

Screaming Jelly Babies Tutorial



Your tutor should have prepared your Raspberry Pi's SD card for you. If not, please read the last section "For Tutors".

Conventions

Things that appear on the screen are written like this.

Things you need to type in, are underlined like this.

At the end of each line, you will usually need to press the Enter key.

Log in and start IDLE

When you turn on your Raspberry Pi, after a few moments it will ask you to log in. You will need to type the username (pi) followed by the Enter key, then the password (raspberrypi), followed by the Enter key.

```
raspberrypi login: pi  
password: raspberrypi
```

Note that the password won't appear on the screen when you type it.

From now on, you should assume that you always need to press the Enter key at the end of every line.

Once you have logged in, the screen will go black for a few moments, then the desktop should display and you should be able to move the pointer with your mouse. If not, enter:

```
startx
```

Find the "IDLE" icon and double-click on it. If you find it difficult to double-click, you can right-click it and select "Open" instead.

You want IDLE and not IDLE3.

IDLE is a program for writing programs, using the Python version 2 programming language. The more advanced you become with Python, the more useful features you'll use. However it is possible to write programs without IDLE, using a text editor such as nano . There are other versions of this tutorial which use nano instead of IDLE.

IDLE3 is for version 3 of Python. This tutorial is written for version 2.



Your first program – Hello World

Type in the following program:

```
print "Hello World"
```

Save it using the File menu, Save, and then double-click into the `python` folder and then the `jellybaby` folder. Save the program as: `jellybaby.py`



LX Terminal

On the desktop, double-click the LX Terminal icon. A new black window will appear.

Switch into the `python/jellybaby` folder by entering:

```
cd python/jellybaby
```

Have a look at what files are in this folder by entering:

```
ls
```

You should see your `jellybaby.py` program, some `.wav` sound files (those are the scream sounds we'll use later) and an `examples` directory (if you want to cheat, you'll find a fully working version of the screaming jelly baby program already there!).

Now run your program:

```
sudo python jellybaby.py
```

The computer should respond:

```
Hello World
```

Let's have a quick look at what we did there:

`sudo` means "superuser do". For many programs this doesn't matter, but in a minute we are going to use electronics attached to the Raspberry Pi's GPIO ports (General Purpose Input and Output), and you need to be the superuser to do that. On the downside, running a program as a superuser allows you to do things which can break the computer –the superuser is like a superhero who can do anything. Since the Raspberry Pi is a very cheap computer, that doesn't matter too much, but on other expensive computers you might want to be more careful with `sudo`.

`python` tells the computer to run a program written in the python programming language. `jellybaby.py` is the name of your program.

Wires

You should always turn off your Raspberry Pi before attaching wires to it. Make sure you tell the Raspberry Pi to shut down safely before you unplug the power. You can either use the red power icon in the bottom-right corner of the desktop, or in the LX Terminal window (the black window) type:

```
sudo shutdown -h now
```

`sudo` is the superuser-do command again. `shutdown` means to turn itself off. `-h` means halt (as opposed to `-r` which means restart) and `now` means, well, right now! (if there were other people using your Pi over the network, you could say `+5` instead of `now`, to give five minutes' warning).

Wait for the Raspberry Pi to turn itself off, then unplug the power cable.

Unwind a metal paper clip so that it is a long straight-ish wire.

Take a female-to-female jumper cable and push the unwound paper clip in one end.

Do this twice, so you have two wires, both with an unwound paper clip in the end.

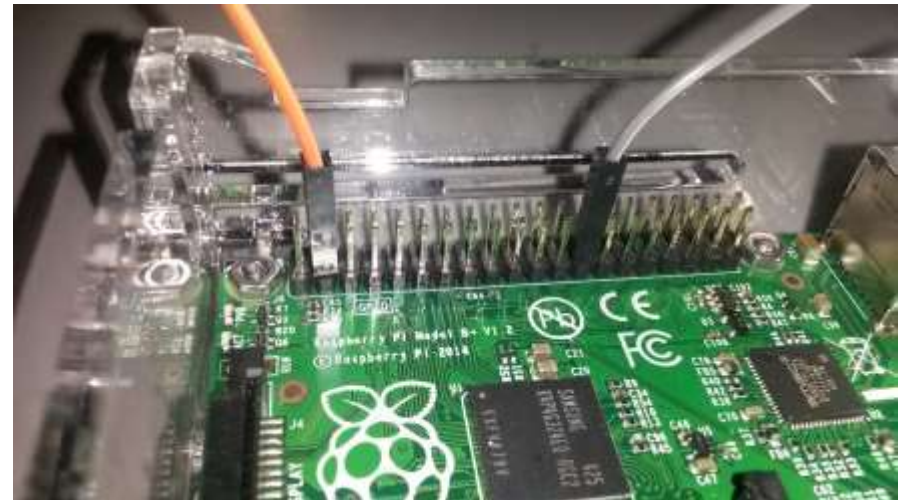
Larger paperclips may be difficult to fit. Your tutor may have already put them in for you, or you might ask your tutor for some pliers to help push them in.

Now attach the other ends of the jumper cables to pin 3 and pin 25 of the Raspberry Pi's GPIO pins.

The pins are in two rows. Even numbers are on the edge of the board and odd numbers are on the other row. You will need to count in 2s starting from 1 on the left, so the count goes 1, 3, 5, 7 and so on.

It's easy to make a mistake identifying pin 25, especially on the "Plus" models of Raspberry Pi which have 40 pins! It's a lot less hard on the old Model B and Model A which only has 26 pins – pin 25 is the one at the end on the right.

Please check and check again that you've plugged in to the right pins. If you get it wrong, you can permanently damage your Raspberry Pi. Ask your tutor to check.



The GPIO Program – Version 1

Put the power lead back in, wait for the Raspberry Pi to turn on, then log in (`pi` / `raspberrypi` ; don't forget `sudo` if your desktop doesn't start automatically).

Start IDLE from the desktop (double-click), then load your old program by going to File menu, Recent Files and selecting `jellybaby.py` .

Change the program so that it reads:

```
import time
import RPi.GPIO as GPIO
import os
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(3,GPIO.IN)

print "Hello World"

while True:
    if GPIO.input(3) == False:
        print "Ouch!"
        time.sleep(1)
```

All the capitals really matter. You need to make sure that everything in capitals really is in capitals, and everything in lower case really is in lower case. This is called being case-sensitive.

The spaces under the `while` and `if` lines are also important. You need to press the spacebar to make sure they line up. These indentations are called blocks. Blocks of code can be repeated, or the program can decide to run or miss out a block. We're using four spaces to indent each block; some programmers prefer two or another number.

The `import` lines allow us to use other people's little programs inside our program. These little programs are called libraries. The `time` library lets us tell the computer to wait (`sleep`). The `GPIO` library lets us use the GPIO pins. The `os` library lets us call other programs, such as the music player, but we're not using that just yet.

`while` tells us to repeat a block of code. `True` tells the computer to always do something.

`if` asks a question. The question being asked is whether the two wires are touching; is pin 3 connected to the electrical ground? Connecting an input pin to ground results in the value `False`. Note the use of two equals signs `==` . One equals sign is used to set a value, such as `a=3` means that every time the computer sees `a` it will know it has the value 3. Two equals signs are used to compare one thing with another.

`time.sleep(1)` means to wait for 1 second.

Save the program, then run it by starting LX Terminal from the desktop (double-click) then enter:

```
cd python/jellybaby  
sudo python jellybaby
```

The program will print `Hello World!` and then it will wait.

Touch the paperclip wires together. You should see it say:

```
Ouch!
```

Release the wires, wait few seconds and then touch the wires together again. It should say `Ouch!` each time.

If it doesn't work, check your program has been typed in right, check you saved it, and check that your wires aren't covered in anything that might be stopping the electricity.

When you've had enough, hold down the Ctrl key and press C to stop the program. This is called a "break" or a "keyboard interrupt" – you're asking the computer to take a break, and you're interrupting the program using the keyboard. Sometimes you might see Ctrl and C written as CTRL-C or even ^C. You don't need to use the shift key – the C is actually a lower case c .

The GPIO Program – Version 2

Let's change the program so that it makes a sound instead of just printing "Ouch!" on the screen.

Change:

```
print "Ouch!"
```

...to:

```
os.system('omxplayer scream-c.wav')
```

Make sure the speakers are plugged in, turned on and have the volume at least half way (some speakers do not have on/off buttons or volume).

Go to the LX Terminal window and run the program again.

Touch the wires together and you should hear a scream!

Wire up the Jelly Baby

Get a jelly baby and push the wires into it.

You need to make them cross over inside the jelly baby, but not quite touch.

Now, when you press down hard on the jelly baby, it will scream!

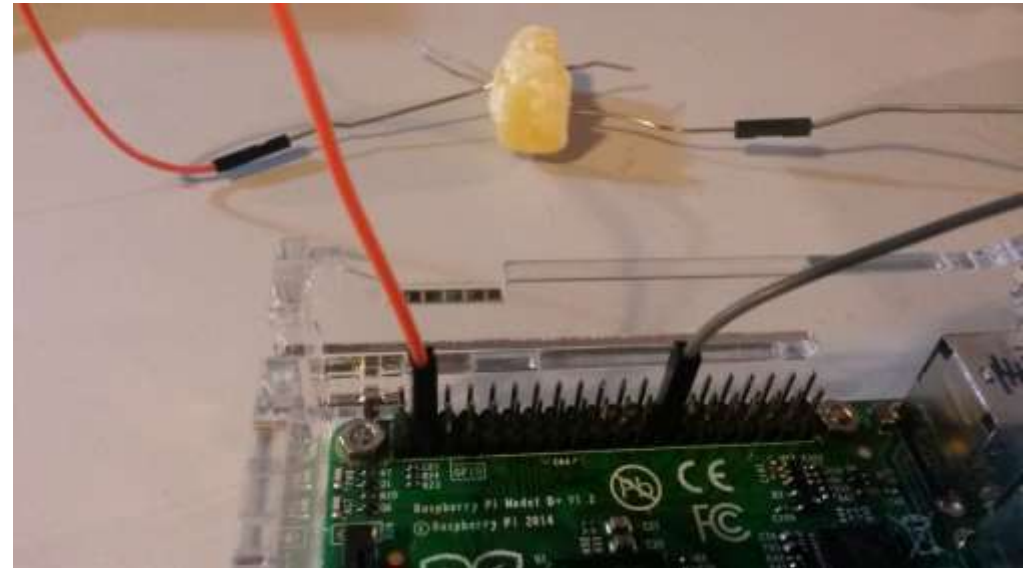
You might have to press quite hard – hard enough to push through the sticky middle of the jelly baby.

Pressing down hard means that the jelly baby will probably be useless after three or four presses. If you're not diabetic or allergic to jelly babies, you could eat the jelly babies after they're squished!

Further fun

Your program plays a scream for the musical note C . There are also scream sound files for the notes D and E. Try changing your program to use a different note. See if you and your classmates can play a tune such as Mary Had A Little Lamb (EDCDEEE,DDD,EEE,EDCDEEE,EDDEDC).

Find a list of GPIO pins (<http://raspi.tv/download/RPi.GPIO-Cheat-Sheet.pdf>). Can you connect more wires and have one Raspberry Pi wired up to two or three jelly babies at the same time, playing different notes?



For Tutors

Equipment – for each workstation:

- 1x Raspberry Pi, any model, with monitor, mouse and keyboard
- 1x SD or Micro SD card with up-to-date Raspbian operating system
- 1x speaker or pair of speakers, powered (rechargeable mobile MP3 speakers are fine, as are proper PC speakers)
- 2x Female to Female jumper wires – recommended minimum length 20cm
- 2x small metal paper clips (not plastic coated) (if you use large clips, you may need to use pliers to push them into the jumper wires)
- Several jelly babies – allow at least 6 per workstation, although probably best to only hand them out one or two at a time!
- A knowledge of which of your pupils are diabetic or allergic to jelly babies

Prerequisites

Download and unzip the folder structure, scream sound samples and example program. To do this, from an internet-connected Raspberry Pi, log in (pi / raspberry) and do:

```
cd
wget http://www.cotswoldjam.org/downloads/2014-11/jellybaby.zip
unzip jellybaby.zip
```

This will create:

```
~/python/
~/python/jellybaby/
~/python/jellybaby/scream-c.wav
~/python/jellybaby/scream-d.wav
~/python/jellybaby/scream-e.wav
~/python/jellybaby/example/
~/python/jellybaby/example/jellybaby.py
```

Note that there is an excellent diagram of GPIO pins available at: <http://raspi.tv/download/RPi.GPIO-Cheat-Sheet.pdf>

Written by Andrew Oakley of Cotswold Raspberry Jam www.cotswoldjam.org CC-BY-SA

Latest version always at: <http://www.cotswoldjam.org/tutorials/>

Based on “Make a Jelly Baby Scream!” by the Raspberry Pi Foundation, CC-BY-SA

<http://www.raspberrypi.org/learning/screaming-jellybaby/>

<https://creativecommons.org/licenses/by-sa/4.0/>