Imagine you want to represent transformations using matrices. Points could be stored as

$$\begin{bmatrix} x \\ y \end{bmatrix}$$
[xy]

and you could represent a rotation as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$
[uv]=[cos(θ)−sin(θ)sin(θ)cos(θ)][xy]

and scaling as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} k1 & 0 \\ 0 & k2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$
[uv]=[k100k2][xy]

These are known as linear transformations and they allow us to do transformations as matrix multiplications. But notice that you cannot do translations as a matrix multiplication. Instead you have to do

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} s \\ t \end{bmatrix}$$
[uv]=[xy]+[st]

This is known as an affine transformation. However this is undesirable (computationally).

Let R and S be rotation and scaling matrices and T be a translation vector. In computer graphics, you may need to do a series of translations to a point. You could imagine how tricky this could get.

Scale, translate, then rotate and scale, then translate again:

$$p' = SR(Sp + T) + T$$
p′=SR(Sp+T)+T

Not too bad but imagine you had do this computation on a million points. What we would like is to represent rotation, scaling, and translation all as matrix multiplications. Then those matrices can be pre multiplied together for a single transformation matrix which is easy to do computations with.

Scale, translate, then rotate and scale, then translate again:

$$M = TSRTS$$
M=TSRTS

$$p' = Mp$$
p′=Mp

We can achieve this by adding another coordinate to our points. I'm going to show all this for 2D graphics (3D points) but you could extend all this to 3D graphics (4D points).

$$p = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
p=[xy1]

Rotation matrix:

$$R = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

R=[cos(θ)−sin(θ)0sin(θ)cos(θ)0001]

Scale matrix:

$$S = \begin{bmatrix} k1 & 0 & 0 \\ 0 & k2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

S=[k1000k20001]

Translation matrix:

$$T = \begin{bmatrix} 1 & 0 & t1 \\ 0 & 1 & t2 \\ 0 & 0 & 1 \end{bmatrix}$$

T=[10t101t2001]

You should work out some examples to convince yourself that these do in fact give you the desired transformation and that you can compose a series of transformations by multiplying multiple matrices together.

You could go further and allow the extra coordinate to take on any value.

$$p = \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

p=[xyw]

and say this this homogeneous (x, y, w) coordinate represents the euclidean (x, y) coordinate at (x/w, y/w). Normally you cannot do division using matrix transformations, however by allowing w to be a divisor, you can set w to some value (through a matrix multiplication) and allow it to represent division. This is useful for doing projection because (in 3D) you will need to divide the x and y coordinates by -z (in a right handed coordinate system). You can achieve this by simply setting w to -z using the following projection matrix:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$