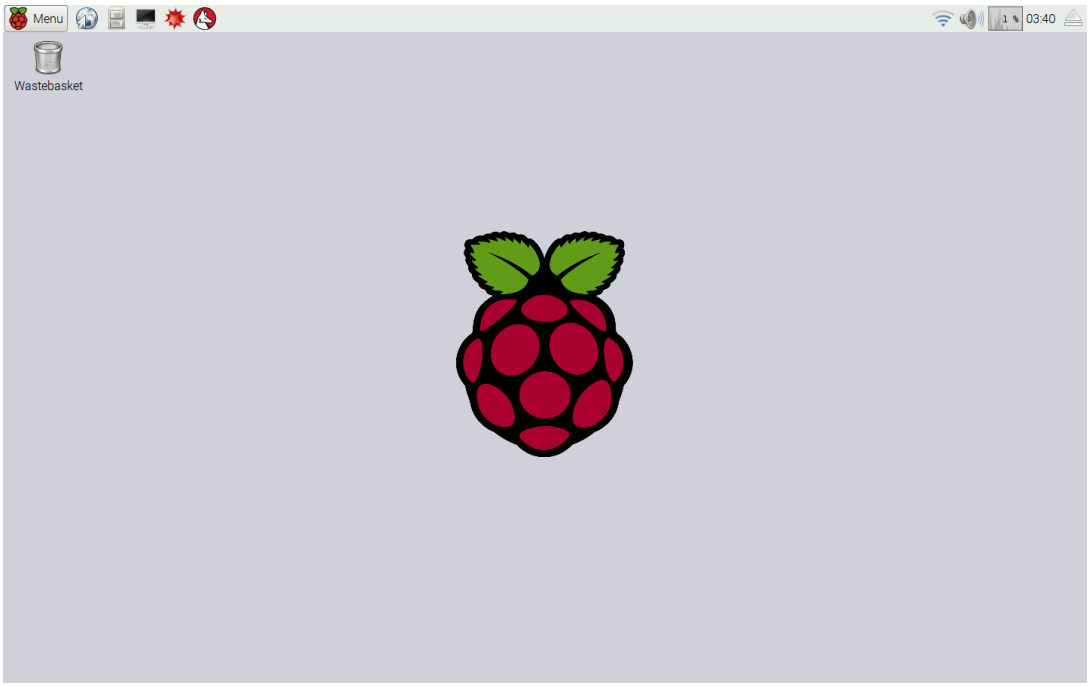
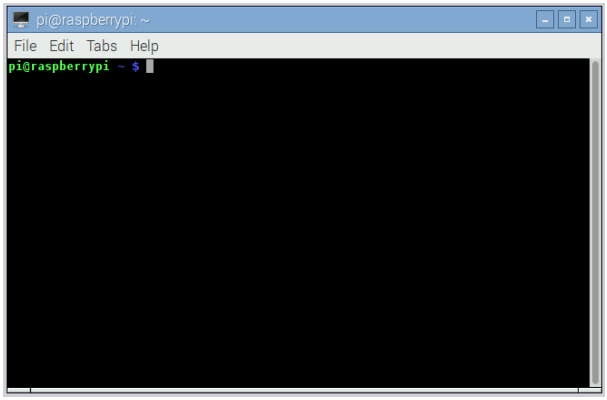
# Touch sensing and sounds with the Raspberry Pi

When you first start your Raspberry Pi it should be showing its start up screen as shown below.

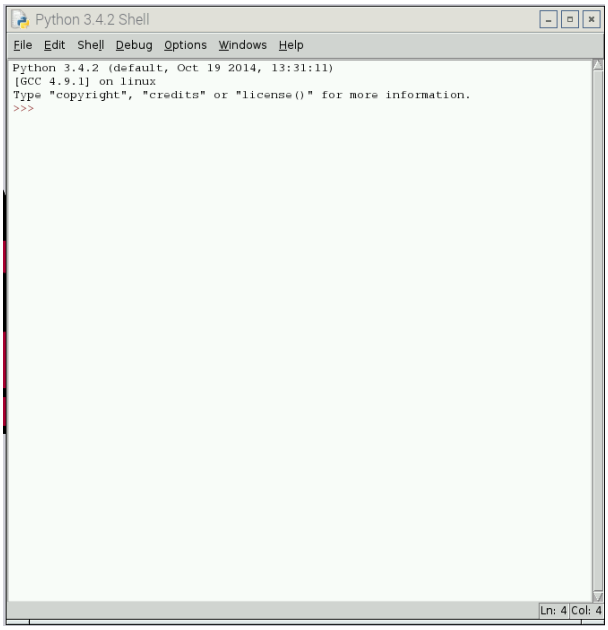



The first thing we need to do is to start running a *terminal*. You start this by clicking on the terminal icon along the panel at the top of the screen. Its icon looks like this:

When the terminal appears it looks like this:

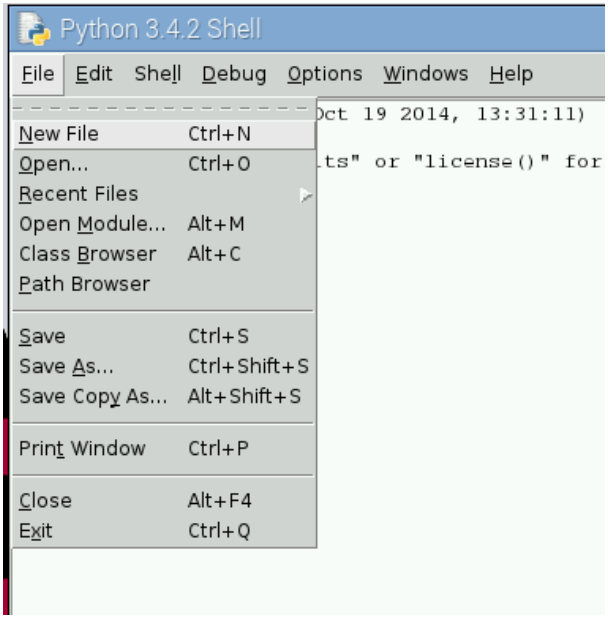


We need to start up IDLE3 (the Python3 development environment) by typing the command `sudo idle3` in the window and hit return. At this point the IDLE3 shell window will appear as below:




The IDLE3 environment is used to write and run the progrms we'll be using in this workshop. We're using the modern Python3 programming languge often used for both teaching and in industry.

We'll need another window to write our program in. We can create this by going to the File tab and selecting New File as shown below:




This open up a new window where we can write our program. Let's start off with a simple test. Type the following into the new (program) window.

```
print ("Hello World")
```

We now need to save and run this program. We do this by using the File menu nd selecting Save as. We use this as we haven't previously named the file. A good name might be `hello.py` (the `.py` suffix indicates that it's a Python program). We can now run this by using the Run menu and selecting Run Module.

The other window should show the `Hello World` output.

You've now run your fist python program!

If you've made a typing mistake (which is very easy to do) then just correct it and run the program again. IDLE3 will ask you if you want to save the program before you run it and you should say Yes to this.

Next we're going to use the Explorer Hat. This needs a slightly more complex program to drive it. Here's that program.

```
import explorerhat

def ohai (channel, event):
   print ("Ohai! I got a touch on button: {}".format (channel))

explorerhat.touch.one.pressed (ohai)
```

You'll need to take care with the indentation (spacing at the start of the line) and the punctuation. Run this and see if it works. You may need to go back and check that you've typed it in properly.

So, how does this work?

The first line `import explorerhat` tell Python3 that we want to use the explorerhat *library*. A library is some software someone else wrote that allows us to perform some activities. One of the best features of Python is that it has a very large amount of libraries for it.

The next two lines:

```
def ohai (channel, event):
   print ("Ohai! I got a touch on button: {}".format (channel))
```

tell Python that we want to `def`ine a function called `ohai` that takes two *parameters*: `channel` and `event`. These tell the code that follows where something happened (`channel`) and what happened (`event`).

The `print` command next is *indented* to show that it's part of the `ohai` function. The part between the outermost brackets indicates what we want to print and in this case it takes the text (known as a *string* in Python) and formats it to put the channel number (in our case the button number) in.

The final line, `explorerhat.touch.one.pressed (ohai)`, tells Python that when button one is pressed we want it to call the `ohai` function. The "backwards" addressing of the name might look a little odd but it's the way that we tell Python where to find things. I think of it as each dot representing ownership. So it says explorerhat's touch sensors's number one sensor's pressed action.

It's also important to understand that the `ohai` function *isn't* called at this point. Instead it is "registered" so that when the sensor is touched later is will be called *then*.

The Explorer Hat has eight touch sensors (one to eight) with numbers five to eight available on the side with the ability to use crocodile clips to extend them. Try replacing the `one` in the program with one of `five` to `eight` and see what happens. You can try connecting tin foil, fruit, play-doh or anything else that comes to hand to see what happens.

Now we need to add some sound. First let's create a new program to just play a sound. Here's the program that just plays a clap.

```
import pygame

pygame.mixer.init ()
clap = pygame.mixer.Sound ("sounds/drums/clap.wav")
pygame.mixer.Sound.play (clap)
```

Try running this program and you should be able to hear a drum clap sound. If you can't then check for any program errors and if there are none then check that your headphones are plugged in properly and the audio output is set to headphones (you'll need to run `raspi-config` for this. It's under advanced options (8) and is option A9.)

Now we combine both of our previous programs. Try out the following:

```
import explorerhat
import pygame

pygame.mixer.init ()
clap = pygame.mixer.Sound ("sounds/drums/clap.wav")

def ohai (channel, event):
   print ("Ohai! I got a touch on button: {}".format (channel))
   pygame.mixer.Sound.play (clap)

explorerhat.touch.five.pressed (ohai)
```

Here we `import` both `explorerhat` and `pygame`, set up the mixer, load the sound, define our function to not only say that a sensor has been touched but to play a sound and finally make sure that the function is called when (in this case) sensor five is touched (pressed).

From here on in it's time to ad-lib. What other sounds are there? How many buttons can you use? What can you connect to the board to make the sensors work?

Depending on how good your python is you can connect different sensors to different sounds in different ways. Give this a try!

There are a lot of possiblities that you can try out from here. What would you like to do?

Finally, as ever, it's time for questions!