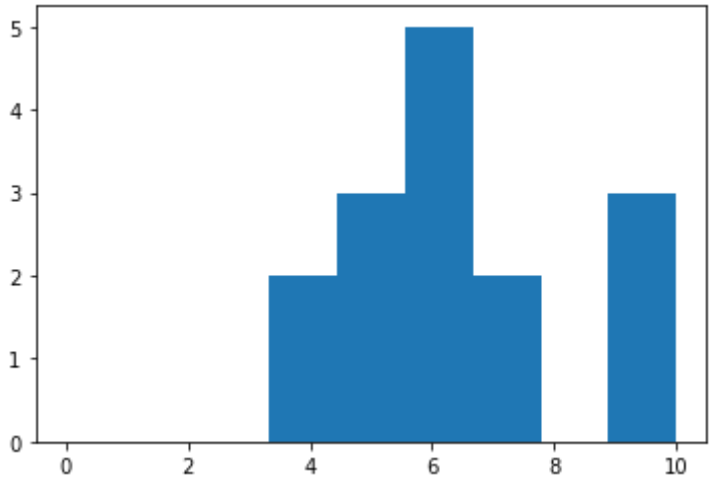
In [9]:
```
import scipy
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
```

In [10]:
```
val = np.array([4,4,5,6,7,9,6,5,6,9,6,10,5,7,6])

hist = plt.hist(val, 9, (0, 10))

print("Mean: ", np.mean(val))
print("Standard deviation: ", np.std(val))
```

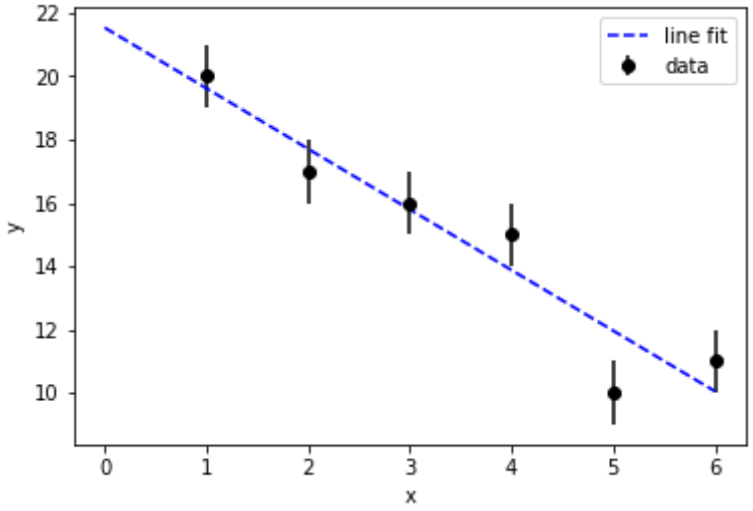
```
Mean:  6.333333333333333
Standard deviation:  1.7384539747207064
```



In [11]:
```
x_list = np.array([1, 2, 3, 4, 5, 6])
y_list = np.array([20, 17, 16, 15, 10, 11])
y_err = np.array([1, 1, 1, 1, 1, 1])

#Plotting x vs. y with error bar
plt.errorbar(x_list, y_list,yerr=y_err,fmt="ko",label="data")

#Linear fit function
def line_func(x, a, b):
    return x*a+b

guess_a = 1.0
guess_b = 0.0
par, cov = optimize.curve_fit(line_func, x_list, y_list,
                              p0=[guess_a, guess_b],sigma=y_err)

fit_a = par[0]
fit_b = par[1]
print("best fit value of a:  ", fit_a)
print("best fit value of b:  ", fit_b)

xf    = np.linspace(0.0,6.0,100)
yf    = fit_b + fit_a * xf
plt.plot(xf,yf,"b--",label="line fit")

plt.xlabel("x")
```

```
plt.ylabel("y")
plt.legend()
plt.show()
```

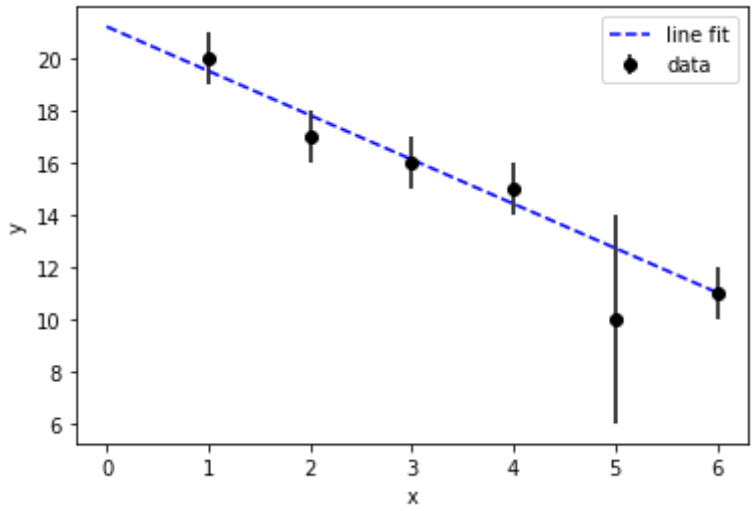
```
best fit value of a:   -1.9142857142920717
best fit value of b:   21.533333333380302
```



In [12]:

```
x_list = np.array([1, 2, 3, 4, 5, 6])
y_list = np.array([20, 17, 16, 15, 10, 11])
y_err = np.array([1, 1, 1, 1, 4, 1])

#Plotting x vs. y with error bar
plt.errorbar(x_list, y_list,yerr=y_err,fmt="ko",label="data")

#Linear fit function
def line_func(x, a, b):
    return x*a+b

guess_a = 1.0
guess_b = 0.0
par, cov = optimize.curve_fit(line_func, x_list, y_list,
                              p0=[guess_a, guess_b],sigma=y_err)

fit_a = par[0]
fit_b = par[1]
print("best fit value of a:  ", fit_a)
print("best fit value of b:  ", fit_b)

xf    = np.linspace(0.0,6.0,100)
yf    = fit_b + fit_a * xf
plt.plot(xf,yf,"b--",label="line fit")

plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.show()
```

```
best fit value of a:   -1.6962962963021764
best fit value of b:   21.194238683173804
```



In [ ]: