

Setting up a USB drive

This step is optional; you don't need a USB drive but it's nice to have. The idea is that the USB drive will hold all of your websites files. The Pi runs off an SD card, and they tend to have small storage capacities and be slow. Best to leave that for the Operating System, and put all of your website files on a USB drive. That should work out more convenient, better performing, and cheaper than trying to get a big enough SD card to do all the work. I use a 32Gb USB flash drive at the moment as that's physically small, fast, cheap, and doesn't require an external power source (or powered USB hub). If I was to need to store a lot of data I could just swap to a more traditional USB HDD. Be aware if you want to do that; the Pi can only supply so much power over USB and a USB powered hard-disk isn't likely to work plugged into the Pi itself; you'd likely need a powered USB hub to supply enough juice to run the drive. Plug your drive into the Pi, then type:

```
sudo fdisk -l
```

You'll see a list of storage devices attached to the Pi; one is the SD card, the other is the drive you just plugged in. The SD card will be the one identified as `/dev/mmcblk0` and will likely have a number of 'partitions' listed under it. We are interested in the other one; for me that is `/dev/sda`, and it has one partition '`/dev/sda1`' yours will likely be the same, but check, and use your value in the following commands rather than mine. We're going to format the partition on the USB drive so Linux has a clean slate to use – this will erase any data on the card. Remember to use the value you got for your USB drive if it's different to mine:

```
sudo mkfs.ext4 /dev/sda1
```

The USB drive is now blank and in a Linux native filesystem format. Now we need to mount it (i.e., let Linux actually use it). First we create a mount point (a directory name we will access the drive from):

```
sudo mkdir /websites
```

and then we actually mount the drive onto that mount point:

```
sudo mount /dev/sda1 /websites
```

The drive is now available to the root user... but no one else has permission to access it. We can change that as follows:

```
sudo chgrp -R users /websites
```

Now any user belonging to the 'users' group can access the drive. But they can't write to it yet:

```
sudo chmod -R g+w /websites
```

Now they can. The last job is to set up auto-mounting. Right now, if you rebooted the Pi then the /websites directory would be inaccessible because the drive would need to be re-mounted. That's annoying, so we'll automate that:

```
sudo nano /etc/fstab
```

You'll see a somewhat complicated looking file. We just need to add a new line to it at the bottom and separate each item on the line with a tab – be sure to press the tab key where you see [tab] instead of writing the phrase [tab]

```
/dev/sda1 [tab] /websites [tab] ext4 [tab] defaults [tab] 0 [tab]  
2
```

Press Ctrl+X to exit, Y then Enter to save. Done.