In order to raise a matrix to some power, use the MatrixPower command. Here is the syntax, along with a couple of examples.

In[1]:=
```
? MatrixPower
```

Out[1]=

Symbol

MatrixPower[m, n] gives the n^{th} matrix power of the matrix m.

MatrixPower[m, n, v] gives the n^{th} matrix power of the matrix m applied to the vector v.

In[2]:=
```
a = {{1, 2, 1}, {-1, 2, 0}, {0, 4, 2}};
MatrixForm[a]
```

Out[3]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 1 \\ -1 & 2 & 0 \\ 0 & 4 & 2 \end{pmatrix}$$

In[4]:=
```
MatrixForm[a.a]
MatrixForm[MatrixPower[a, 2]]
MatrixForm[a.a.a]
MatrixForm[MatrixPower[a, 3]]
MatrixForm[MatrixPower[a, 20]]
```

Out[4]//MatrixForm=

$$\begin{pmatrix} -1 & 10 & 3 \\ -3 & 2 & -1 \\ -4 & 16 & 4 \end{pmatrix}$$

Out[5]//MatrixForm=

$$\begin{pmatrix} -1 & 10 & 3 \\ -3 & 2 & -1 \\ -4 & 16 & 4 \end{pmatrix}$$

Out[6]//MatrixForm=

$$\begin{pmatrix} -11 & 30 & 5 \\ -5 & -6 & -5 \\ -20 & 40 & 4 \end{pmatrix}$$

Out[7]//MatrixForm=

$$\begin{pmatrix} -11 & 30 & 5 \\ -5 & -6 & -5 \\ -20 & 40 & 4 \end{pmatrix}$$

Out[8]//MatrixForm=

$$\begin{pmatrix} 424\,004\,599 & -124\,592\,230 & 185\,511\,743 \\ -185\,511\,743 & 609\,516\,342 & 123\,903\,929 \\ 495\,615\,716 & 246\,431\,256 & 361\,708\,484 \end{pmatrix}$$

Here is the transformation matrix “trans” that we used when discussing the Markov chain for the coin flip game (discussed on p. 301 of the textbook).

In[9]:=
```
trans = {{1, .5, 0, 0, 0, 0}, {0, 0, .5, 0, 0, 0}, {0, .5, 0, .5, 0, 0}, {0, 0, .5, 0, .5, 0}, {0, 0, 0, .5, 0, 0}, {0, 0, 0, 0, .5, 1}};
```

```
MatrixForm[trans]
```

Out[10]//MatrixForm=

$$\begin{pmatrix} 1 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 1 \end{pmatrix}$$

In[11]:=
```
MatrixForm[
 MatrixPower[trans, 5] ]
```

Out[11]//MatrixForm=

$$\begin{pmatrix} 1. & 0.6875 & 0.375 & 0.21875 & 0.0625 & 0. \\ 0. & 0. & 0.15625 & 0. & 0.09375 & 0. \\ 0. & 0.15625 & 0. & 0.25 & 0. & 0. \\ 0. & 0. & 0.25 & 0. & 0.15625 & 0. \\ 0. & 0.09375 & 0. & 0.15625 & 0. & 0. \\ 0. & 0.0625 & 0.21875 & 0.375 & 0.6875 & 1. \end{pmatrix}$$

We can use the Manipulate command to easily see how things change as some paramter changes. Below is the documentation, plus the example of the transformation matrix raised to some power.

In[12]:=
```
? Manipulate
```

Out[12]=

Symbol

Manipulate[*expr*, {*u*, u_{min}, u_{max}}] generates a version of *expr*
with controls added to allow interactive manipulation of the value of *u*.

Manipulate[*expr*, {*u*, u_{min}, u_{max}, *du*}] allows the value of *u* to vary between u_{min} and u_{max} in steps *du*.

Manipulate[*expr*, {{*u*, u_{init}}, u_{min}, u_{max}, ...}] takes the initial value of *u* to be u_{init}.

Manipulate[*expr*, {{*u*, u_{init}, u_{lbl}}, ...}] labels the controls for *u* with u_{lbl}.

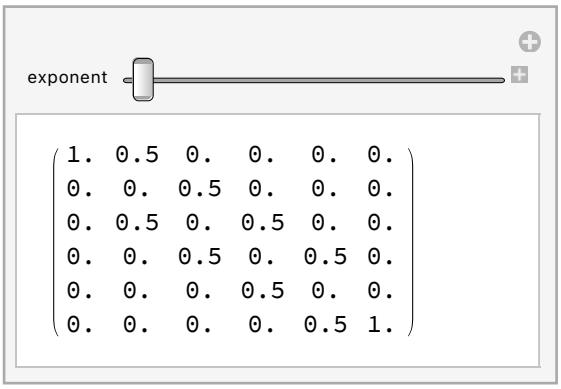
Manipulate[*expr*, {*u*, {u_1, u_2, ...}}] allows *u* to take on discrete values u_1, u_2, ...

Manipulate[*expr*, {*u*, ...}, {*v*, ...}, ...] provides controls to manipulate each of the *u*, *v*, ...

Manipulate[*expr*, $c_u \to$ {*u*, ...}, $c_v \to$ {*v*, ...}, ...] links
the controls to the specified controllers on an external device.

In[13]:=
```
Manipulate[MatrixForm[
  MatrixPower[trans, exponent] ], {exponent, 1, 50, 1}]
```

Out[13]=



$$\begin{pmatrix} 1. & 0.5 & 0. & 0. & 0. & 0. \\ 0. & 0. & 0.5 & 0. & 0. & 0. \\ 0. & 0.5 & 0. & 0.5 & 0. & 0. \\ 0. & 0. & 0.5 & 0. & 0.5 & 0. \\ 0. & 0. & 0. & 0.5 & 0. & 0. \\ 0. & 0. & 0. & 0. & 0.5 & 1. \end{pmatrix}$$

It seems the best way to display numbers to a fixed number of decimal places is to use the NumberForm command, specifically the second example listed in the documentation below. (Note that it behaves a bit unexpectedly when it comes to very small numbers and scientitfic notation, as you will see in examples.)

In[14]:=
```
? NumberForm
```

Out[14]=

Symbol

NumberForm[*expr*, *n*] prints with approximate real numbers in *expr* given to *n*-digit precision.

NumberForm[*expr*, {*n*, *f*}] prints with approximate real numbers having *n* digits, with *f* digits to the right of the decimal point.

NumberForm[*expr*] prints using the default options of NumberForm.

In[15]:=
```
NumberForm[MatrixForm[
  MatrixPower[trans, 5] ], {10, 3}]
```

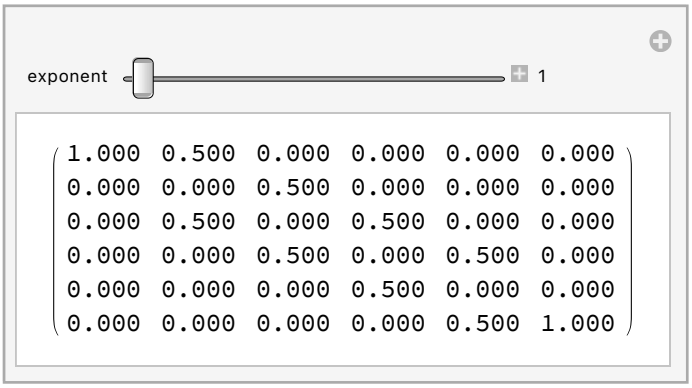
Out[15]//NumberForm=

$$\begin{pmatrix} 1.000 & 0.688 & 0.375 & 0.219 & 0.063 & 0.000 \\ 0.000 & 0.000 & 0.156 & 0.000 & 0.094 & 0.000 \\ 0.000 & 0.156 & 0.000 & 0.250 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.250 & 0.000 & 0.156 & 0.000 \\ 0.000 & 0.094 & 0.000 & 0.156 & 0.000 & 0.000 \\ 0.000 & 0.063 & 0.219 & 0.375 & 0.688 & 1.000 \end{pmatrix}$$

This shows our transformation matrix raised to a power, and we have modified the number format and made it so that the exponent also appears to the right of the slider bar.

In[16]:=
```
Manipulate[NumberForm[MatrixForm[
    MatrixPower[trans, exponent] ], {4, 3}],
 {exponent, 1, 60, 1, Appearance → "Labeled"}]
```

Out[16]=



$$\begin{pmatrix} 1.000 & 0.500 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.500 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.500 & 0.000 & 0.500 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.500 & 0.000 & 0.500 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.500 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.500 & 1.000 \end{pmatrix}$$