4.2.1 Raspberry-Pi Architecture

What is a Raspberry Pi? Raspberry pi is the name of the “credit card-sized computer board” developed by the Raspberry pi foundation, based in the U.K. It gets plugged in a TV or monitor and provides a fully functional computer capability. It is aimed at imparting knowledge about computing to even younger students at the cheapest possible price. Although it is aimed at teaching computing to kids, but can be used by everyone willing to learn programming, thebasics of computing, and building different projects by utilizing its versatility.

Raspberry Pi is developed by Raspberry Pi Foundation in the United Kingdom. The Raspberry Piis a series of powerful, small single board computer

Raspberry Pi is launched in 2012 and there have been several iterations and variations released since then.Various versions of Raspberry Pi have been out till date. All versions consist of a Broadcom system on a chip (SoC) with an integrated ARM-compatible CPU and on-chip graphics processing unit (GPU).

The original device had a single-core Processor speed of device ranges from 700 MHz to 1.2 GHz and a memory range from 256 MB to 1 GB RAM.To store the operating system and program memory Secure Digital (SD) cards are used. Raspbian OS which is a Linux operating system is recommended OS by Raspberry Pi Foundation. Some other third party operating systems like RISC OS Pi. Diet Pi, Kali, Linux can also be run on Raspberry Pi.

Used:

It also provides a set of general purpose input/output pins allowing you to control electronic components for physical computing and explore the Internet of Things (IOT).

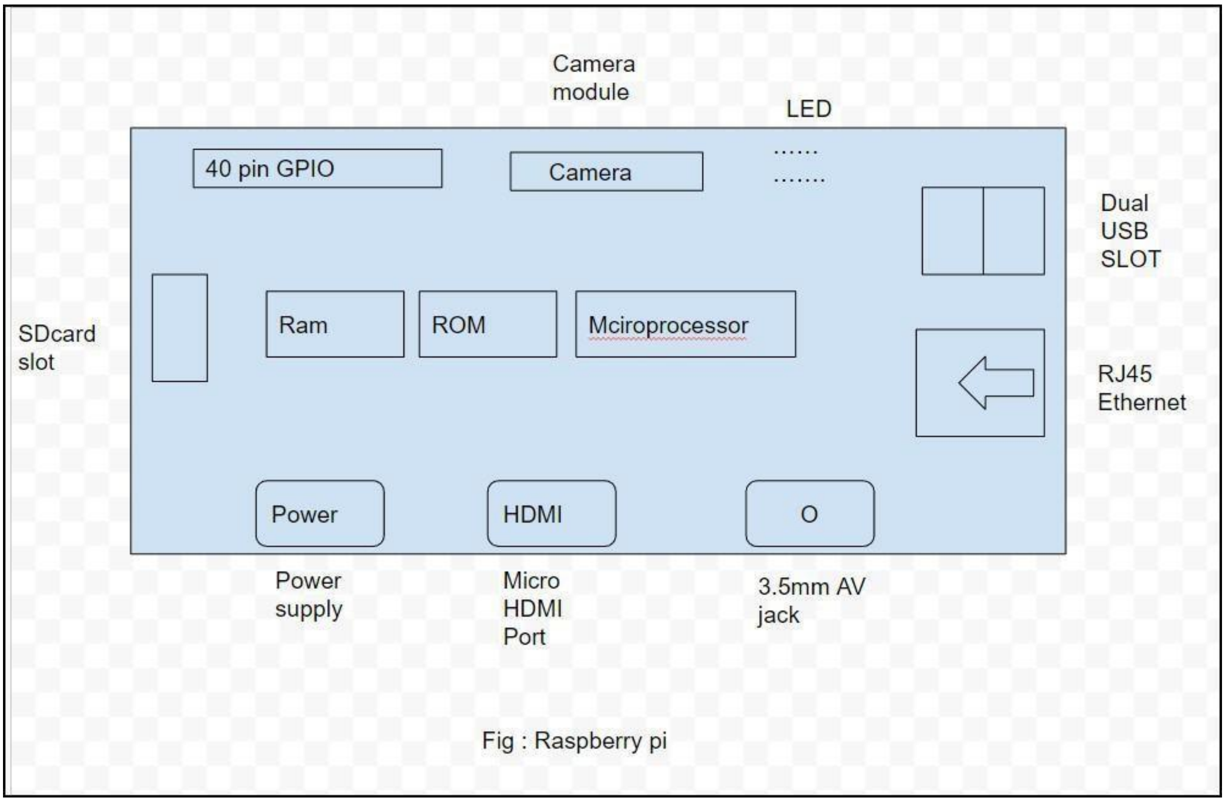
Raspberry pi Model

There have been many generations of raspberry Pi from Pi 1 to Pi 4. There is generally a modelA and model B. Model A is a less expensive variant and it trends to have reduced RAM and dualcores such as USB and Ethernet.

List of Raspberry pi models and releases year:

1. pi 1 model B – 2012
2. pi 1 model A – 2013
3. pi 1 model B+ -2014
4. pi 1 model A+ – 2014
5. Pi 2 Model B – 2015
6. Pi 3 Model B- 2016
7. Pi 3 Model B+ -2018
8. Pi 3 Model A+ -2019
9. Pi 4 Model A – 2019

Raspberry pi Diagram:



Specs of the Computer: – The computer has a quad-core ARM processor that doesn’t support the same instruction as an X86 desktop CPU. It has 1GB of RAM, One HDMI port, four USB ports, one Ethernet connection, Micro SD slot for storage, one combined 3.5mm audio/videoport, and a Bluetooth connection. It has got a series of input and output pins that are used for making projects like – home security camera, encrypted door lock, etc.

Versatility of Raspberry Pi: – It is indeed a versatile computer and can be utilized by people from all age groups, it can be used for watching videos on YouTube, watching movies, and programming in languages like Python, Scratch, and many more. As mentioned above it has a series of I/O pins that give this board the ability to interact with its environment and hence can beutilized to build really cool and interactive projects.

Examples of projects: – It can be turned into a weather station by connecting some instruments to it for check the temperature, wind speed, humidity etc… It can be turned into a home surveillance system due to its small size; by adding some cameras to it the security network will be ready. If you love reading books it can also become a storage device for storing thousands of eBooks and also you can access them through the internet by using this device.

Build Physical Projects With Python on the Raspberry Pi

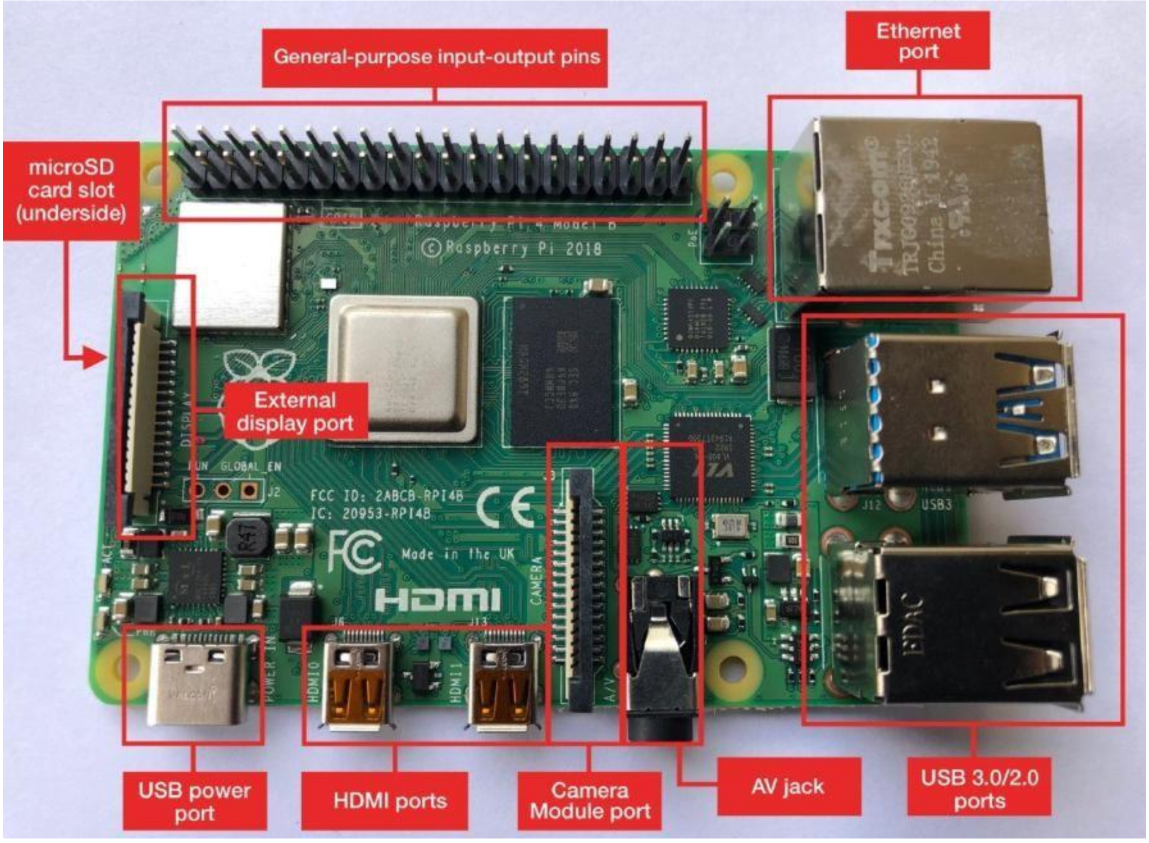
The Raspberry Pi is one of the leading physical computing boards on the market. From hobbyists building DIY projects to students learning to program for the first time, people use the RaspberryPi every day to interact with the world around them. Python comes built in on the Raspberry Pi, so you can take your skills and start building your own Raspberry Pi projects today.

Getting to Know the Raspberry Pi

The Raspberry Pi is a single-board computer developed by the Raspberry Pi Foundation, a UK- based charity organization. Originally designed to provide young people with an affordable computing option to learn how to program, it has developed a massive following in the maker and DIY communities because of its compact size, full Linux environment, and general-purpose input–output (**GPIO**) pins.

With all the features and capabilities that are packed into this small board, there’s no shortage of projects and use cases for the Raspberry Pi.If you can think of a project that would benefit from having a credit card–sized computer attached to it, then someone has probably used a Raspberry Pi to do it. The Raspberry Pi is a fantastic way to bring your Python project ideas to life.
Raspberry Pi Board Overview

Below is the board layout of the Raspberry Pi 4. While this layout is slightly different from previous models of the Raspberry Pi, most of the connections are the same. The setup described in the next section should be the same for both a Raspberry Pi 3 and a Raspberry Pi 4:



The Raspberry Pi 4 board contains the following components:

- **General-purpose input–output pins:** These pins are used to connect the Raspberry Pi toelectronic components.
- **Ethernet port:** This port connects the Raspberry Pi to a wired network. The Raspberry Pi also has Wi-Fi and Bluetooth built in for wireless connections.
- **Two USB 3.0 and two USB 2.0 ports:** These USB ports are used to connect peripherals like a keyboard or mouse. The two black ports are USB 2.0 and the two blue ports are USB 3.0.
- **AV jack:** This AV jack allows you to connect speakers or headphones to the Raspberry Pi.
- **Camera Module port:** This port is used to connect the official Raspberry Pi Camera Module, which enables the Raspberry Pi to capture images.
- **HDMI ports:** These HDMI ports connect the Raspberry Pi to external monitors. The Raspberry Pi 4 features two micro HDMI ports, allowing it to drive two separate monitors at the same time.
- **USB power port:** This USB port powers the Raspberry Pi. The Raspberry Pi 4 hasa **USB Type-C** port, while older versions of the Pi have a **micro-USB** port.
- **External display port:** This port is used to connect the official seven-inch RaspberryPi touch display for touch-based input on the Raspberry Pi.
- **microSD card slot (underside of the board):** This card slot is for the microSD card thatcontains the Raspberry Pi operating system and files.

People often wonder what the difference is between a Raspberry Pi and an Arduino. The Arduinois another device that is widely used in physical computing. While there is some overlap in the capabilities of the Arduino and the Raspberry Pi, there are some distinct differences.

The Arduino platform provides a hardware and software interface for programming microcontrollers. A microcontroller is an integrated circuit that allows you to read input from and send output to electronic components. Arduino boards generally have limited memory, so they’re often used to repeatedly run a single program that interacts with electronics.

The Raspberry Pi is a general-purpose, Linux-based computer. It has a full operating system witha GUI interface that is capable of running many different programs at the same time.
The Raspberry Pi comes with a variety of software preinstalled, including a web browser, an office suite, a terminal, and even Minecraft. The Raspberry Pi also has built-in Wi-Fi and Bluetooth to connect to the Internet and external peripherals.

For running Python, the Raspberry Pi is often the better choice, as you get a full-fledged Python installation out of the box without any configuration.

4.2.2 Raspberry Pi Programming

"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 oftesting 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.

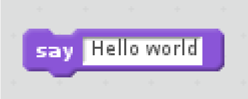
1. Scratch

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!

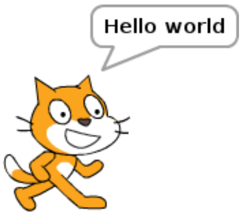
1. Open **Scratch 2** from the main menu
2. Click **Looks**.

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

4. Change the text to Hello world.



5. Click on the block to run the code.



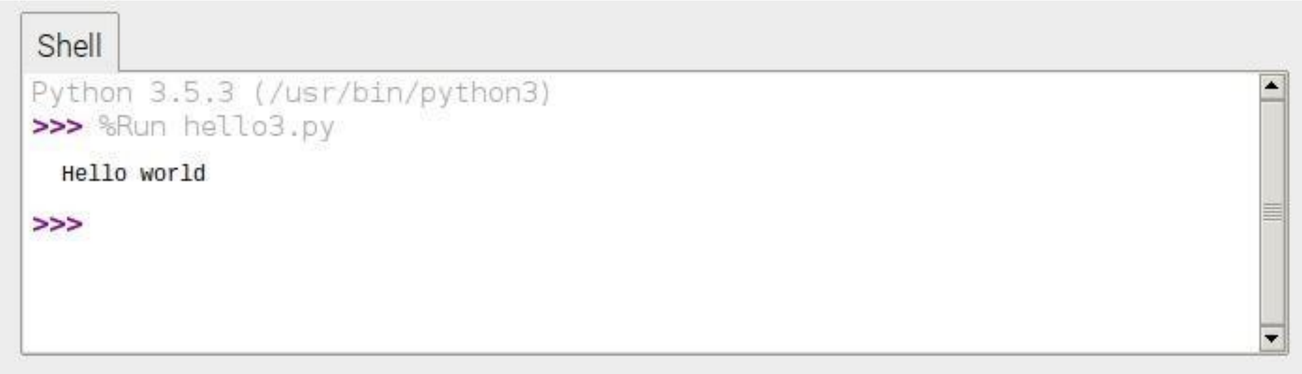
2. Python

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

>>>
```

opensource.co

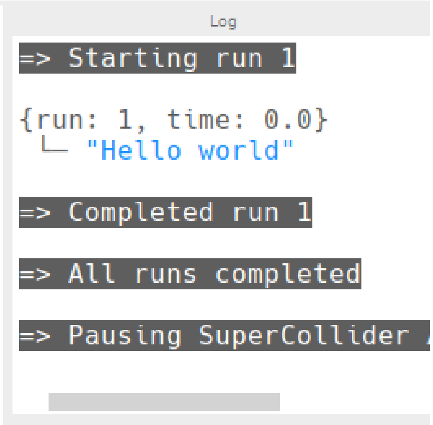
3. Ruby/Sonic Pi

Ruby is another powerful language that's friendly for beginners. Sonic Pi, the live coding musicsynth, 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

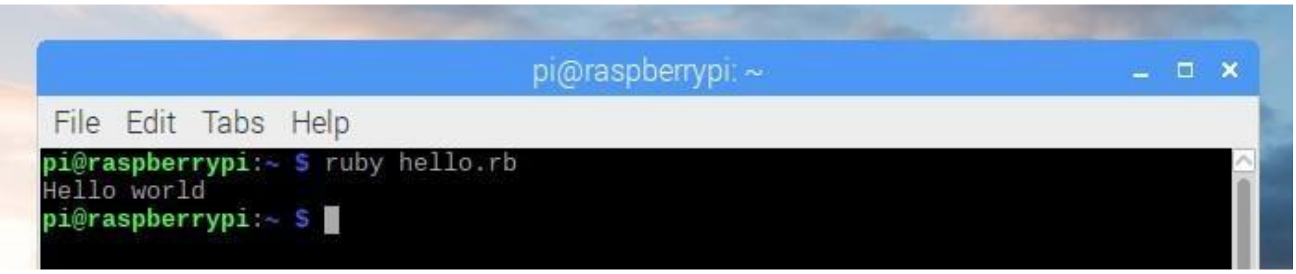
Alternatively, to using the Sonic Pi application for this example, you can write Ruby code in atext editor and run it in the terminal:

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.r
```



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

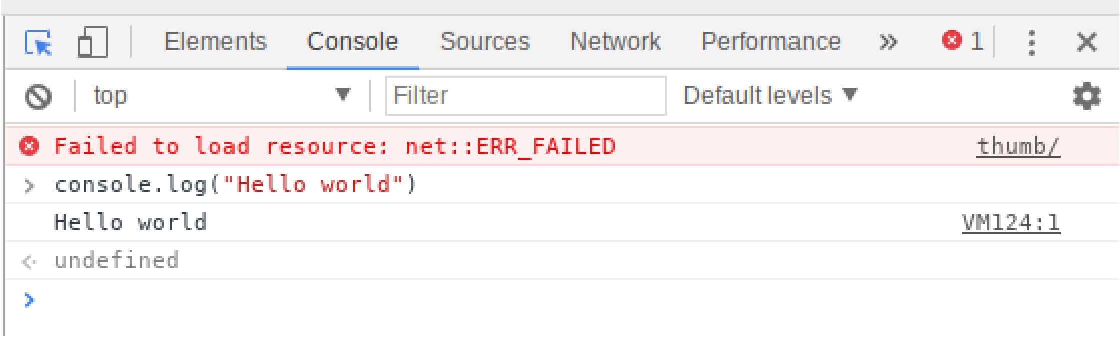
4. JavaScript

This is a bit of a cheat as I just make use of client-side JavaScript within the web browser usingthe 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.

5. Bash

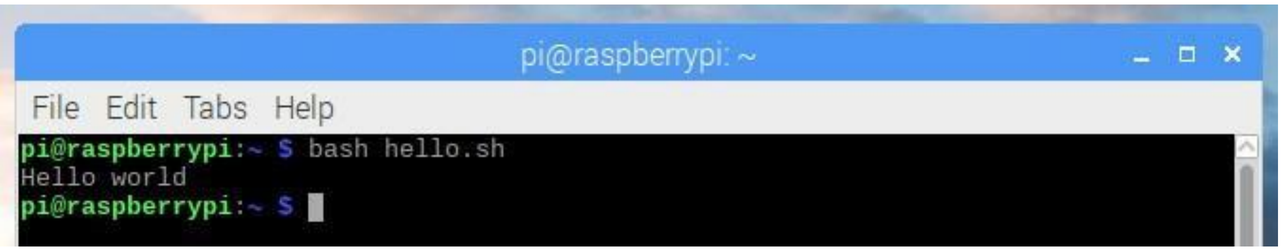
Bash (Bourne Again Shell) is the default Unix shell command language in most Linux distributions, including Raspbian. You can enter Bash commands directly into a terminal window, or script them into a file and execute the file like a programming script.

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:

```
ash
```



```
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:

3. **public class Hello** {

4. **public static void main**(String[] args) {

5. System.out.println("Hello world");

6. }

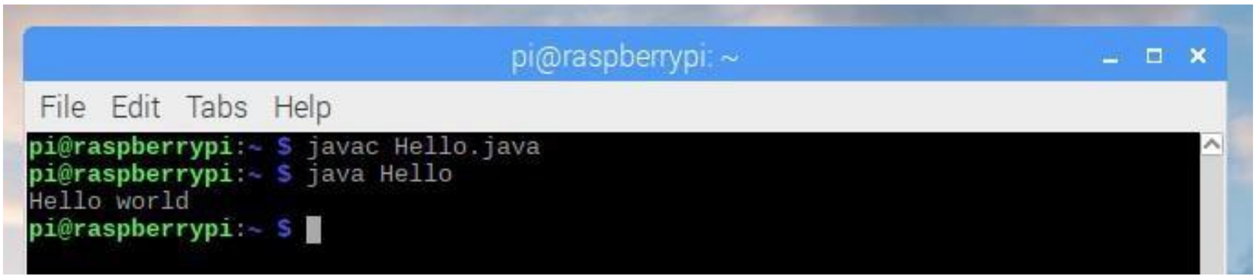
7. }

8.

9. Save the file as Hello.java in the home directory.
10. Open **Terminal** from the main menu.
11. Run the following commands:

12.
```
javac Hello.java
```

```
java Hell o
```



```
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.

Ialways 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 arewritten 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:
3. #include <stdio.h>

4. int main() {

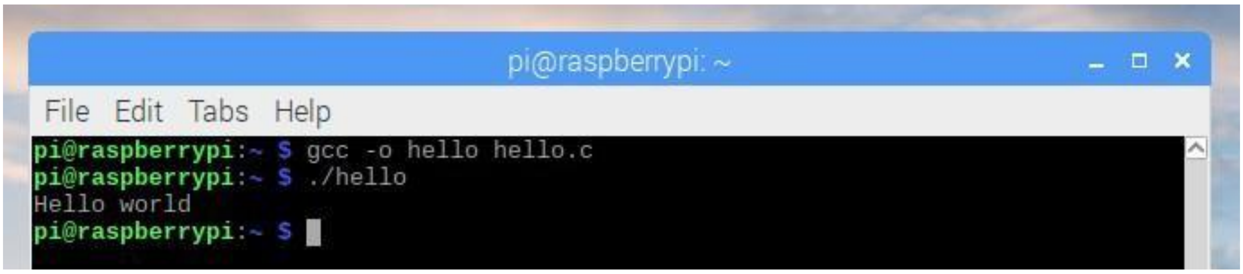
5. printf("Hello world\n");

}

6. Save the file as hello.c in the home directory.
7. Open **Terminal** from the main menu
8. Run the following commands:

9.
```
gcc -o hello
hello.c
```

```
./hel o
```



```
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 codewith 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

rangeof uses, including game development, and chunks of your operating system will be written in C+
+ too.

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

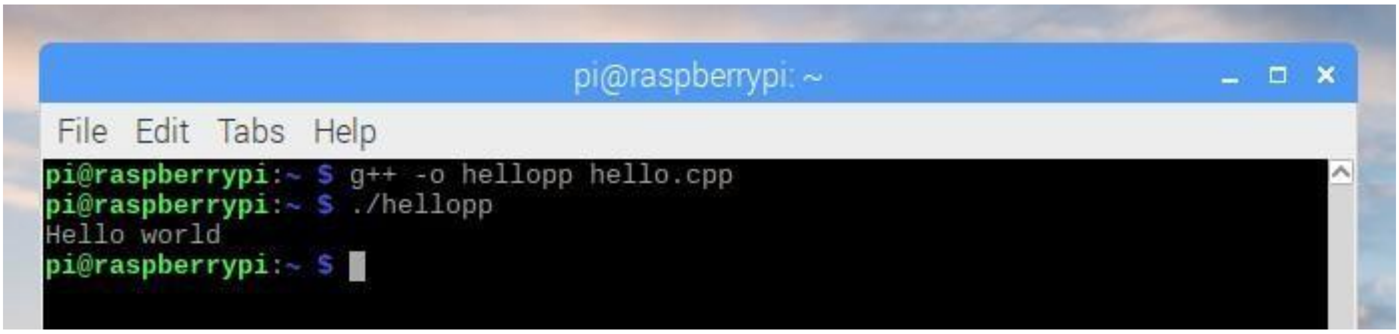
```
#include <iostream>using namespace std;int main() {
cout << "Hello world\n";
}
```

Save the file as hello.cpp in the home directory.Open **Terminal** from the main menu.

Run the following commands:

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

```
./hellocp
```



```
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 inmy examples. This is intentional as to remove boilerplate, which is not strictly necessary.