RASPBERRY PI GUIDE FOR SENIORS

SMART WEBB

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

TABLE OF CONTENT

WHAT IS A RASPBERRY PI?

Raspberry Pi is the name of a series of single-board computers made by the Raspberry Pi Foundation, a UK charity that aims to educate people in computing and create easier access to computing education.

The Raspberry Pi launched in 2012, and there have been several iterations and variations released since then. The original Pi had a single-core 700MHz CPU and just 256MB RAM, and the latest model has a quad-core CPU clocking in at over 1.5GHz, and 4GB RAM. The price point for Raspberry Pi has always been under $100 (usually around $35 USD), most notably the Pi Zero, which costs just $5.

All over the world, people use the Raspberry Pi to learn programming skills, build hardware projects, do home automation, implement Kubernetes clusters and Edge computing, and even use them in industrial applications.

The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins, allowing you to control electronic components for physical computing and explore the Internet of Things (IoT).

What Raspberry Pi models have been released?

There have been many generations of the Raspberry Pi line: from Pi 1 to 4, and even a Pi 400. There has generally been a Model A and a Model

B of most generations. Model A has been a less expensive variant, and tends to have reduced RAM and fewer ports (such as USB and Ethernet). The Pi Zero is a spinoff of the original (Pi 1) generation, made even smaller and cheaper. Here's the lineup so far:

Pi 1 Model B (2012)

Pi 1 Model A (2013)

Pi 1 Model B+ (2014)

Pi 1 Model A+ (2014)

Pi 2 Model B (2015)

Pi Zero (2015)

Pi 3 Model B (2016)

Pi Zero W (2017)

Pi 3 Model B+ (2018)

Pi 3 Model A+ (2019)

Pi 4 Model A (2019)

Pi 4 Model B (2020)

Pi 400 (2021)

WHAT'S THE RASPBERRY PI FOUNDATION?

The Raspberry Pi Foundation works to put the power of computing and digital making into the hands of people all over the world. It does this by providing low-cost, high-performance computers that people use to learn, solve problems, and have fun. It provides outreach and education to help more people access computing and digital making—it develops free resources to help people learn about computing and making things with computers and also trains educators who can guide other people to learn.

Code Club and CoderDojo are part of the Raspberry Pi Foundation, although these programs are platform-independent (they're not tied to Raspberry Pi hardware). The Raspberry Pi Foundation promotes these clubs and helps grow the network around the world in order to ensure every child has access to learning about computing. Similarly, Raspberry Jams are Raspberry Pifocused events for people of all ages to come together to learn about Raspberry Pi and share ideas and projects.

IS THE RASPBERRY PI OPEN SOURCE?

The Raspberry Pi operates in the open source ecosystem: it runs Linux (a variety of distributions), and its main supported operating system, Pi OS, is open source and runs a suite of open source software. The Raspberry Pi Foundation contributes to the Linux kernel and various other open source projects as well as releasing much of its own software as open source.

The Raspberry Pi's schematics are regularly released as documentation, but the board is not open hardware.

The Raspberry Pi Foundation relies on income from the sale of Raspberry Pi units to do its charitable work in the education sector.

WHICH RASPBERRY PI SHOULD YOU CHOOSE FOR YOUR PROJECT?

There are a number of different models and versions of the Raspberry Pi computer. But which one is best for your project?

Raspberry Pi 4 and Raspberry Pi 400

The model with the highest potential specification is the Raspberry Pi 4, so for many general purpose projects this is your best bet. It's the most powerful Pi, with a fast clock speed, the most RAM available on a Pi yet, Wi-Fi and Bluetooth, and the ability to run two screens at 4k resolution. If you need speed and power, you want this one.

The Pi 4 has a price range depending on the amount of RAM you need. You can get up to 8 GB, but more isn't necessarily always better, especially if you don't use what you have. Computers use RAM to store whatever data is open at any one time. This may include open browser tabs, video clips in a video editor, code that's actively being written or compiled, and so on. If you're designing a simple art or utilitarian project, you may be able to settle for 1 or 2 GB. If you intend to use your Pi as a desktop computer or for intensive data proccessing, then more RAM is better.

Pros: Fast, powerful, capable of running dual screens.

Cons: Mini-HDMI ports may require new cables if you're upgrading from previous Pi models.

Perfect for: Desktop PC, home Kubernetes cluster, home server and media center, gaming emulator, computer vision.

A variation on a theme, the Pi 400 is a keyboard with, essentially, a Pi 4 built into it. It's the ultimate in portable computing: throw the Pi 400 and its power supply in your backpack, and plug it in to whatever monitor and mouse already exists at your destination. It's designed for desktop use, but it still has GPIO pins for hardware projects and all the other ports you need for connectivity.

Pros: Keyboard form factor makes this the most conveniently portable Pi yet.

Cons: Keyboard form factor makes this the physically largest Pi yet.

Perfect for: Desktop PC, classrooms, gaming emulator, computer vision.

Raspberry Pi 3

The Pi 3 is gives a genuinely pleasant desktop PC experience, in no small part thanks to four years of extreme work in optimizing the official Pi operating system, Raspbian. The Pi 3 boots in a matter of seconds, the web browser flies, you can open Minecraft and create a world in no time at all, and intensive applications like LibreOffice and Mathematica respond as they should on a decent PC.

It's also perfect for retro gaming emulation. As Stack Overflow co-founder Jeff Atwood wrote recently, "the ascendance of Raspberry Pi has single-handedly revolutionized the emulation scene. It's fast enough to emulate N64 and PSX and Dreamcast reasonably, all for a whopping $35."

If you ever tried to use OpenCV for computer vision on the Pi 1, you may have had limited success. With the more powerful Pi 3, this is much more feasible.

Pros: Fast, powerful, excellent value for money.

Cons: More power hungry.

Perfect for: Desktop PC, media center, web server, gaming emulator, computer vision.

Raspberry Pi 2

If you own a Pi 2, it's more than likely the original BCM2836-based Model B. Recently, the Pi 2 was upgraded to use the BCM3837 CPU found in the Pi 3. The only real difference between this and the 3B is the lack of Wi-Fi and Bluetooth.

If you have the 2B, you'll find it to be a reasonable desktop PC, though not as powerful as the Pi 3. It handles most things almost as well as a Pi 3 would, and has the same RAM and GPU.

If you owned both a Pi 2 and Pi 3, you could use the 3 for projects that need wireless connectivity and the 2 for ones that don't, and you wouldn't notice much difference between them. The Pi 2 runs on less power than the 3, so that's a bonus if you're not pushing it to its limit.

Gaming emulation, playing video and regular use as a desktop PC would also be appropriate for this model.

Pros: Fast, powerful, good value for money.

Cons: Roughly the same price as superior Pi 3, lacking wireless connectivity.

Perfect for: Desktop PC, media center, web server, gaming emulator.

Raspberry Pi 1 Model B+

Technically speaking, the Model B+ was only a slight improvement on the original Model B, but it added two USB ports and expanded the GPIO header from 26 to 40 pins. It also moved the Pi to a new standard form factor, which it has stuck with for its main models ever since. This format is well-received and ideal for building around. With the B+ came the HAT specification, a standard for add-on boards.

The B+ is perfectly good at most routine tasks, though there's a noticeable sluggishness when using intensive applications like the web browser when compared to the quad-core models. Because every Pi has the same GPU, they all play HD video effortlessly, so using it as a media center is a good option. The only thing you'll notice is the menus won't work quite as smoothly as the Pi 2 or 3.

The B+ gives you a 40-pin GPIO header so you can play with HATs or just access the GPIO pins for electronics projects. You have ethernet and can easily add Wi-Fi and Bluetooth via the four USB ports. Its elegant design dropped some ugly components from the original Pi and neatened up the layout of the board, making it more visually appealing.

The Pi 1 CPU has been used in many millions of units. It's probably the most stable and best supported single-board computer SoC in the world. The B+ has even been certified for space flight–it was used by astronaut Tim Peake on the International Space Station this year as part of the Astro Pi schools' competition, which is expanding into a European-wide competition in 2017.

Pros: Standard form factor, stability, price.

Cons: Slower and less powerful than other models, no wireless connectivity.

Perfect for: Media center, GPIO projects, robotics, information displays, International Space Station missions.

Raspberry Pi 1 Model A+

The A+ is a slimmed-down version of the B+ with the bank of USB ports and ethernet jack removed and replaced by a single USB port. At $20, it's cheaper than the main models and still a good value.

If you don't mind it lacking wireless connectivity or it being a bit sluggish on the desktop, you'll get along fine with the A+. If it was your only Pi, you'd probably be frustrated with it, but you can do your programming on a Pi 3 and deploy the code to the A+ to benefit from its smaller size.

It's not as small as the Pi Zero, but it does have a full-sized USB port and HDMI port, making it easy to connect up the essential peripherals. And if you want to embed it inside something like a robot, it's not too bulky, which is nice.

If you have an A+, unless you bought it very recently, it is probably the original version with 256MB RAM. It was recently upgraded to 512MB, which will help with the heavy lifting and make regular desktop use more tolerable, but it's still nowhere near as powerful as a Pi 2 or 3.

One reason you'd use the A+ over the Zero is that it has a DSI display interface, so you can use the official touch screen. If you want to leave a Pi running on low power or have it be battery or solar-powered, the A+ is your best option as it has the lowest power consumption of all models.

Pros: Standard form factor, size, weight, stability, price.

Cons: Slower and less powerful than other models, one USB port, no wired or wireless connectivity.

Perfect for: High altitude ballooning, robotics, wall display.

Raspberry Pi Zero

The Pi Zero is the smallest, lightest, cheapest Pi available. It's not available in bulk (usually one per customer), so it's not ideal if you need lots of them, but it's perfect for embedded projects that don't rely on wireless connectivity, though you can always add a Wi-Fi or Bluetooth dongle.

The Pi Zero's CPU is the Pi 1's BCM2835 overclocked to 1GHz, so it's even faster than a Pi 2 (though only single-core). It also packs 512MB RAM, giving it a surprisingly reasonable desktop experience. In terms of power usage, it's one of the lowest, around the same as a Model A+.

Now that the Pi Zero has a camera interface, it's perfect for projects like high-altitude ballooning, where size and weight really count.

The Pi Zero is not suitable for use as a general purpose PC, as you need adapters to convert from mini HDMI and micro USB, plus a USB hub, although this could still work out cheaper than a full-sized Pi! Like the A+, the Zero is ideal as a device to deploy code to from another.

Pros: Price, size, weight.

Cons: Limited availability, no wireless connectivity, GPIO header unpopulated.

Perfect for: High-altitude ballooning, miniature robotics, espionage.

Compute module

If you're serious about your project, you may wish to consider developing around the Raspberry Pi compute module. It's ideal if you want to ship a product with the Pi as its brain and need to control the layout and the space it takes up. The compute module has been used in products and prototypes for a range of applications, including media centers, digital signage solutions, box set TVs, and even cubesats.

Compute module development kits are available from around $150, but they're only intended to get you started. The idea is you design your own IO board and slot the module into it. Modules are available at around the price of a Pi, roughly $30 depending on number of orders. The upgrade to a Pi 3 CPU module is expected soon. The great thing is you will be able to swap out a CM1 for CM3 with no hassle.

The compute module also allows you to connect two cameras (for stereoscopic photography) and two displays. You could even use it simply to gain access to the massive 120 GPIO pin header to control more physical devices.

You can actually design a PCB which takes multiple compute modules, which can open up possibilities for supercomputing, parallelism, and more!

Pros: Layout customization, extra GPIOs, extra camera and display ports.

Cons: More expensive to get started, CM3 not available yet, intended for more advanced users only.

Perfect for: Mass-produced products requiring custom layout.

SETTING UP A RASPBERRY PI

What you will need

Hardware

- A Raspberry Pi computer with an SD card or micro SD card
- A monitor with a cable (and, if needed, an HDMI adaptor)
- A USB keyboard and mouse
- A power supply
- Headphones or speakers (optional)
- An ethernet cable (optional)

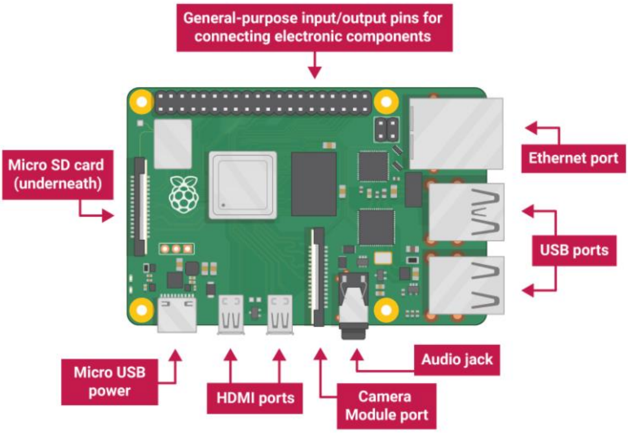
Software

- Raspberry Pi OS, installed using the Raspberry Pi Imager

MEET RASPBERRУ PI

You are going to take a first look at Raspberry Pi! You should have a Raspberry Pi computer in front of you for this. The computer shouldn’t be connected to anything yet.

- Look at your Raspberry Pi. Can you find all the things labelled on the diagram?


General-purpose input/output pins for connecting electronic components
Micro SD card (underneath)
Ethernet port
USB ports
Micro USB power
HDMI ports
Audio jack
Camera Module port

- USB ports — these are used to connect a mouse and keyboard. You can also connect other components, such as a USB drive.
- SD card slot — you can slot the SD card in here. This is where the operating system software and your files are stored.

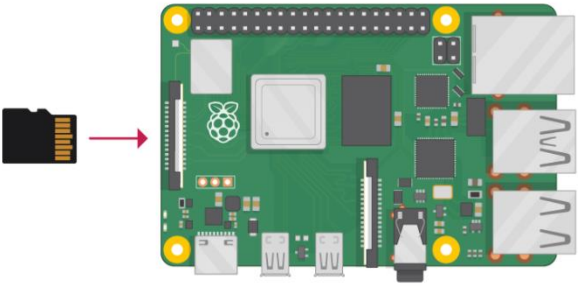
- Ethernet port — this is used to connect Raspberry Pi to a network with a cable. Raspberry Pi can also connect to a network via wireless LAN.
- Audio jack — you can connect headphones or speakers here.
- HDMI port — this is where you connect the monitor (or projector) that you are using to display the output from the Raspberry Pi. If your monitor has speakers, you can also use them to hear sound.

- Micro USB power connector — this is where you connect a power supply. You should always do this last, after you have connected all your other components.
- GPIO ports — these allow you to connect electronic components such as LEDs and buttons to Raspberry Pi.

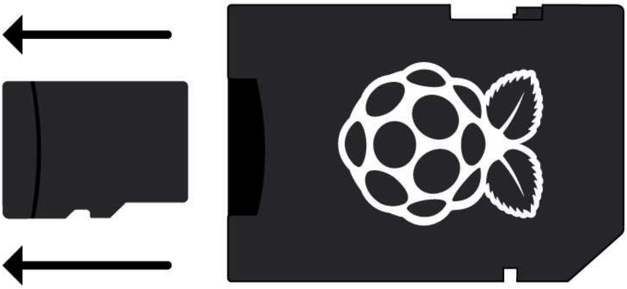
CONNECT YOUR RASPBERRY PI

Let’s connect up your Raspberry Pi and get it running.

- Check the slot on the underside of your Raspberry Pi to see whether an SD card is inside. If no SD card is there, then insert an SD card with Raspbian installed (via NOOBS).



Note: Many microSD cards come inside a larger adapter — you can slide the smaller card out using the lip at the bottom.



INSTALL RASPERRY PI OS ON YOUR SD CARD WITH THE RASPBERRY PI IMAGER

Many vendors sell SD cards with a simple Rasperry Pi OS installer called

NOOBS preinstalled but you can really easily install Rasperry Pi OS yourself using a computer that has an SD card port or using an SD card reader.

Using the Raspberry Pi Imager is the easiest way to install Rasperry Pi OS on your SD card.

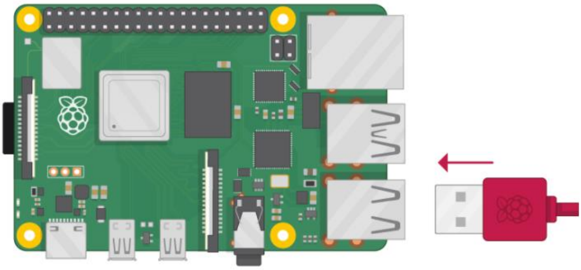
Note: More advanced users looking to install a particular operating system should use this guide to installing operating system images.

Download and launch the Raspberry Pi Imager

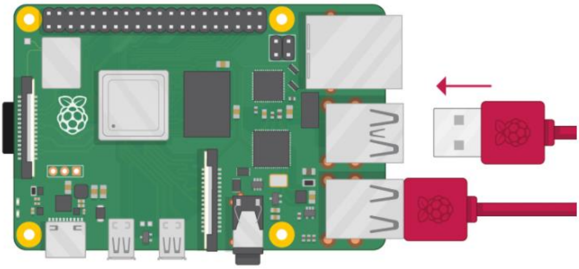
Visit the Raspberry Pi downloads page

Click on the link for the Raspberry Pi Imager that matches your operating system.

- Find the USB connector end of your mouse’s cable, and connect the mouse to a USB port on your Raspberry Pi (it doesn’t matter which port you use).



- Connect the keyboard in the same way

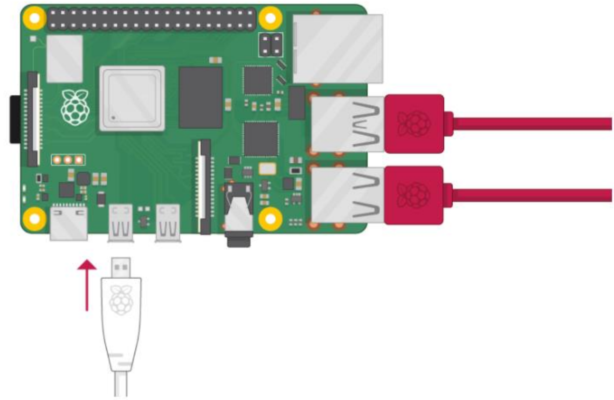


- Make sure your screen is plugged into a wall socket and switched on

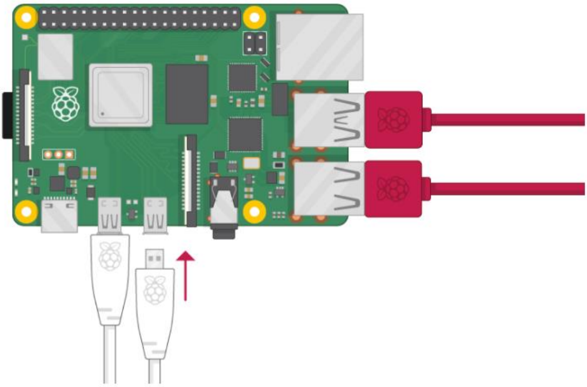
- Look at the HDMI port(s) on your Raspberry Pi — notice that they have a flat side on top.
- Use a cable to connect the screen to the Raspberry Pi’s HDMI port — use an adapter if necessary.

Raspberry Pi 4

Connect your screen to the first of Raspberry Pi 4’s HDMI ports, labelled HDMI0.

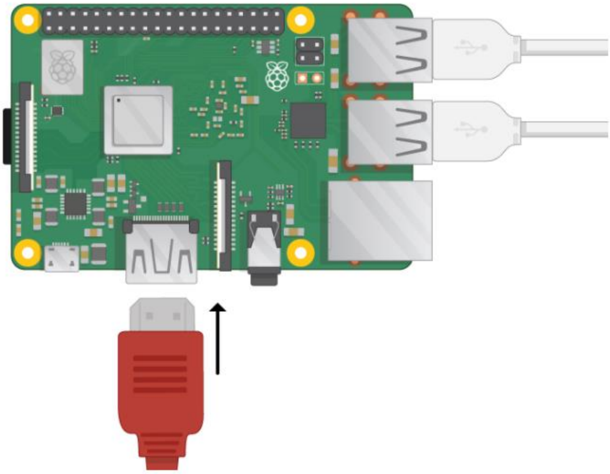


You could connect an optional second screen in the same way. You could connect an optional second screen in the same way.



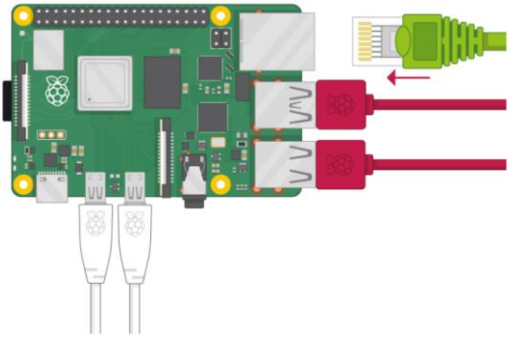
Raspberry Pi 1, 2, 3

Connect your screen to the single HDMI port.

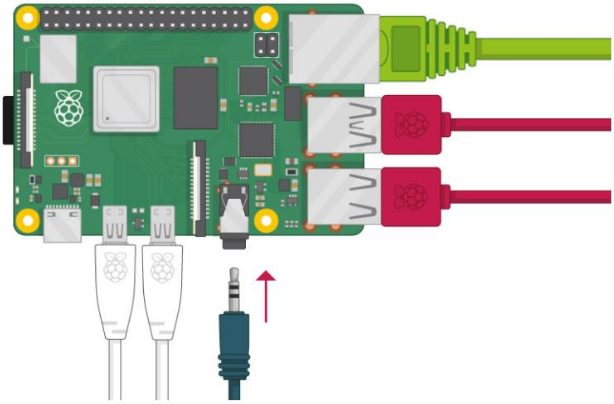


Note: nothing will display on the screen, because the Raspberry Pi is not running yet.

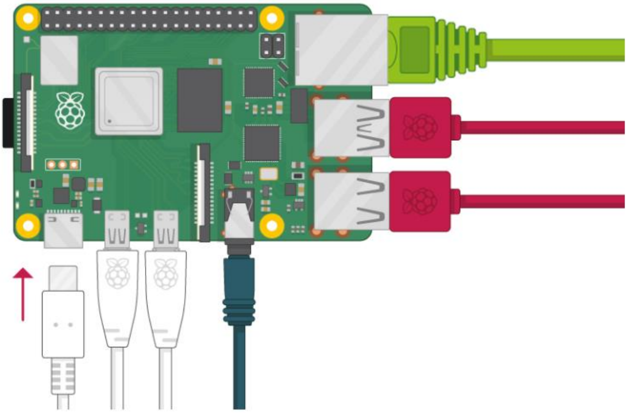
- If you want to connect the Pi to the internet via Ethernet, use an Ethernet cable to connect the Ethernet port on the Raspberry Pi to an Ethernet socket on the wall or on your internet router. You don’t need to do this if you want to use wireless connectivity, or if you don’t want to connect to the internet.



If your screen has speakers, your Raspberry Pi can play sound through these. Or you could connect headphones or speakers to the audio port.



- Plug the power supply into a socket and then connect it to your Raspberry Pi’s USB power port.



You should see a red light on your Raspberry Pi and raspberries on the monitor.

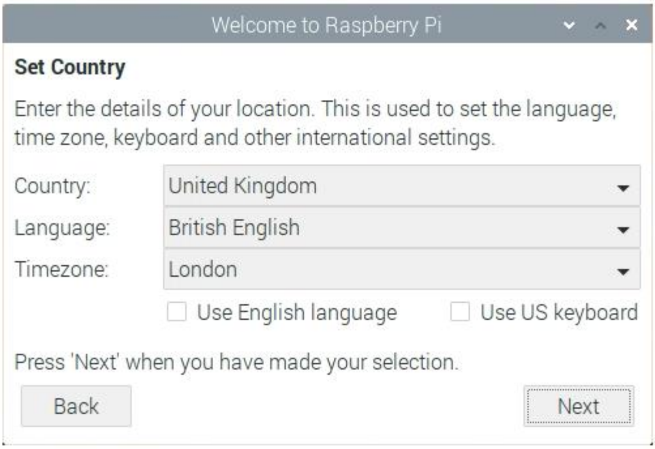
Your Raspberry Pi then boots up into a graphical desktop.

FINISH THE SETUP

When you start your Raspberry Pi for the first time, the Welcome to

Raspberry Pi application will pop up and guide you through the initial setup.

- Click Next to start the setup.
- Set your Country, Language, and Timezone, then click Next again.



Set Country

Enter the details of your location. This is used to set the language, time zone, keyboard and other international settings.

Country: United Kingdom

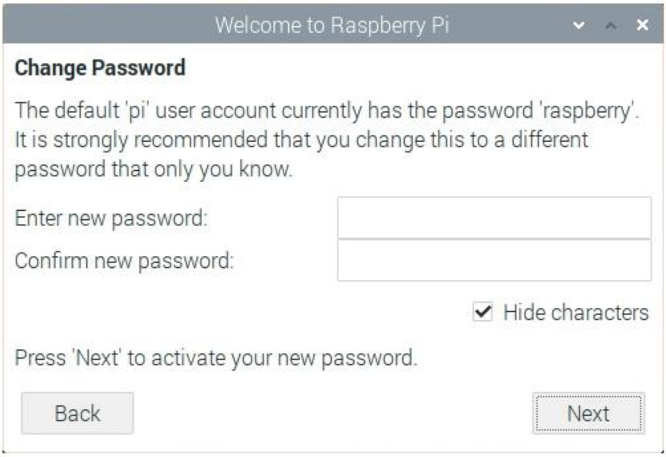
Language: British English

Timezone: London

□ Use English language □ Use US keyboard

Press 'Next' when you have made your selection.

- Enter a new password for your Raspberry Pi and click Next.



Welcome to Raspberry Pi

Change Password

The default 'pi' user account currently has the password 'raspberry'. It is strongly recommended that you change this to a different password that only you know.

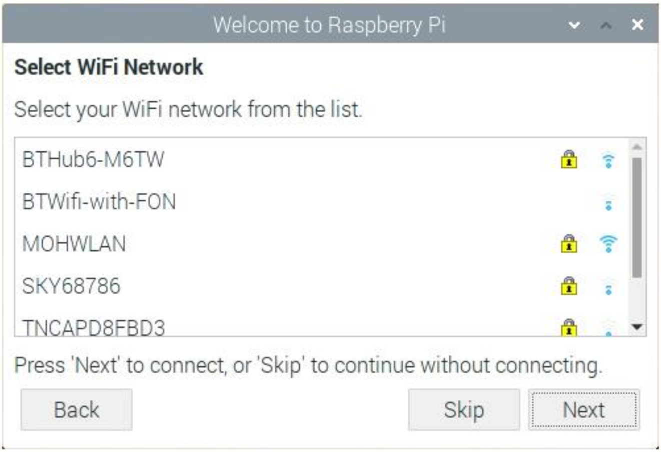
Enter new password:

Confirm new password:

☑ Hide characters

Press 'Next' to activate your new password.

- Connect to your WiFi network by selecting its name, entering the password, and clicking Next.



Select WiFi Network

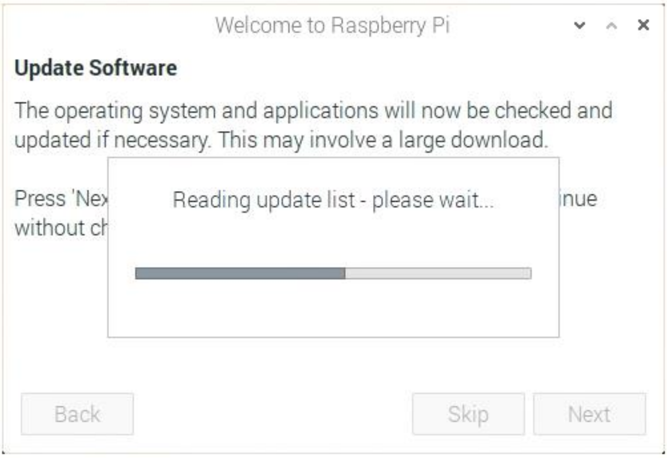
Select your WiFi network from the list.

BTHub6-M6TW
BTWifi-with-FON
MOHWLAN
SKY68786
TNCAPD8FBD3

Press 'Next' to connect, or 'Skip' to continue without connecting.

Note: if your Raspberry Pi model doesn’t have wireless connectivity, you won’t see this screen.

- Click Next let the wizard check for updates to Raspbian and install them (this might take a little while).



Welcome to Raspberry Pi

Update Software

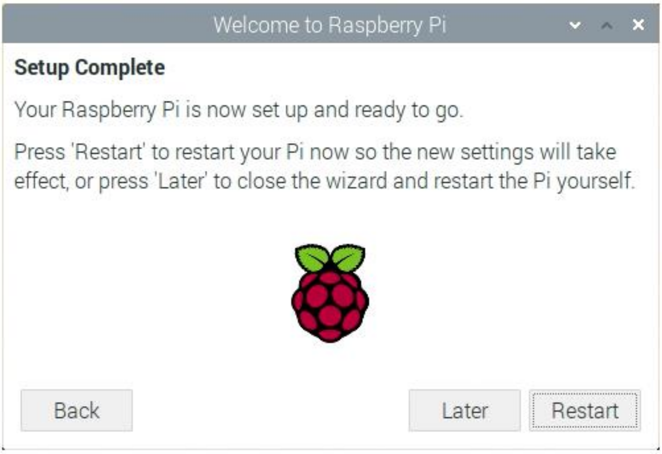
The operating system and applications will now be checked and updated if necessary. This may involve a large download.

Press 'Nex inue without ch

Reading update list - please wait...

- Click Done or Reboot to finish the setup.

Note: you will only need to reboot if that’s necessary to complete an update.



Welcome to Raspberry Pi

Setup Complete

Your Raspberry Pi is now set up and ready to go.

Press 'Restart' to restart your Pi now so the new settings will take effect, or press 'Later' to close the wizard and restart the Pi yourself.

10 HELLO WORLD PROGRAMS FOR YOUR RASPBERRY PI

"Hello world" is the beginning of everything when it comes to computing and programming. It's the first thing you learn in a new programming language, and it's the way you test something out or check to see if something's working because it's usually the simplest way of testing simple functionality.

Warriors of programming language wars often cite their own language's "hello world" against that of another, saying theirs is shorter or more concise or more explicit or something. Having a nice simple readable "hello world" program makes for a good intro for beginners learning your language, library, framework, or tool.

I thought it would be cool to create a list of as many different "hello world" programs as possible that can be run on the Raspberry Pi using its Raspbian operating system, but without installing any additional software than what comes bundled when you download it from the Raspberry Pi website. I've created a GitHub repository of these programs, and I've explained 10 of them for you here.

1. Scratch

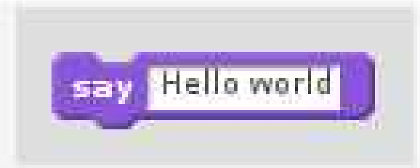
Scratch is a graphical block-based programming environment designed for kids to learn programming skills without having to type or learn the synax of a programming language. The "hello world" for Scratch is simple—and very visual.

1. Open Scratch 2 from the main menu.

2. Click Looks.

3. Drag a say Hello! block into the workspace on the right.

4. Change the text to Hello world.



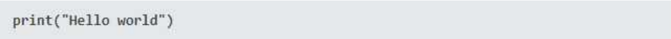
say Hello world

5. Click on the block to run the code.

2. Python

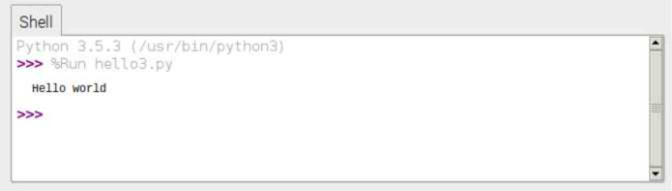
Python is a powerful and professional language that's also great for beginners— and it's lots of fun to learn. Because one of Python's main objectives was to be readable and stick to simple English, its "hello world" program is as simple as possible.

1. Open Thonny Python IDE from the main menu.
2. Enter the following code:



```
print("Hello world")
```

3. Save the file as hello3.py.
4. Click the Run button.

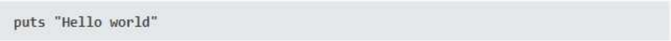


```
Python 3.5.3 (/usr/bin/python3)
>>> %Run hello3.py
  Hello world
>>>
```

3. Ruby/Sonic Pi

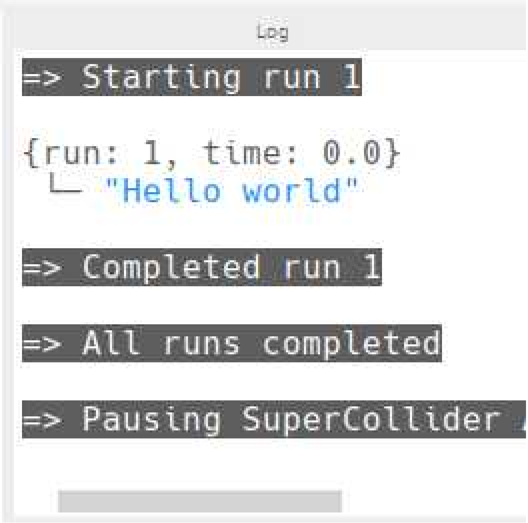
Ruby is another powerful language that's friendly for beginners. Sonic Pi, the live coding music synth, is built on top of Ruby, so what users actually type is a form of Ruby.

1. Open Sonic Pi from the main menu.
2. Enter the following code:



```
puts "Hello world"
```

3. Press Run.

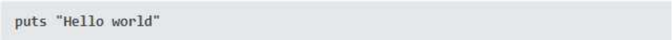


```
=> Starting run 1

{run: 1, time: 0.0}
 └─ "Hello world"

=> Completed run 1

=> All runs completed

=> Pausing SuperCollider
```

Unfortunately, "hello world" does not do Sonic Pi justice in the slightest, but after you've finished this article you should check out its creator Sam Aaron live coding, and see the tutorials on the Sonic Pi website.

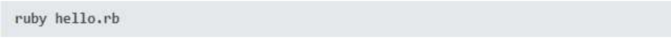
Alternatively, to using the Sonic Pi application for this example, you can write Ruby code in a text editor and run it in the terminal:

1. Open Text Editor from the main menu.
2. Enter the following code:

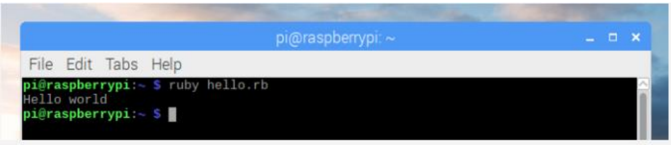


```
puts "Hello world"
```

3. Save the file as hello.rb in the home directory.
4. Open Terminal from the main menu.
5. Run the following command:



```
ruby hello.rb
```



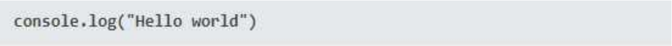
```
pi@raspberrypi:~ $ ruby hello.rb
Hello world
pi@raspberrypi:~ $ 
```

4. JavaScript

This is a bit of a cheat as I just make use of client-side JavaScript within the web browser using the Web Inspector console, but it still counts!

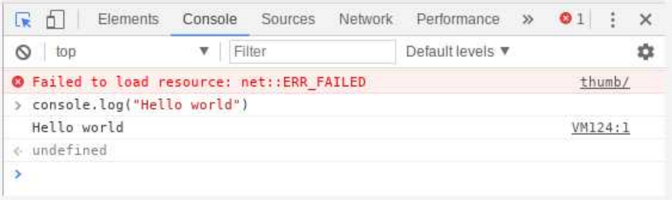
1. Open Chromium Web Browser from the main menu.
2. Right-click the empty web page and select Inspect from the context menu.
3. Click the Console tab.

4. Enter the following code:



```
console.log("Hello world")
```

5. Press Enter to run.



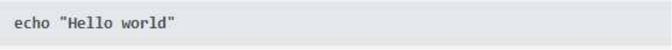
```
Failed to load resource: net::ERR_FAILED                    thumb/
> console.log("Hello world")
  Hello world                                               VM124:1
< undefined
>
```

You can also install NodeJS on the Raspberry Pi, and write server-side JavaScript, but that's not available in the standard Raspbian image.

6. Bash

Bash (Bourne Again Shell) is the default Unix shell command language in most Linux distributions, including Raspbian. You can enter Bash commands directly into a terminal window, or script them into a file and execute the file like a programming script.

1. Open Text Editor from the main menu.
2. Enter the following code:

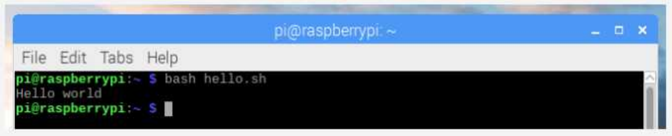


```
echo "Hello world"
```

3. Save the file as hello.sh in the home directory.
4. Open Terminal from the main menu.

5. Run the following command:

```
bash hello.sh
```



```
pi@raspberrypi:~ $ bash hello.sh
Hello world
pi@raspberrypi:~ $ 
```

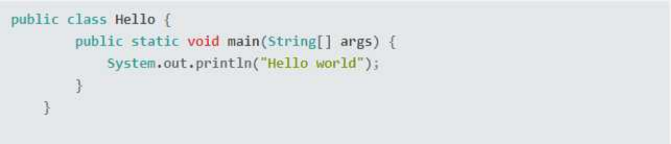
Note you'd usually see a "hashbang" at the top of the script (#!/bin/bash), but because I'm calling this script directly using the bash command, it's not necessary (and I'm trying to keep all these examples as short as possible).

You'd also usually make the file executable with chmod +x, but again, this is not necessary as I'm executing with bash.

6. Java

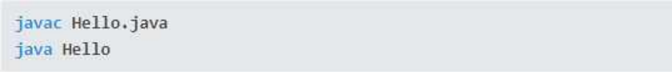
Java is a popular language in industry, and is commonly taught to undergraduates studying computer science. I learned it at university and have tried to avoid touching it since then. Apparently, now I do (very small amounts of) it for fun...

1. Open Text Editor from the main menu.
2. Enter the following code:

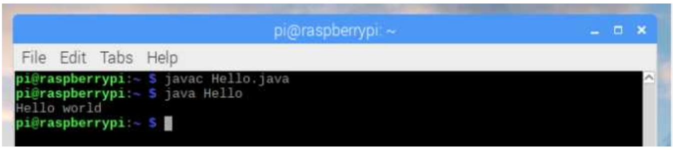


```
public class Hello {
        public static void main(String[] args) {
            System.out.println("Hello world");
        }
    }
```

3. Open Terminal from the main menu.
4. Run the following commands:



```
javac Hello.java
java Hello
```



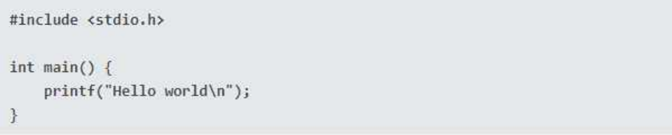
```
pi@raspberrypi:~ $ javac Hello.java
pi@raspberrypi:~ $ java Hello
Hello world
pi@raspberrypi:~ $ 
```

I could almost remember the "hello world" for Java off the top of my head, but not quite. I always forget where the String[] args bit goes, but it's obvious when you think about it...

7. C

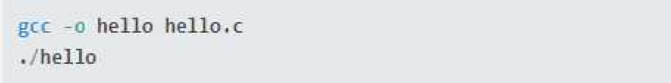
C is a fundamental low-level programming language. It's what many programming languages are written in. It's what operating systems are written in. See for yourself&mdash:take a look at the source for Python and the Linux kernel. If that looks a bit hazy, get started with "hello world":

1. Open Text Editor from the main menu.
2. Enter the following code:

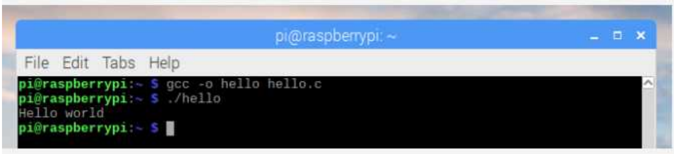


```
#include <stdio.h>

int main() {
    printf("Hello world\n");
}
```

3. Save the file as hello.c in the home directory.
4. Open Terminal from the main menu.
5. Run the following commands:



```
gcc -o hello hello.c
./hello
```



```
pi@raspberrypi:~ $ gcc -o hello hello.c
pi@raspberrypi:~ $ ./hello
Hello world
pi@raspberrypi:~ $ 
```

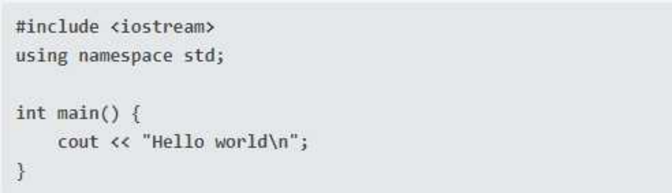
Note that in the previous examples, only one command was required to run the code (e.g., python3 hello.py or ruby hello.rb) because these languages are interpreted rather than compiled. (Actually Python is compiled at runtime but that's a minor detail.) C code is compiled into byte code and the byte code is executed.

If you're interested in learning C, the Raspberry Pi Foundation publishes a book Learning to code with C written by one of its engineers. You can buy it in print or download for free.

8. C++

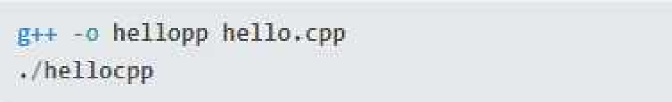
C's younger bother, C++ (that's C incremented by one...) is another fundamental low-level language, with more advanced language features included, such as classes. It's popular in a range of uses, including game development, and chunks of your operating system will be written in C++ too.

1. Open Text Editor from the main menu.
2. Enter the following code:

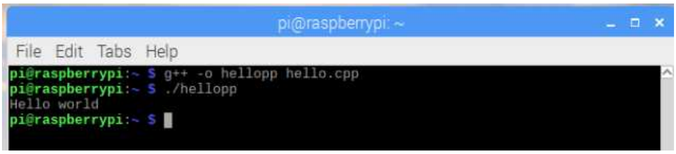


```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello world\n";
}
```

3. Save the file as hello.cpp in the home directory.
4. Open Terminal from the main menu.
5. Run the following commands:



```
g++ -o hellopp hello.cpp
./hellocpp
```



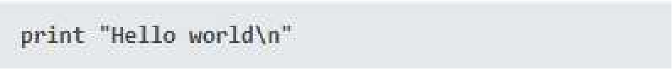
```
pi@raspberrypi:~ $ g++ -o hellopp hello.cpp
pi@raspberrypi:~ $ ./hellopp
Hello world
pi@raspberrypi:~ $ 
```

Readers familiar with C/C++ will notice I have not included the main function return values in my examples. This is intentional as to remove boilerplate, which is not strictly necessary.

9. Perl

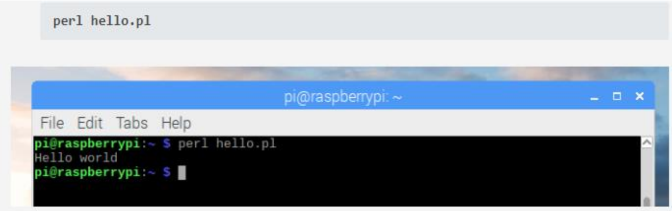
Perl gets a lot of stick for being hard to read, but nothing much gets in the way of understanding its "hello world" program. So far, so good!

1. Open Text Editor from the main menu.
2. Enter the following code:



```
print "Hello world\n"
```

3. Save the file as hello.pl in the home directory.
4. Open Terminal from the main menu.
5. Run the following command:



```
perl hello.pl
```

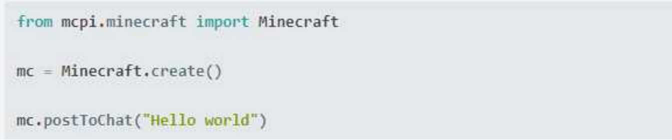
```
pi@raspberrypi:~ $ perl hello.pl
Hello world
pi@raspberrypi:~ $ 
```

Again, I learned Perl at university, but unlike Java, I have managed to successfully avoid using it.

10. Python extras: Minecraft and the Sense HAT emulator

So that's nine different programming languages covered, but let's finish with a bit more Python. The popular computer game Minecraft is available for Raspberry Pi, and comes bundled with Raspbian. A Python library allows you to communicate with your Minecraft world, so open Minecraft and a Python editor side-by-side for some fun hacking your virtual world with code.

1. Open Minecraft Pi from the main menu.
2. Create and enter a Minecraft world.
3. Press Tab to release your focus from the Minecraft window.
4. Open Thonny Python IDE from the main menu.
5. Enter the following code:



```
from mcpi.minecraft import Minecraft

mc = Minecraft.create()

mc.postToChat("Hello world")
```

6. Save the file as hellomc.py.
7. Click the Run button.

CONCLUSION

The Raspberry Pi is a powerful little beast and a great platform for building low-cost, but highly capable, embedded systems.

The interfaces built into its GPIO connector make it easy to bolt on modules using simple low-cost electronics and a bit of configuration to create very functional and flexible systems.

The inclusion of a dedicated camera interface and networking interfaces give you everything you could possible need for an Internet-connected home security system.