```
In [1]: # Python har tre datatyper som representerer tall: Heltall, desimaltall,
   ...: og komplekse tall.

In [2]: # Heltallsdivisjon:

In [3]: 3/2
Out[3]: 1

In [4]: # Vanlig divisjon:

In [5]: 3.0/2
Out[5]: 1.5

In [6]: # For å arbeide med matriser og vektorer numerisk bruker vi modulen

In [7]: # numpy:

In [8]: import numpy as np

In [9]: # Vi definerer to vektorer:

In [10]: v=np.array([1.,2.,3.])

In [11]: w=np.array([0.,-1.,1.])

In [12]: # Komponentene nummereres fra 0 og oppover:

In [13]: v[0]
Out[13]: 1.0

In [14]: # Skalarprodukt:

In [15]: np.dot(v,w)
Out[15]: 1.0

In [16]: # Kommandoen v*w gir derimot komponentvis produkt:

In [17]: v*w
Out[17]: array([ 0., -2.,  3.])

In [18]: # Vi definerer en matrise:

In [19]: A=np.array([[1.,2.,3.],[4.,5.,6.],[7.,8.,9.]])

In [20]: # Første rad:

In [21]: A[0]
Out[21]: array([ 1.,  2.,  3.])

In [22]: # Alternativt for første rad:

In [23]: A[0,:]
Out[23]: array([ 1.,  2.,  3.])

In [24]: # Tredje søyle:

In [25]: A[:,2]
Out[25]: array([ 3.,  6.,  9.])

In [26]: # Tredje element på første rad:

In [27]: A[0,2]
Out[27]: 3.0

In [28]: # Den transponerte:

In [29]: B=A.T

In [30]: # Viser resultatet:
```

```
In [31]: B
Out[31]:
array([[ 1.,  4.,  7.],
       [ 2.,  5.,  8.],
       [ 3.,  6.,  9.]])

In [32]: # Matrise-vektor-produkt:

In [33]: np.dot(A,v)
Out[33]: array([ 14.,  32.,  50.])

In [34]: # Her ble v automatisk tolket som en søylevektor. Skriver man isteden

In [35]: np.dot(v,A)
Out[35]: array([ 30.,  36.,  42.])

In [36]: # blir v tolket som radvektor.

In [37]: # Matrise-matrise-produkt:

In [38]: np.dot(B,A)
Out[38]:
array([[  66.,   78.,   90.],
       [  78.,   93.,  108.],
       [  90.,  108.,  126.]])

In [39]: # Kommandoen B*A gir isteden elementvis produkt:

In [40]: A*B
Out[40]:
array([[  1.,   8.,  21.],
       [  8.,  25.,  48.],
       [ 21.,  48.,  81.]])

In [41]: # Identitetsmatrisen:

In [42]: np.eye(3)
Out[42]:
array([[ 1.,  0.,  0.],
       [ 0.,  1.,  0.],
       [ 0.,  0.,  1.]])

In [43]: # Den inverse matrisen:

In [44]: np.linalg.inv(A)
Out[44]:
array([[ -4.50359963e+15,   9.00719925e+15,  -4.50359963e+15],
       [  9.00719925e+15,  -1.80143985e+16,   9.00719925e+15],
       [ -4.50359963e+15,   9.00719925e+15,  -4.50359963e+15]])

In [45]: # Svaret tyder på at A ikke er inverterbar, og at det er gjort en

In [46]: # avrundingsfeil. Vi sjekker determinanten:

In [47]: np.linalg.det(A)
Out[47]: 6.6613381477509402e-16

In [48]: # Siden elementene i A matematisk sett er hele tall, må determinanten

In [49]: # være 0, så A er ikke inverterbar.

In [50]: # Vi beregner egenverdier og egenvektorer:

In [51]: egenverdier, egenvektorer = np.linalg.eig(A)

In [52]: egenverdier
Out[52]: array([  1.61168440e+01,  -1.11684397e+00,  -1.30367773e-15])
```

```
In [53]: egenvektorer
Out[53]:
array([[-0.23197069, -0.78583024,  0.40824829],
       [-0.52532209, -0.08675134, -0.81649658],
       [-0.8186735 ,  0.61232756,  0.40824829]])

In [54]: # Den tredje potens av A:

In [55]: np.linalg.matrix_power(A,3)
Out[55]:
array([[  468.,   576.,   684.],
       [ 1062.,  1305.,  1548.],
       [ 1656.,  2034.,  2412.]])
```