Kernel: SageMath 10.3

NAME : SUMIT HELONDE

ROLL NO : 58

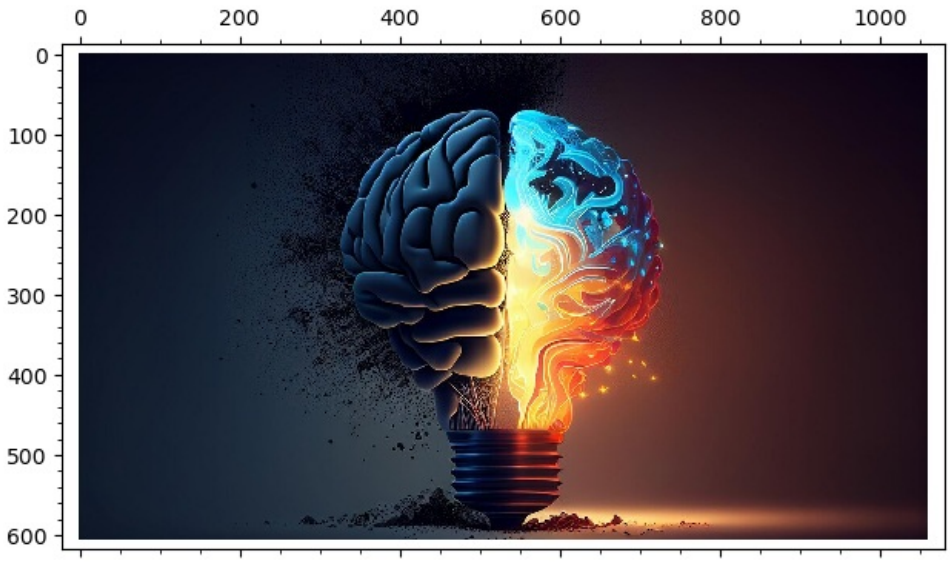
EXPERIMENT NO : 6

AIM : To perform Singular Value Decomposition with SageMath and it's application .

In [3]:
```
from matplotlib.pyplot import imread
import pylab
import numpy as np
img = pylab.imread('brain-light-bulb.png')
```

In [4]:
```
matrix_plot(img)
```

Out[4]:

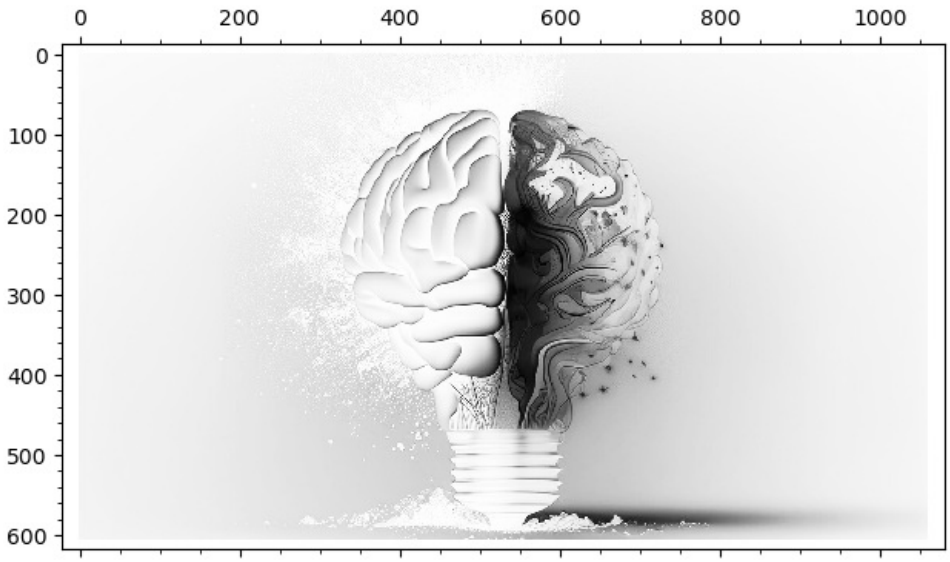


In [5]:
```
img.shape # 461 rows , 637 columns, 3 colors (blue, green, red)
```

Out[5]: (606, 1060, 3)

In [6]:
```
gray = lambda rgb : np.dot(rgb[... , :3] , [0.299 , 0.587, 0.114])
gray_img = gray(img)
matrix_plot(gray_img)
```

Out[6]:



In [7]:
```
gray_img.shape
```

Out[7]: (606, 1060)

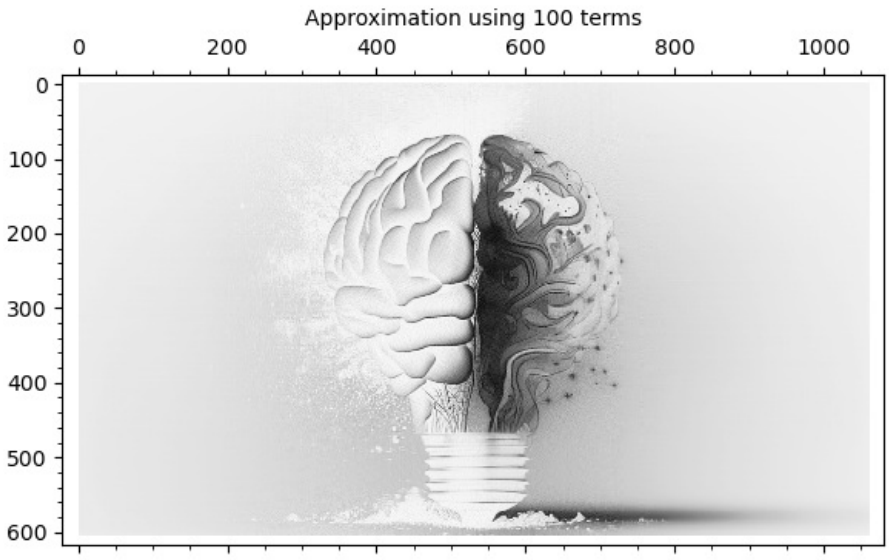
In [8]:
```
U,S,V = matrix(gray_img).SVD()
```

In [9]:
```
U.dimensions(),S.dimensions(),V.dimensions()
```

Out[9]:
```
((606, 606), (606, 1060), (1060, 1060))
```

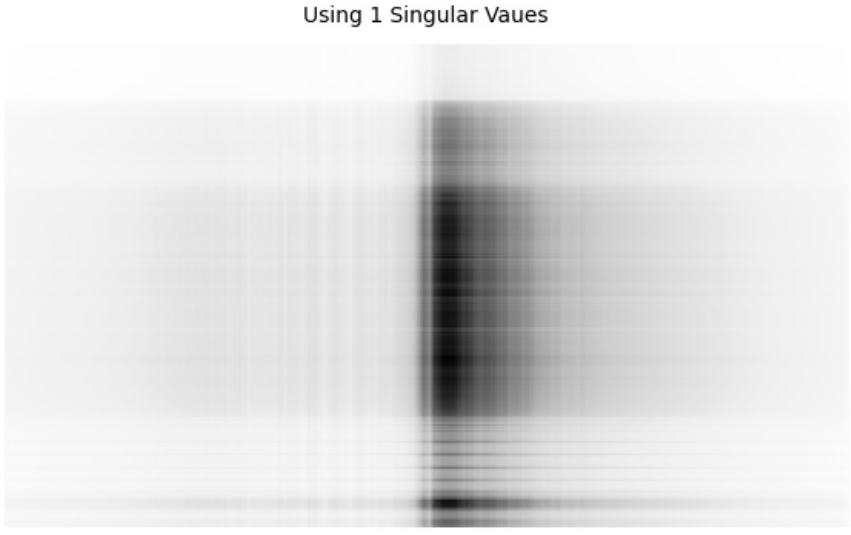
In [10]:
```
n=100
A_approx = U[:,:n]*S[:n,:n]*V.T[:n,:]
#print('Approximation using '+str(n)+ ' terms')
svd_img1=matrix_plot(A_approx,figsize=6,title='Approximation using '+str(n)+' terms')
svd_img1
```

Out[10]:

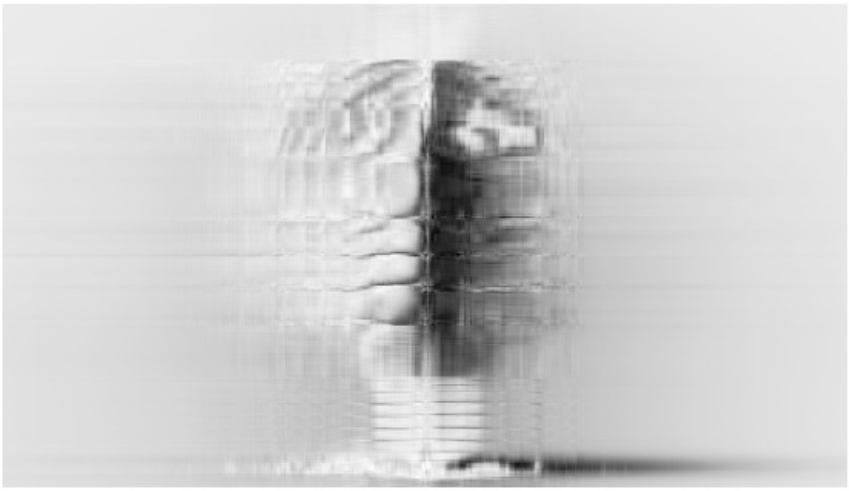


In [12]:
```
appx =[]
for i in range(1,100,10):
 A_approx = U[:,:i]*S[:i,:i]*V.T[:i,:]
 appx_img = matrix_plot(A_approx, title="Using "+str(i)+' Singular Vaues', frame=False)
 show(appx_img,figsize=6)
```

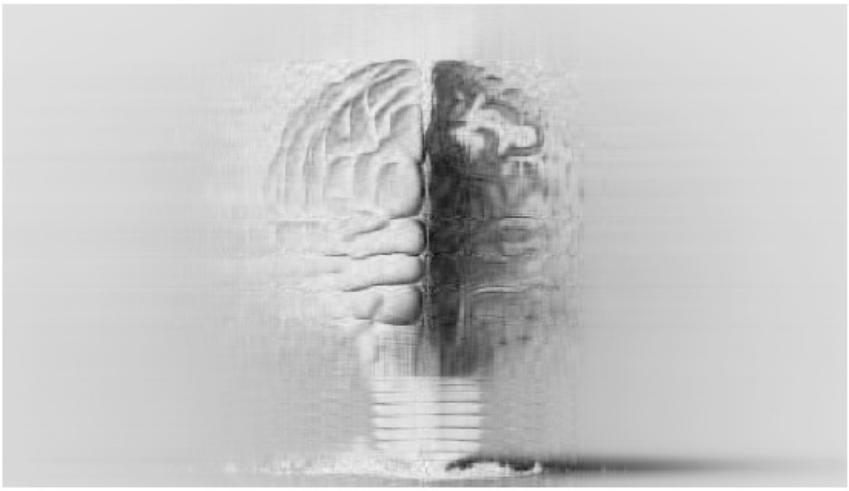
Out[12]:



Using 11 Singular Vaues



Using 21 Singular Vaues



Using 31 Singular Vaues

Using 41 Singular Vaues

Using 51 Singular Vaues

Using 61 Singular Vaues

Using 71 Singular Vaues

Using 81 Singular Vaues

Using 91 Singular Vaues

Conclusion: Singular Value Decomposition is successfully performed by means of its application namely dimensionality reduction.