

the python pit™

With web.py you can present web forms on client browsers (we are using Chrome on Android or Safari on iOS) and accept the form content via HTTP Post back to the web server (your Raspberry Pi). The values returned are fed directly to your Python script, which in turn can then execute any commands that you require. Web.py contains an in-built webserver meaning that it is not necessary to install Apache or Lighttpd.

By way of an example we will use your smartphone's browser connected via WiFi to your local area network and in turn your Pi to perform basic remote control of LedBorg (www.piborg.com). LedBorg is a pre-assembled LED array capable of outputting 27 colours. However for this tutorial we will just demonstrate the principle with red, green and "black" (all LEDs off). The technique described below works equally well for any other Pi remote control project. Hence replace the LedBorg specific code to meet your own requirements.

We can install web.py from the command line via PIP (see The Python Pit in issue 8 for instructions on installing PIP):

```
sudo pip install web.py
```

Next create a directory (pyserver in this

ledborg.py

```
# web.py: controlling your Pi from a smartphone
# using LedBorg (www.piborg.com) as an example
# ColinD 27/12/2012

import web
from web import form
```

In this month's Python Pit we show you how to control your Raspberry Pi with your smartphone by using the web.py framework from www.webpy.org.

This article provides an alternative to using Vala, also covered in this issue.

example) on your Pi to act as the root of the web server that web.py will start. Inside create two other directories and and save the following files within:

```
pyserver/ledborg.py
pyserver/templates/index.html
pyserver/static/styles.css
```

We will concentrate on the first two files: ledborg.py contains our python code and index.html the template web page that will be called when the program executes. The stylesheet, styles.css is optional and changes the usual drab grey buttons found on web forms to be coloured and larger, as seen in the screenshot. The styles were generated at www.cssbuttongenerator.com.

Run the program on your Pi and navigate to the Pi's IP address on your smartphone, appending the port number 8080. This will present you with the web form, enabling commands to be sent to your Python script when you tap the buttons. In the example screenshot the smartphone connects to the Pi via `http://192.168.1.69:8080` - you can



```
PYTHON VERSION: 2.7.3rc2
PYGAME VERSION: 1.9.2a0
O.S.: Debian 7
```

TESTED!