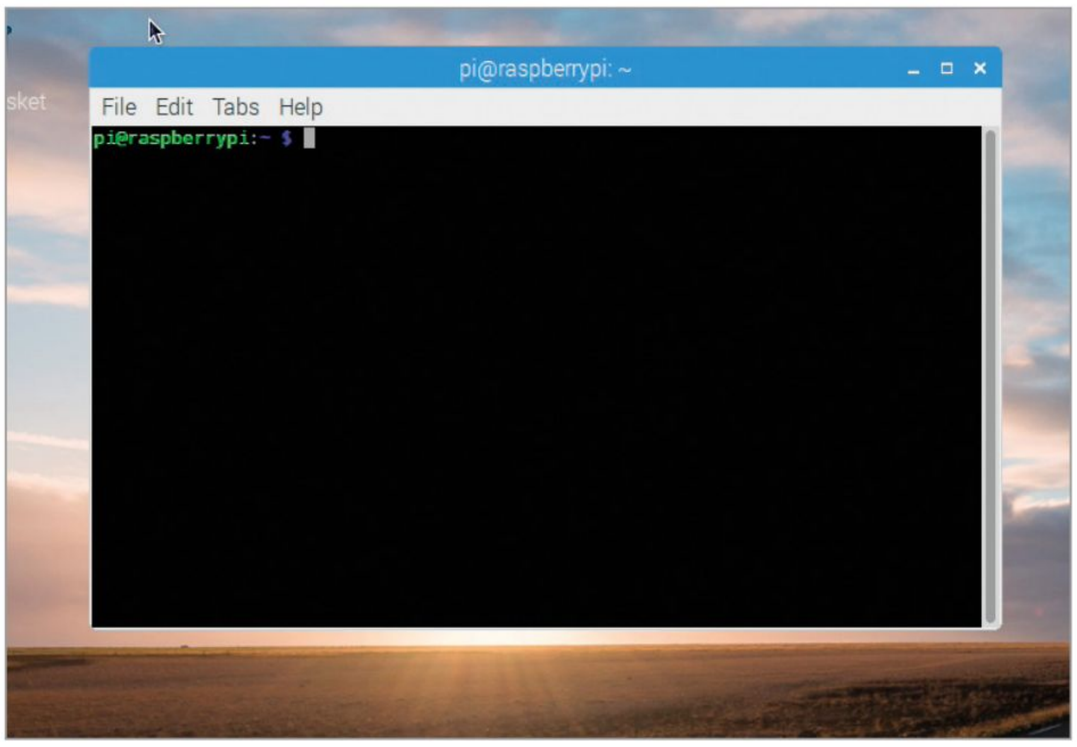
Get More from the Terminal App

Terminal is an app in Raspbian that enables you to access the command line and issue text commands to your Pi. You'll spend a lot of time in Terminal, so it's a great idea to get to know the app and set it up to work your way.

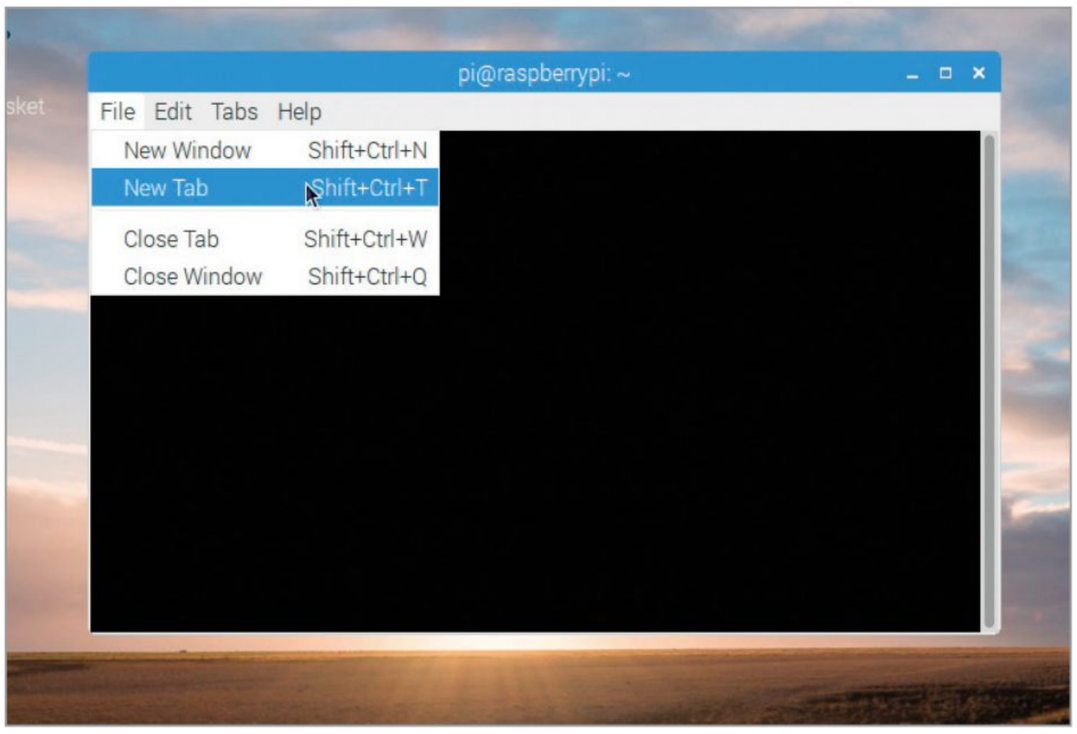
CUSTOM TERMINAL

The Terminal, or command line, is a powerful environment for accessing the system. Careful, here be dragons.

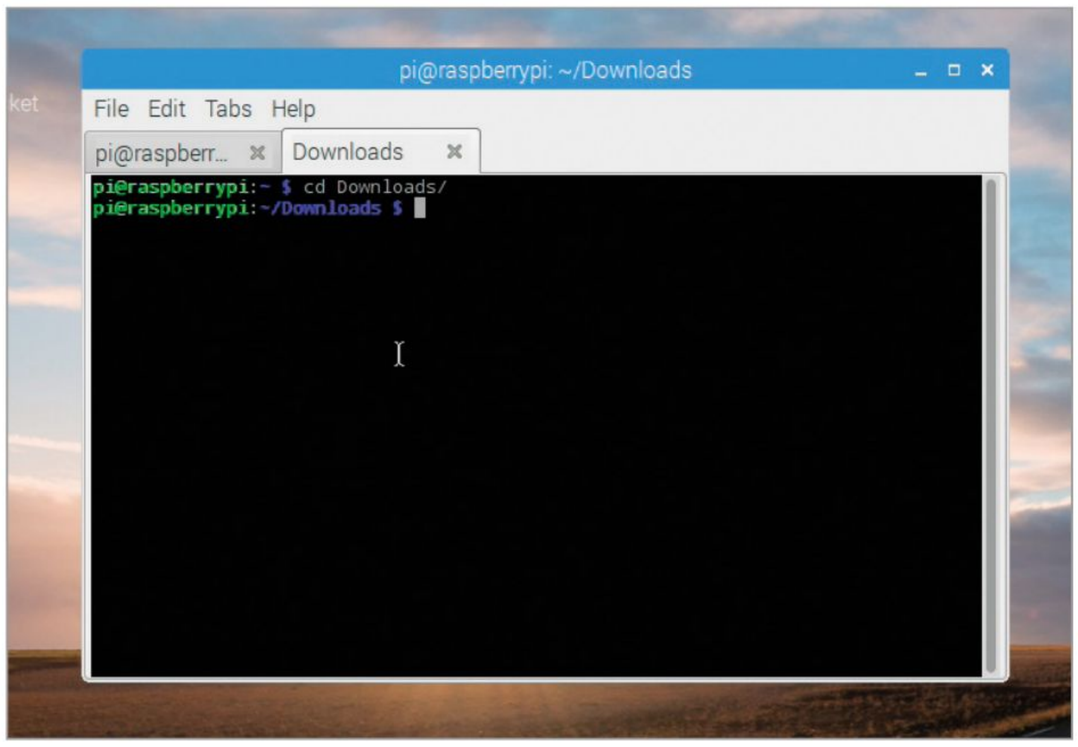
STEP 1 Start Terminal by clicking the Terminal icon in the Application Launch Bar. You can also start Terminal by pressing Control-Alt-T or choosing Menu > Accessories > Terminal. Terminal emulates the old style video terminals from before desktop interfaces; so by default it displays bright (mostly green) on a black background.



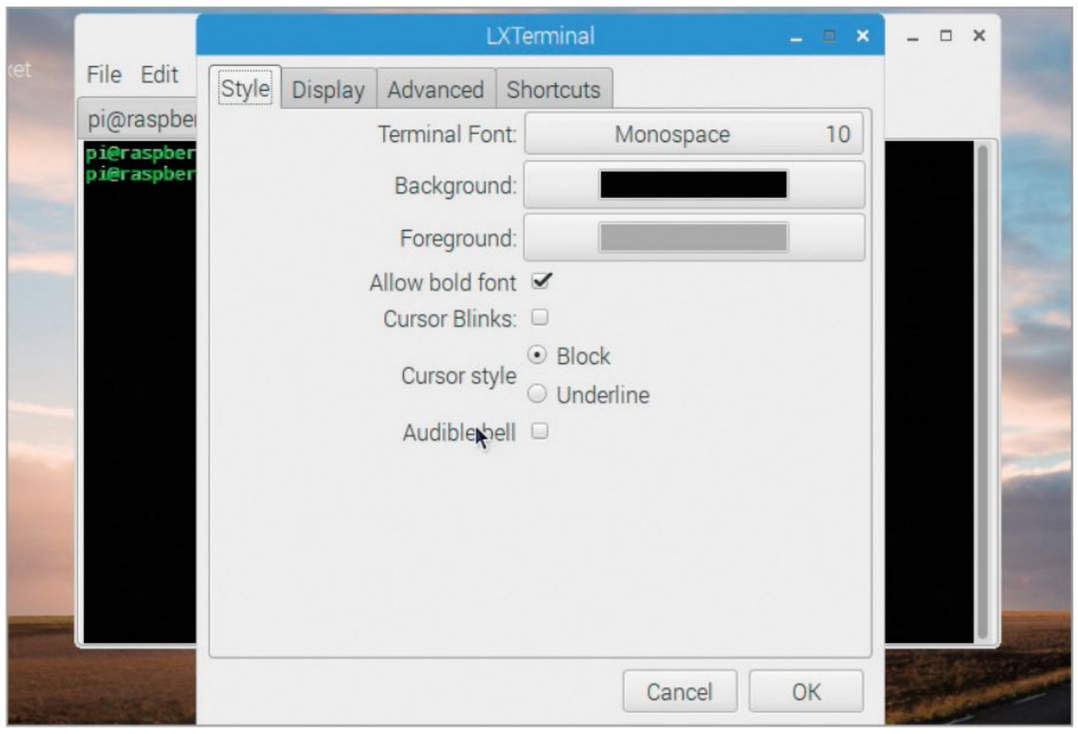
STEP 2 The first thing most people overlook is the ability to run multiple terminals at once in different tabs. Choose File > New Tab (Shift-Control-T) to create a new tab. Navigating multiple directories simultaneously can be a challenge in Terminal and tabs makes it that much easier.



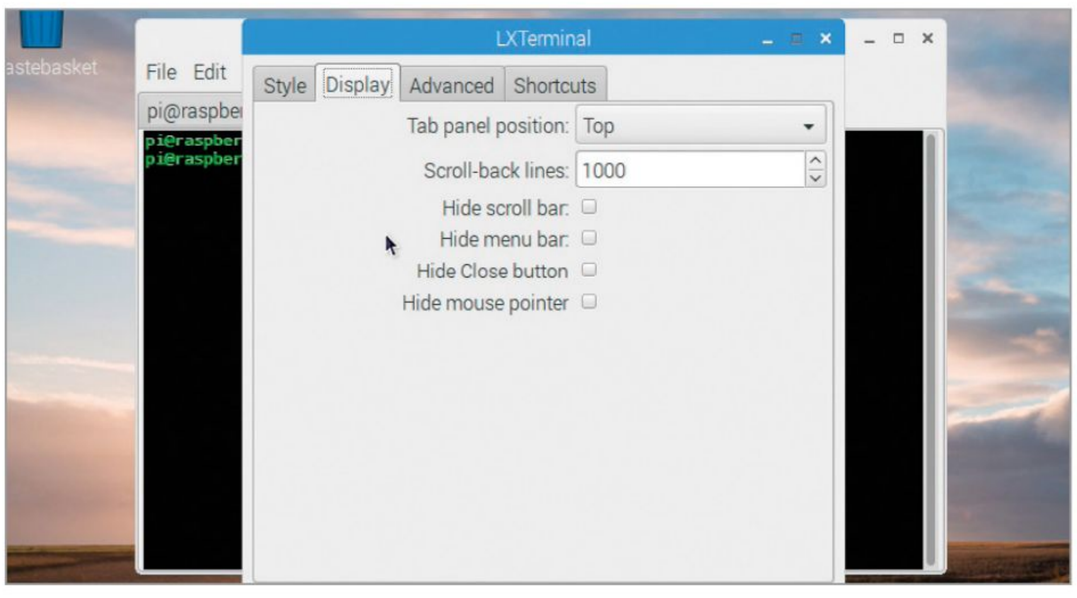
STEP 3 Naming tabs makes them much easier to recognise. Choose Tabs > Name Tab and enter a name for the tab such as "Home" or "Downloads". Click OK to set the name. You can quickly switch between tabs using Control-Page Up and Control-Page Down and rearrange them using the Tabs Move Tab Left / Move Tab Right options.



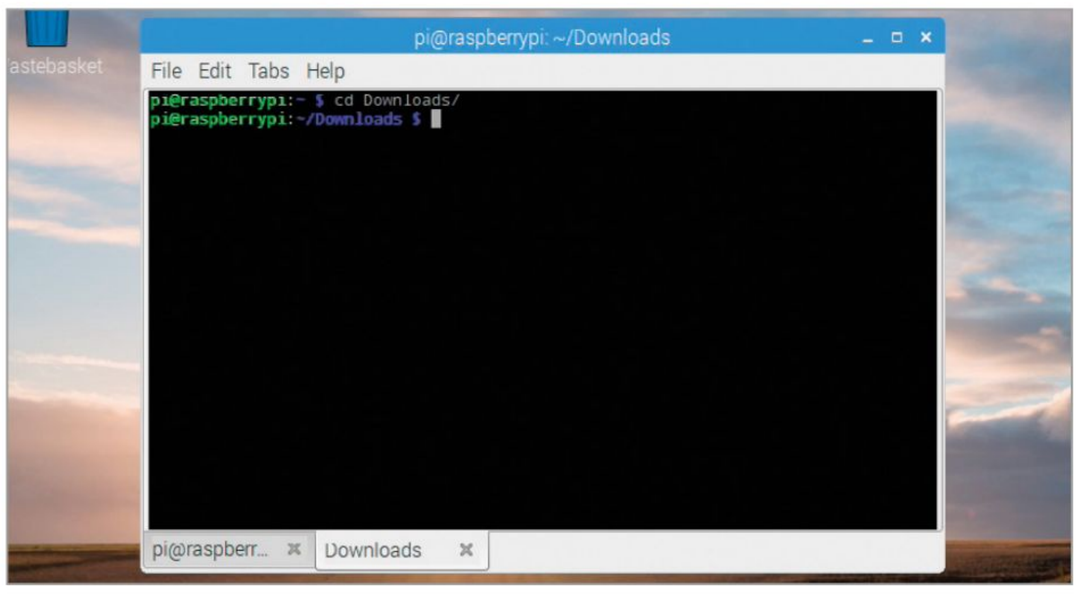
STEP 4 Choose Edit > Preferences to personalise the look and feel of LXTerminal. One of the best tricks is to click Background and drag the Opacity slider to the half-way point. This enables you to view through the terminal background and see the windows below. You can also personalise the colour of the background.



STEP 5 While still in Preferences click the Display tab. Here you can adjust the Tab panel positions, placing them on the Left makes them more prominent. You can also adjust the scroll-back line count. This is the number of lines you can scroll up with using the mouse, not the history accessed by the Up and Down arrows.



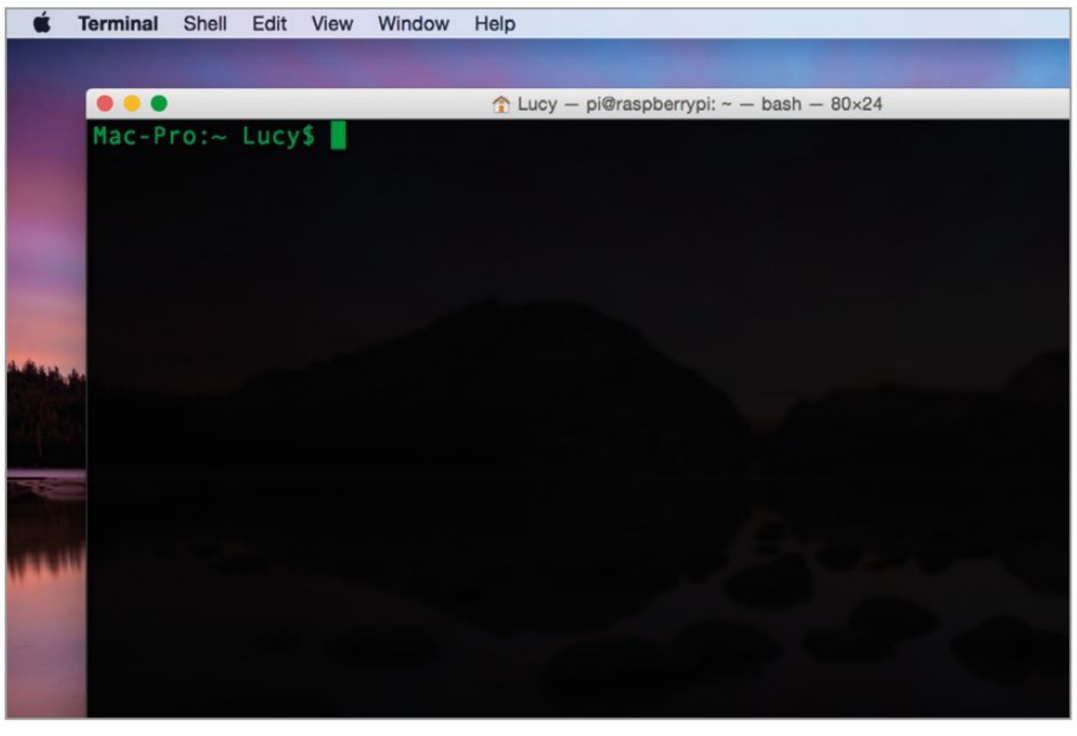
STEP 6 If you prefer a more minimalist approach try moving the Tabs to the bottom and selecting the Hide scroll bar, Hide menu bar and Hide Close button options. When combined with a low opacity background this makes for a subtle terminal window. You can still access menu settings using a right-click on the mouse.



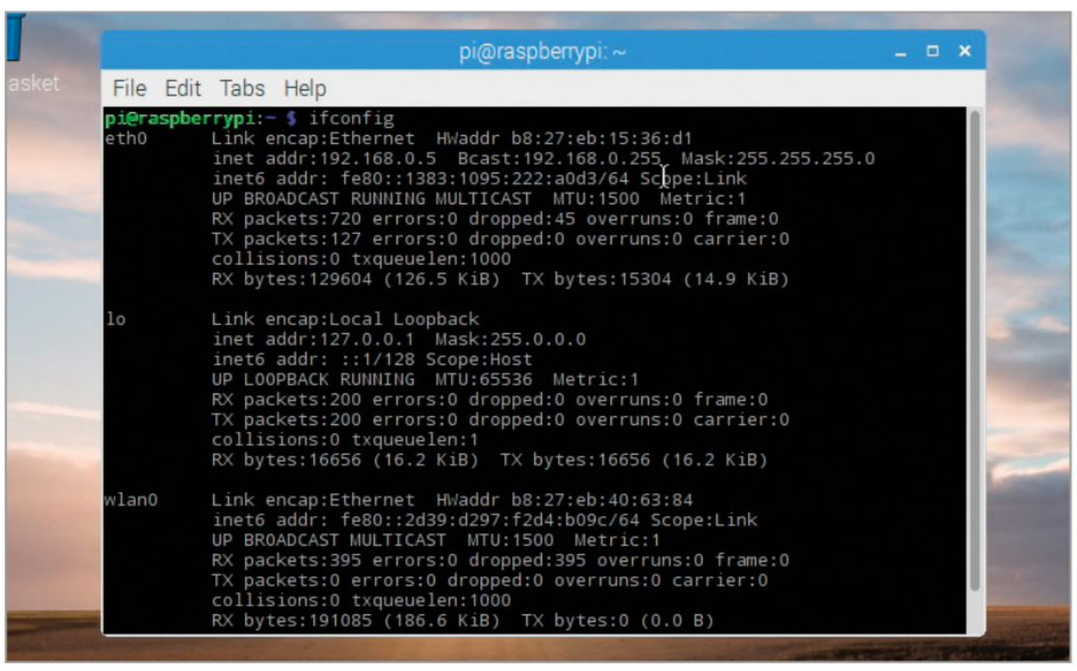
CONNECT USING SSH

Use a different computer to control your Raspberry Pi.

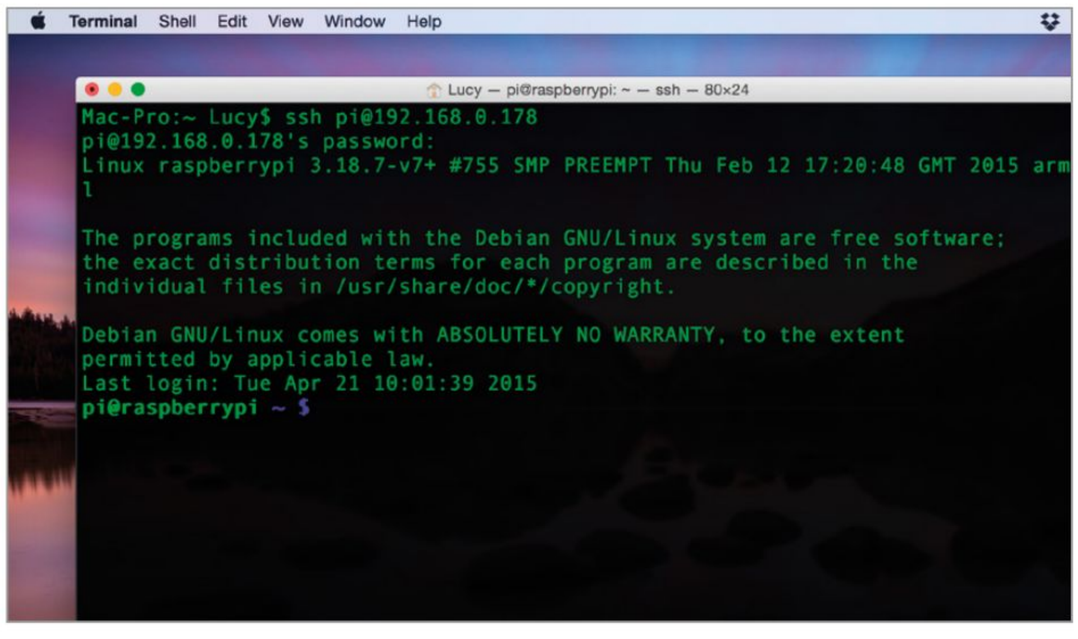
STEP 1 If you have your Raspberry Pi on the same network as a Mac or Linux-based computer, you can control it using the Terminal program on that computer. We're going to use Terminal in macOS here but the process is the same on a Linux machine. Open the Terminal app on your computer.



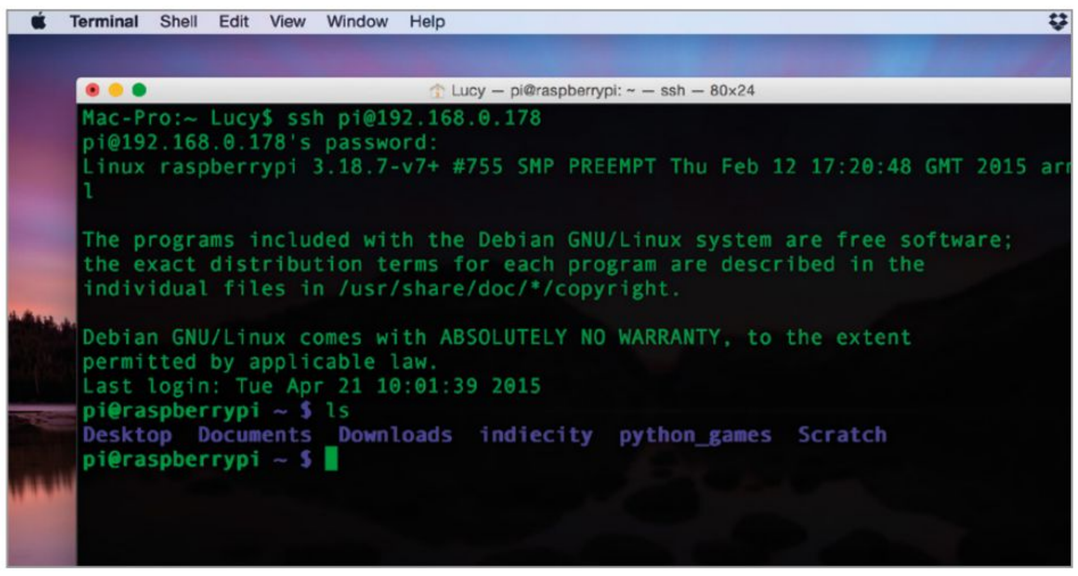
STEP 2 Enter: `ifconfig` into the Terminal app on your Raspberry Pi. This will let you know which IP address it is using. Look for the four blocks of digits following "inet addr:'. They should begin with 192.168.0 followed by a three-digit number. Ours is 192.168.0.179. You need to use that number in Terminal on your Mac to connect.



STEP 3 Switch back to Terminal on your Mac or Linux machine and enter: `ssh pi@192.168.0.178` (using your own IP address). The "pi" bit is the name of the default user account, so if you have changed that it needs to match your user account on the Raspberry Pi. You'll be asked for a password. This is the password that you use to log in to the Raspberry Pi (not your Mac or Linux computer).



STEP 4 You are now logged in to your Raspberry Pi and can enter commands directly into the Terminal on your Mac or Linux computer. Unlike remote networking you don't see the commands being entered on the screen of the Raspberry Pi, you are accessing the computer from behind the scenes. Many people prefer to set up the Raspberry Pi so they can control it from a more powerful computer. Enter: `exit` to close the connection.



Using the File Manager

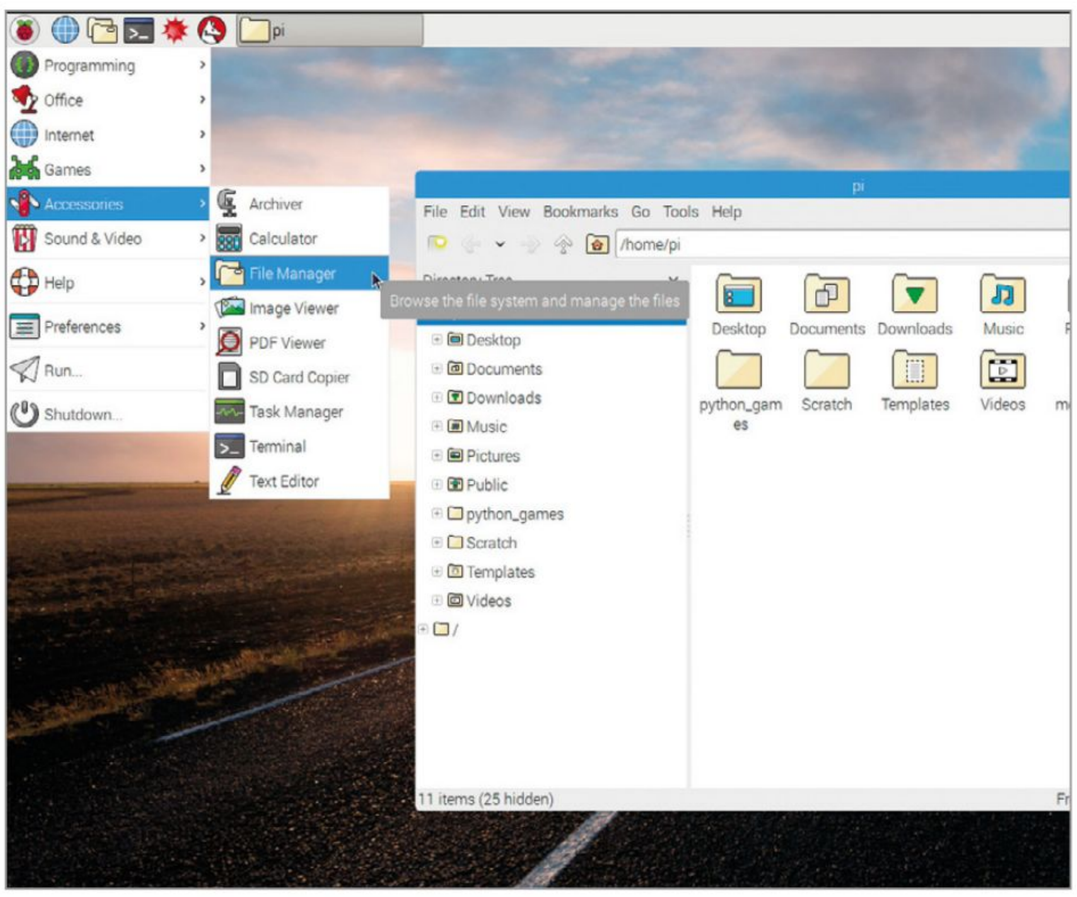
Because you'll be creating programs on your Raspberry Pi, you'll need to know more about the file system than you would on other computers. One vital tool to become familiar with is the File Manager. This is used to find, move and remove files from your Raspberry Pi.

VIEWING AND MOVING FILES

You can move, manage and delete files using the Command Line but like all modern operating systems Raspbian has a program to help you manage your files. Discover how to use the File Manager app.

STEP 1

Raspbian includes a great file management program called "File Manager PCManFM" or just "File Manager" for short. Click on the FileManager PCManFM icon in the Launch Bar or choose Menu > Accessories > File Manager.

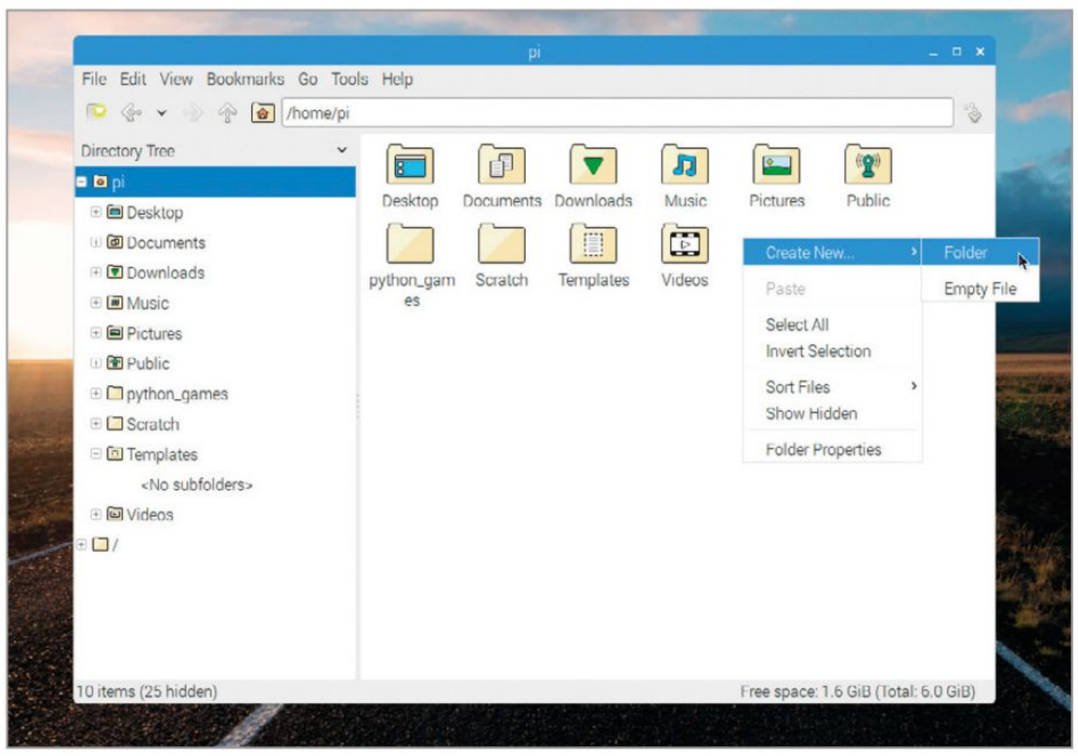


STEP 2

File Manager displays the folders in your Home folder (this should match your user name, 'Pi' by default). By default you should have Desktop, Documents, Downloads, Music, Pictures, Public, python_games, Scratch and folders. Double-click any folder to open it and view its contents. Click the Parent Folder icon or press Alt-Up Arrow to head back up.

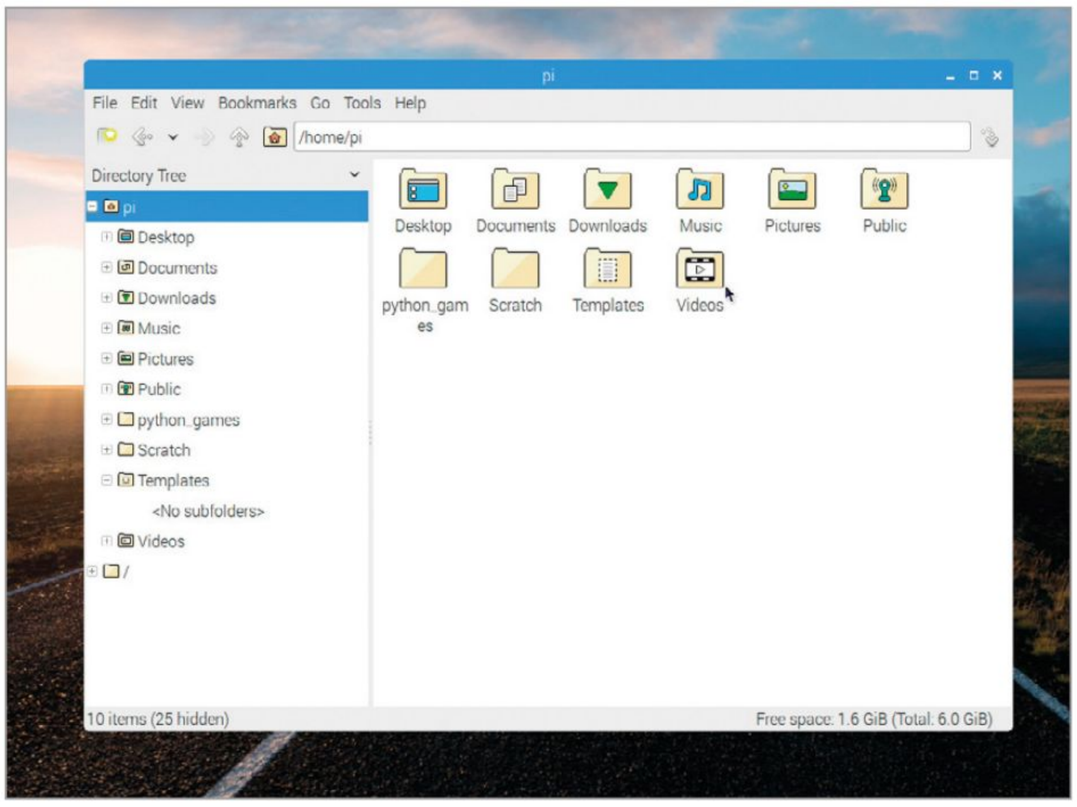
STEP 3

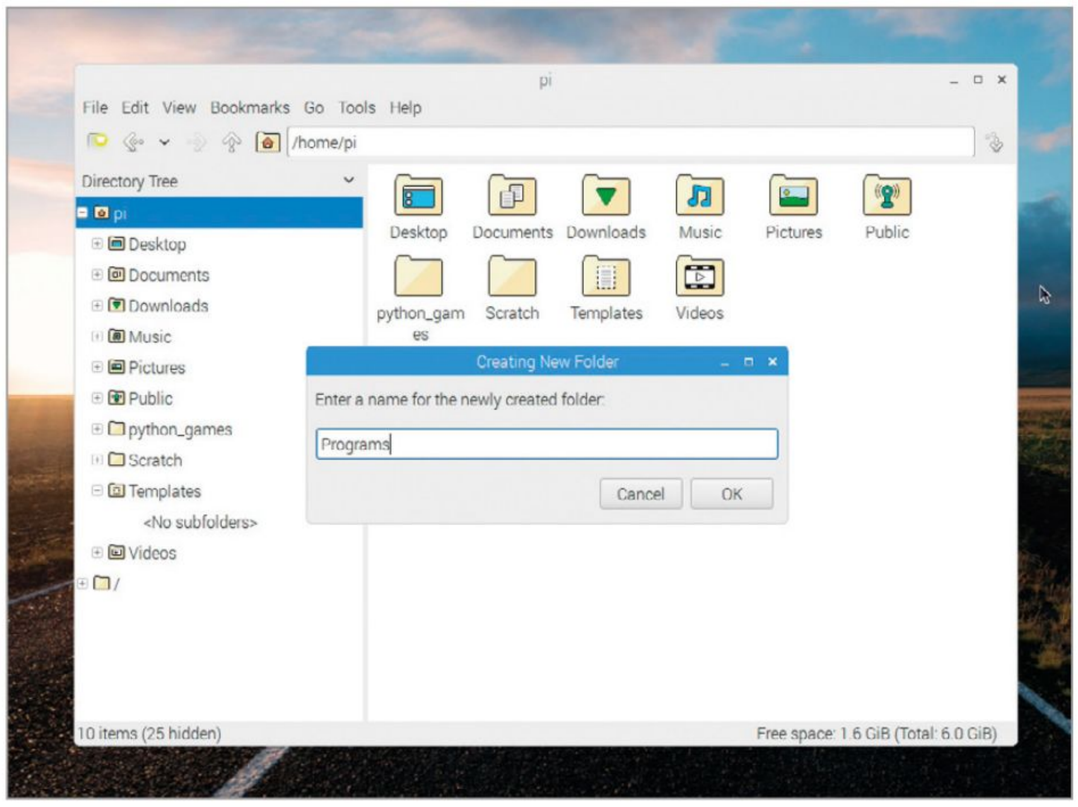
In the left-hand part of the File Manager sits the Side Pane. By default this displays the Directory Tree, which is another way of navigating the folders on your hard drive. Alternatively click the Side Pane menu and choose Places. Now the Side Pane displays common locations such as Home Folder, Desktop, Rubbish Bin (also known as the Wastebasket), Applications and your SD Card.



STEP 4

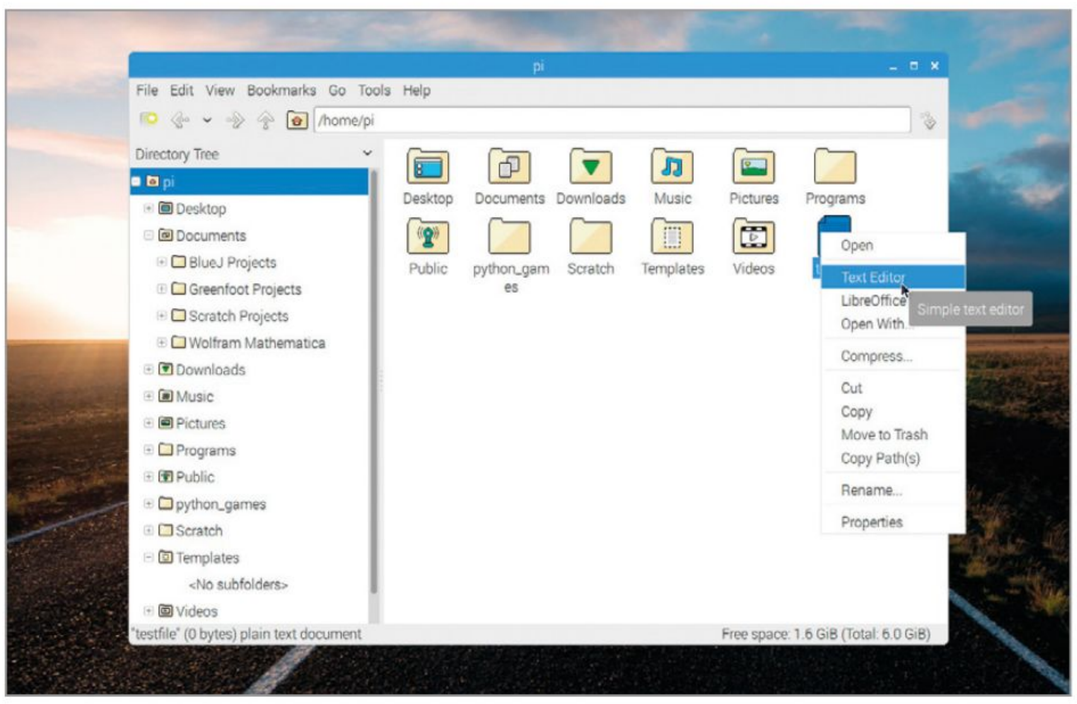
You can create a new folder in the current location by choosing File > Create New > Folder or press Shift-Control-N. Enter a name for the folder and click OK. Files and folders can be dragged on top of one another to move them around. You can also drag files to the folders in the Side Pane, which is an easy way to move them back up the folder tree.





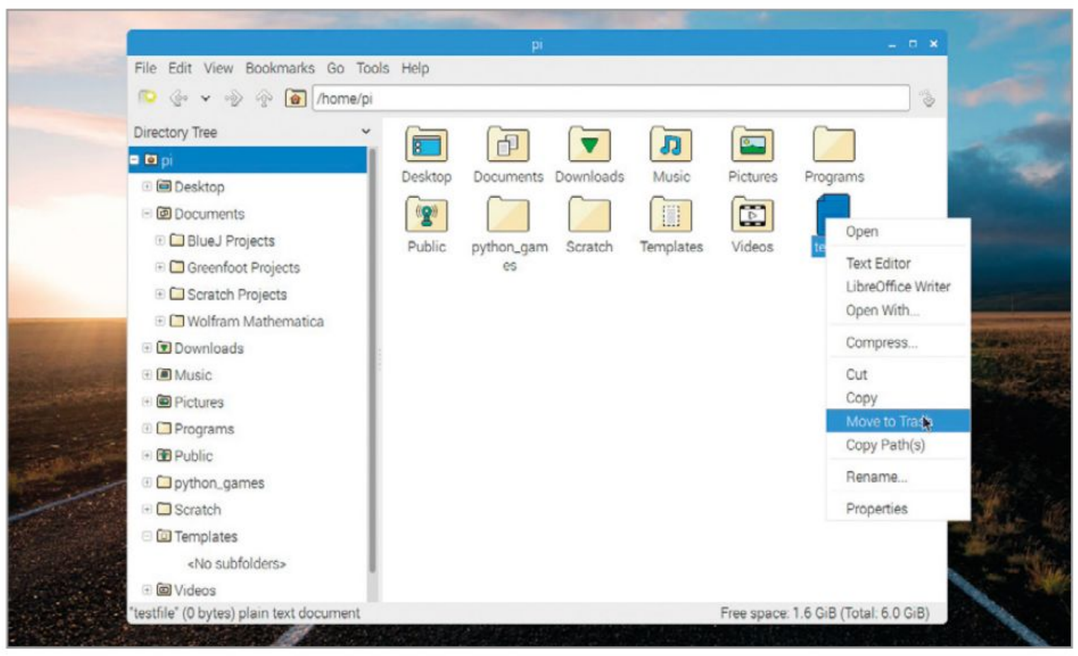
STEP 5

Double-click a file in File Manager to open it with the default app. You can also right-click a file to view compatible applications in Raspbian. Pick an application from the list provided to open it. Click Properties and use the Open with menu to automatically open that type of file with that app in future.



STEP 6

Drag items you no longer want to the Wastebasket to delete them or right-click and choose Move to Trash. To empty the wastebasket and permanently delete the unwanted files double click Wastebasket to open it. Now right-click the black space in File Manager and choose Empty Rubbish Bin. Click Yes in the alert window and the files will be removed.

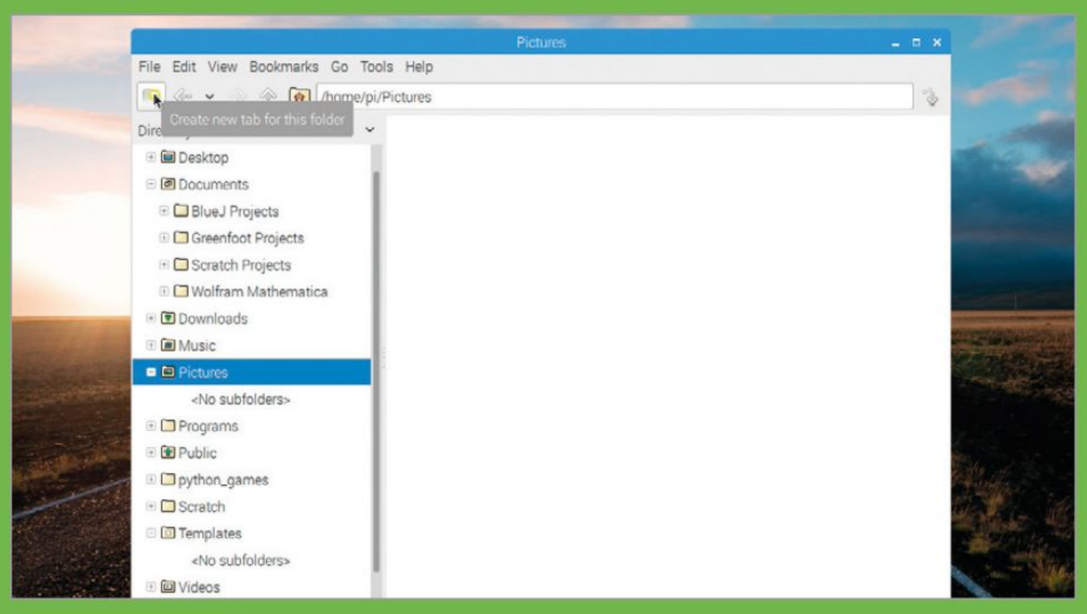


ADVANCED FILE MANAGER TRICKS

These handy tricks and tips make File Manager more powerful.

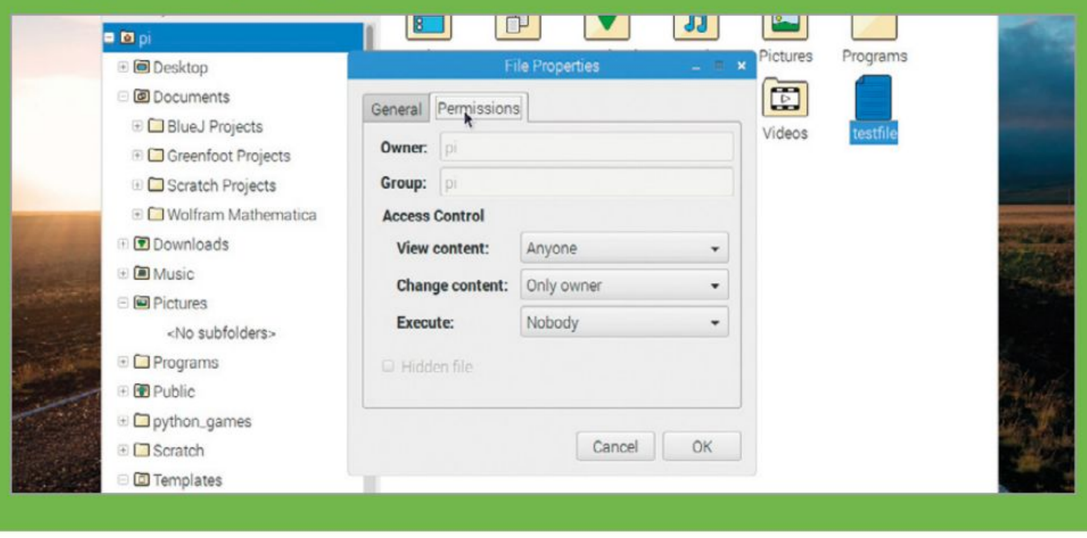
STEP 1

As you start to move files around you will find navigating between the same folders tiresome. It is possible to open locations in File Manager as tabs, so you can quickly jump back and forth. Click the Create new tab for this folder icon, just below the File menu, and the current folder opens as a tab. Click the tabs to jump between locations.



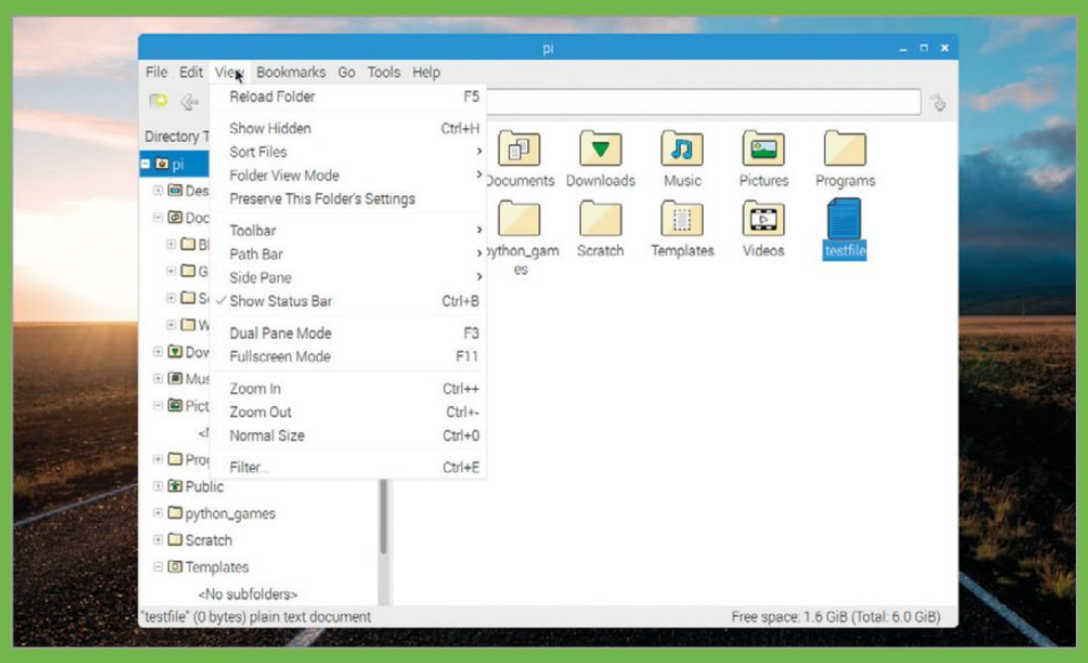
STEP 2

To view the properties of a file right-click and choose Properties. Here you can view information about the File Type and the Open With application. More importantly, if you click Permissions you can view the permissions associated with that file. The concept of file permissions, especially the 'Execute' setting, becomes increasingly important as you become more advanced in using your Raspberry Pi.



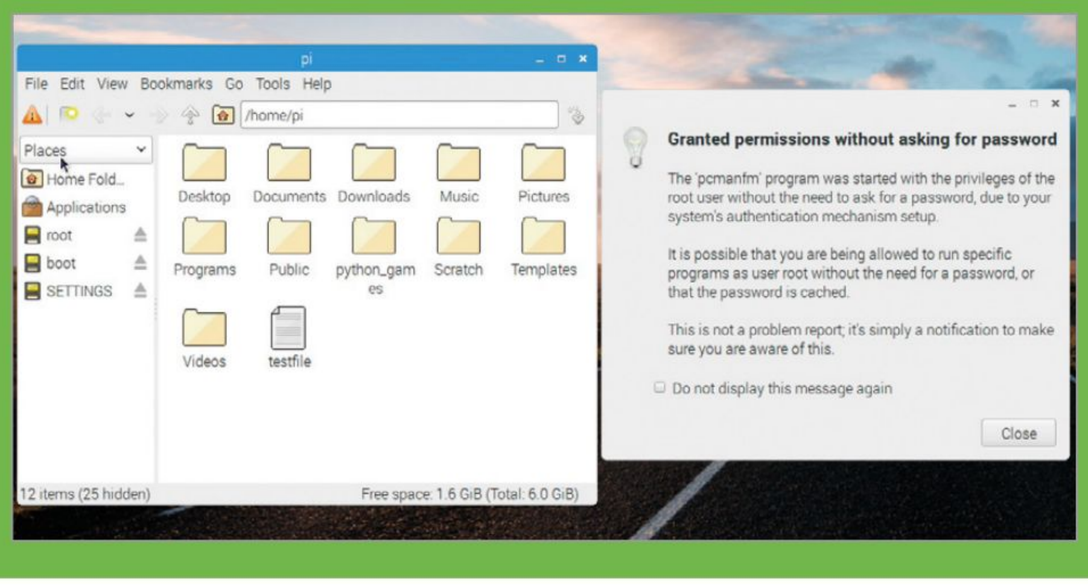
STEP 3

It's worth taking time to explore the View menu in File Manager. Here you'll find a Show Hidden option, which enables you to view hidden files, as well as Sort Files. You can also adjust the Folder View Mode as well as Toolbar, Path Bar, Side Pane and Status Bar.



STEP 4

There are times when you will want to move files but find you can't because you don't have root (sudo) access in File Manager. If you want to open File Manager with root mode choose Menu > Run and enter: `gksu pcmanfm`. File Manager will open and display root and boot() folders. Be careful when running File Manager in root mode, and close it when you are finished.



Edit Images with GIMP

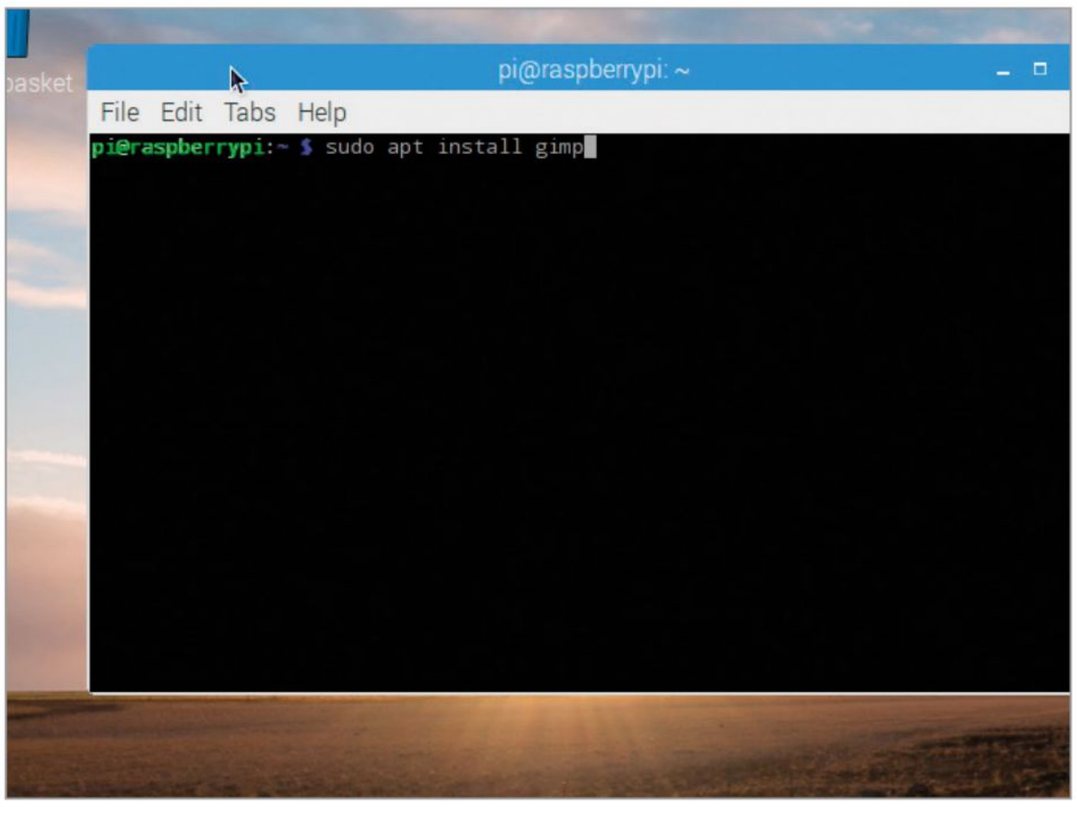
The name may be funny, but this image editing app is incredibly serious. The Raspberry Pi is perfectly adept at photo editing, and you can use GIMP to create icons, images and works of art for your programs.

GET TO KNOW GIMP

Your Raspberry Pi can display images with its built-in Image View app but for any image editing you'll need GIMP (GNU Image Manipulation Program). GIMP is a powerful software package for photo editing and is a great tool to install on your Raspberry Pi.

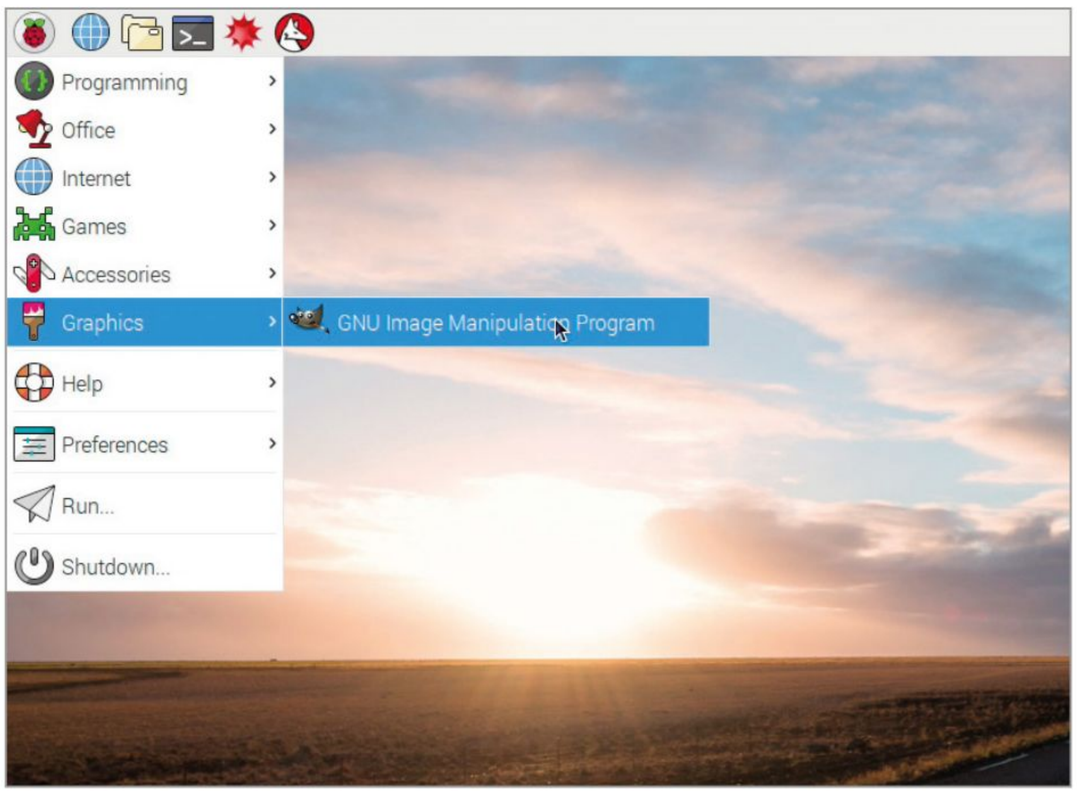
STEP 1

Open terminal and enter: `sudo apt install gimp` to download and install the GIMP package. An alert will appear saying: "after this operation, 117 MB of additional disk space will be used. Do you want to continue [Y/n]?" Enter: `y` and press `Return`. GIMP will now be installed in Raspbian. GIMP is a lot larger than most programs you'll install, so the installation takes longer.



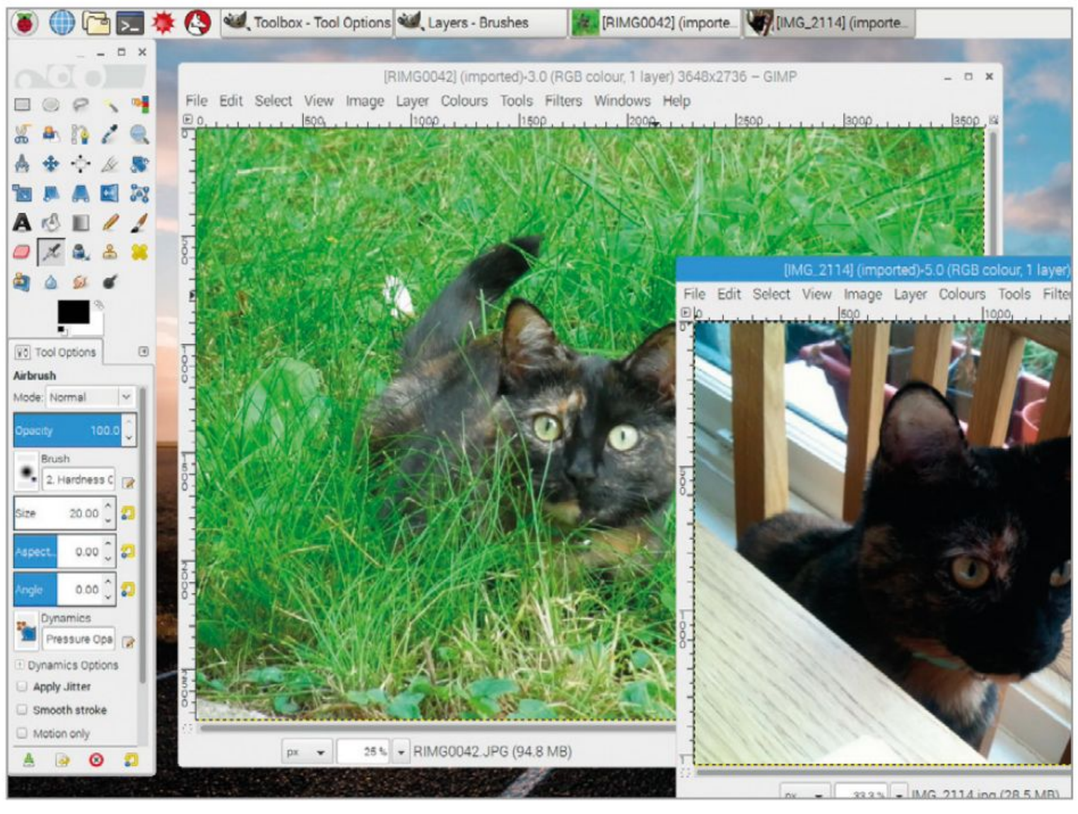
STEP 2

When the installation has completed you will find GIMP under Menu > Graphics > GNU Image Manipulation Program. It has a more detailed interface than many Linux programs with two boxes, Tool Options and Brushes offering a range of image editing options. Users of software like Adobe Photoshop will feel right at home and it's ideal for image editing.



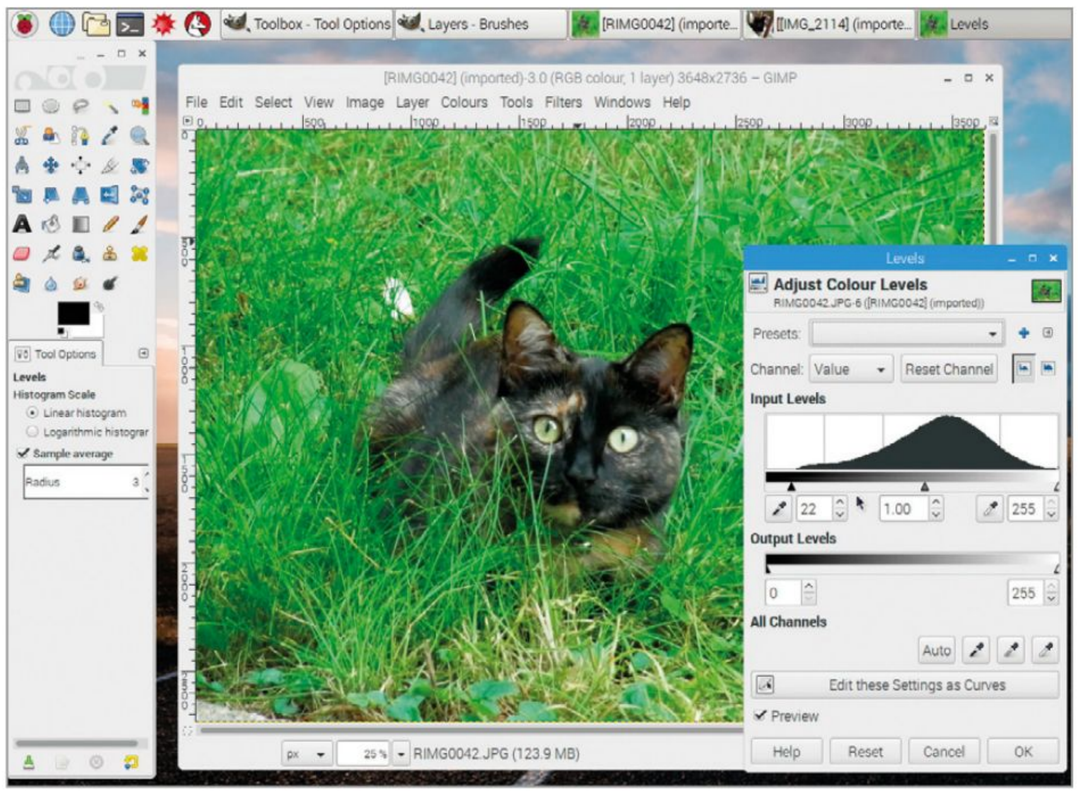
STEP 3

Images can be opened in GIMP using File > Open or by right-clicking on files in File Manager and choosing GNU Image Manipulation Program. You can open multiple images at once in GIMP but each opens inside a different window. Choose Window > Single Window Mode to gather them together. Click on the tabs at the top of the screen to switch from one image to another.

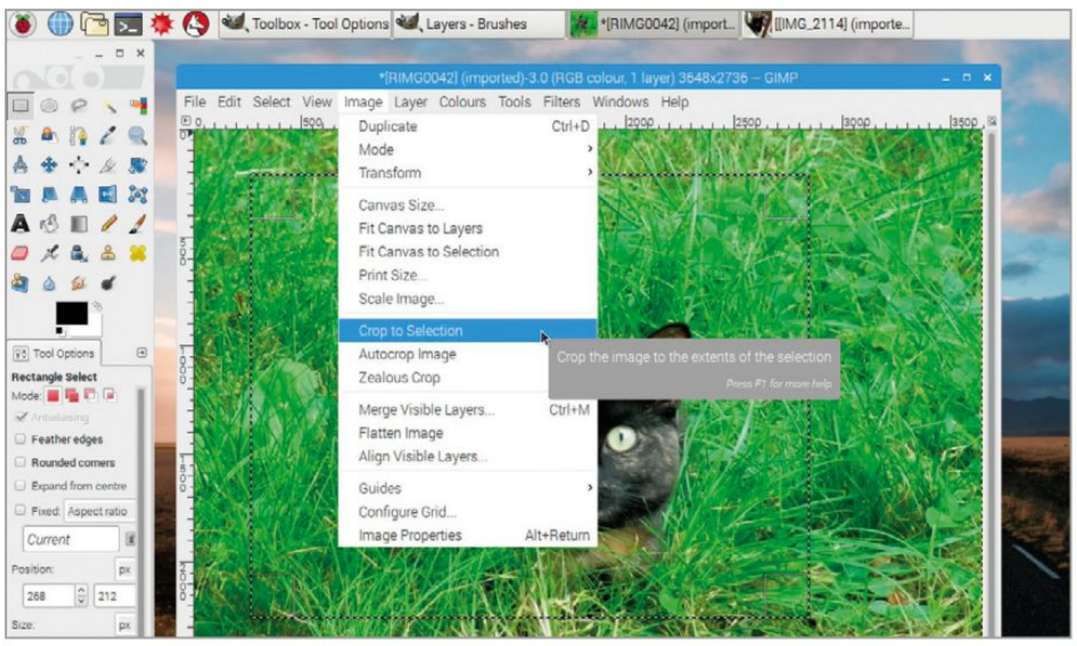


STEP 4

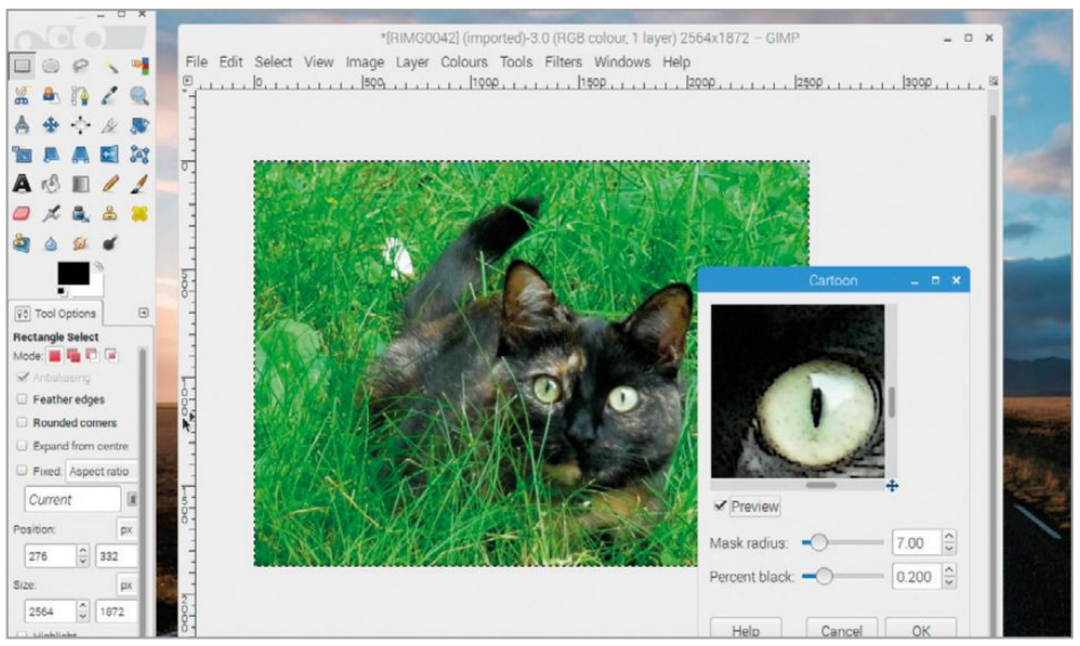
You can perform powerful edits in GIMP but we don't have space to go over them all. Probably the most useful you'll find is Colours > Levels. This window enables you to adjust the tonal range and colour balance of an image. Drag the left and right handles in slightly and slide the centre handle to the right to improve the contrast of an image.



STEP 5 To crop an image click the Rectangle Select Tool and drag a square on the image. Choose Image > Crop To Selection to remove the unwanted parts of the image. Use the Image > Scale Image and Image > Canvas Size options to adjust the image to specific sizes. Cropping and resizing images is a vital technique to know when working on websites.



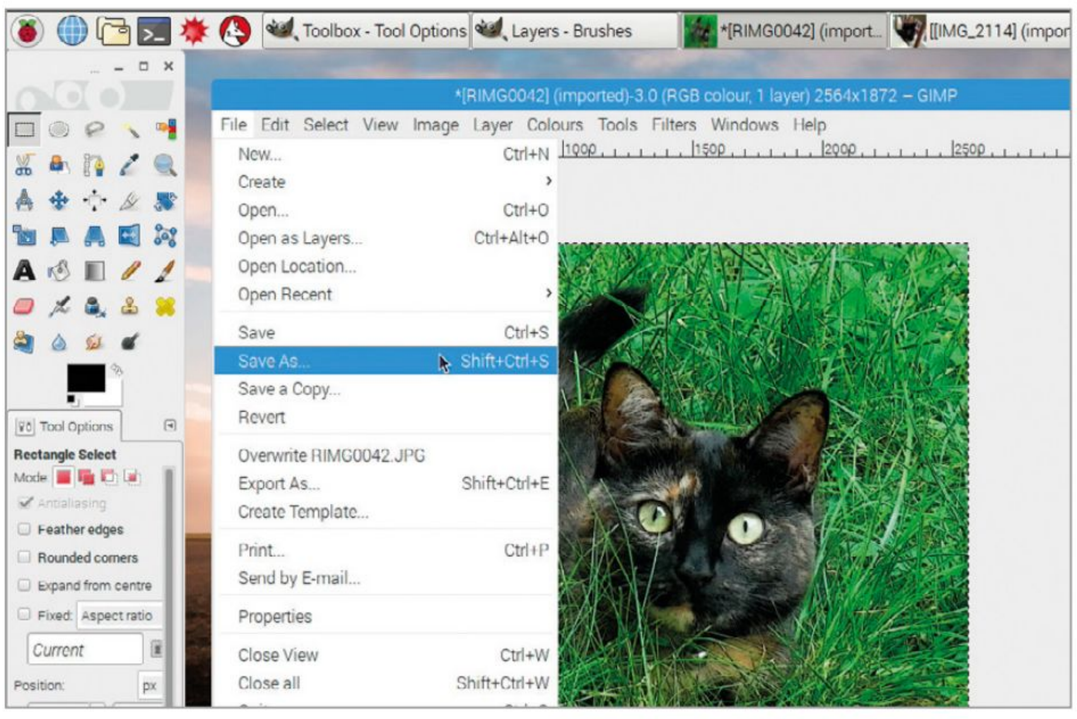
STEP 6 There are a huge range of image effects and filters you can add to images using GIMP. Far more than we have room to cover here. Try Filters > Artistic > Cartoon to give your image a black outline or Filters > Artistic > Oilify to simulate a painted artwork. Take a look at `www.gimp.org/tutorials` for creative inspiration.



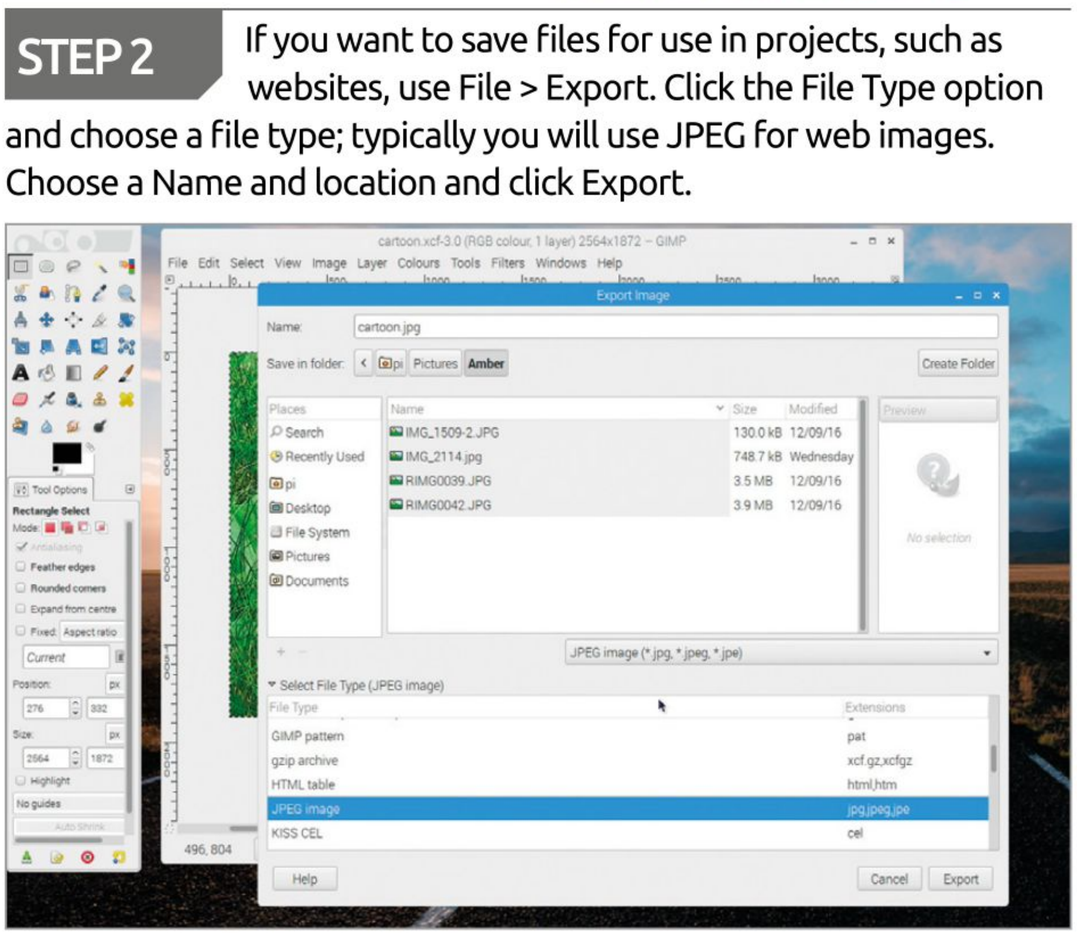
EXPORTING IMAGES

Save your images for use in a website.

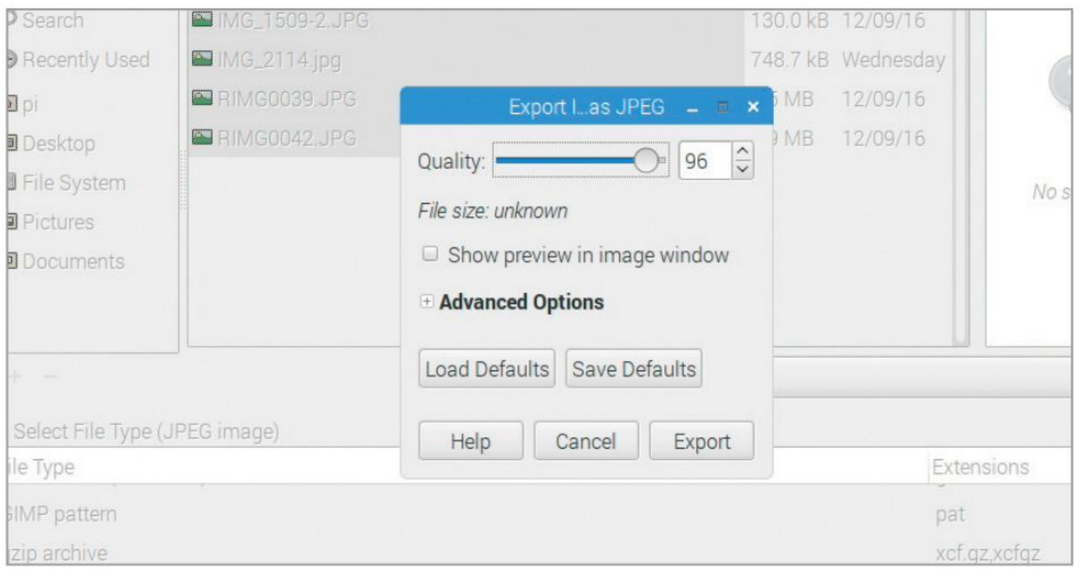
STEP 1 Images are saved using File > Save As in the xcf format (Experimental Computing Facility). This is GIMP's native format so should only be used to save files you want to view and work on in GIMP.



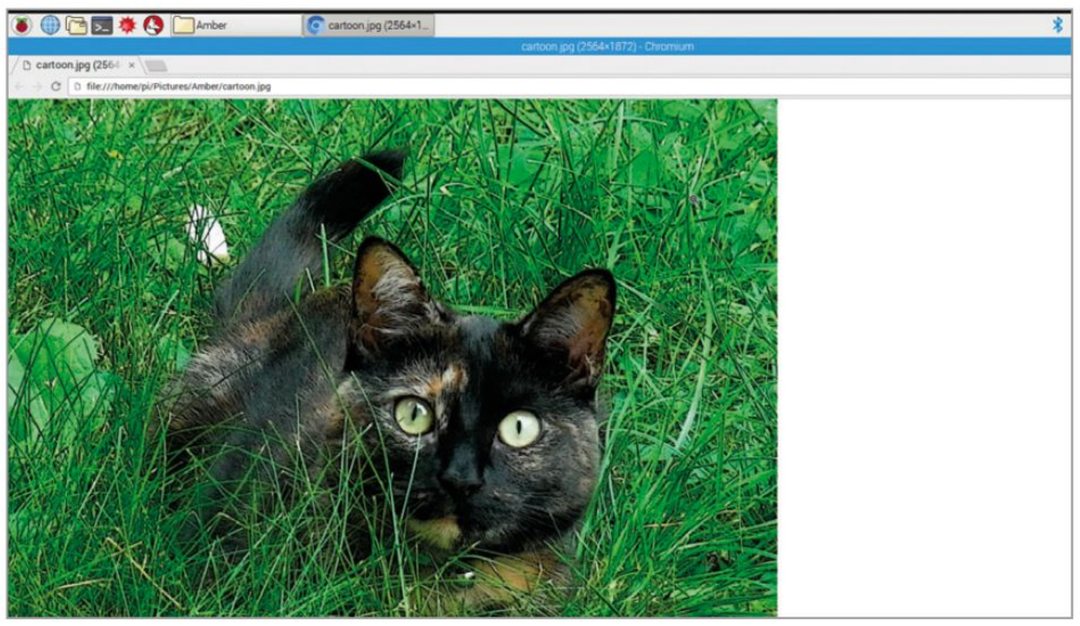
STEP 2 If you want to save files for use in projects, such as websites, use File > Export. Click the File Type option and choose a file type; typically you will use JPEG for web images. Choose a Name and location and click Export.



STEP 3 The Export Image as JPEG window appears, displaying a Quality slider. The Quality range is from 0 (very poor) to 100 (perfect). While you might be tempted to set the quality at 100, reducing it slightly will create much smaller files. Smaller files ensure that your web page loads much more quickly. You can typically reduce the Quality to 80 and get a much smaller file with little discernible difference.



STEP 4 Right-click a JPEG image in the File Manager and choose Open With. Expand Internet and choose Chromium Web Browser and OK. This enables you to see how it will appear when you add it to your website. You can also view images more quickly by right clicking and choosing Image Viewer. You can use Save File As in Image Viewer to quickly change the image format.



More Apps

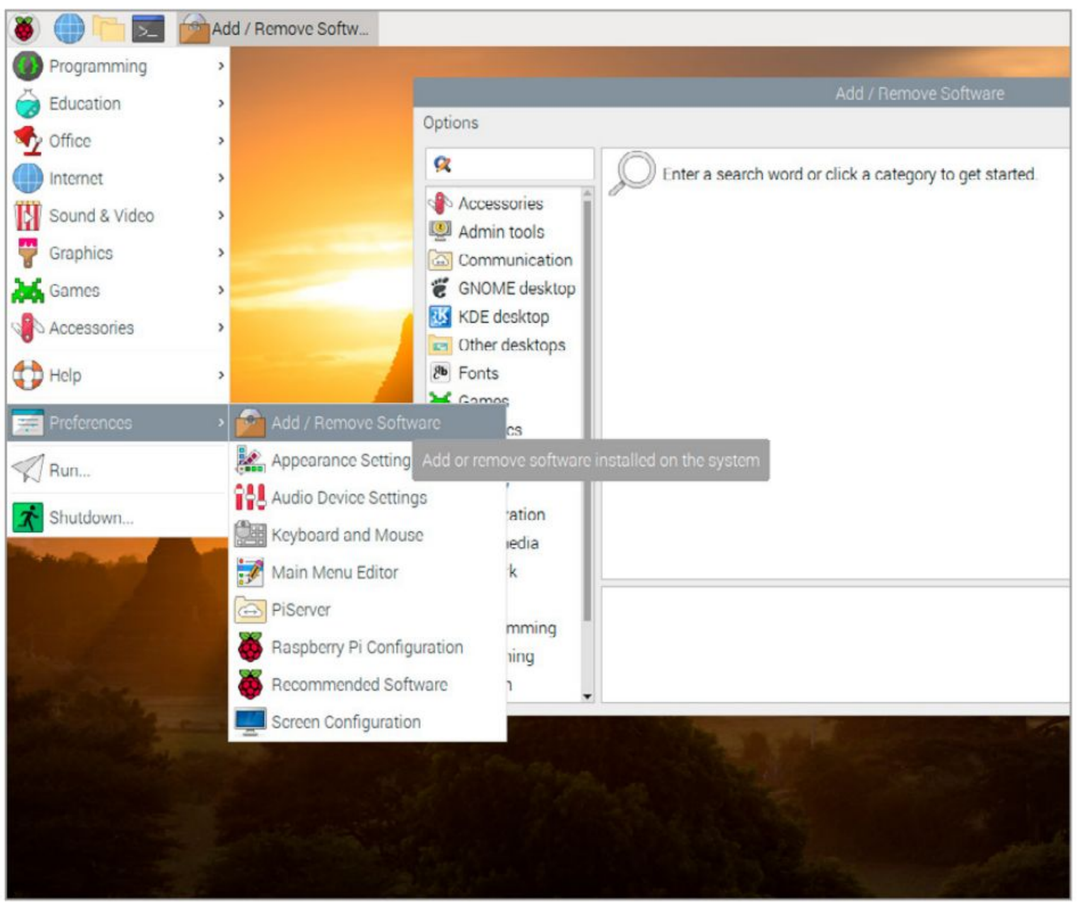
While installing apps, or rather packages, is perfectly fine through the Terminal and the Apt command, there is an easier, graphical way. For most users, the Add/Remove software feature is better than the command line and here's how it works.

ADD/REMOVE

You can use the Add/Remove Software feature both on the Pi version of Raspbian as well as the Raspberry Pi Desktop version – whether it's installed on a PC or in a virtual machine.

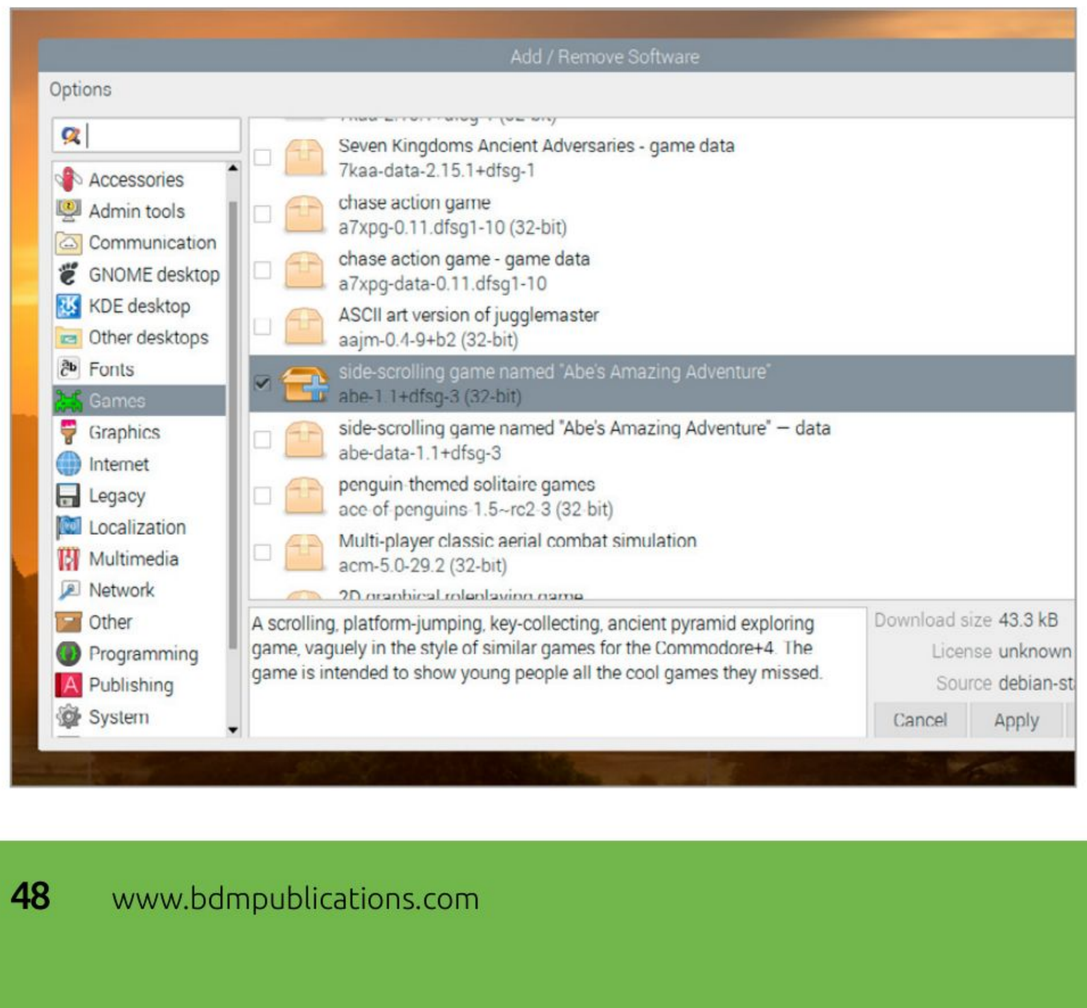
STEP 1

Begin by powering up Raspbian and getting to the desktop. From the menu, select Preferences > Add/Remove Software. This will open a new window with various categories down the side of the screen.



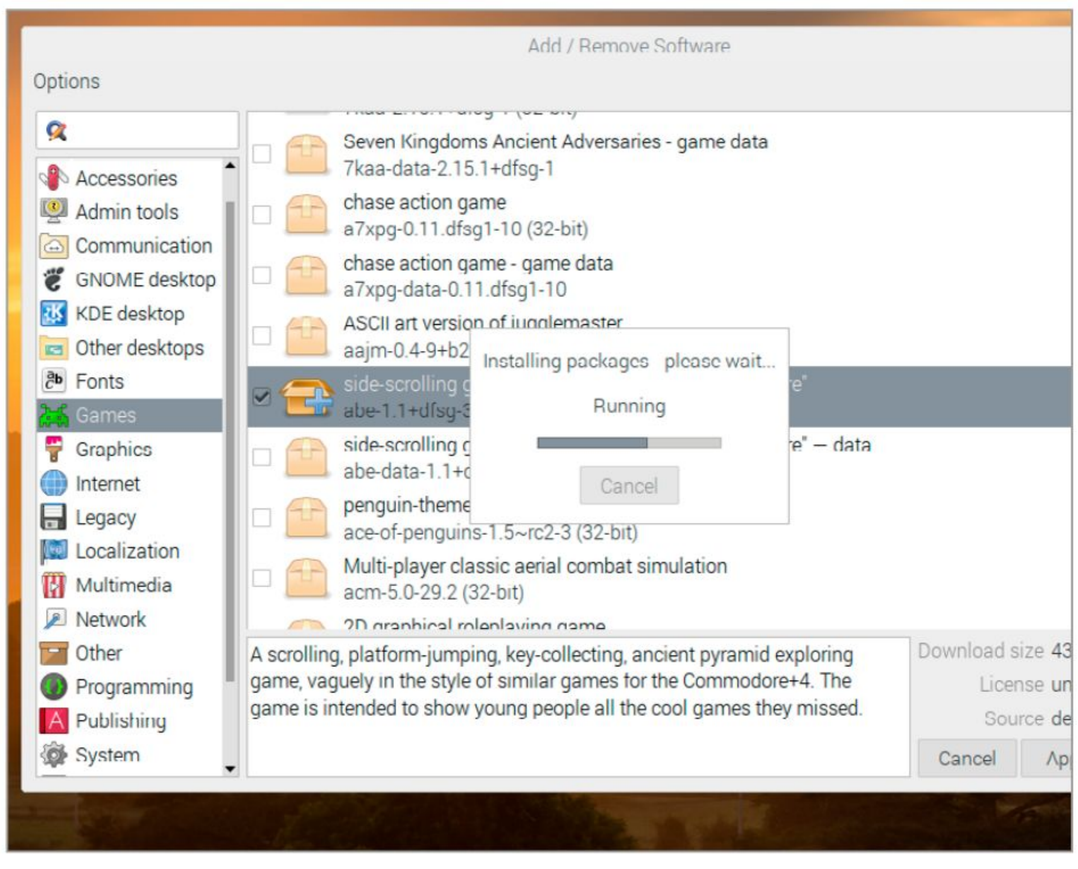
STEP 2

These categories down the side of the Add/Remove Software window denote the type of software package through which you want to search. For example, click on the Games category, then scroll down the list of available packages and tick the box next to 'Side-scrolling game named **Abe's Amazing Adventure**'.



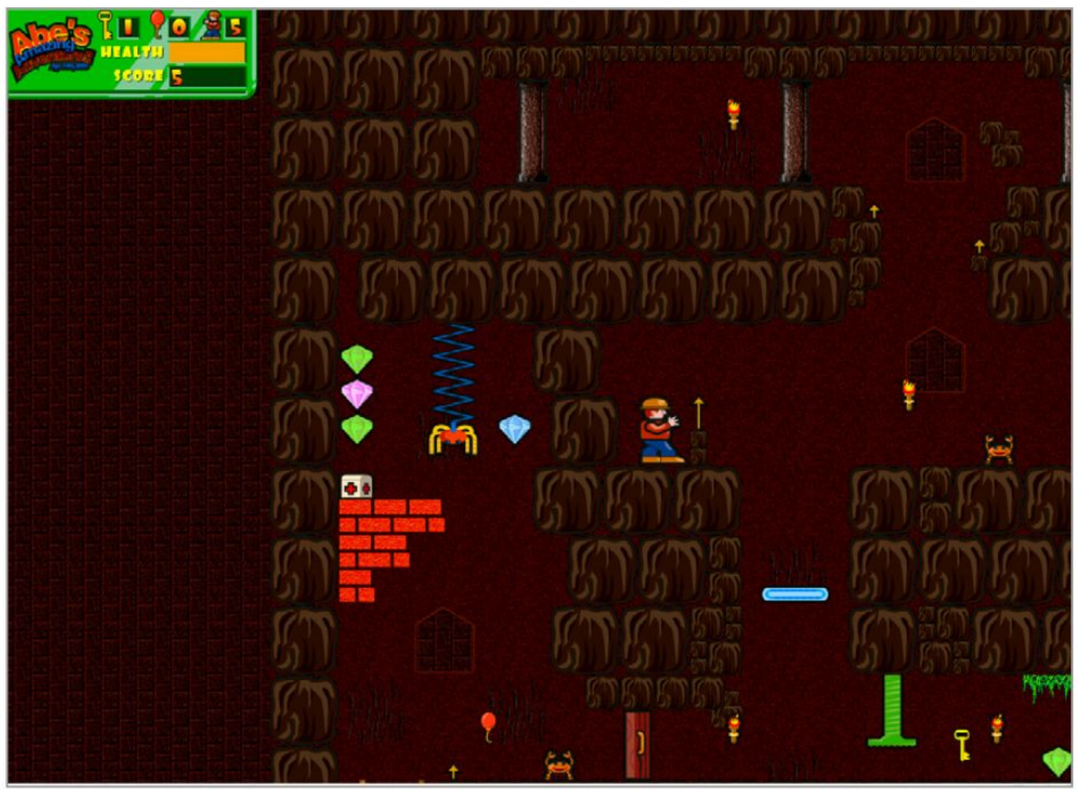
STEP 3

Now click on the OK button, located to the bottom right of the Add/Remove Software window. This will begin the installation of the selected package – note, it'll also auto-select any other required packages in order to make the software work.



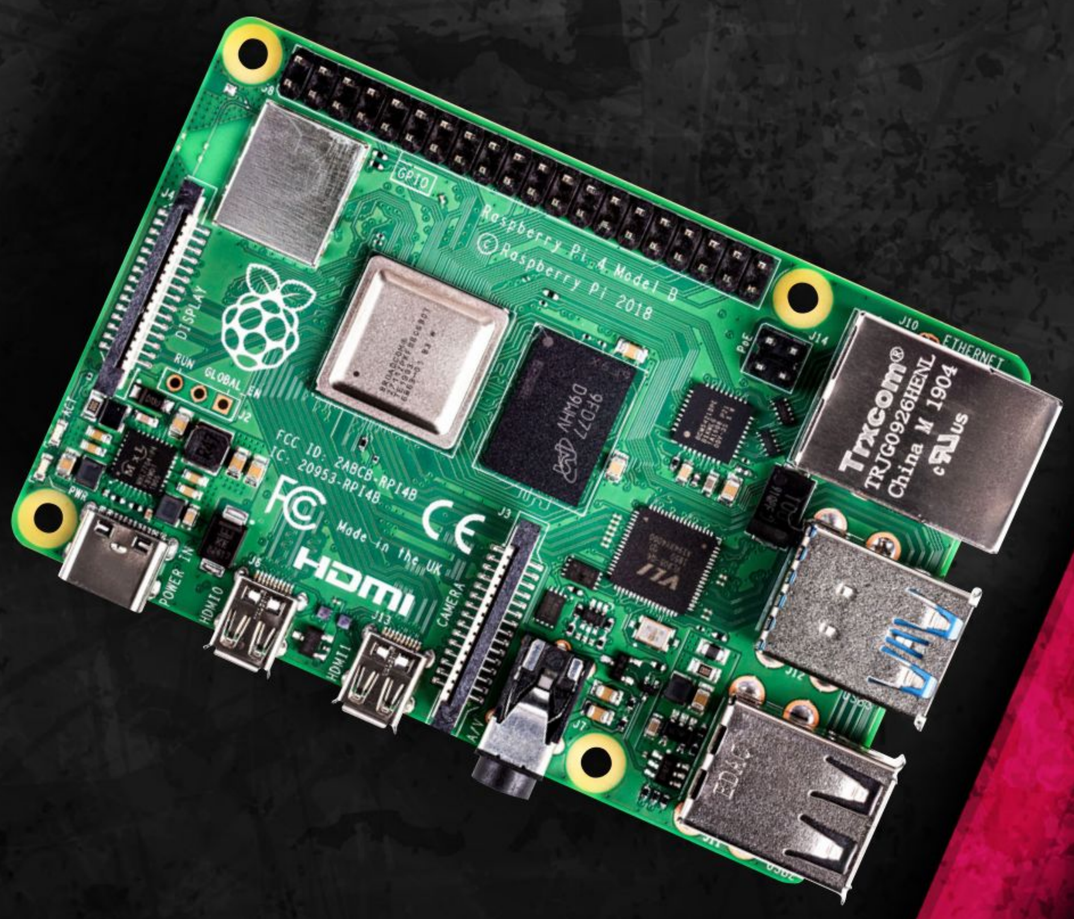
STEP 4

Once the packages have been installed, you can find the software within the Pi Menu items. In this case, Abe's Amazing Adventure is in the Games section of the menu. Click on the icon for the game, and it'll load. To remove the newly installed software, simply re-open Add/Remove Software, un-tick the box and click OK again.



Learning Linux

In this section you will discover how the OS works, how the filesystem is built, and how you can list, move, create and delete files and folders. To truly be able to hack and master the Raspberry Pi, you will need to be familiar with the OS and its inner workings.



What is Linux?

The Raspberry Pi operating system is Raspbian, which is a Linux operating system; but what exactly is Linux? Where did it come from and what does it do? In a world where Windows and macOS have supremacy of the desktop, it's easy to overlook it, but there's more to Linux than you might imagine.

Linux is a surprisingly powerful, fast, secure and capable operating system. It's used as the OS of choice for the Raspberry Pi, in the form of Raspbian OS, as well as in some of the most unlikely places.

Despite only enjoying a 1.96% share (according to netmarketshare.com) of the total desktop operating system market, Linux has a dedicated following of enthusiasts, users and contributors. It was created in 1991 by University of Helsinki student, Linus Torvalds, who had become frustrated with the limitations and licensing of the popular educational system Minix, a miniature version of the Unix operating system, in use at the time.

Unix itself was released in the early '70s, as a multi-tasking, modular-designed operating system originally developed for programmers who needed a stable platform to code on. However, its performance, power and portability meant that it soon became the system of choice for companies and universities where high-end computing tasks were needed.

Torvalds needed a system that could mirror Unix's performance and features, without the licensing cost. Thus was born Linux, the Unix-like operating system which used freely available code from the GNU project. This enabled users around the world to utilise the power of the Unix-like system, completely free of charge, an ethos that still holds today: Linux is free to download, install and use.

Linux is much like any other operating system, such as Windows or macOS in that it manages the computer hardware, provides an interface for the user to access that hardware and comes with programs for productivity, communications, gaming, science, education and more. Linux can be broken up into a number of significant elements:

BOOTLOADER

The bootloader is the software that initialises and boots up your computer. It loads up the various modules the OS uses to begin to access the hardware in the system.

KERNEL

The kernel is the core of the system and the single element that is actually called Linux. The Linux kernel manages the computer processor, memory, storage and any peripherals you have attached to your computer.

DAEMONS

Daemons are background services that start as the operating system is booting. These can enable printing, sound, networking and so on.

GRAPHICAL SERVER

This is a module within Linux that provides a graphical output to your monitor. It's referred to as the X server or simply X.

SHELL

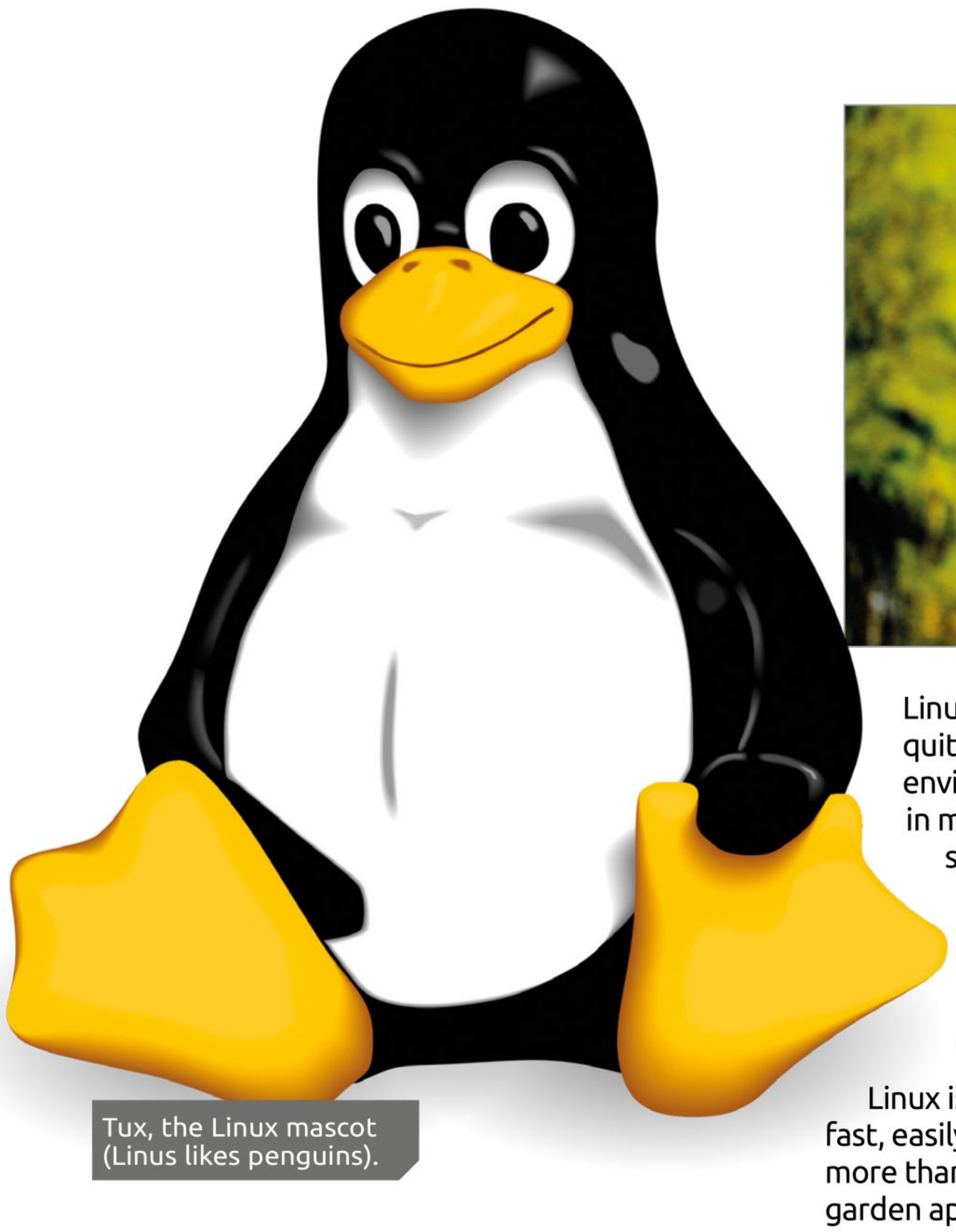
The Linux shell is a command line interface environment, which a Linux user can use to enter commands to the OS that directly affect it. Within the shell you can add new users, reboot the system, create and delete files and folders plus much more. BASH (Bourne-Again Shell) is the most popular shell used in Linux, although others are available. The shell is also known as the Terminal and it's where we're going to work from through this section of the book.

DESKTOP ENVIRONMENT

The Desktop Environment, or DE, is the main Graphical User Interface (GUI) that users interact with. It's the desktop and includes Internet browsers, productivity, games and whatever program or app you're using. There are countless DEs available, however Raspbian uses PIXEL.

PROGRAMS/APPLICATIONS

As Linux is a free, open source operating system, it also makes good use of the tens of thousands of freely available applications. The likes of LibreOffice, GIMP and Python are just the tip of the iceberg.



Tux, the Linux mascot (Linus likes penguins).

Linus Torvalds, the creator of the Linux kernel.

Linux is used throughout the world, in a number of basic and quite unique uses. While it may look radically different from one environment to the next, the actual Linux kernel, can be found in modern smart TVs, in-car entertainment systems and GPS, supercomputers, IoT devices and the Raspberry Pi. It's used by NASA, both in the command centre and on-board the ISS. Linux servers power the backbone of the Internet, along with most of the websites you visit daily. Android utilises components of the Linux kernel, as do set top boxes, games consoles and even your fridge, freezer, oven and washing machine.

Linux isn't just a free to use operating system. It's stable, powerful and fast, easily customised and requires very little maintenance. However, it's more than just performance stats; Linux means freedom from the walled garden approach of other operating systems. It's a lively community of like-minded individuals who want more from their computers without the shackles of price or conformity. Linux means choice.

Raspbian on the Raspberry Pi, is the Linux distribution of choice.

A Desktop Environment can be as complex or as simple as the user desires.

Using the Filesystem

To master Linux, it's important to understand how the filesystem works. What's more, it's also important to become familiar with the Terminal, or shell. This command line environment may appear daunting at first, but with practise, it soon becomes easy to use.

GETTING AROUND

To drop into the Terminal, click on the fourth icon from the left along the top of the Raspberry Pi desktop, the one with a right-facing arrow and an underscore. This is the shell, or Terminal.

STEP 1

First, you're going to look at directories and the directory path. A directory is the same thing as a folder, however in Linux it's always called a directory. These are placed inside each other using a "/" character. So when you see /home/pi it means the pi directory is inside the home directory. Enter: `clear` and press return to clean the screen. Now enter: `pwd`. This stands for Print Working Directory and displays /home/pi.

```
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $
```

STEP 2

When you log in to your Raspberry Pi, you don't start at the base of the hard drive, known as the 'root' (also known as the topmost directory). Instead you begin inside your user directory, which is named 'pi' by default and is itself in a directory called 'home'. Directories are indicated by the '/' symbol. So, "'/home/pi'" tells you that in the root is a directory called home, and the next "'/'" says that inside "home" is a directory called "pi". That's where you start.

```
pi@raspberrypi ~ $ pwd
/home/pi
pi@raspberrypi ~ $
```

STEP 3

Enter: `ls` to view the contents of the current directory. You should see Desktop, Documents, and Downloads and Scratch in Blue. You may also see other items depending on how much you have used your Raspberry Pi. The colour code is worth knowing: directories are blue while most files are white. As you go on you'll see other colours: executable files (programs) are bright green, archived files are red and so on. Blue and white are the two you need to know to get started.

```
pi@raspberrypi ~ $ pwd
/home/pi
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  exit  indiecity  python_games  Scra
pi@raspberrypi ~ $
```

STEP 4

Now you're going to move from the pi directory into the Documents directory. Enter: `cd Documents`. Note the capital "D". Linux is case sensitive, which means you have to enter the exact name including correct capitalisation. The cd command stands for change directory. Now enter: `pwd` again to view the directory path. It will display /home/pi/ Documents. Enter: `ls` to view the files inside the Documents directory.

```
pi@raspberrypi ~ $ pwd
/home/pi
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  exit  indiecity  python_games  Scratch
pi@raspberrypi ~ $ cd Documents
pi@raspberrypi ~/Documents $ pwd
/home/pi/Documents
pi@raspberrypi ~/Documents $ ls
Amber  archive.tar  Crypto101.pdf  dog_jump  euro  fizzbang_backup.py  fizzbang.py
pi@raspberrypi ~/Documents $ _
```

STEP 5 How do you get back up to the pi directory? By using a command "cd ..". In Linux two dots means the directory above, also known as the parent directory. Incidentally, a single dot "." is used for the same directory. You never use "cd ." to switch to the same directory but it's worth knowing because some commands need you to specify the current directory.

```
pi@raspberrypi ~/Documents $ pwd
/home/pi/Documents
pi@raspberrypi ~/Documents $ cd ..
pi@raspberrypi ~ $ pwd
/home/pi
pi@raspberrypi ~ $
```

STEP 6 The "ls" and "cd" commands can also be used with more complex paths. Enter: `ls Documents/Pictures` to view the contents of a Pictures directory inside your Documents directory. You can switch to this directory using `cd Documents/Pictures`; use `cd ../..` to move back up two parent directories.

```
pi@raspberrypi ~ $ ls Documents/Pictures
LEGO  LucyHattersley.jpg  raspberry_pi_2_photographs
pi@raspberrypi ~ $ cd Documents/Pictures
pi@raspberrypi ~/Documents/Pictures $ pwd
/home/pi/Documents/Pictures
pi@raspberrypi ~/Documents/Pictures $ cd ../..
pi@raspberrypi ~ $ pwd
/home/pi
pi@raspberrypi ~ $
```

ABSOLUTE VS RELATIVE PATHS

It is important to know the difference between the working directory, root directory and home. There are also two types of path: Absolute and Relative. These are easier to understand than they sound. Let's take a look…

STEP 1 By default, commands like "ls" use the working directory. This is the current directory that you're looking at and is set to your home directory by default (/users/pi). Using "pwd" (Print Working Directory) lets you know what the working directory is, and using "cd" changes the working directory.

```
pi@raspberrypi ~ $ pwd
/home/pi
pi@raspberrypi ~ $
```

STEP 2 The root directory is always '/'. Entering: `ls /` lists the contents of root, and entering: `cd /` switches to the root directory. This is important because there is a difference between "ls Documents/Pictures" and "ls /Documents/Pictures". The first command lists the contents of the Pictures directory in Documents inside the working directory (which, if you are in the home directory, will work).

```
pi@raspberrypi ~ $ ls /
bin  boot  dev  etc  home  lib  lost+found  media  mnt  opt  pro
pi@raspberrypi ~ $ ls /Documents/Pictures
ls: cannot access /Documents/Pictures: No such file or directory
pi@raspberrypi ~ $ _
```

STEP 3 The second command ("ls /Documents/Pictures") attempts to list the content of Pictures in a directory called Documents inside the root directory (because the path started with '/', which is root). There is typically no Documents directory in root, so you will get a "No such file or directory" error. Starting a path with '/' is known as an "absolute path", while starting without the '/' is known as a "relative path" because it is relative to your working directory.

STEP 4 There is also an absolute path shortcut to your user directory, and that is the tilde "~" character. Entering: `ls ~` always lists the contents of your home directory, while "cd ~" moves straight to your home directory, no matter what your working directory is. You can also use this shortcut wherever you are: enter: `ls ~/Documents/Pictures` to display the contents of the Pictures.

```
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ ls Documents/Pictures
BDM-Web-logo-dark1.jpg  David Hayward.jpg  RPi.png
pi@raspberrypi:~ $
```

```
pi@raspberrypi:~ $ cd ~
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ ls ~/Documents/Pictures
BDM-Web-logo-dark1.jpg  David Hayward.jpg  RPi.png
pi@raspberrypi:~ $
```

Listing and Moving Files

Admittedly, using the desktop GUI to list and move files is much easier than using the Terminal and keyboard. However, it's an important skill that you will appreciate as you advance with the Raspberry Pi and Linux.

LOOKING AT FILES

Operating systems are built on files and folders, or directories if you prefer. While you're used to viewing your own files, most operating systems keep other files out of sight. In Raspbian, you have access to every file in the system.

STEP 1

We've already looked at "ls", which lists the files in the working directory, but you are more likely to use a command like "ls –l". The bit after the command (the '-lah') is known as the argument. This is an option that modifies the behaviour of the command.

```
pi@raspberrypi ~ $ ls -l_
```

STEP 2

The "-l" argument lists files and directories in long format. Each file and directory is now on a single line, and before each file is a lot of text. First you'll see lots of letters and dashes, like 'drwxr-xr-x'. Don't worry about these for now; they are known as 'permissions' and we'll come to those later.

```
pi@raspberrypi ~ $ ls -l
total 24
-rw-r--r-- 1 pi pi    0 May 11 20:56 articles.txt
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 Apr 21 14:50 Documents
drwx------ 2 pi pi 4096 Apr 21 15:23 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
-rw-r--r-- 1 pi pi    0 May 11 20:56 names.txt
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 Apr 17 12:53 Scratch
pi@raspberrypi ~ $
```

STEP 3

After the permission letters come a single number. This is the number of files in the item. If it's a file then it'll be 1, but if it's a directory it'll be at least 2. This is because each directory contains two hidden files; one with a single dot (.) and one with two dots (..). Directories containing files or other directories will have a higher number.

```
pi@raspberrypi ~ $ ls -l
total 24
-rw-r--r-- 1 pi pi    0 May 11 20:56 articles.txt
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 Apr 21 14:50 Documents
drwx------ 2 pi pi 4096 Apr 21 15:23 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
-rw-r--r-- 1 pi pi    0 May 11 20:56 names.txt
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 Apr 17 12:53 Scratch
pi@raspberrypi ~ $
```

STEP 4

Next you'll see the word "pi" listed twice on each line. This refers to the user rather than the name of your computer (your default username is "pi"). The first is the owner of the file, and the second is the group. Typically these will both be the same and you'll see either 'pi' or 'root'. You can enter: `ls -l /` to view the files and directories in the root directory that belong to the root account.

```
pi@raspberrypi ~ $ ls -l
total 28
-rw-r--r-- 1 pi pi    0 May 11 20:56 articles.txt
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 Apr 21 14:50 Documents
drwx------ 2 pi pi 4096 Apr 21 15:23 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
-rw-r--r-- 1 pi pi    0 May 11 20:56 names.txt
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 Apr 17 12:53 Scratch
drwxr-xr-x 3 pi pi 4096 May 11 21:15 test
pi@raspberrypi ~ $ ls -l /
total 74
drwxr-xr-x   2 root root  4096 Jan  1  1970 bin
drwxr-xr-x   3 root root  2048 Jan  1  1970 boot
drwxr-xr-x  12 root root  3280 May 11 09:03 dev
drwxr-xr-x 109 root root  4096 May 11 09:03 etc
drwxr-xr-x   3 root root  4096 Jan  1  1970 home
drwxr-xr-x  12 root root  4096 Jan  1  1970 lib
drwx------   2 root root 16384 Feb 15 11:21 lost+found
drwxr-xr-x   3 root root  4096 May 11 07:42 media
drwxr-xr-x   2 root root  4096 Jan 11 00:02 mnt
drwxr-xr-x   6 root root  4096 Jan  1  1970 opt
dr-xr-xr-x  86 root root     0 Jan  1  1970 proc
drwx------   9 root root  4096 May 11 07:36 root
drwxr-xr-x  10 root root   460 May 11 09:03 run
drwxr-xr-x   2 root root  4096 Jan  1  1970 sbin
```

STEP 5

The next number relates to the size of the file, in bytes. In Linux each text file is made up of letters and each letter takes up a byte, so our names.txt file has 37 bytes and 37 characters in the document. Files and directories can be extremely large and hard to determine, so use "ls – lh". The "h" argument humanises the number, making it easier to read.

```
pi@raspberrypi ~ $ ls -l
total 32
-rw-r--r-- 1 pi pi    0 May 11 20:56 articles.txt
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 Apr 21 14:50 Documents
drwx------ 2 pi pi 4096 Apr 21 15:23 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
-rw-r--r-- 1 pi pi   37 May 11 21:27 names.txt
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 Apr 17 12:53 Scratch
drwxr-xr-x 3 pi pi 4096 May 11 21:15 test
pi@raspberrypi ~ $ ls -lh
total 32K
-rw-r--r-- 1 pi pi    0 May 11 20:56 articles.txt
drwxr-xr-x 2 pi pi 4.0K Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4.0K Apr 21 14:50 Documents
drwx------ 2 pi pi 4.0K Apr 21 15:23 Downloads
drwxr-xr-x 3 pi pi 4.0K Apr 17 18:48 indiecity
-rw-r--r-- 1 pi pi   37 May 11 21:27 names.txt
```

STEP 6

Finally, you should be aware that there are many hidden files in Linux. These are listed using the "-a" argument. Hidden files and directories begin with a dot (.), so you should never start a file or directory with a dot, unless you want to hide it. Typically, you can combine all three arguments together into the command "'s –lah".

```
pi@raspberrypi ~ $ ls -lah
total 520K
drwxr-xr-x 33 pi   pi        4.0K May 11 21:14 .
drwxr-xr-x  3 root root      4.0K Jan  1  1970 ..
drwx------  2 pi   pi        4.0K Apr 20 14:31 .aptitude
-rw-r--r--  1 pi   pi           0 May 11 20:56 articles.txt
-rw-------  1 pi   pi        8.7K May 11 09:03 .bash_history
-rw-r--r--  1 pi   pi         220 Feb 15 14:05 .bash_logout
-rw-r--r--  1 pi   pi        3.2K Feb 15 14:05 .bashrc
drwxr-xr-x 10 pi   pi        4.0K Apr 21 17:08 .cache
drwxr-xr-x 20 pi   pi        4.0K Apr 21 13:33 .config
drwx------  3 pi   pi        4.0K Feb 16 14:16 .dbus
drwxr-xr-x  2 pi   pi        4.0K Apr 21 17:55 Desktop
-rw-r--r--  1 pi   pi          35 Apr 17 12:17 .dmrc
drwxr-xr-x  5 pi   pi        4.0K Apr 21 14:50 Documents
drwx------  2 pi   pi        4.0K Apr 21 15:23 Downloads
drwxr-xr-x  2 pi   pi        4.0K Apr 20 13:45 .dreamchess
drwxr-xr-x  2 pi   pi        4.0K Apr 21 18:15 .fontconfig
drwxr-xr-x  2 pi   pi        4.0K Apr 17 18:56 .freeciv
```

SOME COMMON DIRECTORIES

Now that you know how to view the contents of your hard drive you'll start to notice a lot of directories with names like bin, sbin, var and dev. These are the files and directories that you are kept away from on a Mac, and won't encounter on a Windows PC.

STEP 1

Enter: `ls -lah /` to view all of the files and directories, including the hidden items, in the root directory of your hard drive. Here you will see all the items that make up your Raspbian OS (which is a version of Linux). It's worth taking the time to know some of them.

```
pi@raspberrypi ~ $ ls -lah /
total 82K
drwxr-xr-x  22 root root 4.0K May 11 21:23 .
drwxr-xr-x  22 root root 4.0K May 11 21:23 ..
drwxr-xr-x   2 root root 4.0K Jan  1  1970 bin
drwxr-xr-x   3 root root 2.0K Jan  1  1970 boot
drwxr-xr-x  12 root root 3.3K May 11 09:03 dev
drwxr-xr-x 109 root root 4.0K May 11 09:03 etc
drwxr-xr-x   3 root root 4.0K Jan  1  1970 home
drwxr-xr-x  12 root root 4.0K Jan  1  1970 lib
drwx------   2 root root  16K Feb 15 11:21 lost+found
drwxr-xr-x   3 root root 4.0K May 11 07:42 media
drwxr-xr-x   2 root root 4.0K Jan 11 00:02 mnt
drwxr-xr-x   6 root root 4.0K Jan  1  1970 opt
dr-xr-xr-x  85 root root    0 Jan  1  1970 proc
drwx------   9 root root 4.0K May 11 07:36 root
drwxr-xr-x  10 root root  460 May 11 09:03 run
drwxr-xr-x   2 root root 4.0K Jan  1  1970 sbin
drwxr-xr-x   2 root root 4.0K Jun 20  2012 selinux
drwxr-xr-x   2 root root 4.0K Feb 15 11:23 srv
dr-xr-xr-x  12 root root    0 May 11 21:00 sys
drwxrwxrwt   4 root root 4.0K May 11 21:39 tmp
```

STEP 2

Bin is a directory that stores binaries. This is the Linux way of saying programs or applications. Sbin is for system binaries, which are the programs that make up your system. Dev contains references to your devices: hard drive, keyboard, mouse and so on. Etc contains your system configuration files.

```
pi@raspberrypi ~ $ ls /bin
bash      bzfgrep      chgrp      dash          domainname  fgconsole
bunzip2   bzgrep       chmod      date          dumpkeys    fgrep
bzcat     bzip2        chown      dd            echo        findmnt
bzcmp     bzip2recover chvt       df            ed          fuser
bzdiff    bzless       con2fbmap  dir           egrep       fusermount
bzegrep   bzmore       cp         dmesg         false       grep
bzexe     cat          cpio       dnsdomainname fbset       gunzip
pi@raspberrypi ~ $
```

STEP 3

Entering: `ls /home` displays the contents of your home directory, which contains pi; the directory that you start in. So, entering: `ls/home/pi` is the same as just "ls" from the default home directory. This is where you are expected to place most of the documents you create. Don't confuse home with "usr"; the /usr directory is where find you find program tools and libraries.

```
pi@raspberrypi ~ $ ls
articles.txt  Desktop  Documents  Downloads  indiecity  names.txt  pytho
pi@raspberrypi ~ $ ls /home/pi
articles.txt  Desktop  Documents  Downloads  indiecity  names.txt  pytho
pi@raspberrypi ~ $
```

STEP 4

Lib is a directory that contains libraries of code that are referred to by other programs (different programs share files in Lib). "Var" is short for various, which is mostly files used by the system, but you may need to work with items here. Finally there is a directory called "tmp", which is for temporary files; files placed here are on your system for the short term and can be deleted from the system.

```
pi@raspberrypi ~ $ ls /var
backups  cache  lib  local  lock  log  mail  opt  run  spool  swap  tmp
pi@raspberrypi ~ $
```

Creating and Deleting Files

Being able to create and delete a file is an everyday computing skill. However, when using the Linux Terminal, there's an element of care required, chiefly because any deleted files aren't placed in the system recycle bin.

CREATING FILES

Once you learn to recognise the files and directories that make up Raspbian OS, it's time to discover how to make your own. Knowing how to make, edit and delete files and directories is essential if you want to make your own projects.

STEP 1

We're going to create a file using a command called Touch. Touch is an interesting command that reaches out to a file, or directory, and updates it (this changes the system time as if you'd just opened the file). You can see Touch in access using "ls –l" and checking the time next to a directory (such as Scratch).

```
pi@raspberrypi ~ $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 May 13 10:57 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 Apr 17 12:53 Scratch
pi@raspberrypi ~ $ _
```

STEP 2

Now enter: `touch Scratch` and `ls -l` again and notice that the time has changed. It now matches the current time. You might be wondering what this has to do with creating files or directories. Touch has a second, more popular, use, which is to create files.

```
pi@raspberrypi ~ $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 May 13 10:57 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 Apr 17 12:53 Scratch
pi@raspberrypi ~ $ touch Scratch
pi@raspberrypi ~ $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 May 13 10:57 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 May 13 11:05 Scratch
pi@raspberrypi ~ $ _
```

STEP 3

If you try to touch a file that doesn't exist, you create a blank file with that name. Try it now. Type `touch testfile` and `ls -l` to view the files. You'll now have a new file in your home directory called "testfile". Notice that the size of the file is 0, because it has nothing in it.

```
pi@raspberrypi ~ $ touch testfile
pi@raspberrypi ~ $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 May 13 10:57 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 May 13 11:05 Scratch
-rw-r--r-- 1 pi pi    0 May 13 11:10 testfile
pi@raspberrypi ~ $ _
```

STEP 4

A quick word about file names: remember that Linux is case sensitive, so if you now enter: `touch Testfile` (with a capital T), it doesn't update 'testfile'; instead, it creates a second file called 'Testfile'. Enter: `ls -l` to see both files. This is confusing, so most people stick with using lowercase letters at all times.

```
pi@raspberrypi ~ $ touch testfile
pi@raspberrypi ~ $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 May 13 10:57 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 May 13 11:05 Scratch
-rw-r--r-- 1 pi pi    0 May 13 11:08 testfile
pi@raspberrypi ~ $ touch Testfile
pi@raspberrypi ~ $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 May 13 10:57 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 May 13 11:05 Scratch
-rw-r--r-- 1 pi pi    0 May 13 11:08 testfile
-rw-r--r-- 1 pi pi    0 May 13 11:10 Testfile
pi@raspberrypi ~ $ _
```

STEP 5 Another important thing to know is never to use a space in your file names. If you try to enter: `touch test file`, you create a document called "test" and another called "file". Technically there are ways to create files containing a space but you should always use an underscore character ("_") instead of a space, such as "touch test_file".

```
pi@raspberrypi ~ $ touch test file
pi@raspberrypi ~ $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 May 13 10:57 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
-rw-r--r-- 1 pi pi    0 May 13 11:15 file
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 May 13 11:05 Scratch
-rw-r--r-- 1 pi pi    0 May 13 11:15 test
-rw-r--r-- 1 pi pi    0 May 13 11:10 testfile
-rw-r--r-- 1 pi pi    0 May 13 11:12 Testfile
pi@raspberrypi ~ $ _
```

STEP 6 Here are some other files names to avoid: #%&{}\<>*?/$!'":@+`|=. The full stop (.) is used to create an extension to a file; usually used to indicate a file type, such as textfile.txt or compressedfile.zip, and starting a file with a full stop makes it invisible. Don't use a full stop in place of a space though; stick to underscores.

```
pi@raspberrypi ~ $ touch don't.use{odd}symbols&in<filenames>or=you'll^confu
```

REMOVING FILES

We've created some files that we don't want, so how do we go about removing them? It turns out that deleting files in your Raspberry Pi is really easy, which may be a problem, so be careful.

STEP 1 Enter: `ls -l` to view the files in your home directory. If you've followed the steps before then you should have three files: "test", "testfile", and "Testfile". We're going to get rid of these items because they were created as an example.

```
pi@raspberrypi ~ $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 May 13 10:57 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
-rw-r--r-- 1 pi pi    0 May 13 11:15 file
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 May 13 11:05 Scratch
-rw-r--r-- 1 pi pi    0 May 13 11:15 test
-rw-r--r-- 1 pi pi    0 May 13 11:10 testfile
-rw-r--r-- 1 pi pi    0 May 13 11:46 Testfile
pi@raspberrypi ~ $
```

STEP 2 **To get rid of files you use the "rm" command. Enter: `rm Testfile` to delete the file called "Testfile" (with the uppercase "t"). Enter: `ls -l` and you'll find it's gone. Where is it? It's not in the Trash or Recycle Bin, like on a Mac or Windows PC. It's deleted completely and cannot be recovered. Bear this in mind and always think before deleting files.**

```
pi@raspberrypi ~ $ rm Testfile
pi@raspberrypi ~ $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 5 pi pi 4096 May 13 10:57 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
-rw-r--r-- 1 pi pi    0 May 13 11:15 file
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 May 13 11:05 Scratch
-rw-r--r-- 1 pi pi    0 May 13 11:15 test
-rw-r--r-- 1 pi pi    0 May 13 11:10 testfile
pi@raspberrypi ~ $ _
```

STEP 3 We're going to use a wildcard (*) to delete our next two files, but again this is something you really need to do with care. First use "ls" to list the files and make sure it's the one you want to delete. Enter: `ls test*` to view files that match the word "test" and any other characters. The "*" character is called a "wildcard" and it means any characters here.

```
pi@raspberrypi ~ $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Jul  9 08:36 Desktop
drwxr-xr-x 2 pi pi 4096 Jul  9 08:36 Documents
drwxr-xr-x 2 pi pi 4096 Jul  9 08:36 Downloads
-rw-r--r-- 1 pi pi    0 Jul  9 08:37 file
drwxr-xr-x 2 pi pi 4096 Jul  9 08:36 indiecity
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 Jul  9 08:36 Scratch
-rw-r--r-- 1 pi pi    0 Jul  9 08:37 test
-rw-r--r-- 1 pi pi    0 Jul  9 08:37 testfile
pi@raspberrypi ~ $ ls test*
test  testfile
pi@raspberrypi ~ $ _
```

STEP 4 We see that "ls test*" matches two files: "test" and "testfile", but not the file called "file". That's because it didn't match the "test" part of "test*". Check carefully over groups of files you want to remove (remember you can't recover them) and replace the "ls" with "rm". Enter: `rm test*` to remove both files. Finally enter: `rm file` to get rid of the confusing file.

```
pi@raspberrypi ~ $ rm test*
pi@raspberrypi ~ $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Jul  9 08:36 Desktop
drwxr-xr-x 2 pi pi 4096 Jul  9 08:36 Documents
drwxr-xr-x 2 pi pi 4096 Jul  9 08:36 Downloads
-rw-r--r-- 1 pi pi    0 Jul  9 08:37 file
drwxr-xr-x 2 pi pi 4096 Jul  9 08:36 indiecity
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 Jul  9 08:36 Scratch
pi@raspberrypi ~ $ rm file
pi@raspberrypi ~ $
```

Create and Remove Directories

Creating, moving and deleting directories isn't as easy in the Terminal as it is within a desktop interface. You need to tell Linux to move the directories inside other directories, a process known as recursion. Sounds complex but you should quickly get the hang of it.

MANAGING FILES AND DIRECTORIES

Now that you know how to create files, you'll want to learn how to make directories, which are the same thing as folders, as well as move items around. If you are more used to working with a desktop interface, this can take a bit of getting used to.

STEP 1

Enter: `ls` to quickly view all the directories currently in in the home location. Directories are created using the "mkdir" command (make directory). Enter: `mkdir testdir` to create a new directory in your home directory. Enter: `ls` again to see it.

```
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  python_games  Sc
pi@raspberrypi ~ $ mkdir testdir
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  python_games  Sc
pi@raspberrypi ~ $ _
```

STEP 2

The "mkdir" command is different to touch, in that it doesn't update the timestamp if you use it with a directory that already exists. Enter: `mkdir testdir` again and you'll get the error "mkdir: cannot create directory 'testdir: File exists".

```
pi@raspberrypi ~ $ mkdir testdir
mkdir: cannot create directory `testdir': File exists
pi@raspberrypi ~ $ _
```

STEP 3

Like touch, you can create multiple directories at once with the mkdir command. Enter: `mkdir testdir2 testdir3` and enter: `ls`. You'll now find several directories called testdir. Also, like files, you should know this means you can't (and really shouldn't) create directories with spaces. As with files, use an underscore ("_") character instead of a space.

```
pi@raspberrypi ~ $ mkdir testdir2 testdir3
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  python_games  Sc
pi@raspberrypi ~ $ _
```

STEP 4

You can create directories inside of each other using the directory path. Enter: `mkdir Documents/photos` to create a new directory called "photos" inside your documents directory. The directory has to already exist, though, try to enter: `mkdir articles/reports` and you'll get an error because there is no articles directory.

```
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  python_games  Sc
pi@raspberrypi ~ $ mkdir Documents/photos
pi@raspberrypi ~ $ mkdir articles/reports
mkdir: cannot create directory `articles/reports': No such
pi@raspberrypi ~ $
```

STEP 5 To create a directory path you need to pass in the "p" option to mkdir (which stands for "parents"). Options, if you remember, come after the command and start with a '-'. So enter: `mkdir -p articles/reports`. Enter: `ls` to view the articles directory, or "ls articles" to view the reports directory sitting inside.

```
pi@raspberrypi ~ $ mkdir -p articles/reports
```

STEP 6 Now you're starting to get a bit more advanced, we're going to just reiterate something. In Linux the command structure is always: command, option and argument, in that order. The command is the function, next are the options (typically single letters starting with "-") and finally the argument (often a file, or directory structure). It's always command, option then argument.

```
pi@raspberrypi ~ $ ls -l articles
total 4
drwxr-xr-x 2 pi pi 4096 May 13 12:36 reports
pi@raspberrypi ~ $ _
```

GETTING RID OF DIRECTORIES

Deleting directories is pretty easy in Linux, along with files, and this can be a problem. It's too easy to delete entire directories containing files and these are instantly removed, not sent to a trash directory. Tread carefully.

STEP 1 We're going to remove one of the directories we created earlier using the "rmdir" command. Enter: `ls` to view the files and directories in the current directory. We'll start by getting rid of one of the test directories. Enter: `rmdir testdir3` and `ls` again to confirm the directory has been removed.

```
pi@raspberrypi ~ $ ls
articles  Desktop  Documents  Downloads  indiecity  python
pi@raspberrypi ~ $ rmdir testdir3_
```

STEP 2 Now we'll try to get rid of the articles directory (containing the reports directory). Enter: `rmdir articles` and press return. You'll get an error saying "rmdir: failed to remove 'articles': Directory not empty". This is a puzzler; the rmdir command only removes directories that having nothing in them (no files or other directories).

```
pi@raspberrypi ~ $ rmdir articles
rmdir: failed to remove `articles': Directory not empty
pi@raspberrypi ~ $ _
```

STEP 3 To delete a directory containing files or other directories, you return to the "rm" command used to remove files, only now we need to use the "-R" option (which stands for "recursive".) Using "rm –R" removes all the files and directories to whatever you point it at. Enter: `rm -R articles` to remove the articles directory.

```
pi@raspberrypi ~ $ ls
articles  Desktop  Documents  Downloads  indiecity  python
pi@raspberrypi ~ $ rm -R articles
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  python_games  Sc
pi@raspberrypi ~ $
```

STEP 4 As with multiple files, you can delete multiple directories inside the same directory using the "rm" command with the wildcard character (`*`). This should be done with care though so use the `-I` option (which stands for "interactive"). This will prompt you before each deletion. Enter: `rm -Ri test*` and press `Y` and `return` to each prompt. It's a good idea to use the `-i` option whenever using the `rm` command.

```
pi@raspberrypi ~ $ rm -Ri test*
rm: remove directory `testdir'? y
rm: remove directory `testdir2'? y
rm: remove directory `testdir3'? y_
```

Copying, Moving and Renaming Files

Taking command of the Terminal is essential when learning how your Raspberry Pi's operating system works. The copying, moving and renaming of files is equally important, as you'll be doing a lot of this throughout your Pi projects.

USING THE MOVE COMMAND

In Linux, renaming a file is simply moving it from one name to another and copying a file is moving it without deleting the original. Don't panic, it's quite easy to master.

STEP 1 Before we can move anything around, we need to have a few test items in our home directory. Enter: `touch testfile` and `mkdir testdir` to create a test file and test directory in your home directory. Enter: `ls` to check that they are both present.

```
pi@raspberrypi ~ $ touch testfile
pi@raspberrypi ~ $ mkdir testdir
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  python_games  Sc
pi@raspberrypi ~ $
```

STEP 2 Files and directories are moved using the mv command. This is different to the commands we've looked at so far because it has two arguments (remember Linux command line is command, option, argument). The first argument is the source (the file or directory to be moved) and the second is the destination.

```
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  python_games  Sc
pi@raspberrypi ~ $ mv testfile testdir
```

STEP 3 Enter: `mv testfile testdir` and press return to move the testfile document into the testdir directory. Enter: `ls` to see that it's no longer in the home directory, and `ls testdir` to see the testfile now sitting in the testdir directory. Now enter: `mkdir newparent` to create a new directory.

```
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  python_games  Sc
pi@raspberrypi ~ $ mv testfile testdir
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  python_games  Sc
pi@raspberrypi ~ $ ls testdir
testfile
pi@raspberrypi ~ $
```

STEP 4 Directories with files are moved in the same way. Enter: `mv testdir newparent` to move the testdir directory inside the newparent directory. Let's move into the directory to find the file. Enter: `cd /newparent/testdir` and enter: `ls` to view the testfile sitting inside the directory.

```
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  python_games  Sc
pi@raspberrypi ~ $ mkdir newparent
pi@raspberrypi ~ $ mv testdir newparent
pi@raspberrypi ~ $ cd newparent/testdir
pi@raspberrypi ~/newparent/testdir $ ls
testfile
pi@raspberrypi ~/newparent/testdir $
```

STEP 5

Files and directories can be moved up using the double dot ("..") as an argument. Enter: `ls -la` to view your testfile and the single and double dot files. The single dot is the current directory and the double dot is the parent directory. Enter: `mv testfile ..` to move the testfile up into the newparent directory. Enter: `cd ..` to move up to the parent directory.

```
pi@raspberrypi ~ $ cd newparent/testdir
pi@raspberrypi ~/newparent/testdir $ ls
testfile
pi@raspberrypi ~/newparent/testdir $ mv testfile ..
pi@raspberrypi ~/newparent/testdir $ cd
```

STEP 6

You can also move files using longer paths. Enter: `cd ~` to return to the home directory and `mv newparent/testfile newparent/testdir/testfile` to move the testfile from its current location back inside the testdir directory. Enter: `ls newparent/testdir` to view the file back in its current directory.

```
pi@raspberrypi ~/newparent $ cd ~
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  newparent  pytho
pi@raspberrypi ~ $ mv newparent/testfile newparent/testdir
pi@raspberrypi ~ $ ls newparent/testdir
testfile
pi@raspberrypi ~ $ _
```

RENAMING FILES AND DIRECTORIES

The mv command isn't used just to move files; it also serves the purpose of renaming files (effectively it moves it from its old name to a new name). Let's see how to use mv to rename items.

STEP 1

Let's start by making a new test file called "names". Enter: `touch testfile` and then `ls` to make sure the testfile is present. We're going to turn this into a file that contains the names of some people. So let's call it something more appropriate, like "names".

```
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  newparent  pytho
pi@raspberrypi ~ $ mv testfile names
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  names  newparent
pi@raspberrypi ~ $
```

STEP 2

Enter: `mv testfile names` and `ls`. Now we can see the new "names" file in our directory. The mv command can also be used to rename directories. We should still have our newparent directory in our home directory. Enter: `mv newparent people` to rename the newparent directory. Enter: `ls` to view it.

```
pi@raspberrypi ~ $ touch testfile
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  newparent  pytho
pi@raspberrypi ~ $ mv newparent people
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  people  python_g
pi@raspberrypi ~ $
```

STEP 3

You can rename directories inside other directories using paths. Let's rename the testdir directory, which is now inside the people directory. Enter: `mv names/testdir names/friends`. Now enter: `mv names people/friends` to move the names file inside the friends directory.

```
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  names  people  p
pi@raspberrypi ~ $ mv people/testdir people/friends
```

STEP 4

It is easy to overwrite files using the mv command, so if you have files with the same name use the "-n" option, which stands for "no overwrite". Enter: `touch testfile` to create a new file and `mv -n testfile people/friends`. There's no error report though, enter: `ls` and you'll find testfile still there.

```
pi@raspberrypi ~ $ touch testfile
pi@raspberrypi ~ $ mv -n testfile people/friends
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  people  python_g
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  people  python_g
pi@raspberrypi ~ $ ls people/friends
names  testfile
pi@raspberrypi ~ $
```

Using the Man Pages

Linux comes with man (manual) pages that explain each command and show you all the options you can use. Once you get the hang of reading the man pages, you'll be able to find and do just about anything in Linux.

HEY, MAN!

The man pages are one of the best features of Linux, and as a built-in tool it's invaluable for both beginner and senior level Linux administrators. Let's see how it works

STEP 1

Linux has a built-in manual, known as man for short. Using the man command you can obtain information on all the Linux commands we've talked about. Simply enter: `man` and the name of the command you want to learn more about. Start by entering: `man ls` in the command line.

```
pi@raspberrypi ~ $ man ls
```

STEP 2

The man pages are a bit more detailed than you might be used to. First you have a name, which tells you what the command is called; in this case "list directory contents" and then the synopsis shows you how it works. In this case: "ls [OPTION].. [FILE..]". So you enter: `ls` followed by options (such as `-la`) and the file or directory to list.

```
LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List information about the FILEs (the current directory by default).  Sort entries alph

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              scale sizes by SIZE before printing them.  E.g., `--block-size=M' prints sizes i

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last modification of file status inf

       -C     list entries by columns

       --color[=WHEN]
              colorize the output.  WHEN defaults to `always' or can be `never' or `auto'.  Mo

       -d, --directory
              list directory entries instead of contents, and do not dereference symbolic link

       -D, --dired
              generate output designed for Emacs' dired mode

       -f     do not sort, enable -aU, disable -ls --color

       -F, --classify
              append indicator (one of */=>@|) to entries

       --file-type
              likewise, except do not append `*'

       --format=WORD
              across -x, commas -m, horizontal -x, long -l, single-column -1, verbose -l, vert

       --full-time
              like -l --time-style=full-iso

 Manual page ls(1) line 1 (press h for help or q to quit)
```

STEP 3

Most commands are pretty easy to figure out how to use, so what you spend most of the time in the man pages is looking under the Description. Here you will see all the options and the letters used to activate them. Most man pages are longer than a single page, so press any key, such as the space bar, to move to the next page of content.

```
       -g     like -l, but do not list owner

       --group-directories-first
              group directories before files.

              augment with a --sort option, but any use of --sort=none (-U) disables grouping

       -G, --no-group
              in a long listing, don't print group names

       -h, --human-readable
              with -l, print sizes in human readable format (e.g., 1K 234M 2G)

       --si   likewise, but use powers of 1000 not 1024

       -H, --dereference-command-line
              follow symbolic links listed on the command line

       --dereference-command-line-symlink-to-dir
              follow each command line symbolic link that points to a directory

       --hide=PATTERN
              do not list implied entries matching shell PATTERN (overridden by -a or -A)

       --indicator-style=WORD
              append indicator with style WORD to entry names: none (default), slash (-p), file-type (-

       -i, --inode
              print the index number of each file

       -I, --ignore=PATTERN
```

STEP 4

Press the H key while looking at a man page to view the commands you can use to control the view. This is called the Summary of Less Commands (the less command is something we'll come to when we look at editing text). For now realise that you can move back and forward with Z and W. Press Q to quit this help screen and return to the man page.

```
                    SUMMARY OF LESS COMMANDS

      Commands marked with * may be preceded by a number, N.
      Notes in parentheses indicate the behavior if N is given.

  h  H                 Display this help.
  q  :q  Q  :Q  ZZ     Exit.
 ---------------------------------------------------------------------------

                           MOVING

  e  ^E  j  ^N  CR  *  Forward  one line   (or N lines).
  y  ^Y  k  ^K  ^P  *  Backward one line   (or N lines).
  f  ^F  ^V  SPACE  *  Forward  one window (or N lines).
  b  ^B  ESC-v      *  Backward one window (or N lines).
  z                 *  Forward  one window (and set window to N).
  w                 *  Backward one window (and set window to N).
  ESC-SPACE         *  Forward  one window, but don't stop at end-of-file.
  d  ^D             *  Forward  one half-window (and set half-window to N).
  u  ^U             *  Backward one half-window (and set half-window to N).
  ESC-)  RightArrow *  Left  one half screen width (or N positions).
  ESC-(  LeftArrow  *  Right one half screen width (or N positions).
  F                    Forward forever; like "tail -f".
  r  ^R  ^L            Repaint screen.
  R                    Repaint screen, discarding buffered input.
        ---------------------------------------------------
        Default "window" is the screen height.
        Default "half-window" is half of the screen height.
 ---------------------------------------------------------------------------

                         SEARCHING

  /pattern          *  Search forward for (N-th) matching line.
  ?pattern          *  Search backward for (N-th) matching line.
  n                 *  Repeat previous search (for N-th occurrence).
  N                 *  Repeat previous search in reverse direction.
  ESC-n             *  Repeat previous search, spanning files.
  ESC-N             *  Repeat previous search, reverse dir. & spanning files.
  ESC-u                Undo (toggle) search highlighting.
  &pattern          *  Display only matching lines
        ---------------------------------------------------
        Search patterns may be modified by one or more of:
        ^N or !  Search for NON-matching lines.
        ^E or *  Search multiple files (pass thru END OF FILE).
        ^F or @  Start search at FIRST file (for /) or last file (for ?).
        ^K       Highlight matches, but don't move (KEEP position).
```

STEP 5 Scroll to the bottom of the man page to discover more information. Typically you will find the author's name and information on reporting bugs, including web links that can be useful for more information. Press Q to exit the man page and return to the command line.

```
       assume tab stops at each COLS instead of 8

  -u   with -lt: sort by, and show, access time with -l: show access time and sort by name oth

  -U   do not sort; list entries in directory order

  -v   natural sort of (version) numbers within text

  -w, --width=COLS
       assume screen width instead of current value

  -x   list entries by lines instead of by columns

  -X   sort alphabetically by entry extension

  -Z, --context
       print any SELinux security context of each file

  -1   list one file per line

  --help display this help and exit

  --version
       output version information and exit

  SIZE may be (or may be an integer optionally followed by) one of following: KB 1000, K 1024, M
```

STEP 6 The man command can be used for just about every command you use in Linux. You can even enter: man man to get information on using the man tool. From now on, whenever you come across a new command in this book, such as "nano" or "chmod", take time to enter: `man nano` or `man chmod` and read the instructions.

```
MAN(1)

NAME
       man - an interface to the on-line reference manuals

SYNOPSIS
       man  [-C  file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale] [-m system[,
       [-r prompt] [-7] [-E encoding] [--no-hyphenation] [--no-justification] [-p string] [-t]
       man -k [apropos options] regexp ...
       man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
       man -f [whatis options] page ...
       man -l [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale] [-P pager]
       man -w|-W [-C file] [-d] [-D] page ...
       man -c [-C file] [-d] [-D] page ...
       man [-hV]

DESCRIPTION
       man is the system's manual pager. Each page argument given to man is normally the name o
       section,  if  provided,  will  direct man to look only in that section of the manual.  T
       page found, even if page exists in several sections.

       The table below shows the section numbers of the manual followed by the types of pages t

       1   Executable programs or shell commands
```

USING MAN OPTIONS

Because man doesn't change anything, like mv or mkdir, it is tempting not to see it as a command. But it is, and like all other commands it has options. These can be very handy to learn.

STEP 1 Entering: `man man` enables you to view some of the options, but sometimes you'll just want a quick overview. Fortunately man has a built-in help option that quickly lists the options. Press Q if you're in a man page and enter: `man -h` at the command line.

```
                          print physical location of cat file(s)

   -c, --catman           used by catman to reformat out of date cat pages
   -R, --recode=ENCODING  output source page encoded in ENCODING

  Finding manual pages:
   -L, --locale=LOCALE    define the locale for this particular man search
   -m, --systems=SYSTEM   use manual pages from other systems
   -M, --manpath=PATH     set search path for manual pages to PATH

   -S, -s, --sections=LIST   use colon separated section list

   -e, --extension=EXTENSION  limit search to extension type EXTENSION

   -i, --ignore-case      look for pages case-insensitively (default)
   -I, --match-case       look for pages case-sensitively

       --regex            show all pages matching regex
       --wildcard         show all pages matching wildcard

       --names-only       make --regex and --wildcard match page names only,
                          not descriptions

   -a, --all              find all matching manual pages
   -u, --update           force a cache consistency check

       --no-subpages      don't try subpages, e.g. 'man foo bar' => 'man
                          foo-bar'

  Controlling formatted output:
   -P, --pager=PAGER      use program PAGER to display output
   -r, --prompt=STRING    provide the `less' pager with a prompt

   -7, --ascii            display ASCII translation of certain latin1 chars
   -E, --encoding=ENCODING    use selected output encoding
       --no-hyphenation, --nh turn off hyphenation
       --no-justification,                    --nj   turn off justification
   -p, --preprocessor=STRING  STRING indicates which preprocessors to run:
                          e - [n]eqn, p - pic, t - tbl,
g - grap, r - refer, v - vgrind

   -t, --troff            use groff to format pages
   -T, --troff-device[=DEVICE]   use groff with selected device

   -H, --html[=BROWSER]   use www-browser or BROWSER to display HTML output
   -X, --gxditview[=RESOLUTION]   use groff and display through gxditview
                          (X11):
                          -X = -TX75, -X100 = -TX100, -X100-12 = -TX100-12
   -Z, --ditroff          use groff and force it to produce ditroff

   -?, --help             give this help list
       --usage            give a short usage message
```

STEP 2 If you're fast you may have noticed the start of the text flew up off the page. This is because the "man –h" option doesn't use the less command by default (less is what enables you to move down text one screen at a time). We'll look into pipes ("|") later on, but for now just use "man -h | less" to read long text one page at a time.

```
pi@raspberrypi ~ $ man -h | less
```

STEP 3 One of the most powerful man options is the -k option, which is for "apropos". This enables you to search a wider range of man pages than the exact command. Enter: `man -k directory` to view all of the man pages relating to directories "(man -k directory | less" to view one page at a time). Here you'll find commands like "ls", "mkdir" and "cd" along with their description.

```
fchdir (2)           - change working directory
fchmodat (2)         - change permissions of a file relative to a directory file descriptor
fchownat (2)         - change ownership of a file relative to a directory file descriptor
fdopendir (3)        - open a directory
find (1)             - search for files in a directory hierarchy
fstatat (2)          - get file status relative to a directory file descriptor
fstatat64 (2)        - get file status relative to a directory file descriptor
futimesat (2)        - change timestamps of a file relative to a directory file descriptor
get_current_dir_name (2) - get current working directory
get_current_dir_name (3) - get current working directory
getcwd (2)           - get current working directory
getcwd (3)           - get current working directory
getdents (2)         - get directory entries
getdents64 (2)       - get directory entries
getdirentries (3)    - get directory entries in a file system-independent format
getwd (3)            - get current working directory
git-clone (1)        - Clone a repository into a new directory
git-mv (1)           - Move or rename a file, a directory, or a symlink
git-stash (1)        - Stash the changes in a dirty working directory away
helpztags (1)        - generate the help tags file for directory
linkat (2)           - create a file link relative to directory file descriptors
lookup_dcookie (2)   - return a directory entry's path
ls (1)               - list directory contents
mkdir (2)            - create a directory
mkdirat (2)          - create a directory relative to a directory file descriptor
mkdtemp (3)          - create a unique temporary directory
mkfifoat (3)         - make a FIFO (named pipe) relative to a directory file descriptor
mkfontdir (1)        - create an index of X font files in a directory
mklost+found (8)     - create a lost+found directory on a mounted Linux second extended file s
mknodat (2)          - create a special or ordinary file relative to a directory file descript
mktemp (1)           - create a temporary file or directory
modprobe.d (5)       - Configuration directory for modprobe
mountpoint (1)       - see if a directory is a mountpoint
oldfind (1)          - search for files in a directory hierarchy
openat (2)           - open a file relative to a directory file descriptor
opendir (3)          - open a directory
pam_mkhomedir (8)    - PAM module to create users home directory
pwd (1)              - print name of current/working directory
pwdx (1)             - report current working directory of a process
readdir (2)          - read directory entry
readdir (3)          - read a directory
readdir_r (3)        - read a directory
readlinkat (2)       - read value of a symbolic link relative to a directory file descriptor
remove (3)           - remove a file or directory
renameat (2)         - rename a file relative to directory file descriptors
rewinddir (3)        - reset directory stream
rmdir (2)            - delete a directory
run-parts (8)        - run scripts or programs in a directory
scandir (3)          - scan a directory for matching entries
scandirat (3)        - scan a directory relative to a directory file descriptor
```

STEP 4 Entering the man page for all the commands you come across can be a little long-winded, although ultimately productive. If you simply want to know what a command does you can read just the description using the "whatis" command. Enter: `whatis pwd` to read the description of the "pwd" command ("print name of current/working directory").

```
pi@raspberrypi ~ $ whatis pwd
pwd (1)              - print name of current/working directory
pi@raspberrypi ~ $
```

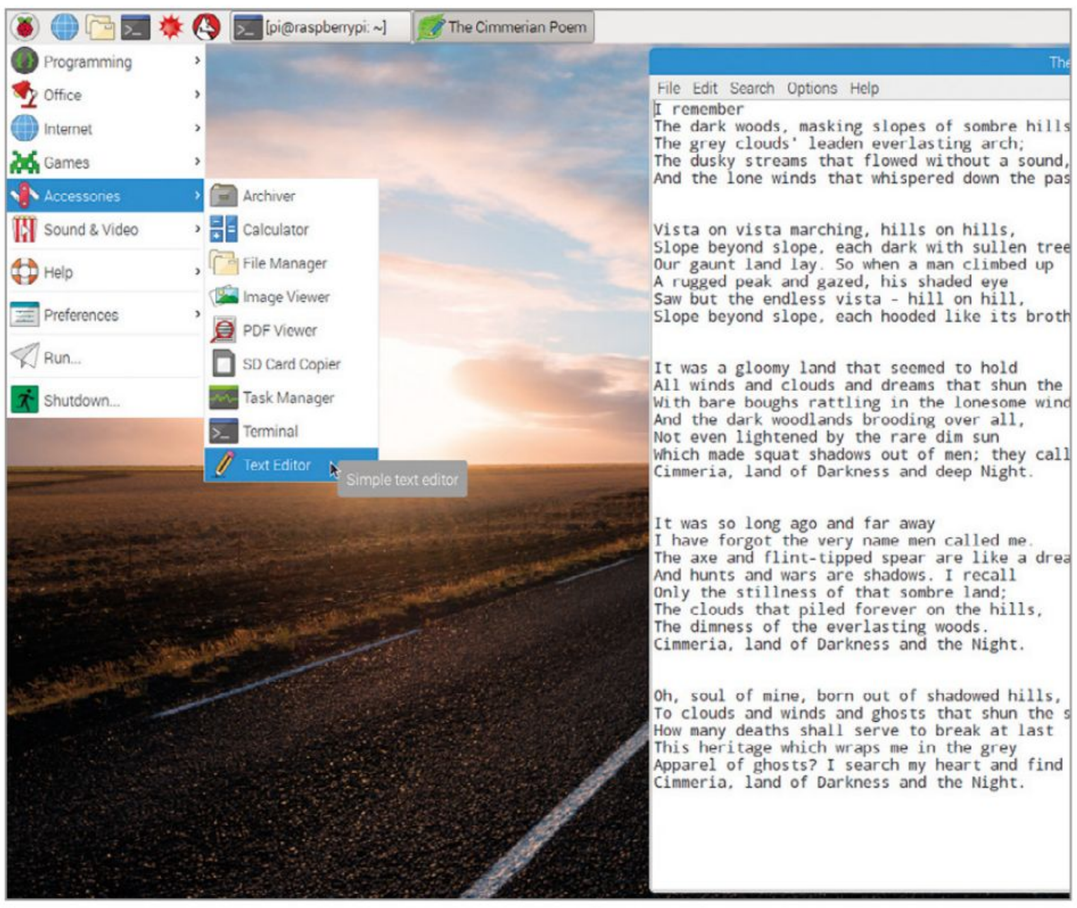
Editing Text Files

A text file in Linux can be anything from a simple set of instructions on how to use an app, to some complex Python, C++ or other programming language code. Text files can be used for scripting, automated executable files, as well as configuration files too.

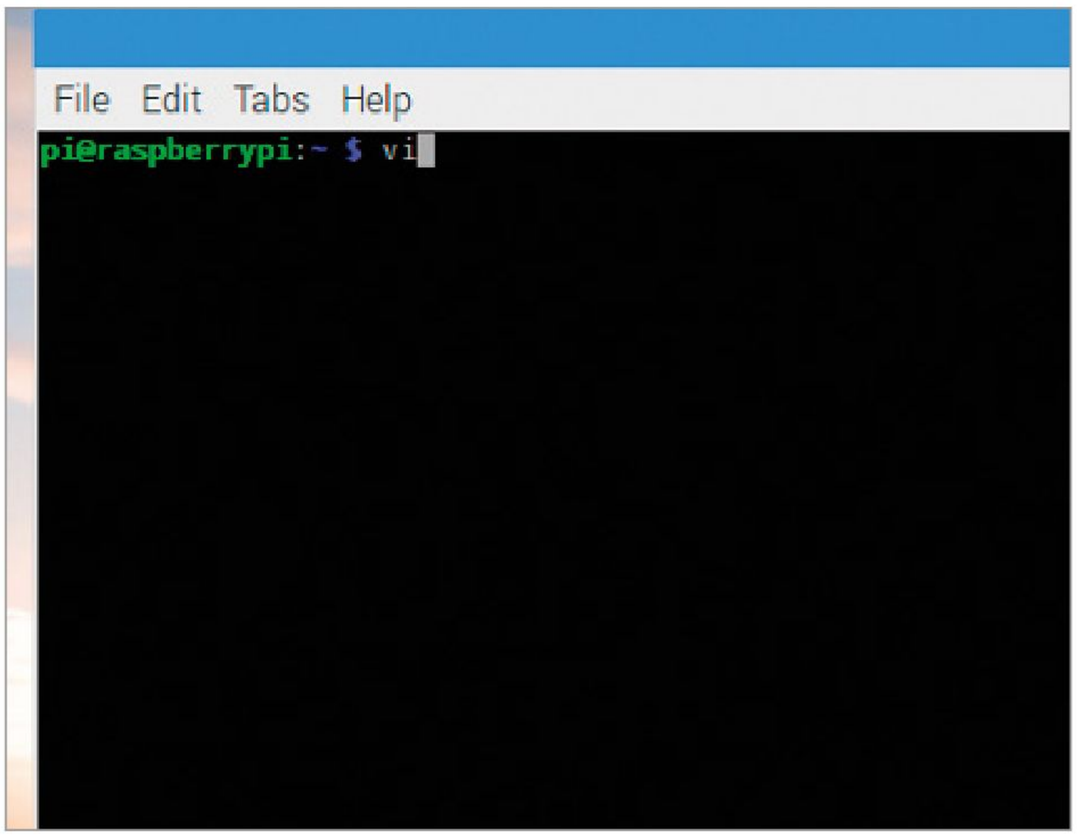
THE JOY OF TEXT

To be able to edit or create a text file, you need a good text editor. Linux has many but here are some in action on the Raspberry Pi.

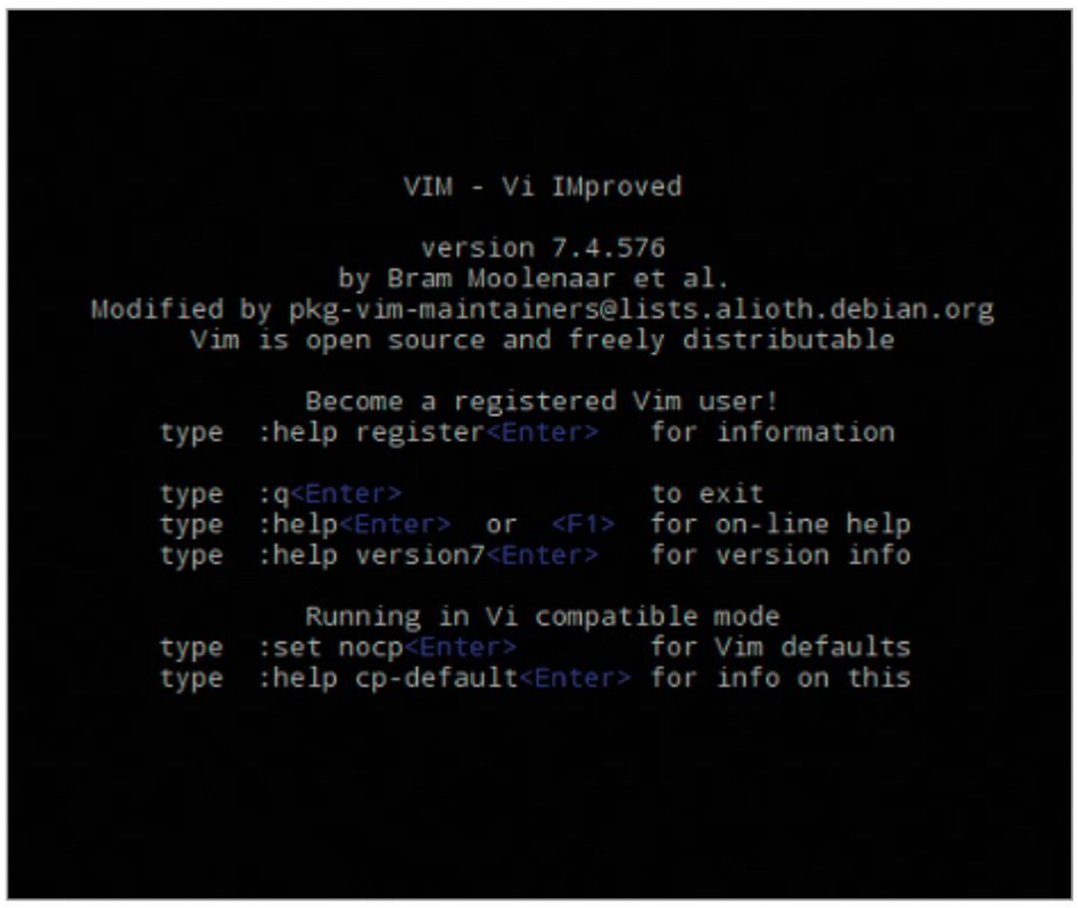
STEP 1 The first text editor for the Raspberry Pi is the default desktop environment app: Leafpad. To use, you can either double-click an existing text file or click the Raspberry Pi menu icon (in the top left of the desktop) and from the Accessories menu, choose Text Editor.



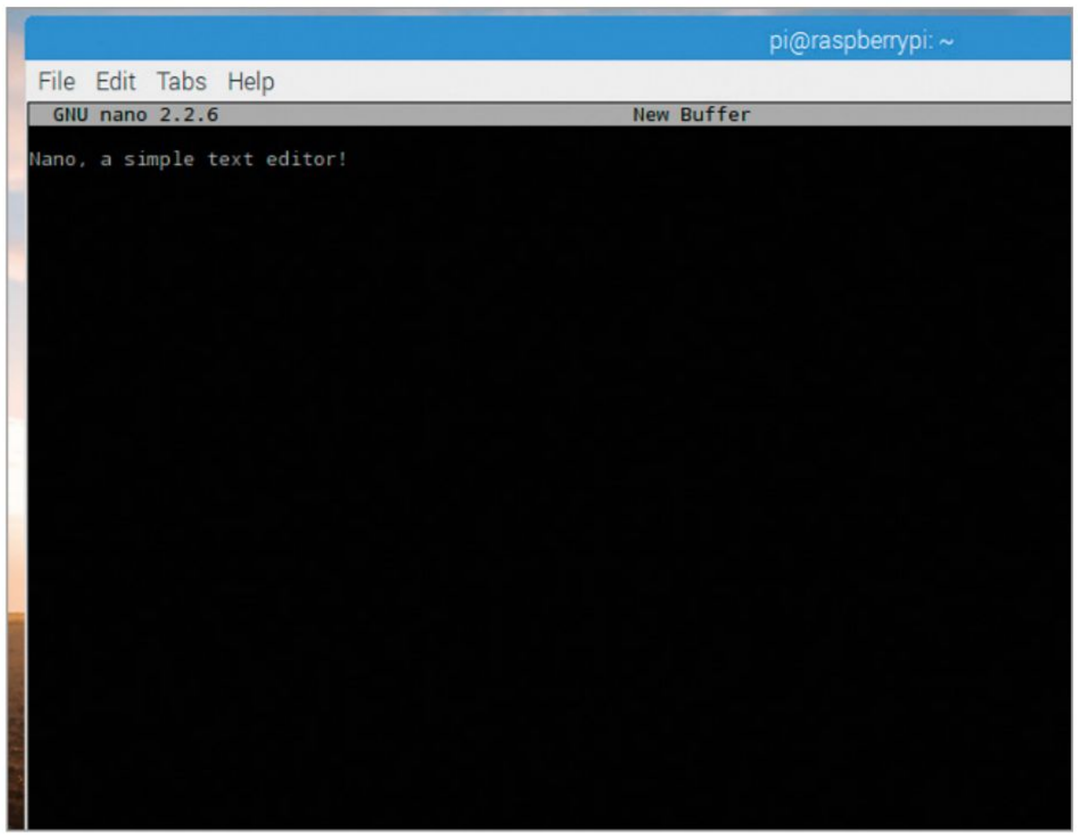
STEP 2 From the Terminal there are even more options, although using the correct command, you can launch any of the desktop apps via the Terminal. One of the simplest, and a classic text editor that's carried over from the Unix days, is vi. In the Terminal, enter: `vi`.



STEP 3 Vi is the original Unix command but in this case it launches VIM, the new Linux version of Vi. Although simple looking, Vi is considered, even by today's standards, to be one of the most widely used text editors, There's a lot you can do with it, so check out the man pages for more Vi information.

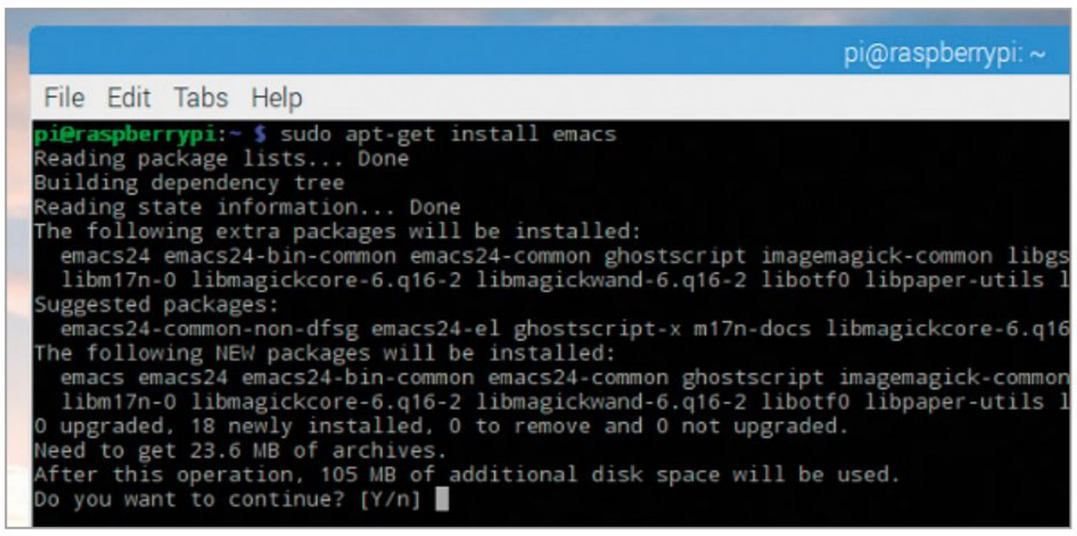


STEP 4 Nano is another favourite, and simple, text editor available for Linux. Enter: `nano` into the Terminal to launch it. You can use Nano for editing code, creating scripts or writing your own help files. To exit Nano, press Ctrl + X, followed by Y to save the file or N to exit without saving.

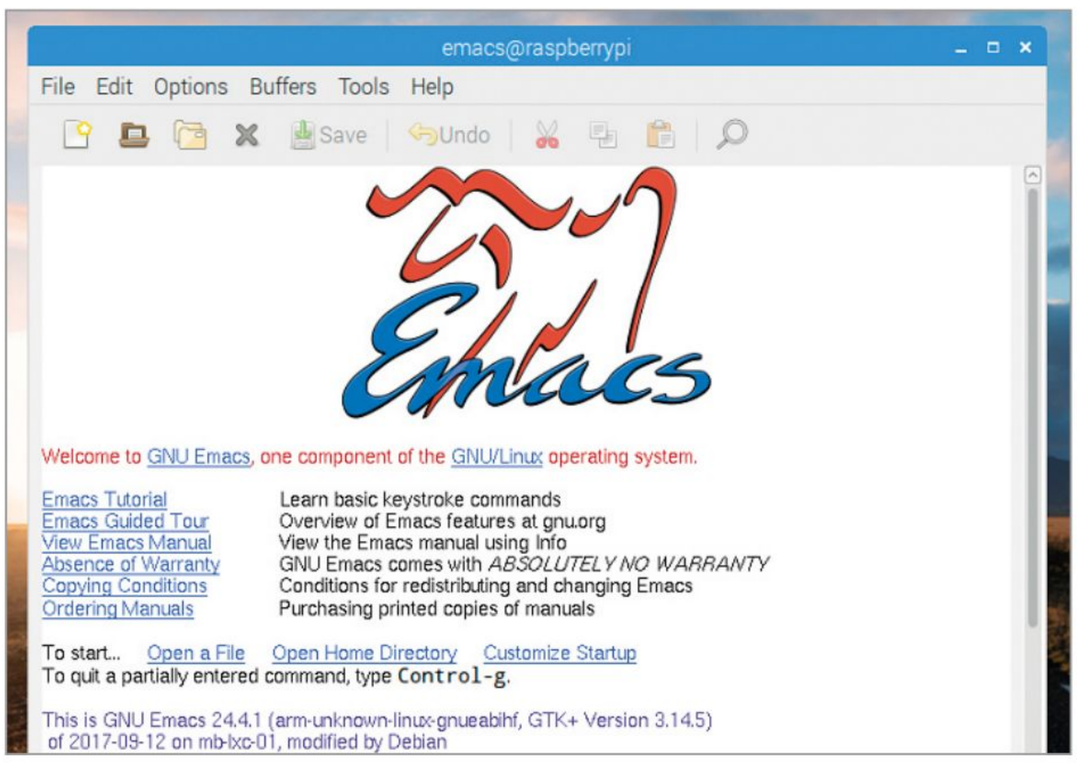


STEP 5 Emacs, or GNU Emacs, is an extensible and customisable, self-documenting, real-time display editor. It's a fantastic text editor and one that's worth getting used to as soon as you can. Sadly, it's not installed on the Pi by default, so you'll need to install it. In the Terminal, enter:

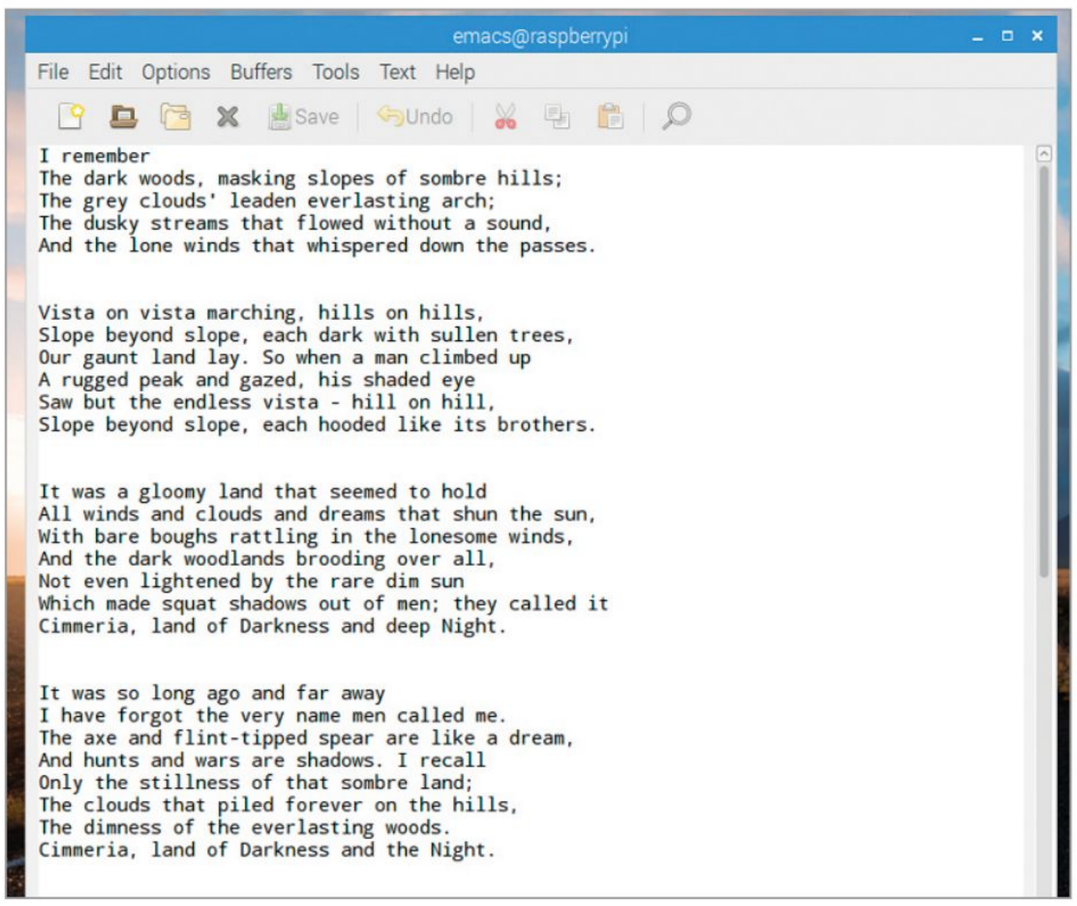
```
sudo apt-get install emacs
```



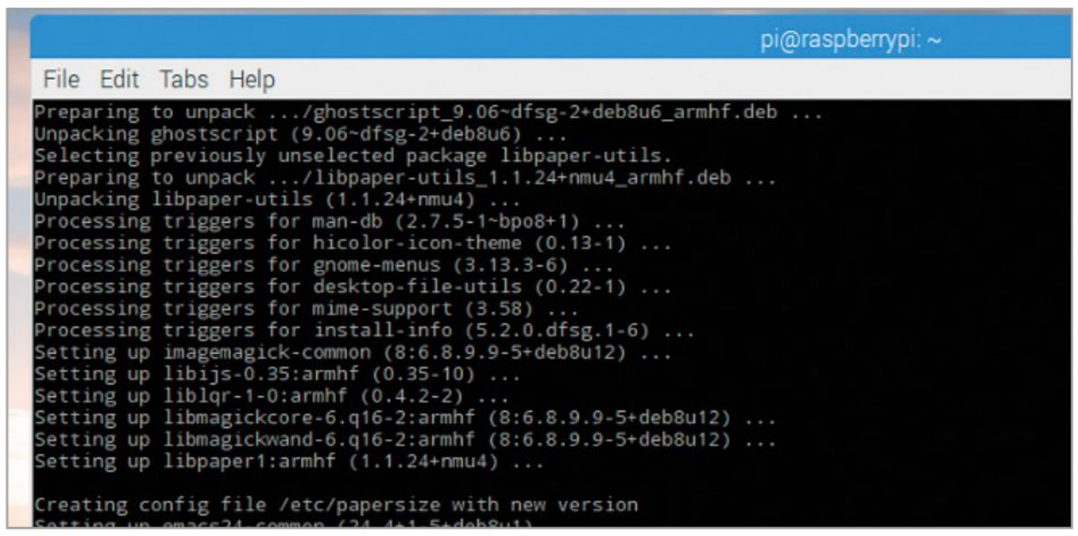
STEP 6 The previous command contacts the Debian (Raspbian is based on a Debian Linux distribution) repositories and pulls down the information needed to install Emacs. When the Pi asks to continue with the installation, press Y. This installs the latest version and when it's done, you'll be back to the command prompt.



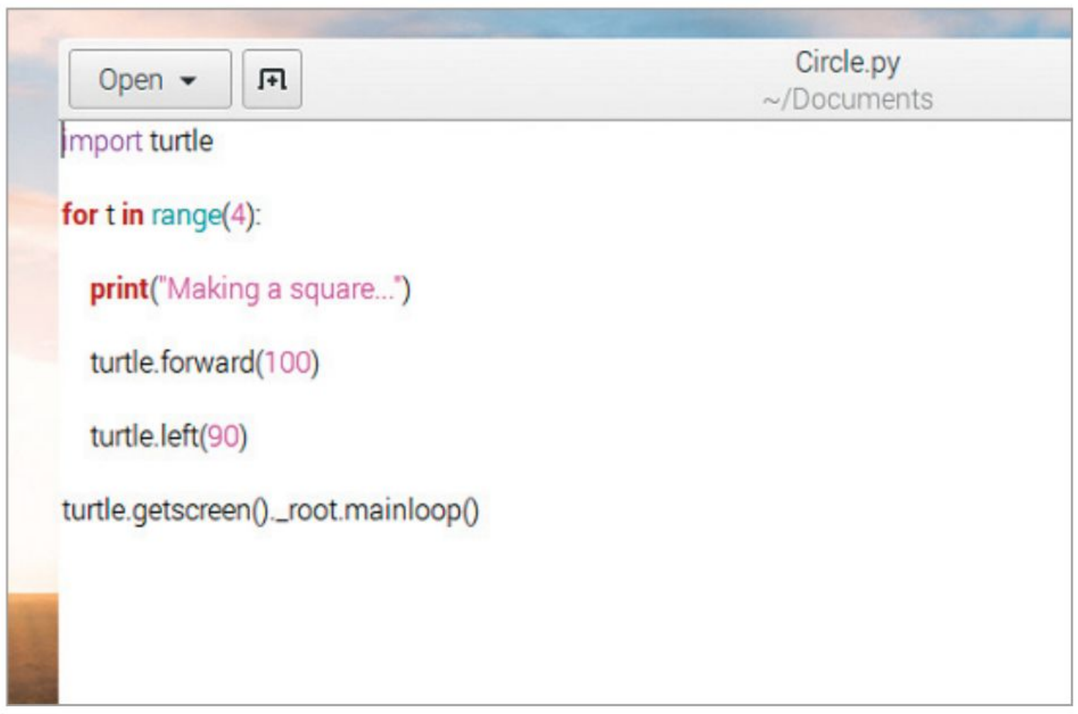
STEP 7 Once the installation is complete, enter: `emacs` into the Terminal. The Emacs splash screen opens in a new window, offering a tutorial (which we recommend you run through) and a guided tour amongst other information.



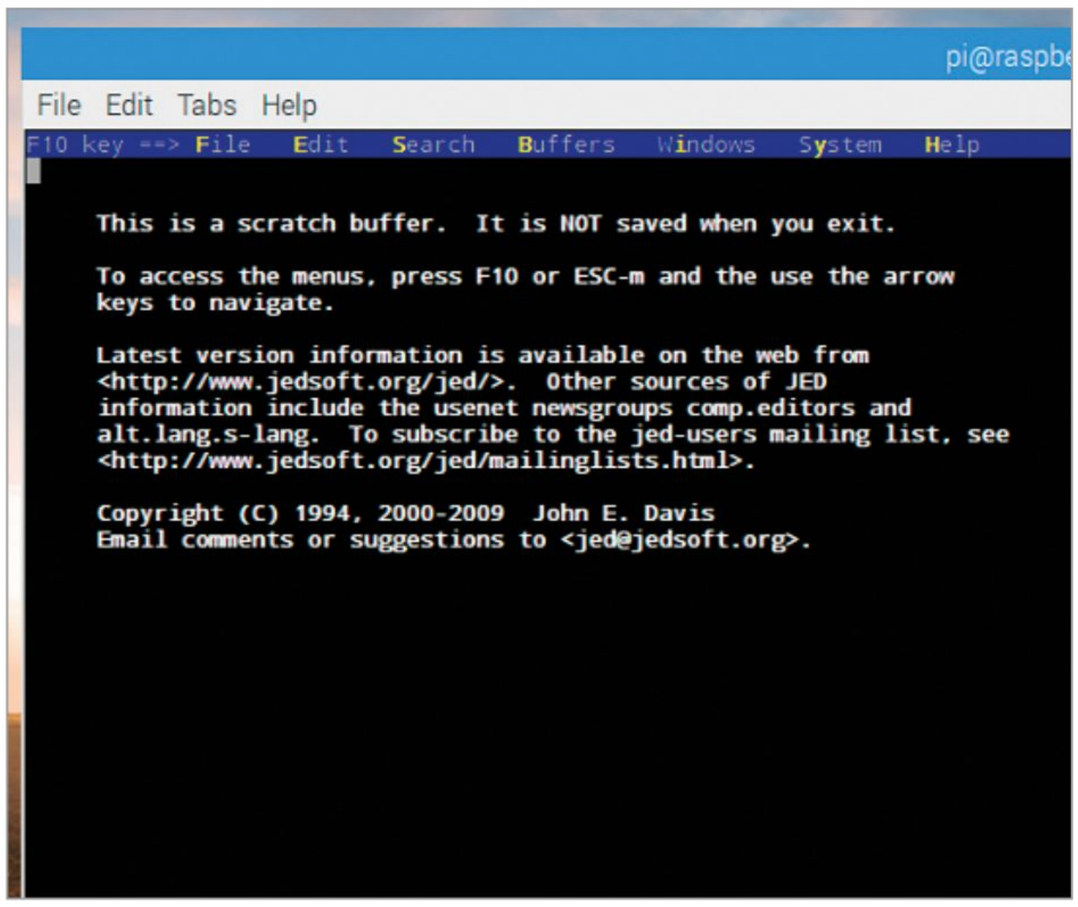
STEP 8 Emacs can offer an uncomplicated view of your text file or one with a plethora of information regarding the structure of the file in question; it's up to you to work out your own preference. There's also a hidden text adventure in Emacs, which we cover later in this book, why not see if you can find it without our help.



STEP 9 Gedit is another excellent text editor for Linux. Again, it's not installed by default on the Raspberry Pi; however, by entering: `sudo apt-get install gedit` and accepting the installation, the program can be on the Pi in a matter of seconds. Once it's installed, use `gedit` in the Terminal to launch it. Gedit is a great text editor for coding.



STEP 10 Finally, Jed is an Emacs-like, cross-platform text editor that's lightweight and comes with a wealth of features. To install it, enter: `sudo apt-get install jed`. Accept the installation and when it's complete, use: `jed` to launch.



Getting to Know Users

You might think you're the only person using your Raspberry Pi but there are several different users and even groups of users. Your main account is normally called Pi and there is an account above it called root, which is more powerful. You can also create users and groups.

WHAT IS A USER?

An important part of using Linux is the concept of users and understanding which user you are and which group you belong to. Like all modern computers, you can have multiple user accounts with each having different levels of access.

STEP 1 The first thing you need to do is get a concept of which user you are. Enter: `whoami` into the command line and press return. It should say "pi" (unless you set up your account name differently during setup). The "whoami" command might seem a bit simplistic, but it comes in very handy sometimes.

```
pi@raspberrypi ~ $ whoami
pi
pi@raspberrypi ~ $ _
```

STEP 2 When you are working in Linux, from time to time a 'Permission denied' error will occur, typically when you try to create, edit or execute (run) a file or directory outside of your area of privilege. If you want to see this, enter: `mkdir /testdir`. Attempting to create a new directory in your root directory isn't allowed.

```
pi@raspberrypi ~ $ mkdir /test
mkdir: cannot create directory `/test': Permissi
pi@raspberrypi ~ $
```

STEP 3 To allow this, you need to use the sudo command. Sudo loosely stands for Substitute User Do; essentially it's the highest level of access to the system and you've already installed text editors using sudo. You'll come across sudo frequently in Linux, so let's create a second account to get the hang of it. Enter: `sudo useradd -m lucy` (or pick your name).

```
pi@raspberrypi ~ $ sudo useradd -m lucy
```

STEP 4 Now add a password for the new account. Enter: `sudo passwd lucy` and enter: a short password. Retype the same password and you'll now have two accounts on your Raspberry Pi. Now enter: `ls -l /home` to view the home directories for both users. Notice that the lucy directory lists lucy as the owner and group; and pi directory is belongs to pi.

```
pi@raspberrypi ~ $ sudo passwd lucy
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi ~ $ ls -l /home
total 8
drwxr-xr-x  2 lucy lucy 4096 May 13 19:01 lu
drwxr-xr-x 33 pi   pi   4096 May 13 18:32 pi
pi@raspberrypi ~ $ _
```

STEP 5 Let's try switching to our new account. Enter: `su lucy` and enter the password you just created for that account. Notice that the command line now says "lucy@raspberrypi" but the working directory is "still /home/pi" (check this using "pwd"). Enter: `whoami` to confirm that you are now the new user.

```
lucy@raspberrypi /home/pi $ su lucy
Password:
lucy@raspberrypi /home/pi $ pwd
/home/pi
lucy@raspberrypi /home/pi $ _
```

STEP 6 We'll look at permissions in the next tutorial, but for now try to create a file as before. Enter: `touch testfile` to create a file. It will say "touch: cannot touch 'testfile': Permission denied". This is because your new user account doesn't have the right to create files in the /home/pi directory. Enter: `su pi` to switch back to your pi account.

```
pi@raspberrypi ~ $ su lucy
Password:
lucy@raspberrypi /home/pi $ touch testfile
touch: cannot touch `testfile': Permission denied
lucy@raspberrypi /home/pi $ _
```

GETTING SUDO

We now have two accounts on our Raspberry Pi: lucy and pi. The lucy account can edit files in /home/lucy and the pi account can edit files in /home/pi. But there's also a third account, called "root", that sits above both lucy and pi. It can edit files anywhere.

STEP 1 The root account is all-powerful. It is possible, but not recommended, to switch to the root account, although you'll need to give it a password first (using "sudo passwd root"). Then just type "su" to switch to root. Please don't do this though: knowledge is a good thing but it's safer and wiser to use sudo instead.

```
pi@raspberrypi ~ $ sudo passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi ~ $ su
Password: _
```

STEP 2 Most people think sudo stands for "super user", but it stands for "substitute user do". It enables you to perform a command as another user. Enter: `sudo -u lucy touch /home/lucy/test` to create a file inside the lucy home directory. You won't get an error because the lucy user has permission to edit that directory.

```
pi@raspberrypi ~ $ sudo -u lucy touch /home/lucy/te
pi@raspberrypi ~ $ _
```

STEP 3 It's rare that you use sudo to substitute another user. If you don't specify a user using the "-"u option with a username it defaults to the root account, as if you'd typed "sudo -u root". Enter: `sudo touch /home/lucy/anothertestfile` to create a file in the lucy directory while still using the pi account.

```
pi@raspberrypi ~ $ sudo touch /home/lucy/anothertest
pi@raspberrypi ~ $ ls /home/lucy/
anothertestfile  pistore.desktop  test
pi@raspberrypi ~ $ _
```

STEP 4 This step is optional. Only the pi user can use sudo. If we want to give the lucy account sudo privileges, it needs to be added to the sudoers file. Enter: `sudo visudo` to view the sudoers file. Add `lucy ALL=(ALL) NOPASSWD: ALL` to the last line and use Control+O to output the file. Remove the ".tmp" that is added to the file name as a security measure. Note that most accounts are not added to the sudoers file as a matter of course.

```
 GNU nano 2.2.6                                  File: /etc/sudoers.tmp

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
pi ALL=(ALL) NOPASSWD: ALL
lucy ALL=(ALL) NOPASSWD: ALL
```

Ownership and Permissions

Once you've got the hang of users, you need to learn about ownership and permissions. Different users have different areas of ownership and can do different things with each file. Permissions in Linux can be quite complex but with careful thought it's not too difficult.

OWNER AND PRIVILEGE

Each user account in Linux is an owner of a section of the filesystem: their Home area. Within this area, they do what they like (within reason), as they have owner privileges. Elsewhere though, they usually just have read-only privileges.

STEP 1

If you followed the previous tutorial you should now have two accounts on your Raspberry Pi. One called "pi" and the other with a name (Lucy in our case). An essential aspect of Linux is the idea of file and directory ownership; who owns, and has access, to what. You need a test file so enter: `touch testfile`.

```
pi@raspberrypi ~ $ touch testfile.txt
```

STEP 2

Now enter: `ls -l` and let's have a good look at the default permissions file. Our testfile.txt files starts with the text "-rw-r--r—". Start with the first letter, which is a dash '-'. All the other items in our home directory are directories. You can tell because they are blue, and our testfile.txt file is white.

```
pi@raspberrypi ~ $ touch testfile.txt
pi@raspberrypi ~ $ ls -l
total 28
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 3 pi pi 4096 May 13 18:08 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxr-xr-x 3 pi pi 4096 May 13 14:53 people
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 May 13 11:05 Scratch
-rw-r--r-- 1 pi pi    0 May 13 21:14 testfile.txt
pi@raspberrypi ~ $
```

STEP 3

The first letter in the permissions also indicates a directory or file. Notice that all the other files start with a 'd' and our textfile.txt file starts with a '-'. That's what the first letter means. It's either a 'd', in which case it's a directory, or a '-', in which case it's not; it's a file. Enter: `ls -l testfile.txt` to view the permissions for just this file.

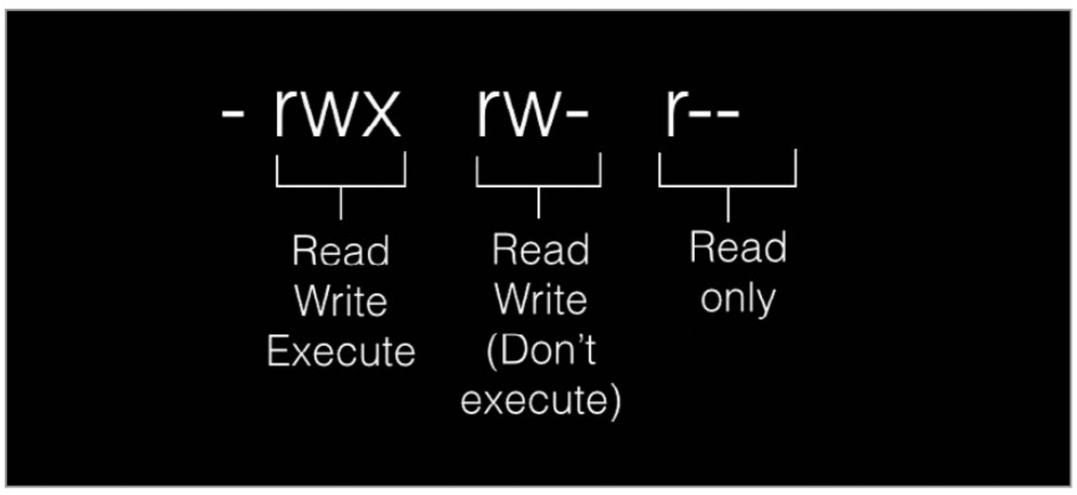
```
pi@raspberrypi ~ $ ls -l
total 28
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 3 pi pi 4096 May 13 18:08 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxr-xr-x 3 pi pi 4096 May 13 14:53 people
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 May 13 11:05 Scratch
-rw-r--r-- 1 pi pi    0 May 13 21:14 testfile.txt
pi@raspberrypi ~ $ ls -l testfile.txt
-rw-r--r-- 1 pi pi 0 May 13 21:14 testfile.txt
pi@raspberrypi ~ $ _
```

STEP 4

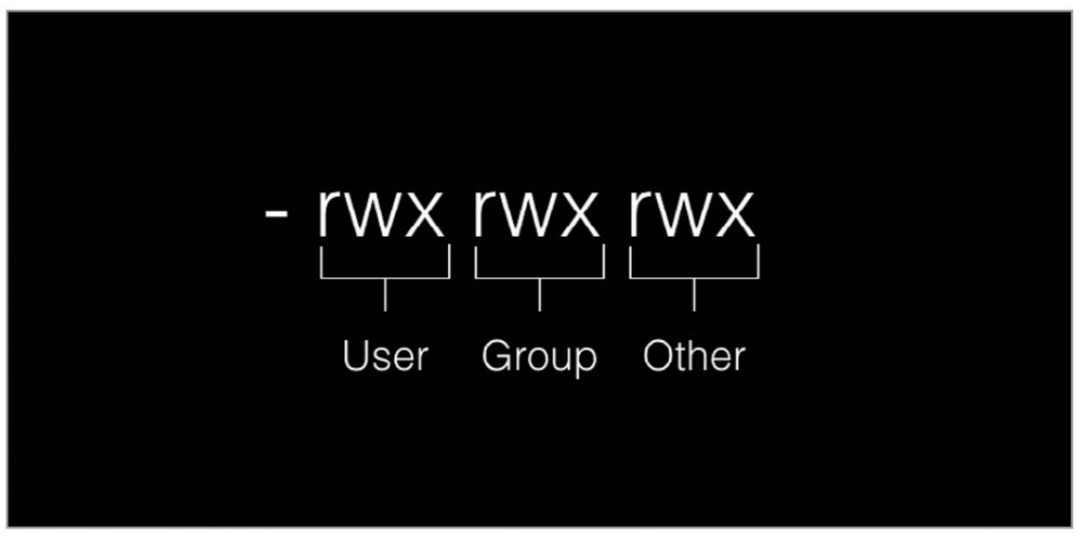
The next nine letters of the permissions are known as "alpha notation" because they let you know the permissions using letters. Each permission is either on, in which case you see a letter, or it is off, in which case you see a dash. The letter doesn't change for each place. So the first permission - the second letter after the directory one - is either an 'r' or a '-'. It's never any other letter.

```
pi@raspberrypi ~ $ ls -l testfile.txt
-rw-r--r-- 1 pi pi 0 May 13 21:14 testfile.txt
pi@raspberrypi ~ $ _
```

STEP 5 The 'r' means that particular permission is read and it's set to On. The nine letters here are divided into three groups of three letters: r, w, x. They stand for read, write and execute (run). Read means the file can be viewed (using cat or nano); w means the file can be edited or moved, and x means the file - typically a script or program - can be run.



STEP 6 The presence of r, w, or x means that this aspect is possible, a dash means it isn't. Our testfile.txt has no x letter; so if it were a script it wouldn't run. So why are there so many letters? Why not just three; read, write and execute? The three blocks of three letters are for different sets of people: user, group and other.



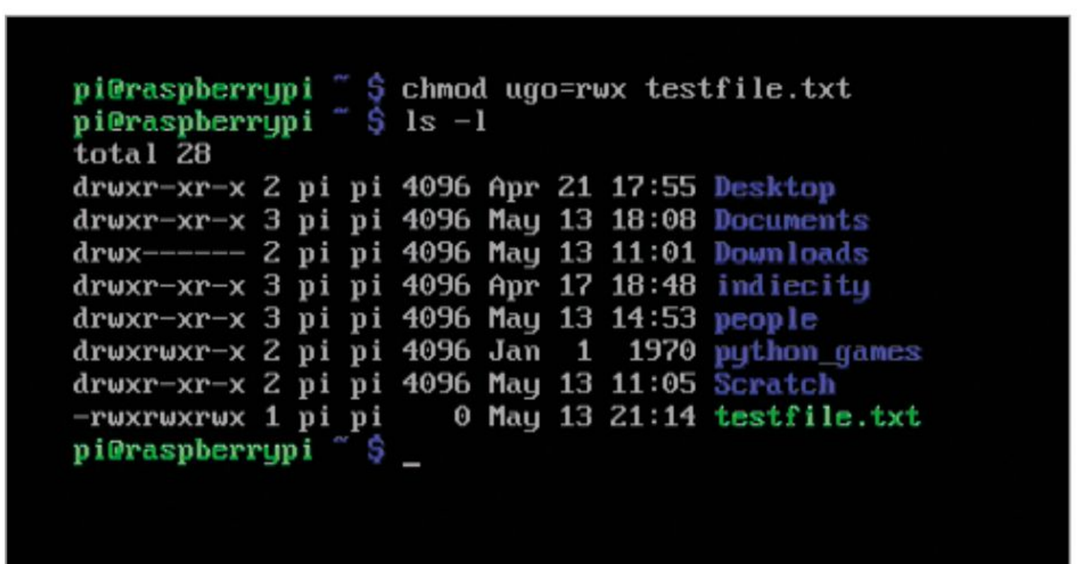
CHANGING PERMISSIONS

Now that you know how groups of permissions work, it's time to look at how to change them.

STEP 1 The first block of three is the most important. This is the user who owns the file (typically pi); the second is for other people in the same group as the user, and the third is for other people on the system. Permissions are changed using the chmod (change file mode bit) command. Enter: `man chmod` to look at the manual.

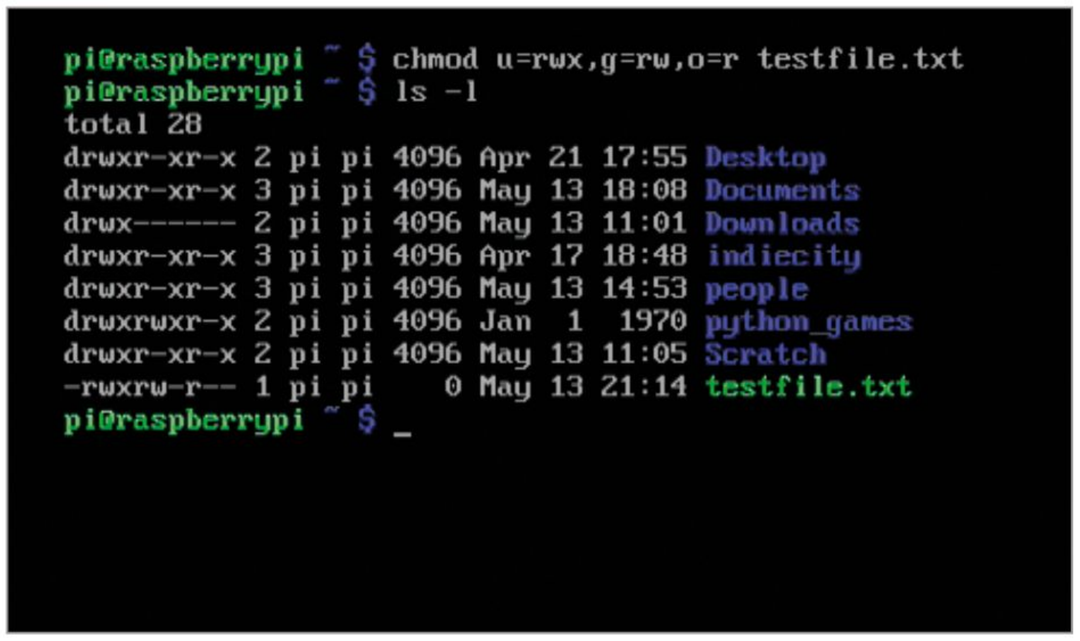
```
CHMOD(1)

NAME
       chmod - change file mode bits

SYNOPSIS
       chmod [OPTION]... MODE[,MODE]... FILE...
       chmod [OPTION]... OCTAL-MODE FILE...
       chmod [OPTION]... --reference=RFILE FILE...

DESCRIPTION
       This manual page documents the GNU version of chmod.  chmod changes the file mode
       representing the bit pattern for the new mode bits.

       The format of a symbolic mode is [ugoa...][[+-=][perms...]...], where perms is ei
       rated by commas.

       A  combination of the letters ugoa controls which users' access to the file will
       (a).  If none of these are given, the effect is as if a were given, but bits that

       The operator + causes the selected file mode bits to be added to the existing fil
       except that a directory's unmentioned set user and group ID bits are not affected

       The  letters  rwxXst select file mode bits for the affected users: read (r), writ
       for some user (X), set user or group ID on execution (s), restricted deletion fla
       granted  to  the  user who owns the file (u), the permissions granted to other us
       gories (o).

       A numeric mode is from one to four octal digits (0-7), derived by adding up the b
       set  group ID (2) and restricted deletion or sticky (1) attributes.  The second d
       other users in the file's group, with the same values; and the fourth for other u
```

STEP 2 The chmod command is one of the trickier ones to understand. There are two ways you can adjust permissions; the first is using chmod with an option to target one of the three groups: owner, group, other. For these you use u, g or o followed by = and the letters or dashes you want. So enter: `chmod ugo=rx testfile.txt` to make all three groups read, write and execute.



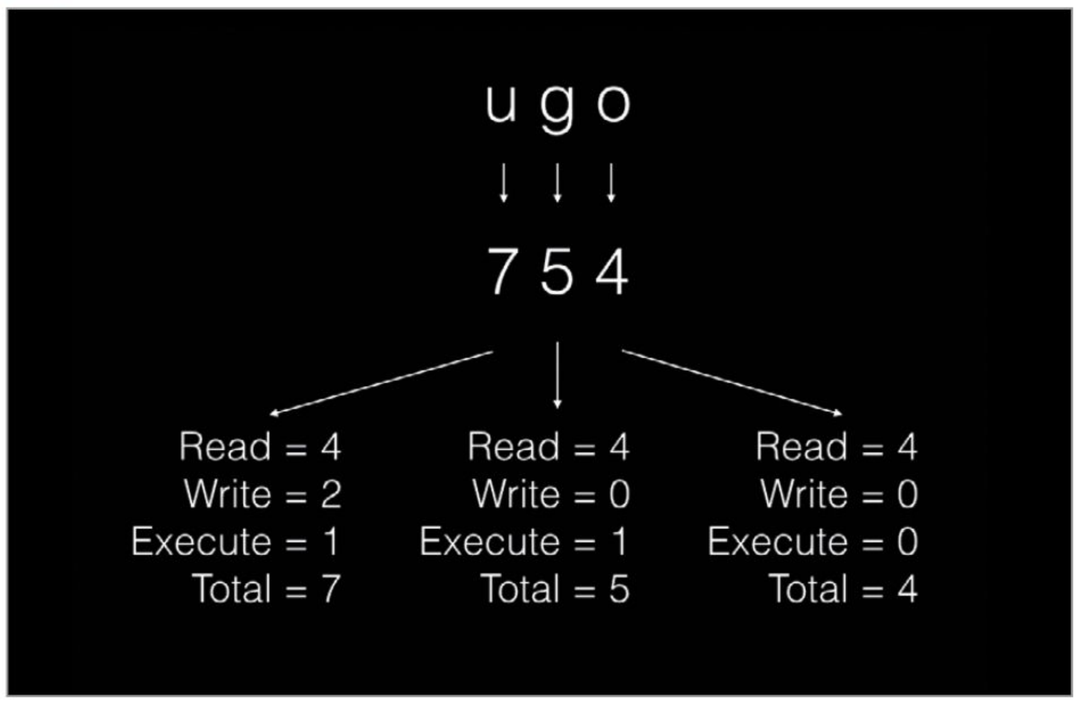
```
pi@raspberrypi ~ $ chmod ugo=rwx testfile.txt
pi@raspberrypi ~ $ ls -l
total 28
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 3 pi pi 4096 May 13 18:08 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxr-xr-x 3 pi pi 4096 May 13 14:53 people
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 May 13 11:05 Scratch
-rwxrwxrwx 1 pi pi    0 May 13 21:14 testfile.txt
pi@raspberrypi ~ $ _
```

STEP 3 Turning everything on is probably overkill, so you need to target each group. Do this by putting commas between each mode option. Enter: `chmod u=rwx,g=rw,o=r testfile.txt` to give users read, write and execute privileges, user read and write and other just read. Enter: `ls -l` to see your handiwork.



```
pi@raspberrypi ~ $ chmod u=rwx,g=rw,o=r testfile.txt
pi@raspberrypi ~ $ ls -l
total 28
drwxr-xr-x 2 pi pi 4096 Apr 21 17:55 Desktop
drwxr-xr-x 3 pi pi 4096 May 13 18:08 Documents
drwx------ 2 pi pi 4096 May 13 11:01 Downloads
drwxr-xr-x 3 pi pi 4096 Apr 17 18:48 indiecity
drwxr-xr-x 3 pi pi 4096 May 13 14:53 people
drwxrwxr-x 2 pi pi 4096 Jan  1  1970 python_games
drwxr-xr-x 2 pi pi 4096 May 13 11:05 Scratch
-rwxrw-r-- 1 pi pi    0 May 13 21:14 testfile.txt
pi@raspberrypi ~ $ _
```

STEP 4 Alpha notation is fine, but many Linux admins use octal notation instead. This is a three-digit number that represents permissions. This is the formula: read=4, write=2 and execute=1 and you add them up for each group, therefore if a group is read, write and execute it's 7, if it's read and write it's 6 or if it's just execute it's 1. A popular option is 755. Enter: `chmod 755 testfile.txt` to change the file using octal notation.



Useful System and Disk Commands

Understanding these core Linux commands will enable you to not only master the inner workings of your Raspberry Pi but also to transfer those skills to other Linux distros, such as Ubuntu or Linux Mint.

LOTS OF LINUX

Linux is a huge and versatile command line language and there are hundreds of commands you can learn and use. Here are a few that can help you get more from your Raspberry Pi.

STEP 1

The Raspberry Pi is a great little computer, so let's start by getting some information. Enter: `cat /proc/cpuinfo` to view some details on your Raspberry Pi processors. If you have a Raspberry Pi 3 you will see four processors, along with the model name and other info.

```
pi@raspberrypi ~ $ cat /proc/cpuinfo
processor       : 0
model name      : ARMv7 Processor rev 5 (v7l)
BogoMIPS        : 38.40
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpa
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 5

processor       : 1
model name      : ARMv7 Processor rev 5 (v7l)
BogoMIPS        : 38.40
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpa
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 5

processor       : 2
model name      : ARMv7 Processor rev 5 (v7l)
BogoMIPS        : 38.40
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpa
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 5

processor       : 3
model name      : ARMv7 Processor rev 5 (v7l)
BogoMIPS        : 38.40
```

STEP 2

Remember that cat is used to list the contents of a text file, which is what cpuinfo is. There are other text files with system info available. Try "cat /proc/meminfo" to get information about your memory, "cat /proc/partitions" for information about your SD card, and "cat /proc/version" shows which version of Raspberry Pi you are using.

```
pi@raspberrypi ~ $ cat /proc/meminfo
MemTotal:         884304 kB
MemFree:          784576 kB
MemAvailable:     834088 kB
Buffers:           11660 kB
Cached:            56340 kB
SwapCached:            0 kB
Active:            49148 kB
Inactive:          30072 kB
Active(anon):      11256 kB
Inactive(anon):      228 kB
Active(file):      37892 kB
Inactive(file):    29844 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:        102396 kB
SwapFree:         102396 kB
Dirty:                36 kB
Writeback:             0 kB
AnonPages:         11132 kB
Mapped:             7464 kB
Shmem:               260 kB
Slab:              10384 kB
SReclaimable:       4512 kB
SUnreclaim:         5872 kB
KernelStack:         672 kB
PageTables:          564 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:      544548 kB
Committed_AS:      43712 kB
VmallocTotal:    1171456 kB
VmallocUsed:        3880 kB
VmallocChunk:     922360 kB
pi@raspberrypi ~ $ cat /proc/partitions
major minor  #blocks  name

 179        0    7761920 mmcblk0
 179        1     835890 mmcblk0p1
 179        2          1 mmcblk0p2
 179        3      32768 mmcblk0p3
 179        5      61440 mmcblk0p5
 179        6    6819840 mmcblk0p6
```

STEP 3

Enter: `uname` to view the name of the operating system's kernel, this is the element that sits between the interface and hardware. Just as you would suspect, the response from the command is Linux, as Raspbian is a Linux distro, which in itself is based on another Linux distro called Debian. While it may sound complicated, it actually demonstrates how versatile Linux is.

```
pi@raspberrypi ~ $ uname
Linux
pi@raspberrypi ~ $
```

STEP 4

Enter: `uname -a` to view some more detailed information. Here you'll see the kernel name, hostname and kernel version (3.18.7-v7 on ours). If you have a Raspberry Pi 2 you'll see SMP (symmetric multiprocessing), followed by the system date, CPU architecture and operating system (GNU/Linux).

```
pi@raspberrypi ~ $ uname
Linux
pi@raspberrypi ~ $ uname -a
Linux raspberrypi 3.18.7-v7+ #755 SMP PREEMPT Thu Feb 12 1
pi@raspberrypi ~ $ _
```

STEP 5 Enter: `vcgencmd measure_temp` to view the current operating system temperature of your Raspberry Pi. Enter: `vcgencmd get_mem arm` to view the RAM available, and `vcgencmd get_mem gpu` to view the memory available to the graphics chip. Finally try `ls usb` to view a list of attached USB devices.

```
pi@raspberrypi ~ $ vcgencmd measure_temp
temp=36.9'C
pi@raspberrypi ~ $ vcgencmd get_mem arm
arm=880M
pi@raspberrypi ~ $ vcgencmd get_mem gpu
gpu=128M
pi@raspberrypi ~ $ lsusb
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 04d9:1503 Holtek Semiconductor, Inc. Shortboard I
Bus 001 Device 005: ID 1a40:0101 Terminus Technology Inc. 4-Port HUB
Bus 001 Device 006: ID 276d:1105
pi@raspberrypi ~ $
```

STEP 6 One command you might be wondering about is how to switch off or restart your Raspberry Pi from the command line. Don't just hit the power switch. Enter: `sudo showdown -h` now to shut down the Raspberry Pi (the "-h" option stands for "halt"), or enter: `sudo shutdown -r` now to restart your Raspberry Pi.

```
pi@raspberrypi ~ $ sudo shutdown -r now

Broadcast message from root@raspberrypi (tty1) (Thu May 14 12:20:29 2015):
The system is going down for reboot NOW!
_
```

DISK COMMANDS

Learn the two commands that enable you to view your disk space and the files on it: df (disk free space) and du (disk usage). With these two commands you can view the file usage on your SD card.

STEP 1 Start by entering: `df` in the command line. It returns a list of the volumes contained on your SD card. You might be wondering what a volume is. It's best to think of your SD card as the drive. This contains partitions, which is where you split one drive to act like two or more drives. And each partition can contain volumes, which are storage spaces.

```
pi@raspberrypi ~ $ df
Filesystem     1K-blocks    Used Available Use% Mounted on
rootfs           6581636 3484164   2740096  56% /
/dev/root        6581636 3484164   2740096  56% /
devtmpfs          437856       0    437856   0% /dev
tmpfs              88432     260     88172   1% /run
tmpfs               5120       0      5120   0% /run/lock
tmpfs             176860       0    176860   0% /run/shm
/dev/mmcblk0p5     60479   14536     45943  25% /boot
pi@raspberrypi ~ $
```

STEP 2 Enter: `df -h` to get the list in human readable form. The first two lines should read "rootfs" and "/dev/root" and have matching Size, Used, Avail and Use% listings. This is the main drive, and is an indication of how much space you have used, and have free, on your Raspbian OS. The other volumes are for booting and initialising devices (you can ignore these for now).

```
pi@raspberrypi ~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
rootfs          6.3G  3.4G  2.7G  56% /
/dev/root       6.3G  3.4G  2.7G  56% /
devtmpfs        428M     0  428M   0% /dev
tmpfs            87M  260K   87M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           173M     0  173M   0% /run/shm
/dev/mmcblk0p5   60M   15M   45M  25% /boot
pi@raspberrypi ~ $
```

STEP 3 Now enter: `du`. You should see lots of text fly up the screen. This is the disk usage for the files contained in your home directory and their sub-directories. As with df, it is better to use du with the "-h" option to humanise the output. If you want to slow down the output, you'll also need to pipe it through less. Enter: `df -h | less` to view the files and their respective usage one page at a time.

```
pi@raspberrypi ~ $ du -h | less
22M     ./.minecraft/games/com.mojang/minecraftWorlds/world
22M     ./.minecraft/games/com.mojang/minecraftWorlds
22M     ./.minecraft/games/com.mojang
22M     ./.minecraft/games
4.0K    ./.pulse
16K     ./.config/gedit
8.0K    ./.config/libfm
1.4M    ./.config/epiphany/adblock
1.5M    ./.config/epiphany
8.0K    ./.config/lxsession/LXDE-pi
12K     ./.config/lxsession
8.0K    ./.config/dconf
8.0K    ./.config/rncbc.org
8.0K    ./.config/lxterminal
8.0K    ./.config/uk.ac.cam.cl
8.0K    ./.config/IndieCity
4.0K    ./.config/enchant
8.0K    ./.config/lxpanel/LXDE-pi/panels
16K     ./.config/lxpanel/LXDE-pi
24K     ./.config/lxpanel
28K     ./.config/openbox
```

STEP 4 You don't typically enter: `du` on its own; most of the time you want to view the disk usage of a specific directory. Enter: `du -h python_games` to view how much space the python_games directory (installed alongside Raspbian) takes up. It should be 1.8M. If you want a more comprehensive breakdown of the files contained, use the "-a" option (all). Enter: `du -ha python_games` to view all the files contained and their disk usage.

```
pi@raspberrypi ~ $ ls
Desktop  Documents  Downloads  indiecity  people  python_games  Scratch  test
pi@raspberrypi ~ $ du -h python_games
1.8M    python_games
pi@raspberrypi ~ $ du -ha python_games
12K     python_games/RedSelector.png
12K     python_games/4row_board.png
12K     python_games/Star.png
28K     python_games/4row_humanwinner.png
12K     python_games/Wall_Block_Tall.png
8.0K    python_games/princess.png
12K     python_games/Selector.png
8.0K    python_games/4row_black.png
4.0K    python_games/catanimation.py
20K     python_games/flippy.py
36K     python_games/match3.wav
24K     python_games/starpusher.py
4.0K    python_games/grass1.png
40K     python_games/beep2.ogg
12K     python_games/slidepuzzle.py
8.0K    python_games/gem5.png
16K     python_games/fourinarow.py
8.0K    python_games/gem2.png
336K    python_games/flippyboard.png
12K     python_games/Grass_Block.png
```

Managing Programs and Processes

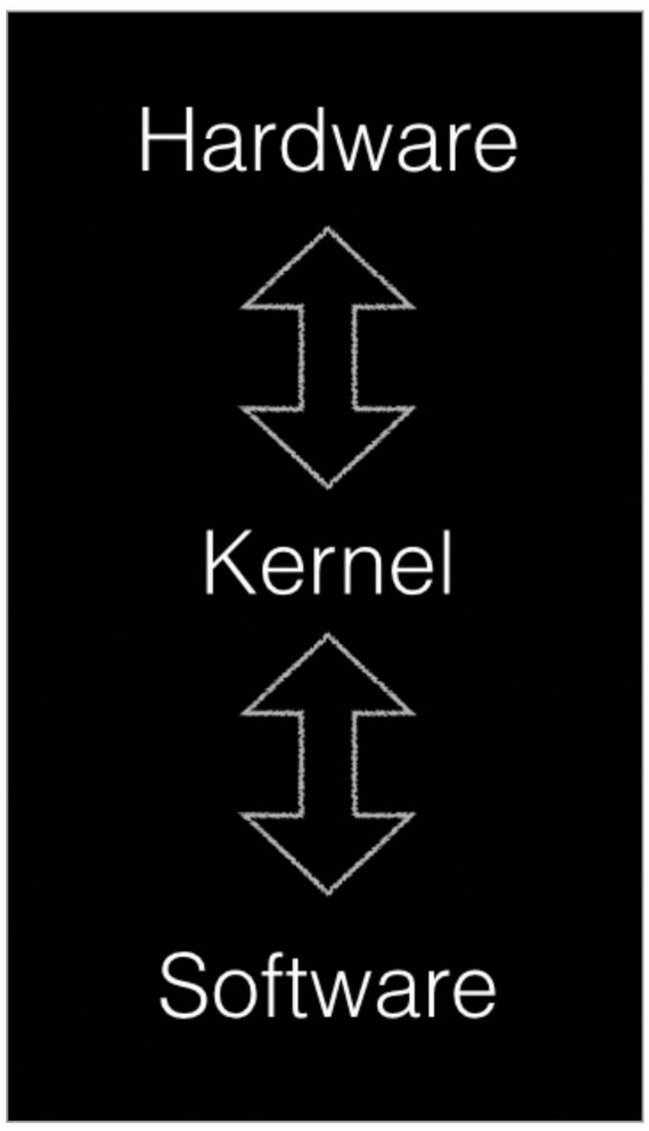
Being able to effectively manage the active programs and processes on your Raspberry Pi allows you to change the way the systems work. If you have a project requiring more memory, you can kill a process to free up the available system resources.

PROGRAMS AND PROCESSES

Linux has a trick up its sleeve when it comes to being able to manage programs and processes. When Windows closes a program, the allocated memory often isn't freed up again. However, Linux is far more streamlined.

STEP 1

As you get into Linux you'll start to hear more about processes and another thing called the "kernel". The kernel sits beneath the software and hardware. It passes instructions to the hardware running processes, which takes up memory, and when the process is finished it closes it and reclaims the memory.



STEP 2

You're probably used to thinking in terms of programs and most OS's tend to keep processes out of sight. In Linux on the other hand, you're right in at the deep end. A process is like a program, only it's a single task running on your computer. Programs as you know them, may be a single process, or multiple processes working together. In Linux, you should learn to manage processes. This is done using the "ps" (process status) command.

```
pi@raspberrypi ~ $ ps_
```

STEP 3

If you type in "ps" on its own you don't see much. You should see two items: bash and ps. Bash (Bourne Again Shell) is the command line environment you are typing in, and ps is the command you just entered. If that seems a little light, that's because it is. These are just the processes owned by the user and are running in the foreground.

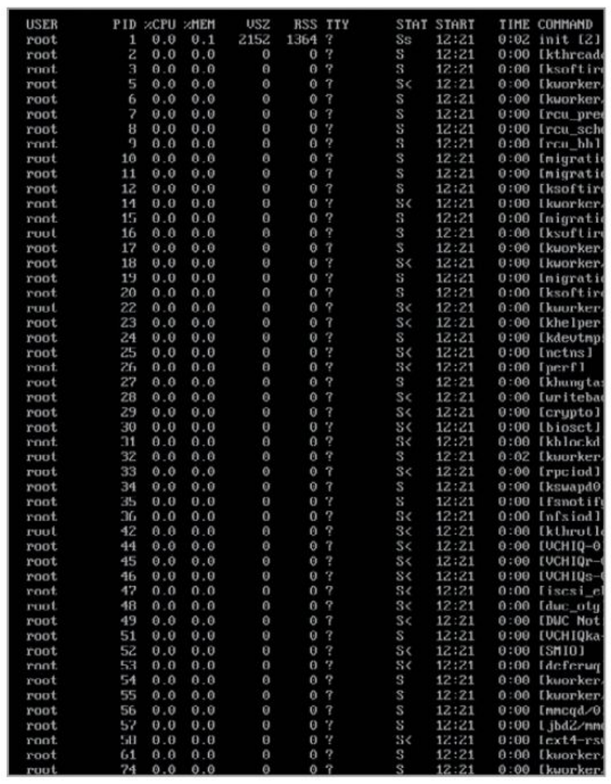
```
pi@raspberrypi ~ $ ps
  PID TTY          TIME CMD
 2165 tty1     00:00:01 bash
 2376 tty1     00:00:00 ps
pi@raspberrypi ~ $ _
```

STEP 4

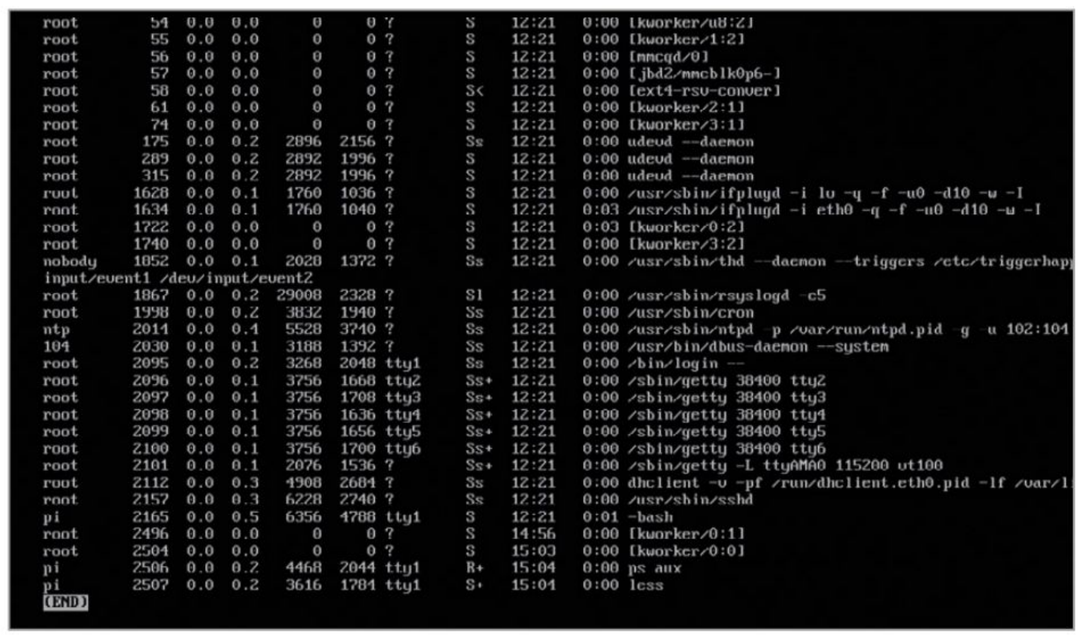
If you want to see processes used by other users (including those started by root) enter: `ps -a`. The option stands for all users. This still isn't everything though because it doesn't include background processes. For this you can enter either "ps –A" or "ps –e". This will show you every process on the system including the background processes. You may need to pipe it through less using "ps -e | less".

```
  PID TTY          TIME CMD
    1 ?        00:00:02 init
    2 ?        00:00:00 kthreadd
    3 ?        00:00:00 ksoftirqd/0
    5 ?        00:00:00 kworker/0:0H
    6 ?        00:00:00 kworker/u8:0
    7 ?        00:00:00 rcu_preempt
    8 ?        00:00:00 rcu_sched
    9 ?        00:00:00 rcu_bh
   10 ?        00:00:00 migration/0
   11 ?        00:00:00 migration/1
   12 ?        00:00:00 ksoftirqd/1
   14 ?        00:00:00 kworker/1:0H
   15 ?        00:00:00 migration/2
   16 ?        00:00:00 ksoftirqd/2
   17 ?        00:00:00 kworker/2:0
   18 ?        00:00:00 kworker/2:0H
   19 ?        00:00:00 migration/3
   20 ?        00:00:00 ksoftirqd/3
   22 ?        00:00:00 kworker/3:0H
   23 ?        00:00:00 khelper
   24 ?        00:00:00 kdevtmpfs
   25 ?        00:00:00 netns
   26 ?        00:00:00 perf
   27 ?        00:00:00 khungtaskd
   28 ?        00:00:00 writeback
   29 ?        00:00:00 crypto
   30 ?        00:00:00 bioset
   31 ?        00:00:00 kblockd
   32 ?        00:00:02 kworker/1:1
   33 ?        00:00:00 rpciod
   34 ?        00:00:00 kswapd0
   35 ?        00:00:00 fsnotify_mark
   36 ?        00:00:00 nfsiod
   42 ?        00:00:00 kthrotld
   44 ?        00:00:00 VCHIQ-0
```

STEP 5 The ps command is one of the oldest known, and it has a lot of legacy. Because of this, it also has alternative options that can be used without the dash at the start. The most common way to use ps is to enter: `ps aux`. Piping this command through less is the best way to go about using ps. Enter: `ps aux | less` to view your processes.



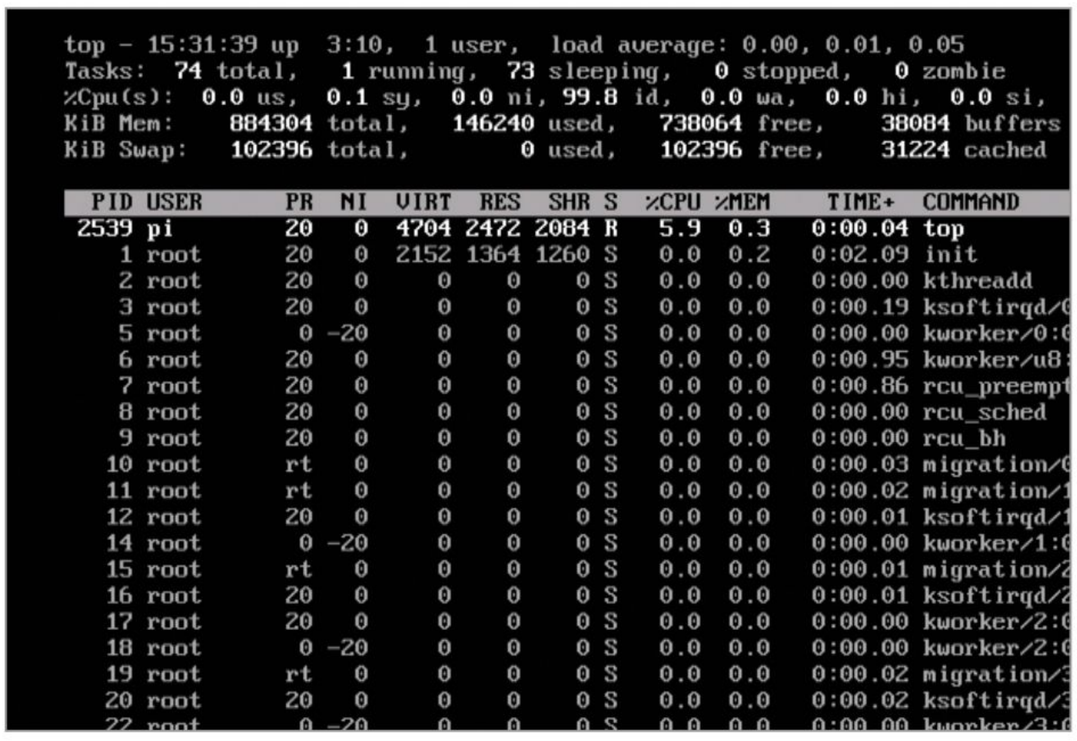
STEP 6 The "ps aux" command displays the processes in columns. The first is the User followed by the Process ID (PID), then we see the percentage of CPU and memory.



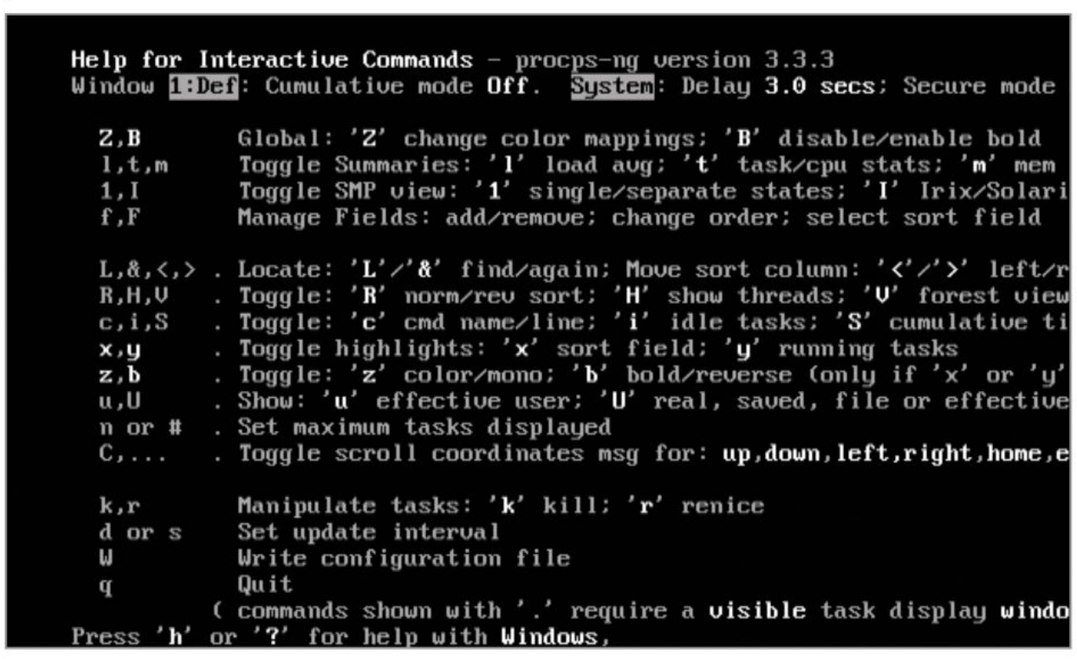
VIEWING AND QUITTING PROCESSES

Now that you've got a handle on processes, you need to know what to do with them. You can view processes running in real-time, and quit ones that you no longer need, or those that are problematic.

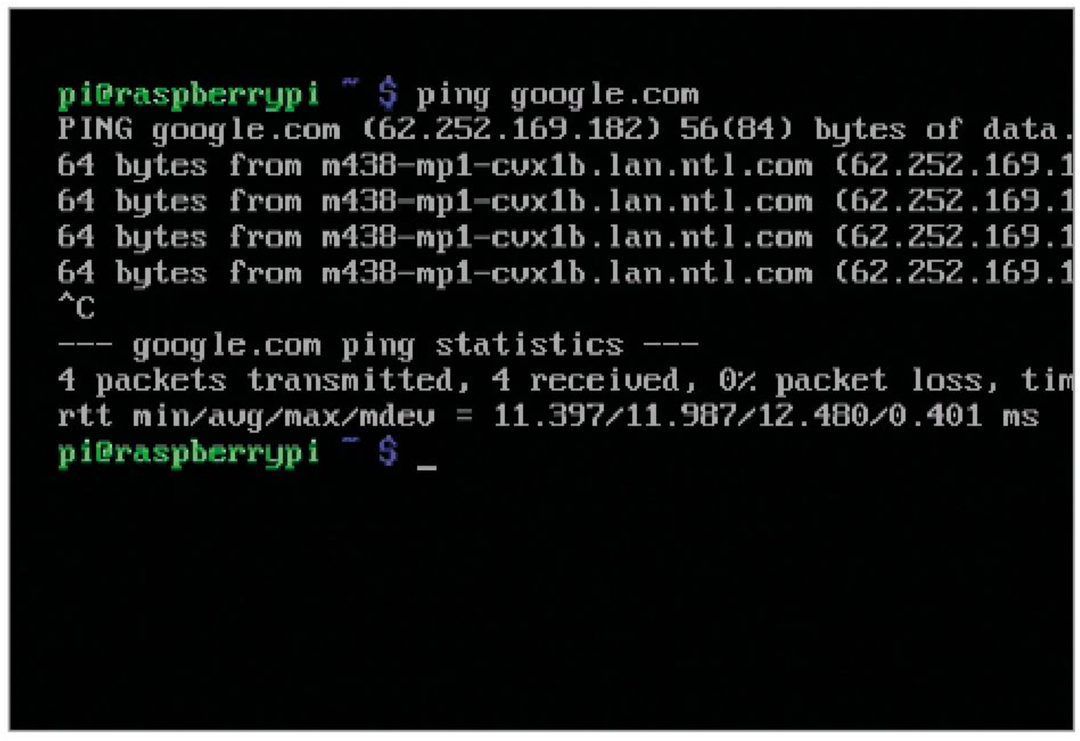
STEP 1 While ps is a great tool for checking all the processes on your Raspberry Pi, sometimes you need to view them in real-time. In particular you may want to check the CPU and memory usage of processes and see if any process is hogging all your computer's resources. Use "top" to do this.



STEP 2 Top fills the display with processes and it fits as many as it can on one screen. These run in real-time so as you watch, the display items will move up and down according to their usage. You can interact with top as it is running: use < and > to change the sort column. Press H to view a help page with all the commands you can use.



STEP 3 Now that you've got an idea of what processes are, how do you go about stopping them? Some processes run in the foreground until you quit them. Enter: `ping www.google.com` and you'll start a ping process continuously contacting Google. This sort of process runs until you press Control+C.



STEP 4 Processes are stopped using the kill command. Enter: `sleep 100 &` to create a dummy process. Enter: `pgrep sleep` to find its PID number (on ours it is 2192). Now enter: `kill 2192` and the sleep process will be stopped. You can also use `pkill sleep` if you're sure you know the process name.

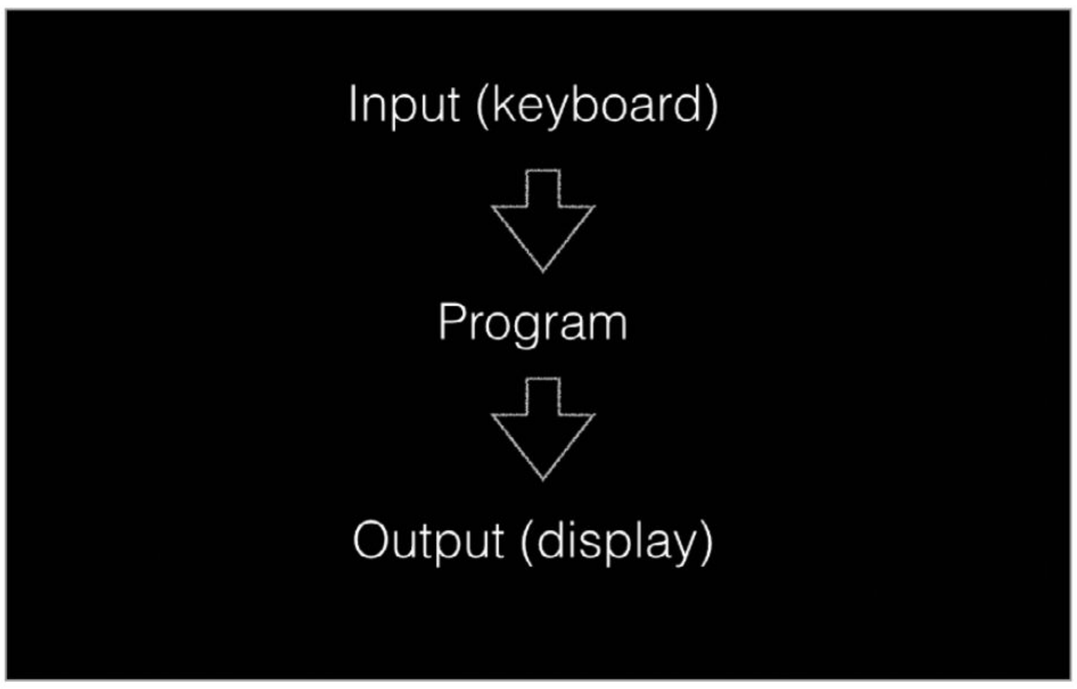
Input, Output and Pipes

Most operating systems allow you to direct the output of something on the screen to, for example, a file. Linux, with its ancestral history of Unix-like command-based arguments, goes a step further and offers much more control.

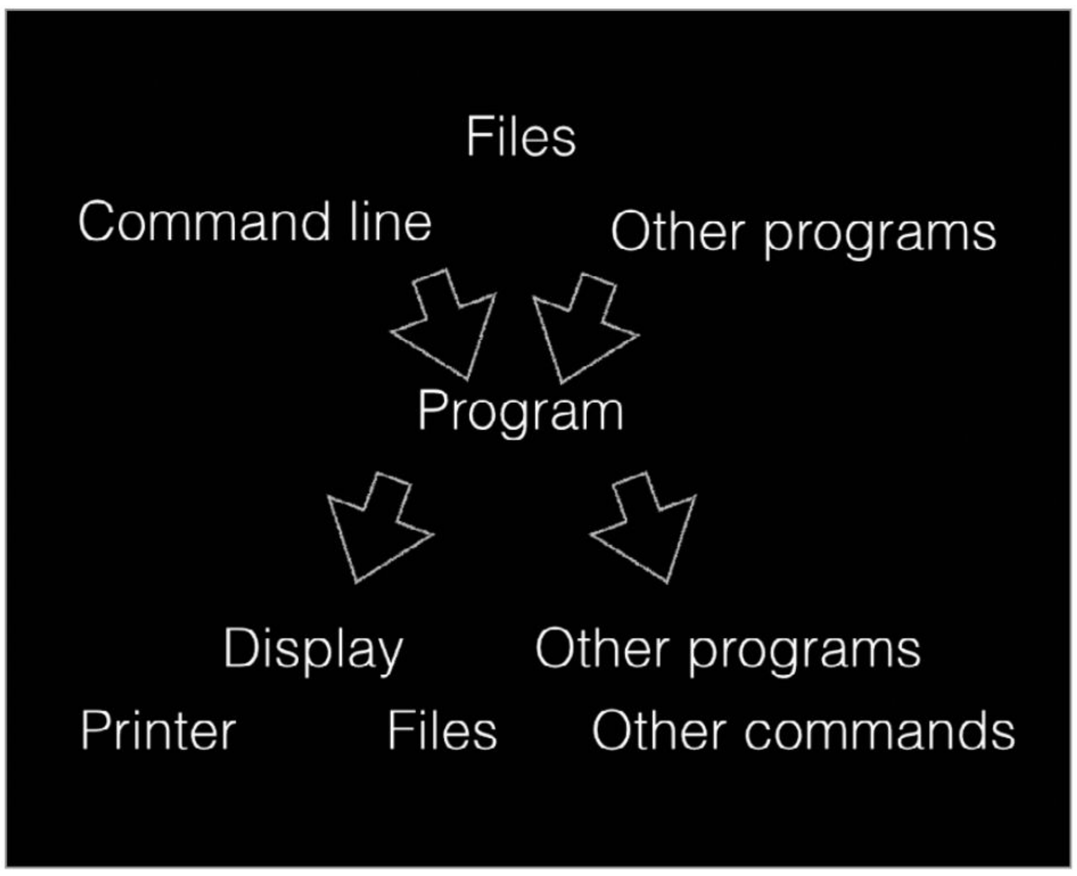
IN, OUT, LINUX ALL ABOUT

Everything on a computer is about input and output. You input things into the computer (press keys, move the mouse) and it makes calculations and outputs content (changes the display, makes a noise, for example).

STEP 1 When you enter commands into Linux (in the command line), you are using standard input and output. This is so obvious that most people don't even think about it. You enter commands using the keyboard (that's input) and it responds to the screen (output). This is the regular way of doing it, which is why it's "standard input and output" (often called "stdin" and "stdout" for short).



STEP 2 As far as the computer is concerned, input and output can be to and from a whole range of different sources that you might never even think about. A program can get input from other programs, from files stored on the hard drive and a whole host of other areas. It outputs back to the display line, but also to files, other programs and even other commands.



STEP 3 You can change the standard output to something else using the ">" character after your command. If we wanted to see all the items in the python_games directory we could use "ls -l python_games". Using "ls -l python_games > games.txt" outputs the list of items to a new text file called "games.txt".

```
pi@raspberrypi ~ $ ls -l python_games > games.txt
```

STEP 4 The games.txt file now contains the output from the ls command. You can check it using "nano games.txt". It's an editable text file containing all the permissions, user and file size information and the names of the files. The output from the ls -l command, normally displayed on the screen, was instead sent to this file. Press Control+X to quit nano.

```
 GNU nano 2.2.6

total 1800
-rw-rw-r-- 1 pi pi   9731 Jan 27 08:34 4row_arrow.png
-rw-rw-r-- 1 pi pi   7463 Jan 27 08:34 4row_black.png
-rw-rw-r-- 1 pi pi   8666 Jan 27 08:34 4row_board.png
-rw-rw-r-- 1 pi pi  18933 Jan 27 08:34 4row_computerwinner.png
-rw-rw-r-- 1 pi pi  25412 Jan 27 08:34 4row_humanwinner.png
-rw-rw-r-- 1 pi pi   8562 Jan 27 08:34 4row_red.png
-rw-rw-r-- 1 pi pi   8912 Jan 27 08:34 4row_tie.png
-rw-rw-r-- 1 pi pi  36908 Jan 27 08:34 badswap.wav
-rw-rw-r-- 1 pi pi  39782 Jan 27 08:34 beep1.ogg
-rw-rw-r-- 1 pi pi  39284 Jan 27 08:34 beep2.ogg
-rw-rw-r-- 1 pi pi  38581 Jan 27 08:34 beep3.ogg
-rw-rw-r-- 1 pi pi  39214 Jan 27 08:34 beep4.ogg
-rw-rw-r-- 1 pi pi    308 Jan 27 08:34 blankpygame.py
-rw-rw-r-- 1 pi pi   6581 Jan 27 08:34 boy.png
-rw-rw-r-- 1 pi pi   1034 Jan 27 08:34 catanimation.py
-rw-rw-r-- 1 pi pi   7270 Jan 27 08:34 catgirl.png
-rw-rw-r-- 1 pi pi  12574 Jan 27 08:34 cat.png
-rw-rw-r-- 1 pi pi   1178 Jan 27 08:34 drawing.py
-rw-rw-r-- 1 pi pi  83036 Jan 27 08:34 flippybackground.png
-rw-rw-r-- 1 pi pi 340760 Jan 27 08:34 flippyboard.png
-rw-rw-r-- 1 pi pi  19479 Jan 27 08:34 flippy.py
-rw-rw-r-- 1 pi pi  13080 Jan 27 08:34 fourinarow.py
-rw-rw-r-- 1 pi pi   2134 Jan 27 08:34 gameicon.png
-rw-rw-r-- 1 pi pi   4382 Jan 27 08:34 gem1.png
-rw-rw-r-- 1 pi pi   5665 Jan 27 08:34 gem2.png
-rw-rw-r-- 1 pi pi   3454 Jan 27 08:34 gem3.png
-rw-rw-r-- 1 pi pi   4217 Jan 27 08:34 gem4.png
-rw-rw-r-- 1 pi pi   5507 Jan 27 08:34 gem5.png
-rw-rw-r-- 1 pi pi   1681 Jan 27 08:34 gem6.png
-rw-rw-r-- 1 pi pi   3132 Jan 27 08:34 gem7.png
-rw-rw-r-- 1 pi pi  22489 Jan 27 08:34 gemgem.py
-rw-rw-r-- 1 pi pi   3009 Jan 27 08:34 grass1.png
-rw-rw-r-- 1 pi pi   3019 Jan 27 08:34 grass2.png
-rw-rw-r-- 1 pi pi   3009 Jan 27 08:34 grass3.png
-rw-rw-r-- 1 pi pi   3032 Jan 27 08:34 grass4.png
-rw-rw-r-- 1 pi pi   8244 Jan 27 08:34 Grass_Block.png
-rw-rw-r-- 1 pi pi   7186 Jan 27 08:34 horngirl.png
-rw-rw-r-- 1 pi pi  18855 Jan 27 08:34 inkspilllogo.png
-rw-rw-r-- 1 pi pi  18739 Jan 27 08:34 inkspill.py
-rw-rw-r-- 1 pi pi   7855 Jan 27 08:34 inkspillresetbutton.png
```

STEP 5 So > enables you to output to files, but you can also get input from a file. Make a new directory called music ("mkdir music") and switch to it ("cd music"). Enter: `nano bands.txt` to create a new text file. Enter some band names and press Control+O to output the file. Press Control+X to quit nano.

```
  GNU nano 2.2.6

The Beatles
The Who
The Kinks
Credence Clearwater_Revival
Jefferson Airplane
Aerosmith
ZZ Top
_
```

STEP 6 We're going to use this text file as input to the sort command. Enter: `sort < bands.txt` and the content from the text file is used as input to sort. Because the output isn't specified, it uses the standard output (the screen) but you use input and output together. Enter: `sort < bands.txt > bands_sorted.txt` to create a new file with the band names in order to switch back to your pi account.

```
pi@raspberrypi ~/music $ sort < bands.txt
Aerosmith
Credence Clearwater_Revival
Jefferson Airplane
The Beatles
The Kinks
The Who
ZZ Top
pi@raspberrypi ~/music $ sort < bands.txt > bands_
```

USING PIPES

As well as directing input and output to and from files, you can send the output from one command directly into another. This is known as piping, and uses the pipe character "|".

STEP 1 As you start to get more advanced in Linux, you begin to create more powerful commands, and one way you do this is by using the pipe character ("|"). Take some time to find this character if you haven't already: it usually sits above or to the left of the Return key on most keyboards.

```
pi@raspberrypi ~/music $ ps aux | less
```

STEP 2 We've used the pipe a few times in the book ("ps aux | less"), but you might not have understood what's actually happening. Enter: `cat bands.txt | wc`. The output from the cat command (the text inside the document) isn't displayed on the screen, instead it is piped into the wc (word count) function. This then tells us how many lines, words and characters are in the document.

```
pi@raspberrypi ~/music $ cat bands.txt | wc
       7      13      94
pi@raspberrypi ~/music $ _
```

STEP 3 You can pipe commands multiple times. Enter: `cat bands.txt | sort | grep The*` to get the bands starting with "The" in alphabetical order. The output of the text from the bands.txt document is passed into sort, and the output from sort is passed into grep which filters out the bands starting with "The". These bands form the output.

```
pi@raspberrypi ~/music $ cat bands.txt | sort | grep The*
The Beatles
The Kinks
The Who
pi@raspberrypi ~/music $ _
```

STEP 4 You can combine pipes with input and output to create complex expressions. You can also use >> to append outputted data to a file that already exists. Enter: `cat bands.txt | wc >> bands.txt`. This takes the output from the bands.txt file and pipes it into the wc (word count) function. The output from wc is appended to the end of the bands.txt file. Enter: `cat bands.txt` to view it.

```
pi@raspberrypi ~/music $ cat bands.txt | wc >> bands.txt_
```

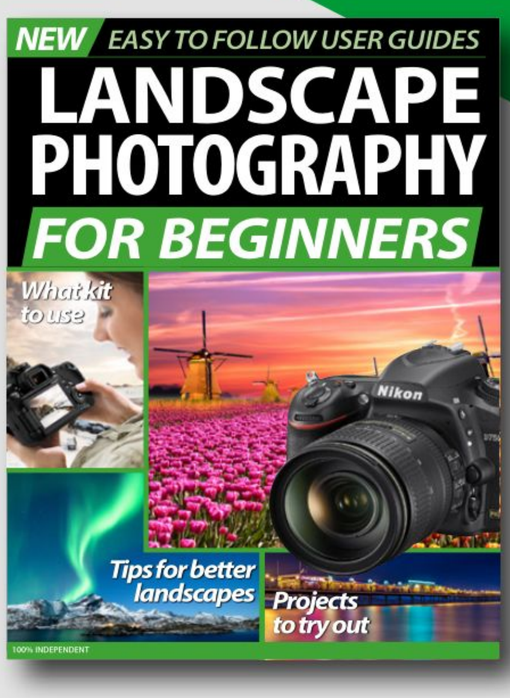
Now you've got the basics down, you can improve and learn more essential skills in our next guide...

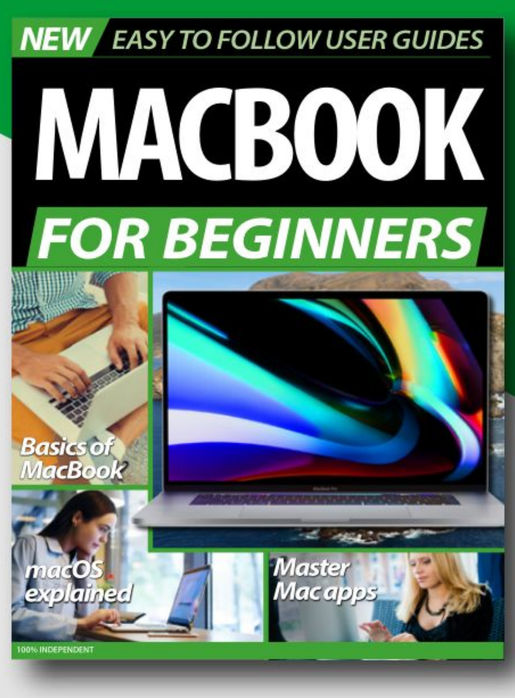
RASPBERRY PI

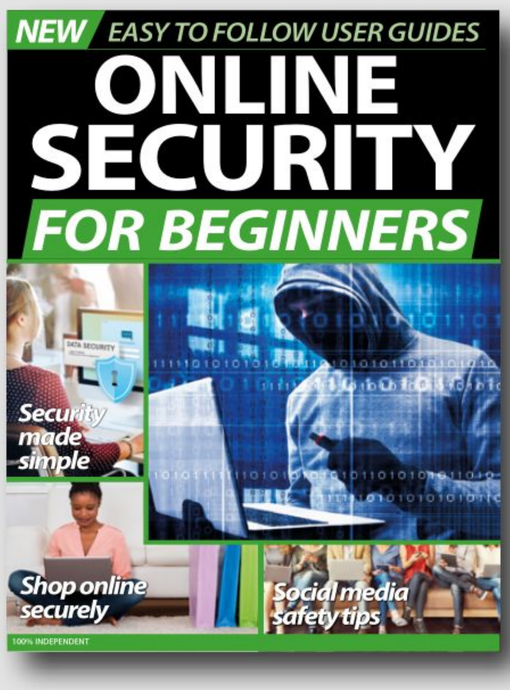
TRICKS AND TIPS

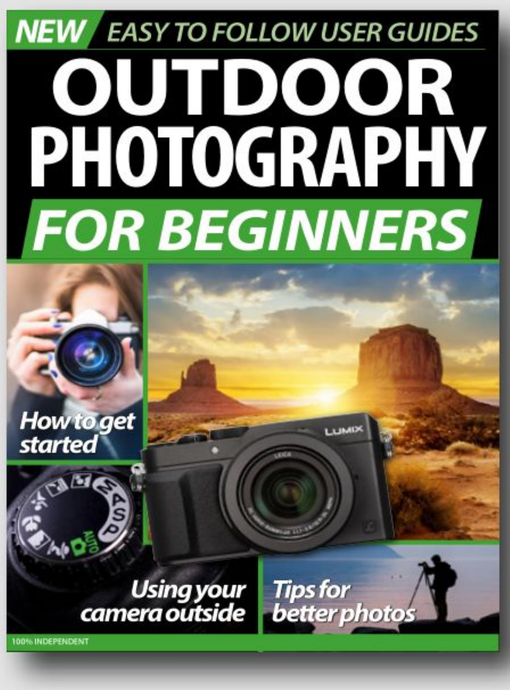
Coming soon!

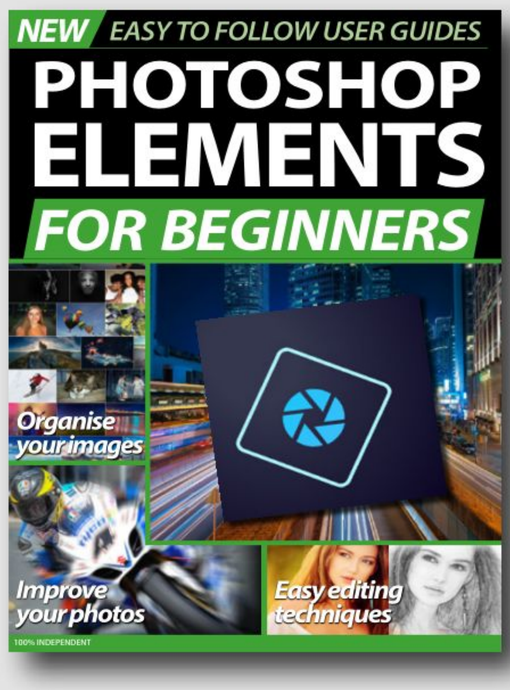
Discover more of our guides...

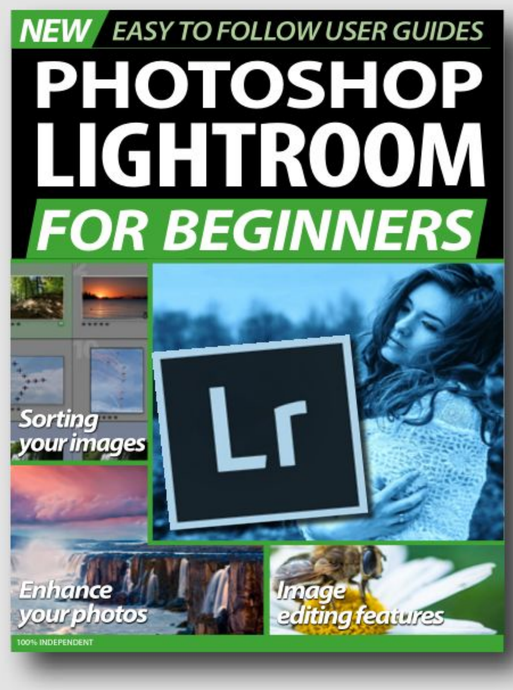


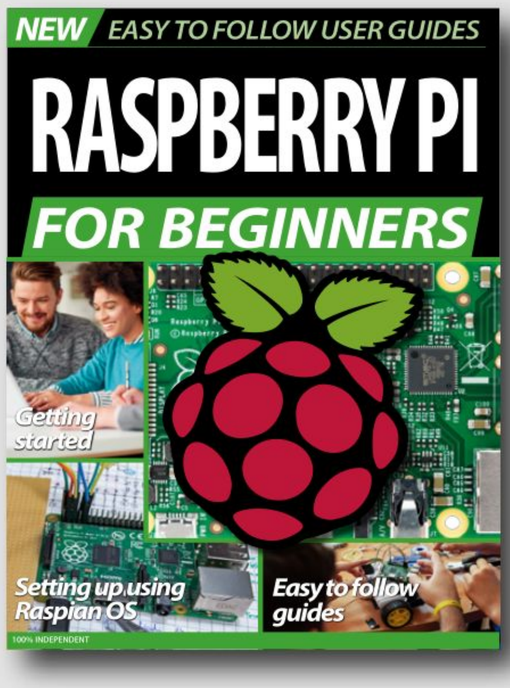












Raspberry Pi For Beginners

ISBN: 978-1-912847-33-4

Published by: Papercut Limited

Digital distribution by:

Readly AB, Zinio, Magzter, Cafeyn, PocketMags

Papercut Limited

Registered in England & Wales No: 4308513