Accelerometeres don't 'naturally' spit out a "meters per second squared" value. Instead, they give you a raw value. In this sensor's case, its a number ranging from -32768 and 32767 (a full 16-bit range). Depending on the sensor range, this number scales between the min and max of the range. E.g. if the accelerometer is set to **+-2g** then 32767 is +2g of force, and -32768 is -2g. If the range is set to **+-16g**, then those two number correlate to +16g and -16g respectively. So knowing the range is key to deciphering the data!

lis.setRange(LIS3DH\_RANGE\_4\_G); // 2, 4, 8 or 16 G!

Serial.print("Range = "); Serial.print(2 << lis.getRange());

In the first line, you can use **LIS3DH\_RANGE\_2\_G**, **LIS3DH\_RANGE\_4\_G**, **LIS3DH\_RANGE\_8\_G**, or **LIS3DH\_RANGE\_16\_G**

When reading the range back from the sensor, **0** is ±2g, **1**is ±4g, **2** is ±8g and **3** is ±16g range

Raw data readings

You can get these raw readings by calling lis.read() which will take a snapshot at that moment in time. You can then grab the x, y and z data by reading the signed 16-bit values from lis.x, lis.y and lis.z. When you are done with that data, call read() again to get another snapshot.

Normalized readings

If you dont want to noodle around with range readings and scalings, you can use **Adafruit\_Sensor** to do the normalization for you. Its a nice way to have consistant readings betwixt multiple types of accelerometers.

It's up to you whether you want the raw numbers or normalized data - there's times you want to keep it simple and avoid floating point numbers, and other times you may want to avoid doing the math for force conversion.

https://github.com/adafruit/Adafruit\_LIS3DH/blob/master/Adafruit\_LIS3DH.cpp