In [1]:

```
# Exercício de Aula sobre Regressão linear
```

In [35]:

```
df = pd.read_csv('exercicio.csv', delimiter=',', encoding='iso8859_2')
```

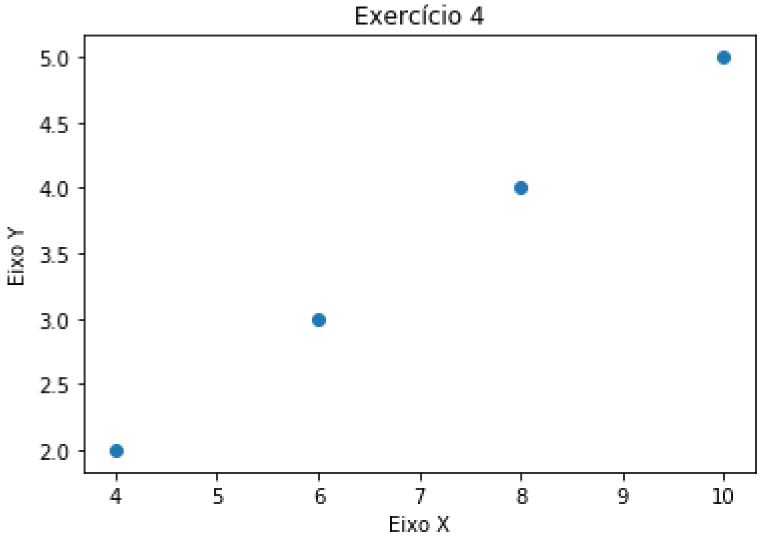
In [36]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 2 columns):
4      3 non-null int64
 2     3 non-null int64
dtypes: int64(2)
memory usage: 128.0 bytes
```

In [39]:

```
from numpy import *
import matplotlib.pyplot as plt

points = genfromtxt('exercicio.csv', delimiter=',')

#Extract columns
x = array(points[:,0])
y = array(points[:,1])

#Plot the dataset
plt.scatter(x,y)
plt.xlabel('Eixo X')
plt.ylabel('Eixo Y')
plt.title('Exercício 4')
plt.show()
#Sem a Regressão liner calculada
```



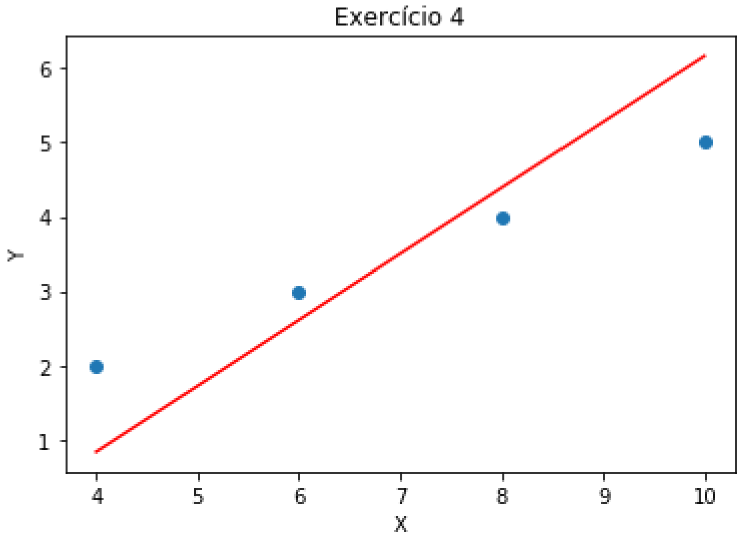
In [47]:

```
m = (len(x) * cov(x, y)[0,1]) / ((len(x) - 1) * var(x))
#m = np.cov(x, y)[0,1]/np.var(x)

b = mean(y) - (m * mean(x))

print(m)
print(b)
```

```
0.8888888888888888
-2.7222222222222214
```

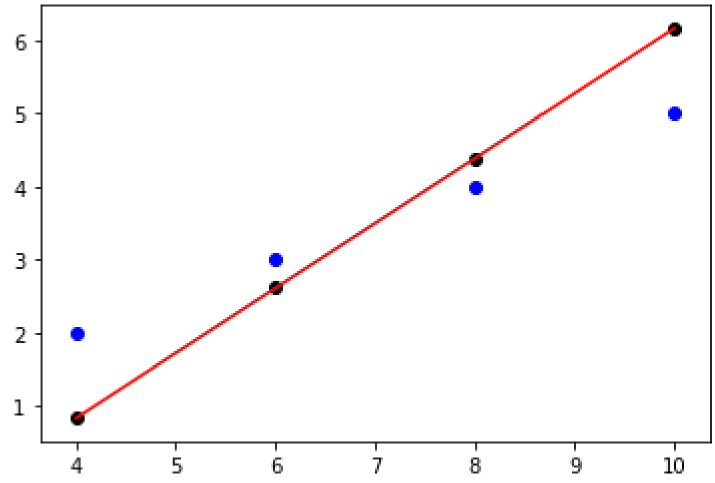
In [48]:

```
#b = 0.02963934787473239
#m = 1.4774173755483797

#Plot dataset
plt.scatter(x, y)
#Predict y values
pred = m * x + b
#pred1 = m1 * x + b1
#Plot predictions as line of best fit
plt.plot(x, pred, c='r')
#plt.plot(x, pred1, c='r')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Exercício 4')
plt.show()
```



In [51]:

```
#Previsões da Curva de Regressão
x_inteiro = [4, 6, 8, 10]
y_inteiro = [2, 3, 4, 5]

y_regressao =  [m * x + b for x in x_inteiro]

plt.scatter(x_inteiro, y_inteiro, c='b')
plt.scatter(x_inteiro, y_regressao, c='k')

pred = m * x + b
plt.plot(x, pred, c='r')

plt.show()

erro_proporcional = [round(100*(1 - i/j),2) for i, j in zip(y_inteiro, y_regressao)]
print(erro_proporcional)
```



```
[-140.0, -14.89, 8.86, 18.92]
```