In this exercise you'll try to build a neural network that predicts the price of a house according to a simple formula.

So, imagine if house pricing was as easy as a house costs 50k + 50k per bedroom, so that a 1 bedroom house costs 100k, a 2 bedroom house costs 150k etc.

How would you create a neural network that learns this relationship so that it would predict a 7 bedroom house as costing close to 400k etc.

Hint: Your network might work better if you scale the house price down. You don't have to give the answer 400...it might be better to create something that predicts the number 4, and then your answer is in the 'hundreds of thousands' etc.

In [2]:

```
import tensorflow as tf
import numpy as np
from tensorflow import keras
```

In [15]:

```
# GRADED FUNCTION: house_model
def house_model(y_new):
    xs = np.array([1,2,3,4,5,6])# Your Code Here#
    ys = np.array([1,1.5,2,2.5,3,3.5])# Your Code Here#
    model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])# Yo
    model.compile(optimizer='sgd', loss='mean_squared_error')# Your Code Here#
    model.fit(xs, ys, epochs=200)# Your Code here#)
    return model.predict(y_new)[0]
```

In [16]:

```
prediction = house_model([7.0])
print(prediction)
```

```
6/6 [==============================] - 0s 362us/sample - loss: 0.004
0
Epoch 146/200
6/6 [==============================] - 0s 397us/sample - loss: 0.004
0
Epoch 147/200
6/6 [==============================] - 0s 291us/sample - loss: 0.003
9
Epoch 148/200
6/6 [==============================] - 0s 278us/sample - loss: 0.003
9
Epoch 149/200
6/6 [==============================] - 0s 14ms/sample - loss: 0.0039
Epoch 150/200
6/6 [==============================] - 0s 253us/sample - loss: 0.003
9
Epoch 151/200
6/6 [==============================] - 0s 338us/sample - loss: 0.003
8
Epoch 152/200
```

In [4]:

```
# Now click the 'Submit Assignment' button above.
# Once that is complete, please run the following two cells to save your work and c
```

In [ ]:

```
%%javascript
<!-- Save the notebook -->
IPython.notebook.save_checkpoint();
```

In [ ]:

```
%%javascript
<!-- Shutdown and close the notebook -->
window.onbeforeunload = null
window.close();
IPython.notebook.session.delete();
```