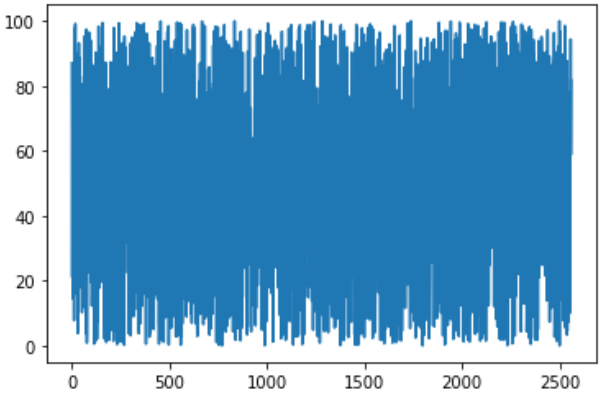
In [2]:
```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

Απλή Γραμμική Παλινδρόμηση

Διάνυσμα με 256 στοιχεία με τιμές από το 0 μέχρι το 100 (από ομοιόμορφη κατανομή)

In [69]:
```
x = 100 * np.random.rand(10*256)
plt.plot(x)
```

Out[69]:
```
[<matplotlib.lines.Line2D at 0x7f22704d4a90>]
```



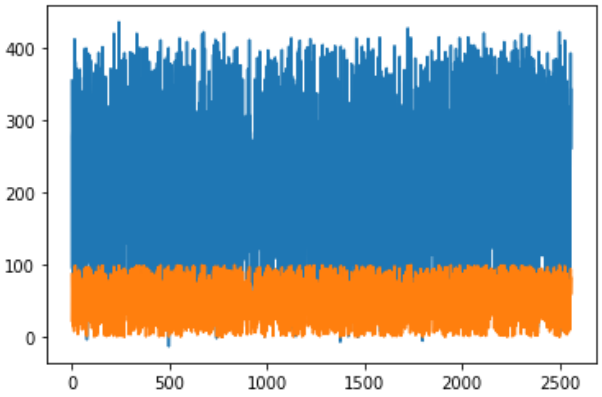
Δημιουργία των δεδομένων χρησιμοποιώντας τη διαδικασία

$$y = A + Bx + \epsilon$$

In [89]:
```
A = 10
B = 4
sigma_epsilon = 10
```

In [90]:
```
y = B * x + sigma_epsilon * np.random.randn(10*256) + A
plt.plot(y)
plt.plot(x)
```

Out[90]: [<matplotlib.lines.Line2D at 0x7f2270255b10>]

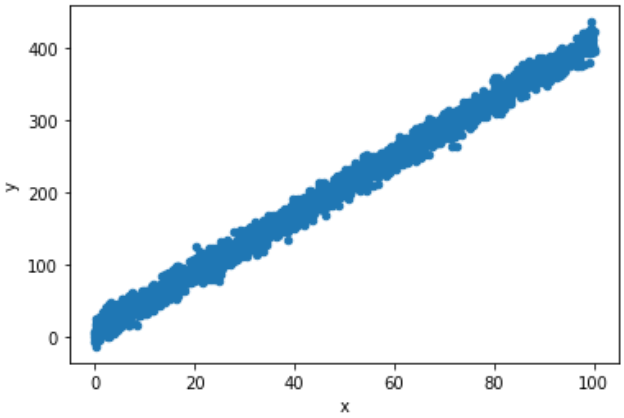


In [91]:
```
dataset = pd.DataFrame()
```

In [92]:
```
dataset['x'] = x
dataset['y'] = y
```

In [93]:
```
dataset.plot.scatter(x='x', y='y')
```

Out[93]: <matplotlib.axes._subplots.AxesSubplot at 0x7f22701f2510>



Χρήση της υλοποίησης του μοντέλου γραμμικής παλινδρόμησης από sklearn

In [94]:
```
reg = LinearRegression()
```

In [95]:
```
x.shape
```

Out[95]: (2560,)

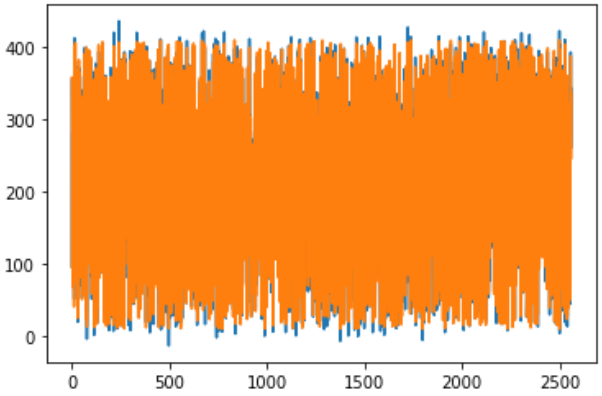
Προσαρμογή της ευθείας στις παρατηρήσεις (συνάρτηση fit)

In [96]:
```
reg.fit(np.expand_dims(x,1),y)
```

Out[96]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [97]:
```
y_hat = reg.predict(np.expand_dims(x,1))
plt.plot(y)
plt.plot(y_hat)
```

Out[97]:
```
[<matplotlib.lines.Line2D at 0x7f22701a0950>]
```



Υπολογισμός του a, είναι $a = \hat{y}(0)$

In [98]:
```
a = reg.predict(np.expand_dims(np.array([0]), -1))[0]
```

In [99]:
```
a
```

Out[99]:
```
9.796666410542372
```

Η παράμετρος της κλίσης b περιέχεται στο

In [100]:
```
b = reg.coef_[0]
```

In [101]:
```
b
```

Out[101]:
```
4.005808410087647
```

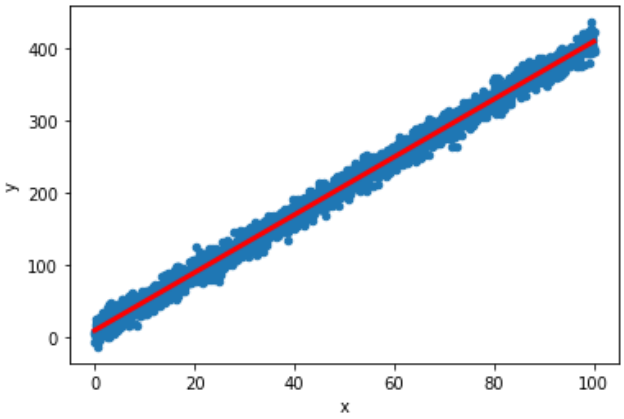
Υπολογισμός του $\hat{y}(100)$ ώστε να έχουμε 2ο σημείο (το πρώτο είναι το $(0, a)$) για να σχεδιάσουμε την ευθεία

In [102]:
```
y_hat_100 = reg.predict(np.expand_dims(np.array([100]), -1))[0]
y_hat_100
```

Out[102]:
```
410.3775074193071
```

In [103]:
```
dataset.plot.scatter(x='x', y='y')
plt.plot((0,100),(a, y_hat_100), color = 'red', lw=3)
```

Out[103]: [<matplotlib.lines.Line2D at 0x7f22701460d0>]



In [104]:
```
R2 = sum((y_hat - np.mean(y))**2) / sum((y - np.mean(y))**2)
```

In [105]:
```
R2
```

Out[105]: 0.9929030353043043

Πολλαπλή Γραμμική Παλινδρόμηση και Ψευδομεταβλητές

In [106]:
```
dataset = pd.read_csv('startups.csv')
```

In [107]:
```
dataset.head()
```

Out[107]:

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

In [108]:
```
Y = dataset['Profit'].values
```

In [120]:
```
X = dataset.iloc[:, :-1].values
```

Δημιουργία των dummy variables

In [114]:
```
labelencoder_X = LabelEncoder()
num_labels = labelencoder_X.fit_transform(X[:, 3])
num_labels.shape
```

Out[114]:
```
(46,)
```

In [113]:
```
onehotencoder = OneHotEncoder(sparse=False)
```

In [115]:
```
D = onehotencoder.fit_transform(np.expand_dims(num_labels,-1))
```

In [117]:
```
D = D[:,:-1]
```

In [133]:
```
Xnew = np.zeros((46, 5))
```

In [134]:
```
Xnew
```

Out[134]:
```
array([[0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.]])
```

In [137]:
```
Xnew[:,0:3] = X[:, 0:3]
```

In [138]:
```
Xnew[:,3::] = D
```

Προσαρμογή του μοντέλου

In [140]:
```
mreg = LinearRegression()
```

In [141]:
```
mreg.fit(Xnew, Y)
```

Out[141]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [142]:
```
mreg.coef_
```

Out[142]:
```
array([ 7.75070130e-01, -5.75881549e-02,  2.10212014e-02,  1.59722937e+03,
       -1.55204999e+02])
```

In [143]:
```
test = [[150000, 50000, 400000, 1, 0],
        [200000, 100000, 300000, 0, 0],
        [0,0,0,0,0]]
```

In [144]:
```
mreg.predict(test)
```

Out[144]:
```
array([181562.68526491, 213737.43451613,  58175.86353248])
```

$$p = (X^T X)^{-1} X^T y$$

In [146]:
```
Xnn = np.ones((46,6))
```

In [150]:
```
Xnn[:,1:] = Xnew
```

In [152]:
```
p = np.dot(np.dot(np.linalg.inv(np.dot(Xnn.T, Xnn)), Xnn.T), Y)
```

In [153]:
```
p
```

Out[153]:
```
array([ 5.81758635e+04,  7.75070130e-01, -5.75881549e-02,  2.10212014e-02,
        1.59722937e+03, -1.55204999e+02])
```