Today we’re going to get some hands on experience with the Raspberry Pi. This group is about physical computing.
What is Physical Computing

Installing Putty

We’re going to connect to our Raspberry Pi using SSH. Secure Shell, or SSH, is an encrypted network protocol to allow remote login and other network services to operate securely over an insecure network. Common applications include remote command-line login and remote command execution.

The SSH client we are going to use is called Putty. It’s available here. http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
Raspberry Pi advantages

To connect to our Raspberry Pi we need to find out what its IP Address is. To do this we can use a network scanner such as Advanced IP Scanner or Fing.

Once we have our IP Address we enter it into Putty then click open.

The login is pi and the password is raspberry. When we get something like this we are ready to go.

```
login as: pi
pi@192.168.1.201's password:
```
Our First Device
Raspberry Pi GPIO

These pins are a physical interface between the Pi and the outside world. At the simplest level, you can think of them as switches that you can turn on or off (input) or that the Pi can turn on or off (output). Seventeen of the 26 pins are GPIO pins; the others are power or ground pins.
What is a Resistor

https://www.youtube.com/watch?v=Gc1wVdbVI0E
Programming from the command line
In putty type the following

```
pi@raspberrypi - $ nano hello.py
```

This creates a file called hello.py and opens it in the Nano text editor.

Type the following

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

Press ctrl and x

Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
Y Yes
N No
^C Cancel

Press y
Press ENTER

In the command line type python hello.py

pi@raspberrypi ~ $ python hello.py
Hello, World
Blinking a Led

Type nano led.py
In Nano enter the following code

When finished press CTRL x to exit
Then press y, then enter to save it

We will test our code after every change

```python
#!/usr/bin/python3
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

led = 4
GPIO.setup(led, GPIO.OUT)

GPIO.output(led, 1)
time.sleep(5)
GPIO.output(led, 0)
GPIO.cleanup()
```
Next Session

I will recap what we covered today and go into more detail as I skimmed over some important details.

We will also look at setting up a remote desktop so we won’t be stuck with the command line all the time.

If you have a Raspberry pi yourself bring it along and I’ll help you get it set up.