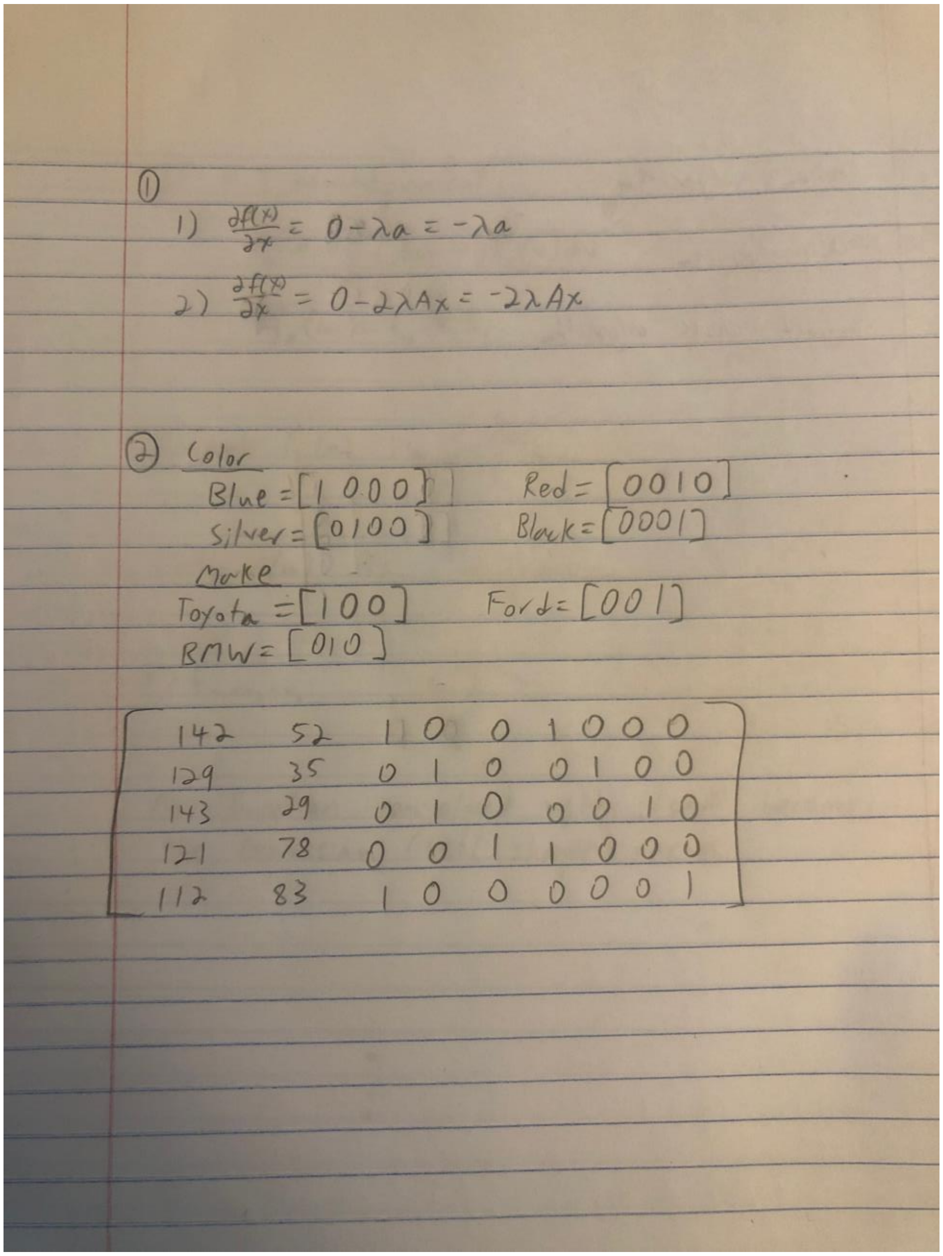


①

1) $\frac{\partial f(x)}{\partial x} = 0 - \lambda a = -\lambda a$

2) $\frac{\partial f(x)}{\partial x} = 0 - 2\lambda A x = -2\lambda A x$

② Color

Blue = [1 0 0 0] Red = [0 0 1 0]

Silver = [0 1 0 0] Black = [0 0 0 1]

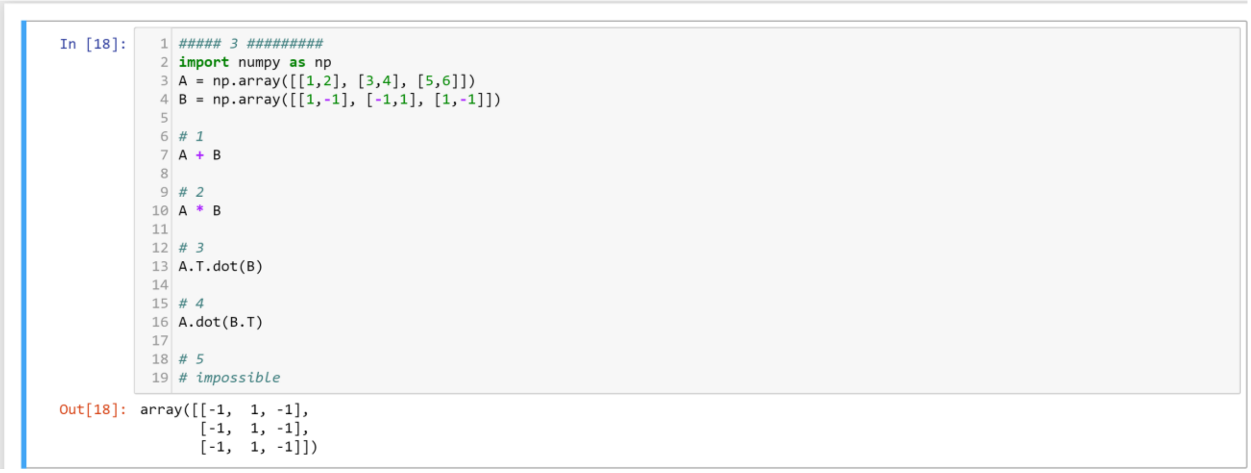
Make

Toyota = [1 0 0] Ford = [0 0 1]

BMW = [0 1 0]

$$\begin{bmatrix} 142 & 52 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 129 & 35 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 143 & 29 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 121 & 78 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 112 & 83 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

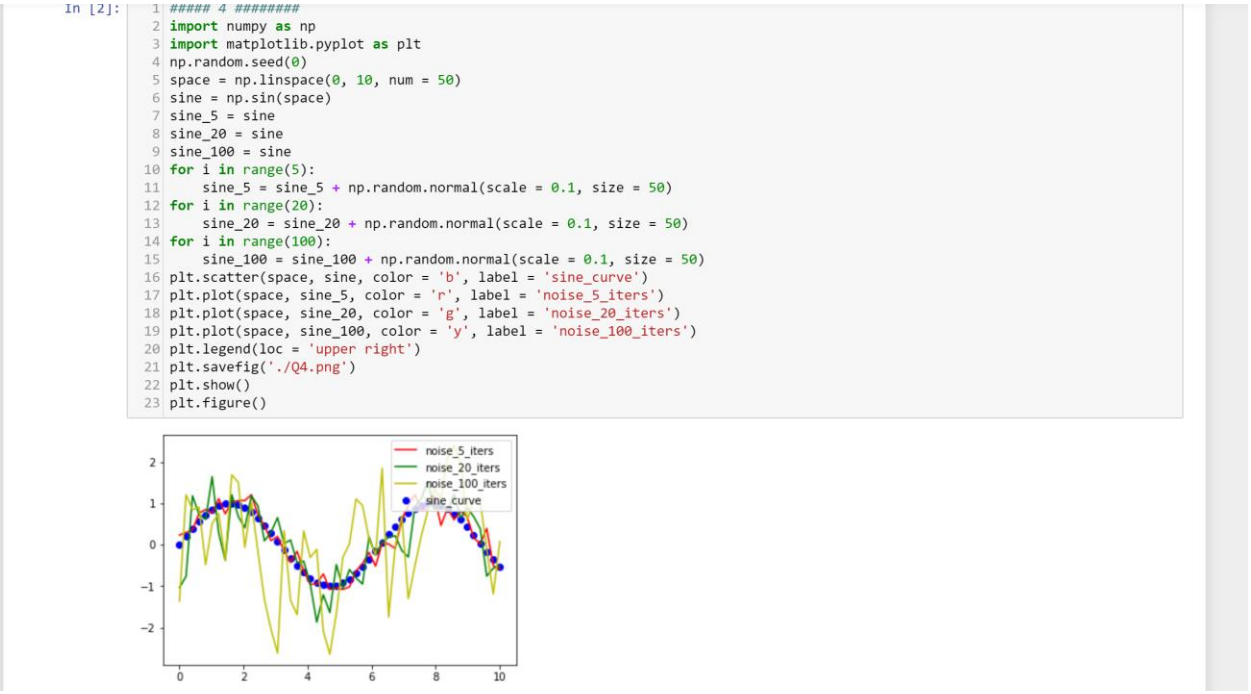
3)



```
In [18]:
##### 3 #########
import numpy as np
A = np.array([[1,2], [3,4], [5,6]])
B = np.array([[1,-1], [-1,1], [1,-1]])

# 1
A + B

# 2
A * B

# 3
A.T.dot(B)

# 4
A.dot(B.T)

# 5
# impossible
```

```
Out[18]: array([[-1,  1, -1],
                [-1,  1, -1],
                [-1,  1, -1]])
```

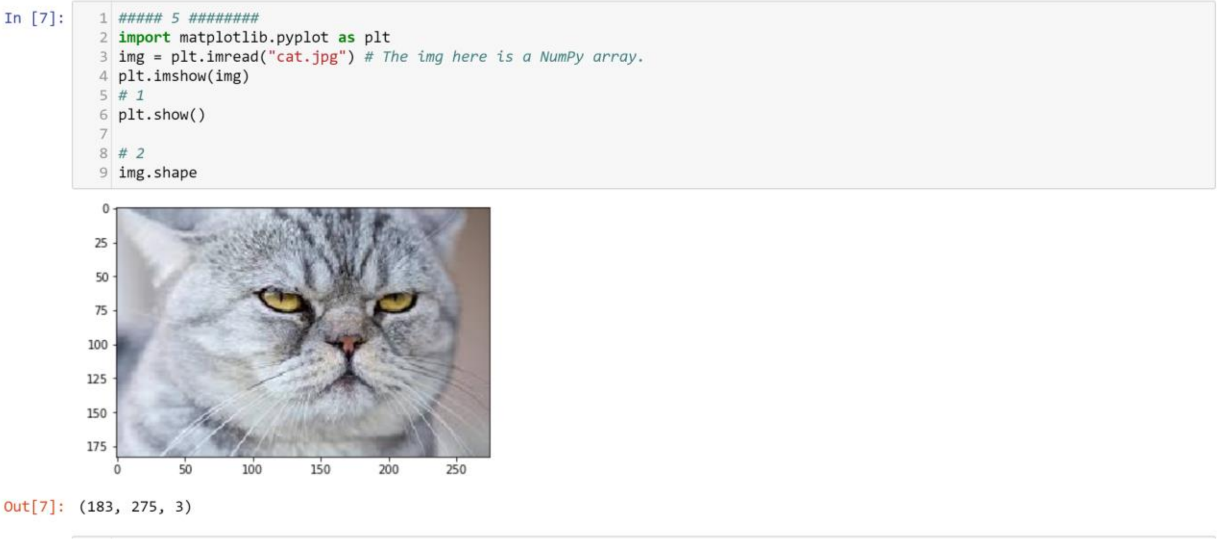
4)



In [2]:

```
##### 4 ########
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
space = np.linspace(0, 10, num = 50)
sine = np.sin(space)
sine_5 = sine
sine_20 = sine
sine_100 = sine
for i in range(5):
    sine_5 = sine_5 + np.random.normal(scale = 0.1, size = 50)
for i in range(20):
    sine_20 = sine_20 + np.random.normal(scale = 0.1, size = 50)
for i in range(100):
    sine_100 = sine_100 + np.random.normal(scale = 0.1, size = 50)
plt.scatter(space, sine, color = 'b', label = 'sine_curve')
plt.plot(space, sine_5, color = 'r', label = 'noise_5_iters')
plt.plot(space, sine_20, color = 'g', label = 'noise_20_iters')
plt.plot(space, sine_100, color = 'y', label = 'noise_100_iters')
plt.legend(loc = 'upper right')
plt.savefig('./Q4.png')
plt.show()
plt.figure()
```

5)



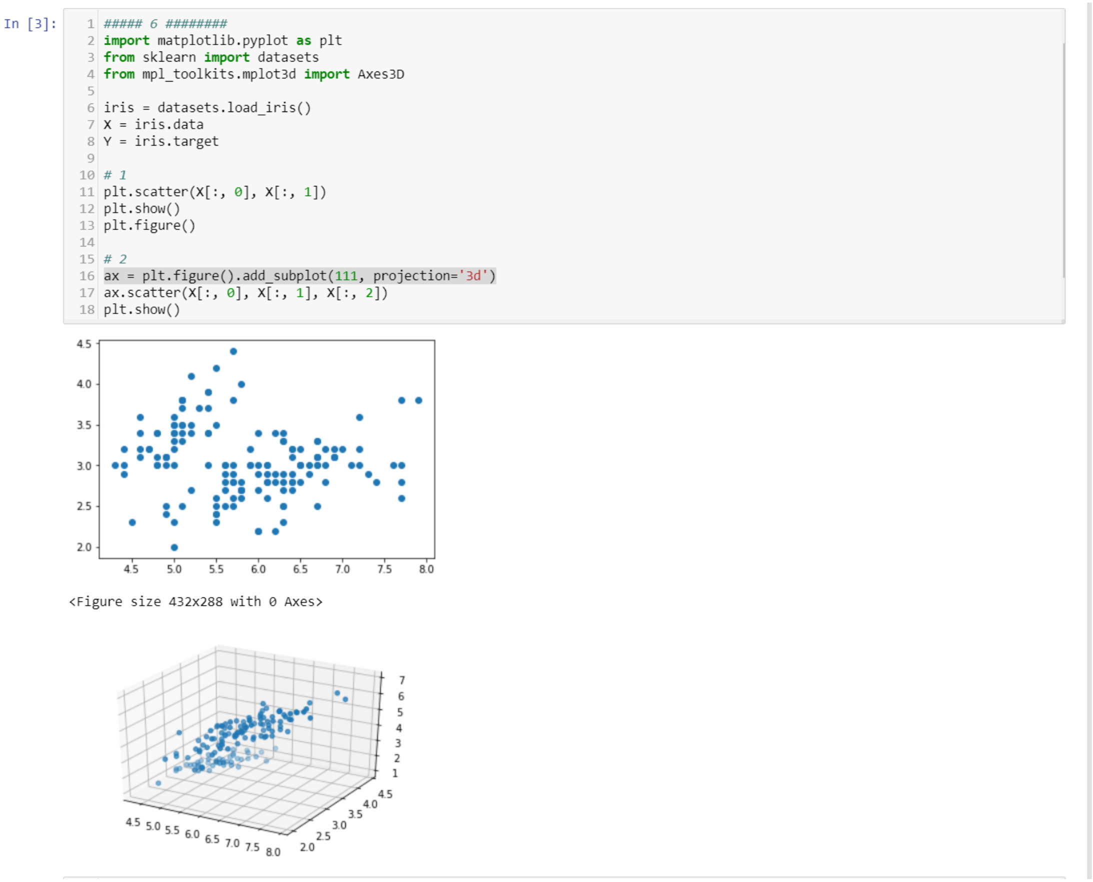
In [7]:

```
##### 5 ########
import matplotlib.pyplot as plt
img = plt.imread("cat.jpg") # The img here is a NumPy array.
plt.imshow(img)
# 1
plt.show()

# 2
img.shape
```

Out[7]:

```
(183, 275, 3)
```

6)

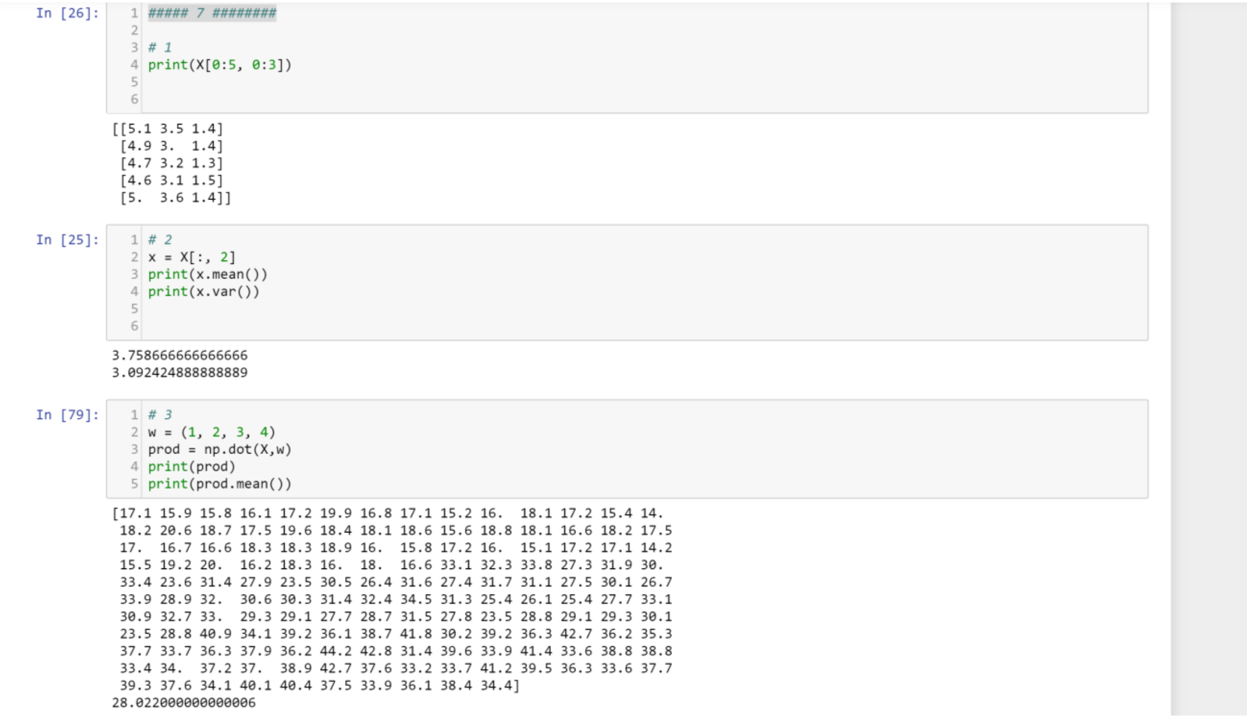


In [3]:

```
##### 6 ########
import matplotlib.pyplot as plt
from sklearn import datasets
from mpl_toolkits.mplot3d import Axes3D

iris = datasets.load_iris()
X = iris.data
Y = iris.target

# 1
plt.scatter(X[:, 0], X[:, 1])
plt.show()
plt.figure()

# 2
ax = plt.figure().add_subplot(111, projection='3d')
ax.scatter(X[:, 0], X[:, 1], X[:, 2])
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

7)



In [26]:
```
##### 7 ########

# 1
print(X[0:5, 0:3])

```

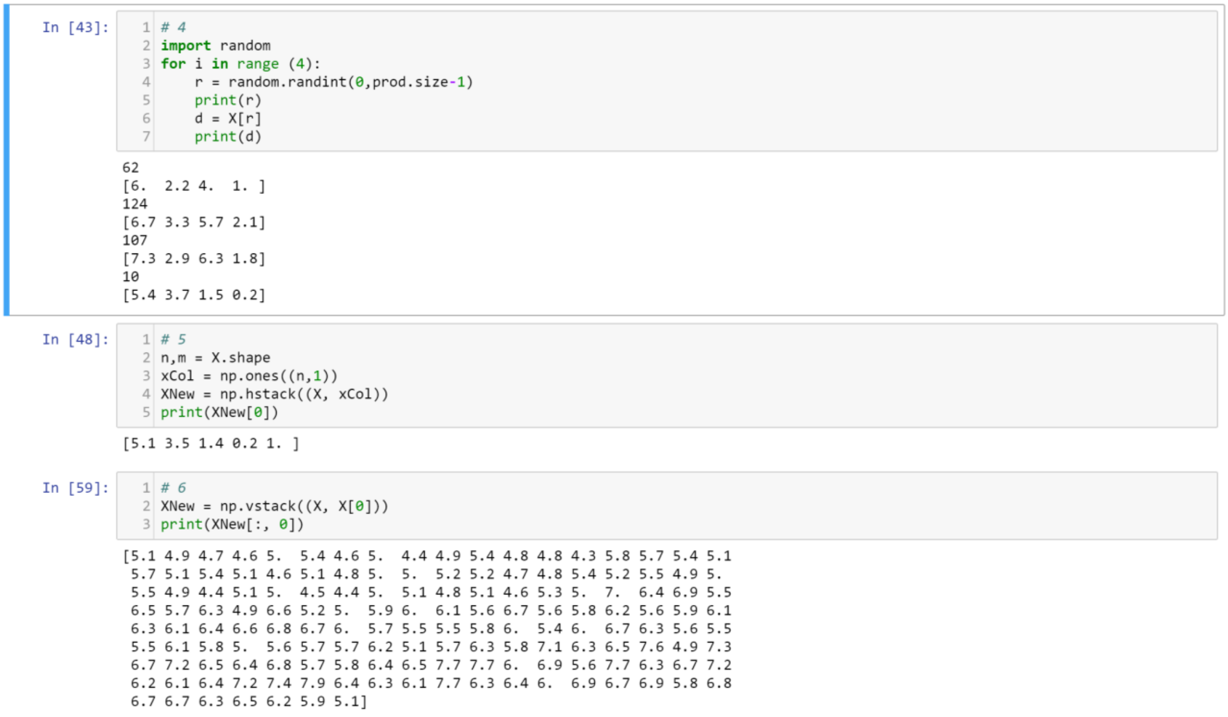
```
[[5.1 3.5 1.4]
 [4.9 3.  1.4]
 [4.7 3.2 1.3]
 [4.6 3.1 1.5]
 [5.  3.6 1.4]]
```

In [25]:
```
# 2
x = X[:, 2]
print(x.mean())
print(x.var())

```

```
3.758666666666666
3.092424888888889
```

In [79]:
```
# 3
w = (1, 2, 3, 4)
prod = np.dot(X,w)
print(prod)
print(prod.mean())
```

```
[17.1 15.9 15.8 16.1 17.2 19.9 16.8 17.1 15.2 16.  18.1 17.2 15.4 14.
 18.2 20.6 18.7 17.5 19.6 18.4 18.1 18.6 15.6 18.8 18.1 16.6 18.2 17.5
 17.  16.7 16.6 18.3 18.3 18.9 16.  15.8 17.2 16.  15.1 17.2 17.1 14.2
 15.5 19.2 20.  16.2 18.3 16.  18.  16.6 33.1 32.3 33.8 27.3 31.9 30.
 33.4 23.6 31.4 27.9 23.5 30.5 26.4 31.6 27.4 31.7 31.1 27.5 30.1 26.7
 33.9 28.9 32.  30.6 30.3 31.4 32.4 34.5 31.3 25.4 26.1 25.4 27.7 33.1
 30.9 32.7 33.  29.3 29.1 27.7 28.7 31.5 27.8 23.5 28.8 29.1 29.3 30.1
 23.5 28.8 40.9 34.1 39.2 36.1 38.7 41.8 30.2 39.2 36.3 42.7 36.2 35.3
 37.7 33.7 36.3 37.9 36.2 44.2 42.8 31.4 39.6 33.9 41.4 33.6 38.8 38.8
 33.4 34.  37.2 37.  38.9 42.7 37.6 33.2 33.7 41.2 39.5 36.3 33.6 37.7
 39.3 37.6 34.1 40.1 40.4 37.5 33.9 36.1 38.4 34.4]
28.022000000000006
```



In [43]:
```
# 4
import random
for i in range (4):
    r = random.randint(0,prod.size-1)
    print(r)
    d = X[r]
    print(d)
```

```
62
[6.  2.2 4.  1. ]
124
[6.7 3.3 5.7 2.1]
107
[7.3 2.9 6.3 1.8]
10
[5.4 3.7 1.5 0.2]
```

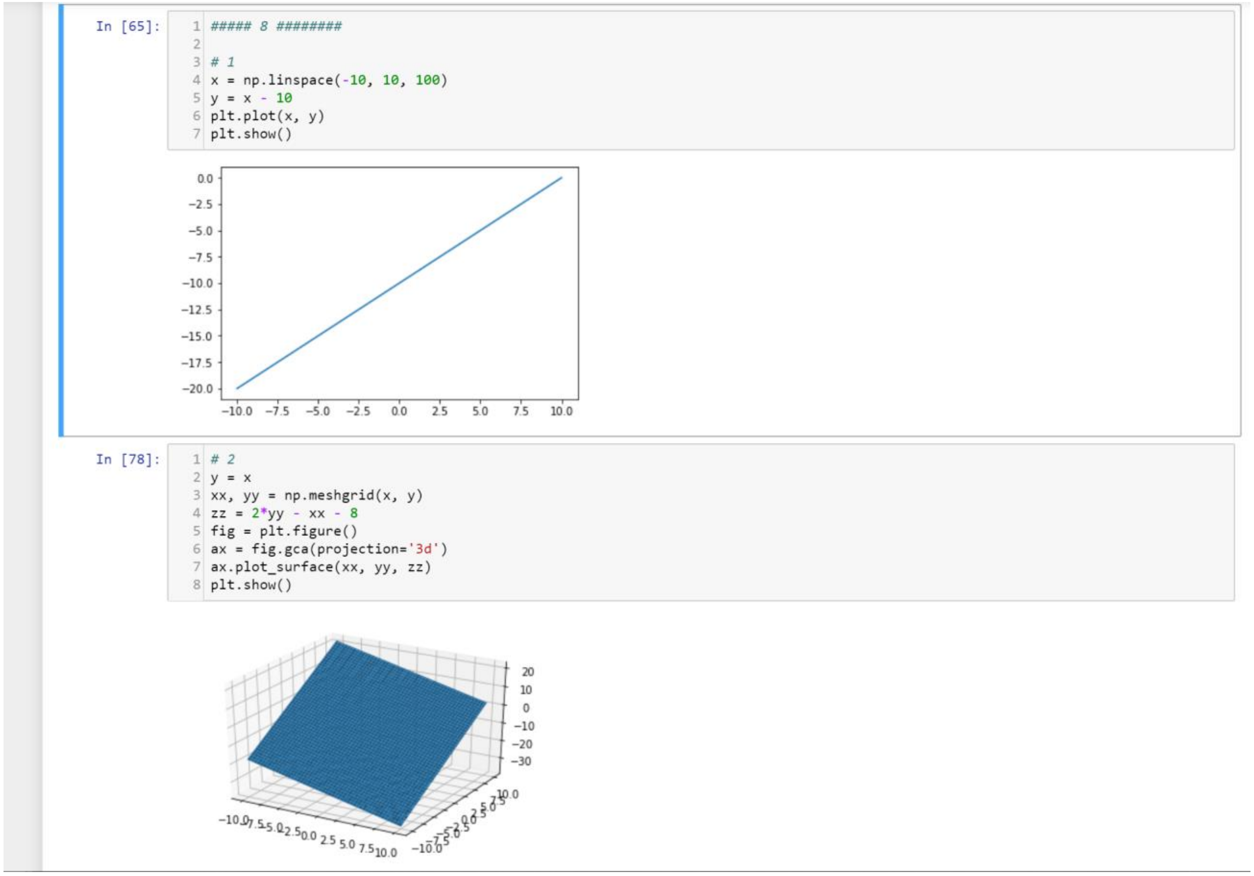
In [48]:
```
# 5
n,m = X.shape
xCol = np.ones((n,1))
XNew = np.hstack((X, xCol))
print(XNew[0])
```

```
[5.1 3.5 1.4 0.2 1. ]
```

In [59]:
```
# 6
XNew = np.vstack((X, X[0]))
print(XNew[:, 0])
```

```
[5.1 4.9 4.7 4.6 5.  5.4 4.6 5.  4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4 5.1
 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.  5.  5.2 5.2 4.7 4.8 5.4 5.2 5.5 4.9 5.
 5.5 4.9 4.4 5.1 5.  4.5 4.4 5.  5.1 4.8 5.1 4.6 5.3 5.  7.  6.4 6.9 5.5
 6.5 5.7 6.3 4.9 6.6 5.2 5.  5.9 6.  6.1 5.6 6.7 5.6 5.8 6.2 5.6 5.9 6.1
 6.3 6.1 6.4 6.6 6.8 6.7 6.  5.7 5.5 5.5 5.8 6.  5.4 6.  6.7 6.3 5.6 5.5
 5.5 6.1 5.8 5.  5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3
 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7 6.  6.9 5.6 7.7 6.3 6.7 7.2
 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7 6.3 6.4 6.  6.9 6.7 6.9 5.8 6.8
 6.7 6.7 6.3 6.5 6.2 5.9 5.1]
```

8)



In [65]:

```
##### 8 ########

# 1
x = np.linspace(-10, 10, 100)
y = x - 10
plt.plot(x, y)
plt.show()
```

In [78]:

```
# 2
y = x
xx, yy = np.meshgrid(x, y)
zz = 2*yy - xx - 8
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.plot_surface(xx, yy, zz)
plt.show()
```