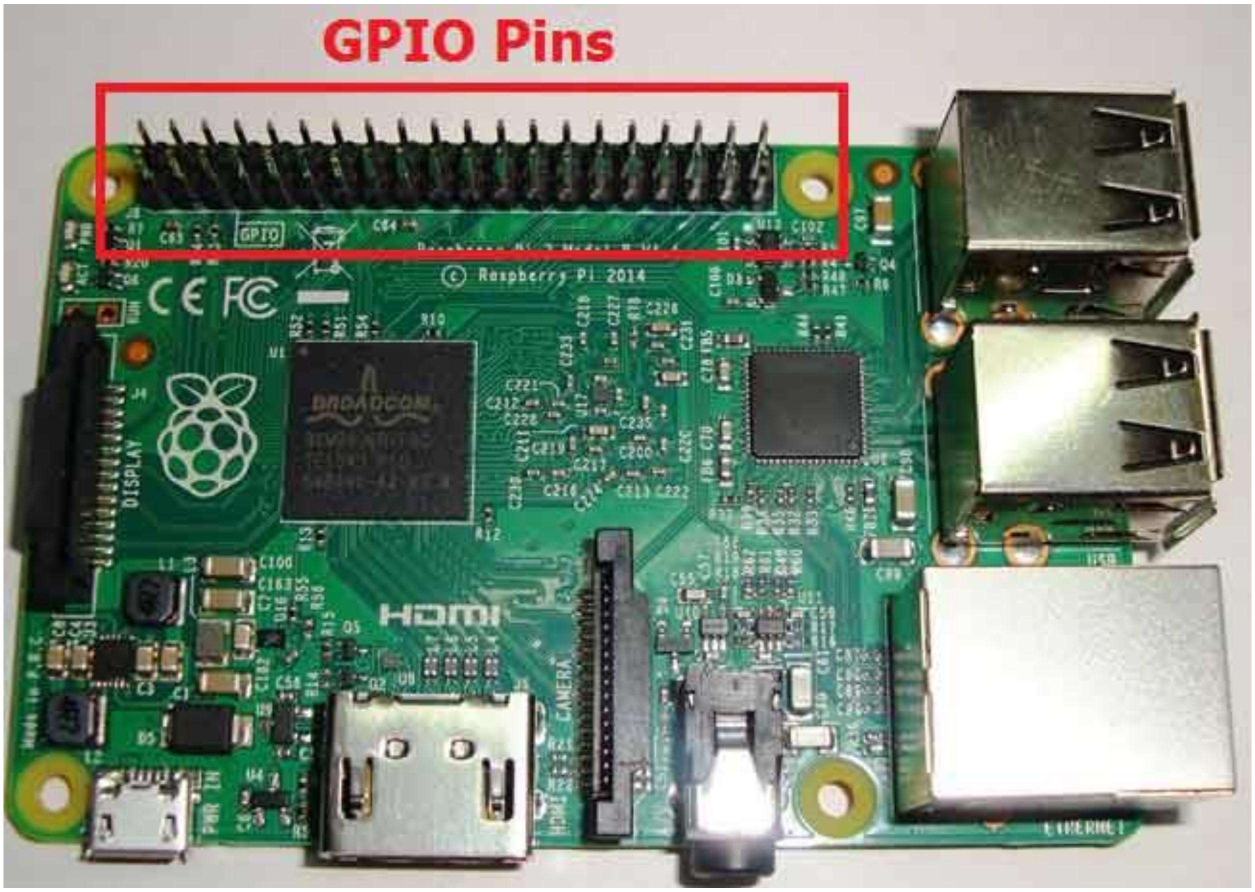
LED INTERFACING WITH RASPBERY PI

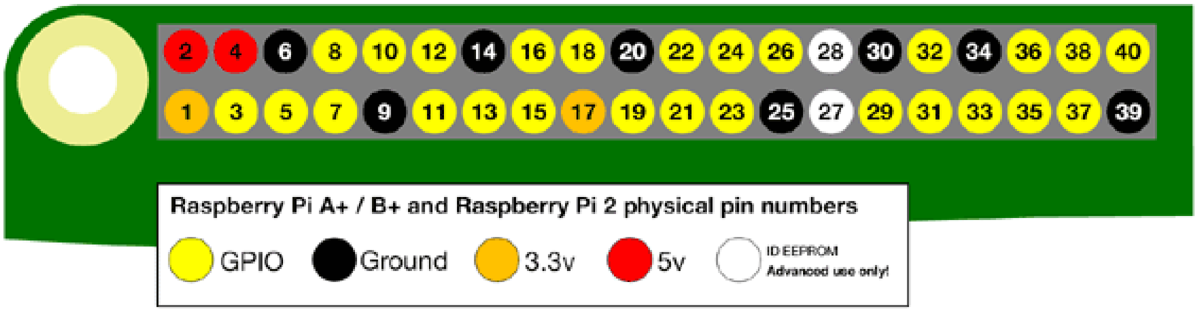
Raspberry Pi is an ARM architecture processor based board designed for electronic engineers and hobbyists. The PI is one of most trusted project development platforms out there now. With higher processor speed and 1 GB RAM, the PI can be used for many high profile projects like Image processing and Internet of Things.

For doing any of high profile projects, one need to understand the basic functions of PI. That is why we are here, we will be teaching all the basic functionalities of Raspberry Pi in these tutorials. In each tutorial series we will discuss one of functions of PI. By the end of tutorial series you will be able to do high profile projects by yourself. Check these for Getting Started with Raspberry Pi and Raspberry Pi Configuration.

In this tutorial of PI series, we will understand the concept of writing and executing programs on PYTHON. We will start with Blink LED using Raspberry Pi. Blinky is done by connecting an LED to one of GPIO pins of PI and turning it ON and OFF.

We will discuss a bit about PI GPIO Pins before going any further,

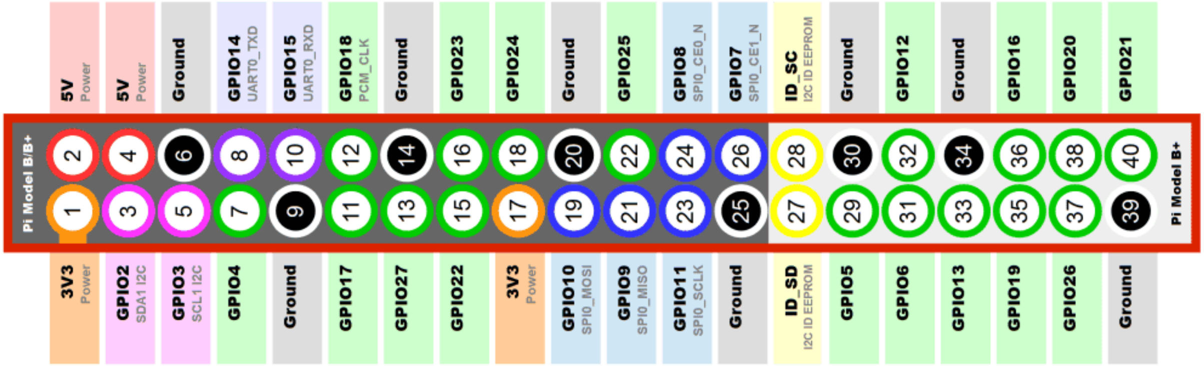




As shown in above figure, there are 40output pins for the PI. But when you look at the second figure, you can see not all 40 pin out can be programmed to our use. These are only 26 GPIO pins which can be programmed. These pins go from GPIO2 to GPIO27.

These 26 GPIO pins can be programmed as per need. Some of these pins also perform some special functions, we will discuss about that later. With special GPIO put aside, we have 17 GPIO remaining (Light green Cirl).

Each of these 17 GPIO pins can deliver a maximum of 15mA current. And the sum of currents from all GPIO cannot exceed 50mA. So we can draw a maximum of 3mA in average from each of these GPIO pins. So one should not tamper with these things unless you know what you are doing.



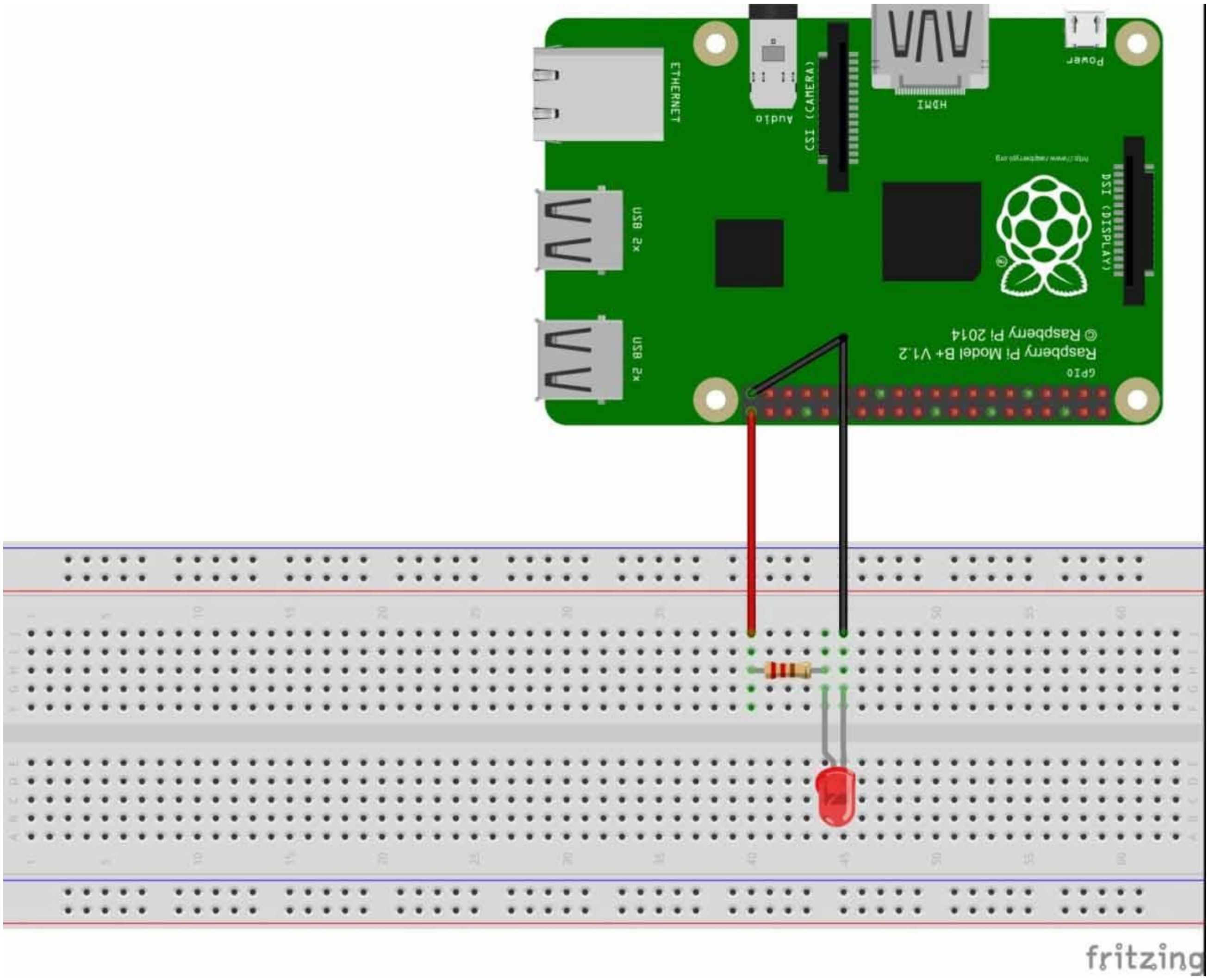
Components Required:

Here we are using Raspberry Pi 2 Model B with Raspbian Jessie OS. All the basic Hardware and Software requirements are previously discussed, you can look it up in the Raspberry Pi Introduction, other than that we need:

- Connecting pins
- 220Ω or 1KΩresistor
- LED
- Bread Board

Circuit Explanation:

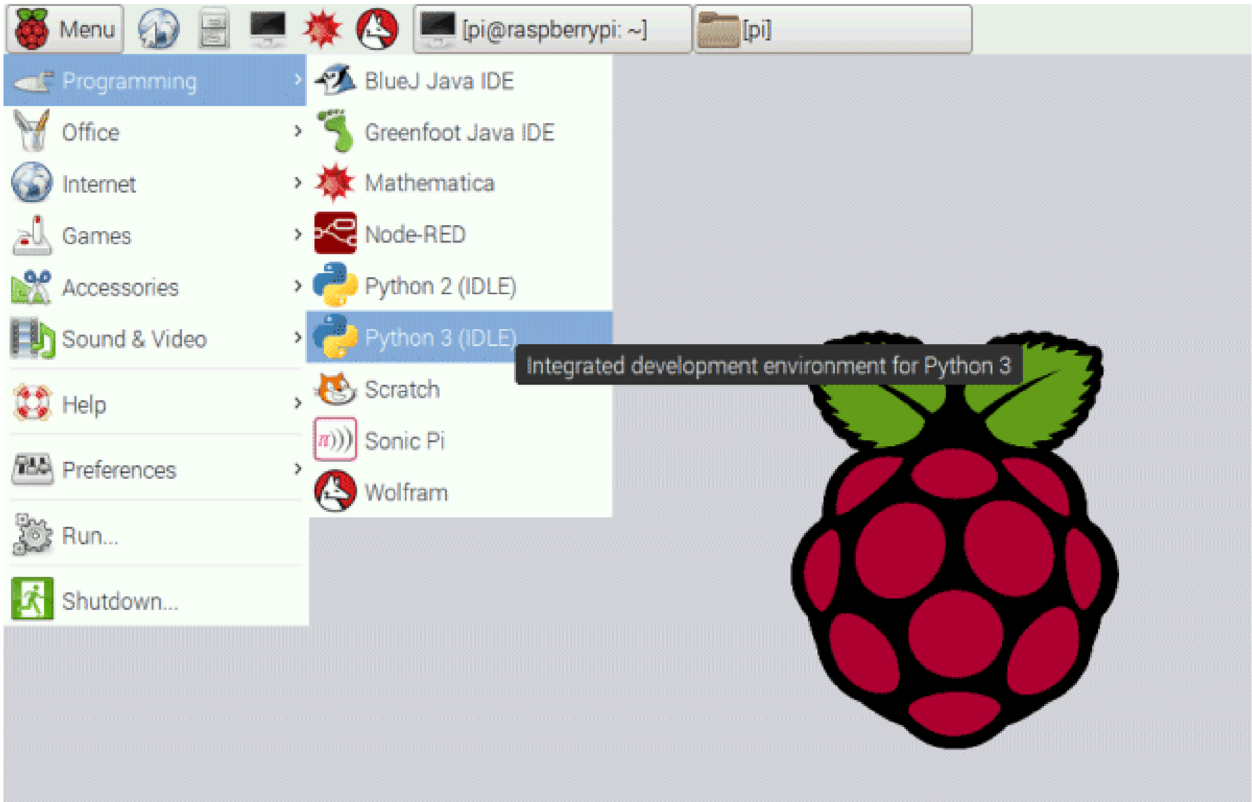
As shown in the circuit diagram we are going to connect an LED between PIN40 (GPIO21) and PIN39 (GROUND). As said earlier, we cannot draw more than 15mA from any one of these pins, so to limit the current we are connecting a 220Ω or 1KΩ resistor in series with the LED.



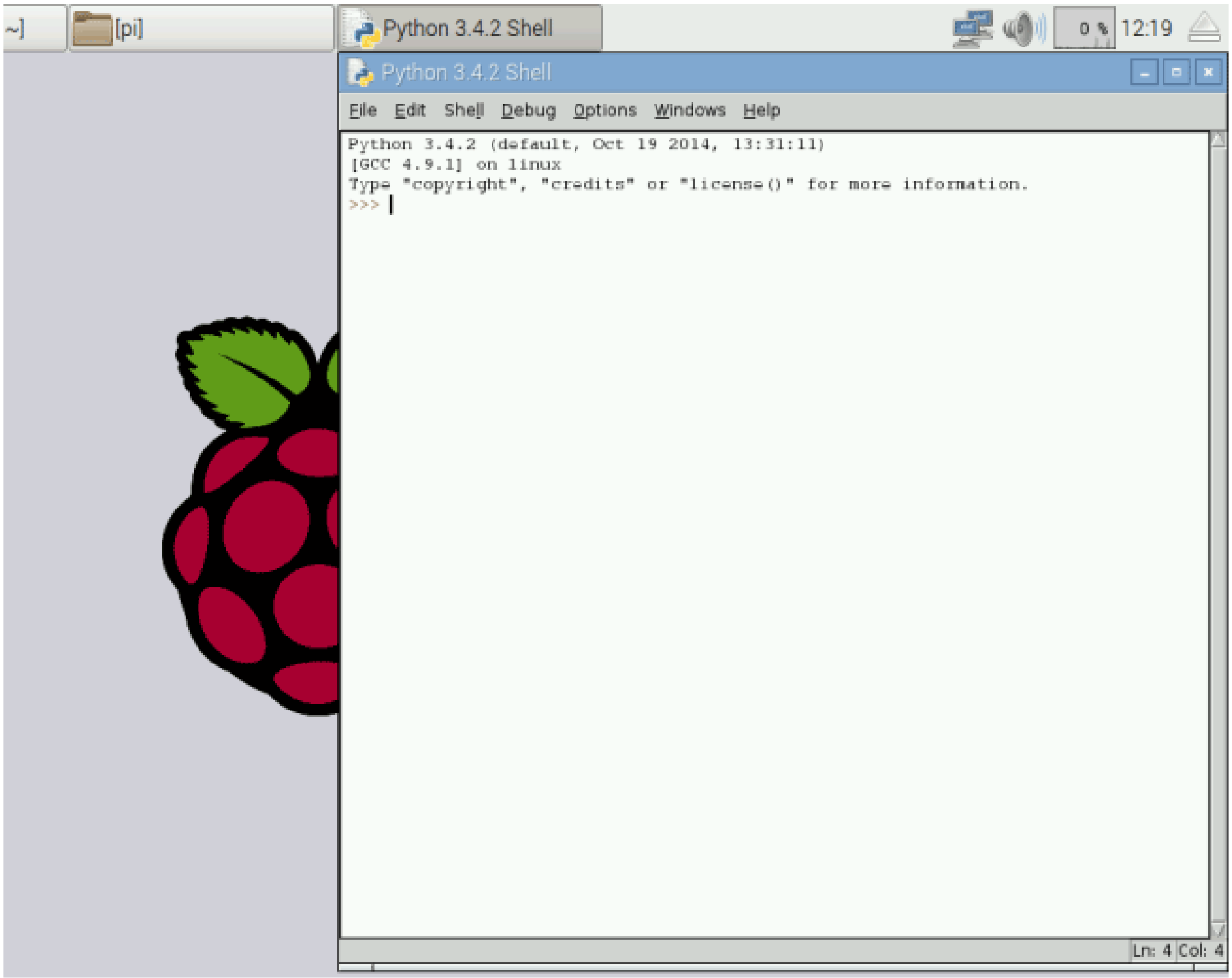
Working Explanation:

Since we have everything ready, turn ON your PI and go to the desktop.

1. On the desktop, go the Start Menu and choose for the PYTHON 3, as shown in figure below.

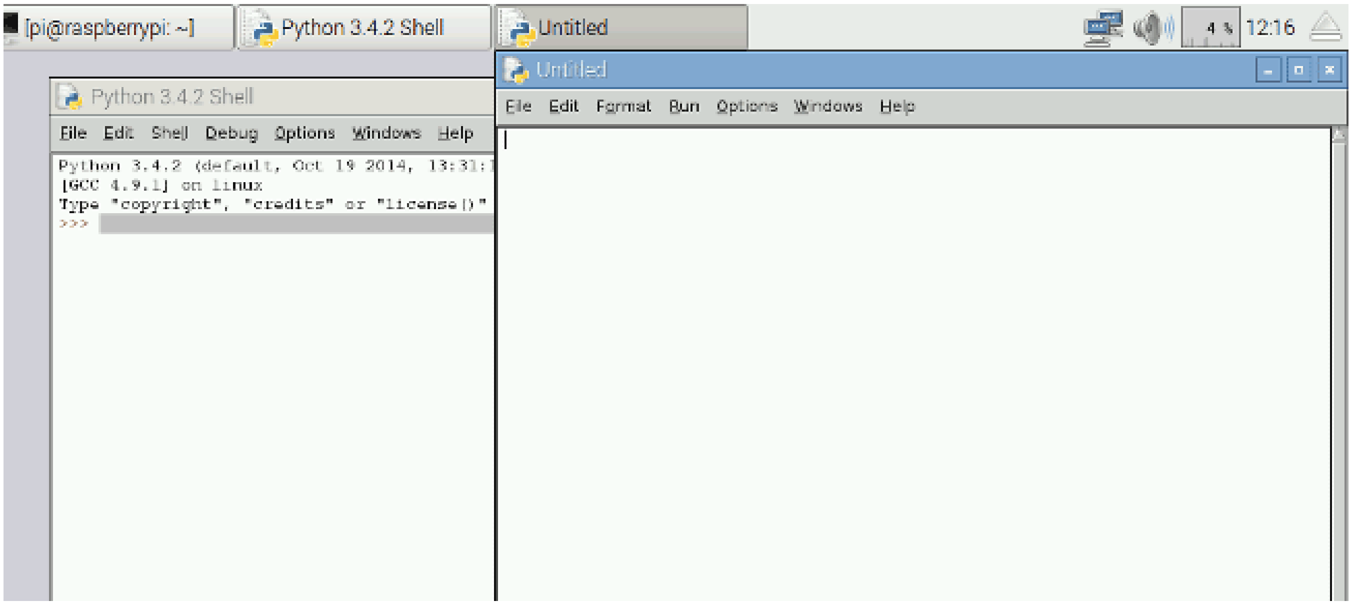


2. After that, PYHON will run and you will see a window as shown in below figure.



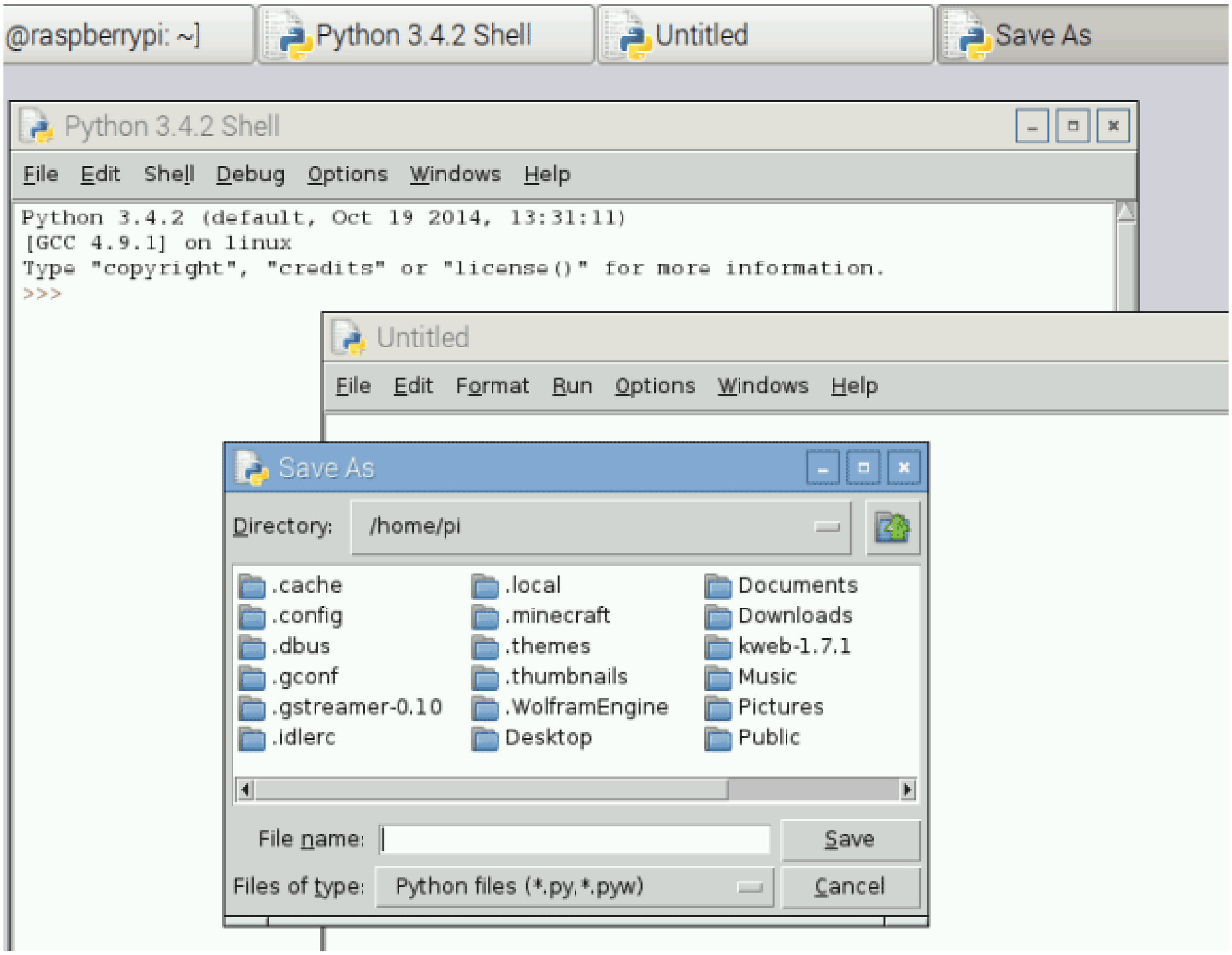
```
Python 3.4.2 (default, Oct 19 2014, 13:31:11)
[GCC 4.9.1] on linux
Type "copyright", "credits" or "license()" for more information.
>>> 
```

3. After that, click on *New File* in *File* Menu, You will see a new Window open,


[pi@raspberrypi: ~]
Python 3.4.2 Shell
Untitled
4 %
12:16
Untitled
File Edit Format Run Options Windows Help
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help

```
Python 3.4.2 (default, Oct 19 2014, 13:31:1
[GCC 4.9.1] on linux
Type "copyright", "credits" or "license()"
>>>
```

4. Save this file as blinky on the desktop,



```
Python 3.4.2 (default, Oct 19 2014, 13:31:11)
[GCC 4.9.1] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
```

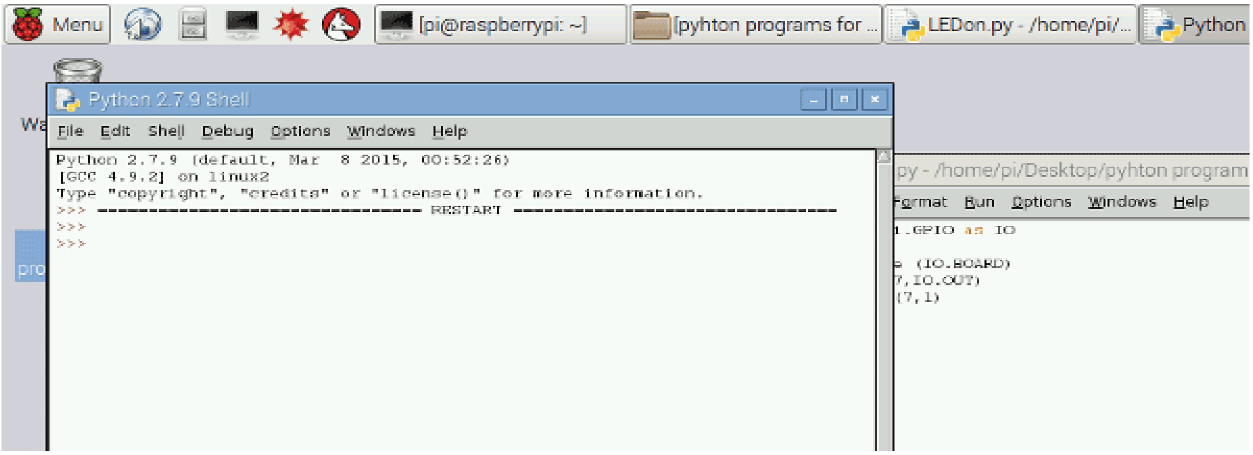
Directory: /home/pi

.cache	.local	Documents
.config	.minecraft	Downloads
.dbus	.themes	kweb-1.7.1
.gconf	.thumbnails	Music
.gstreamer-0.10	.WolframEngine	Pictures
.idlerc	Desktop	Public

File name:

Files of type: Python files (*.py,*.pyw)

5. After that write the program for blinky as given below and execute the program by clicking on "RUN" on 'DEBUG' option.


Menu
[pi@raspberrypi: ~]
[pyhton programs for ...
LEDon.py - /home/pi/...
Python
Python 2.7.9 Shell
File Edit Shell Debug Options Windows Help
py - /home/pi/Desktop/pyhton program
Format Run Options Windows Help

```
Python 2.7.9 (default, Mar  8 2015, 00:52:26)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
>>>
```

```
1.GPIO as IO

s (IO.BOARD)
7,IO.OUT)
(7,1)
```

If the program has no errors in it, you will see a ">>>", which means the program is executed successfully. By this time you should see the LED blinking three times. If there were any errors in the program, the execution tells to correct it. Once the error is corrected execute the program again.

We will see the PYTHON program Code for LED Blinking, in detail, below.

Code

```
import RPi.GPIO as IO            # calling header file for GPIO’s of PI
import time                          # calling for time to provide delays in program

IO.setmode (IO.BOARD)       # programming the GPIO by BOARD pin numbers, GPIO21 is
called as PIN40
IO.setup(40,IO.OUT)            # initialize digital pin40 as an output.
IO.output(40,1)                   # turn the LED on (making the voltage level HIGH)
time.sleep(1)                      # sleep for a second
IO.cleanup()                      # turn the LED off (making all the output pins LOW)
time.sleep(1)                     #sleep for a second

#loop is executed second time
IO.setmode (IO.BOARD)
IO.setup(40,IO.OUT)
IO.output(40,1)
time.sleep(1)
IO.cleanup()
time.sleep(1)

#loop is executed third time
IO.setmode (IO.BOARD)
IO.setup(40,IO.OUT)
IO.output(40,1)
time.sleep(1)
IO.cleanup()
time.sleep(1)
```