For the Raspberry Pi Introduction, we had several Raspberry Pi’s attached to switches, with one of them set up as a file and DHCP server. (We had the first step be to actually plug in the pi’s and turn them on, so initially everything was laid out but not connected.)

Additionally, the pi’s all had two scripts in the pi user’s home directory, checkip.sh and desktop.sh. checkip.sh asks for the IP address and compares it to their actual IP address, telling the user whether or not they have the correct address. desktop.sh asks for a password and then runs startx.

Please check these scripts ahead of time! Future updates to Raspbian may break them.

(**Note**: Our setup relied on monitors using VGA, so we have the extra parts and steps of hooking up a HDMI to VGA adapter. Please modify the instructions if you do not have such limitations.)

INSERT SETUP INFO HERE FOR FILE SERVER AND DHCP SERVER

(**Note**: You might want to change the password used. :P)

When we had students actually following the instructional handout, we had at least one instructor per 2 students (we actually had about a 1 to 1 ratio because of small group size). Each instructor helped a student with any questions they had, or problems with our setup (we made a few mistakes in our rush to set everything up).

Before beginning, we introduced ourselves and played a video from the Raspberry Pi Foundation’s website introducing the pi.

On step 1, answer any questions they may have, and/or explain what certain cables do (for instance, explaining that an Ethernet cable is used for connecting to networks, or going online).

On step 3, take a moment to explain what an IP address is (we used the analogy that it’s like a home address, how do you know where to go? You get an address).

(Additionally, perhaps explain why commands are just words, and ./checkip.sh is so different (that it’s a script and it needs to be told where it is, and that you tell it “it’s right here” by putting ./).)

On step 4, further explain the relation between IP addresses and the pi’s (and what the output of ping is showing). Also at this point, ping will be just going on and on. It’s a good point to tell them about how to quit a command by pressing Ctrl+C.

On step 5, perhaps explain what sudo actually means (SuperUser do), and what it is used for (administrative access without having to log into another account (keep it simple)).

(**Note**: In our workshop, the students tended to not quite make the connection that these words they were typing were commands and arguments *to* those commands. Try to explain and reiterate this.)

(On step 7, perhaps mention hostnames, or if you set up different hostnames for your pi’s, show how that changes?)

On step 9, explain how wget and a web browser are similar (downloading a file from a server over HTTP / the World Wide Web) and how they are different (it saves a file instead of displaying it). Also explain how 10.1.1.1 is the address of the server, and the forward slash means to start looking in the server, and password is the file you’re getting from the server.

A few extra thoughts:

1. Perhaps expand this to cover a few more basics: What is a file? Basic file management commands (ls, mv, rm, cp). This is pretty fundamental, but we just didn’t have time to go into it more.
2. What is a command? What are arguments? Some more examples of running commands to do something interesting would be nice. (Note: fortune and cowsay are pretty entertaining commands! If you haven’t heard of them, look them up!)
3. Specifically with fortune / cowsay, one can explain piping input between commands in an interesting way. Expand on this by showing how one can make a file (echo > file perhaps? nano file?) and then pipe it to cowsay (cat file | cowsay). (Also, tac!)