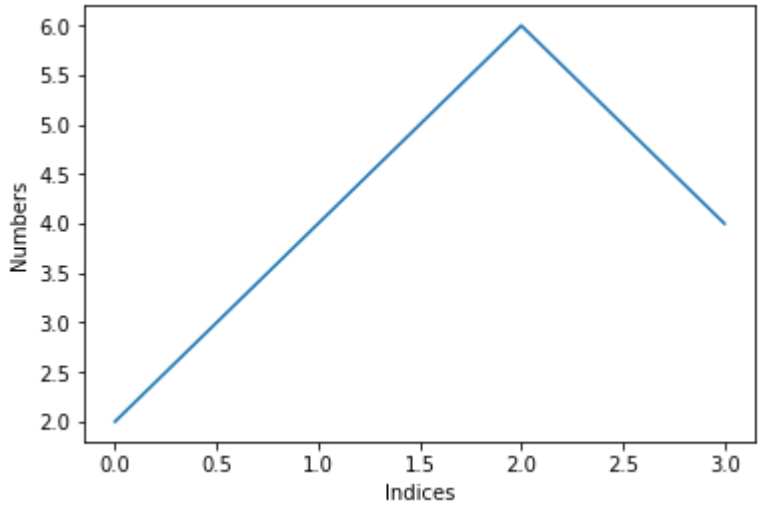
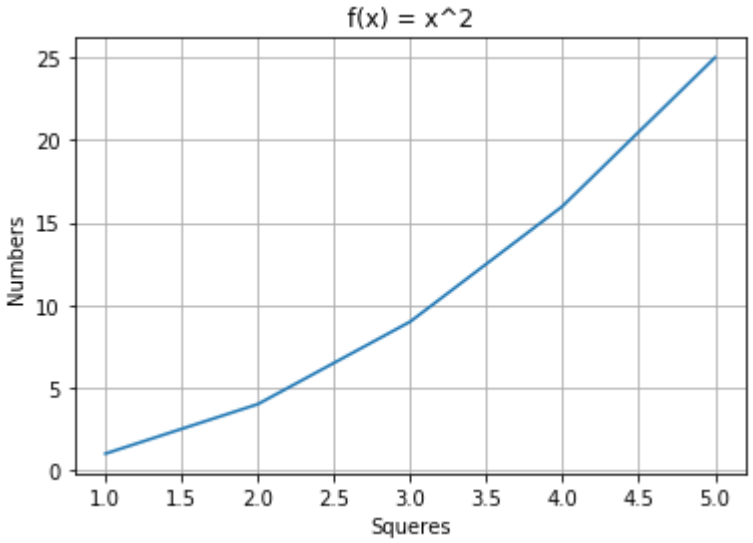
In [1]:
```
import matplotlib.pyplot as plt
```

In [2]:
```
x = [2, 4, 6, 4]
plt.plot(x)

plt.ylabel('Numbers')
plt.xlabel('Indices')

plt.show()

"""Here, in plot function we pass one list. So, the function plot a plane wher
e the value of the list place in Y-axis and the indices of the value will plac
e in the X-axis"""
```

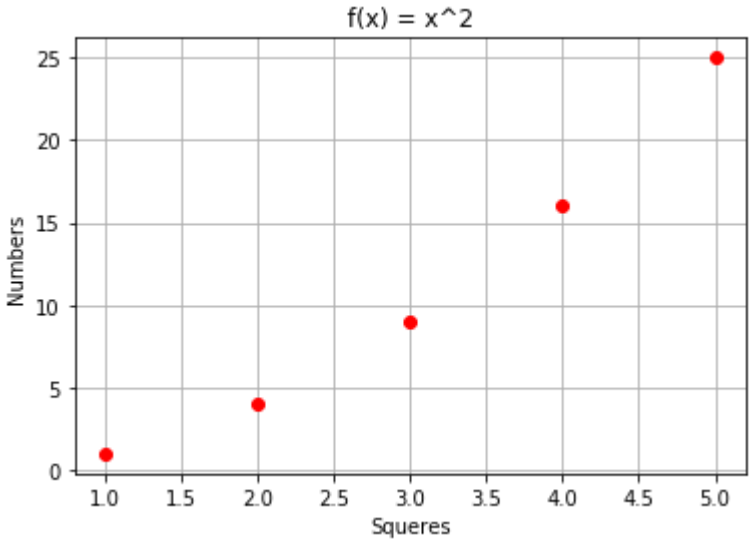


Out[2]:
```
'Here, in plot function we pass one list. So, the function plot a plane where
the value of the list place in Y-axis and the indices of the value will place
in the X-axis'
```

In [3]:
```
#plot data with two list

x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]

plt.plot(x, y)

plt.title('f(x) = x^2')
plt.ylabel('Numbers')
plt.xlabel('Squeres')

plt.grid()
plt.show()
```

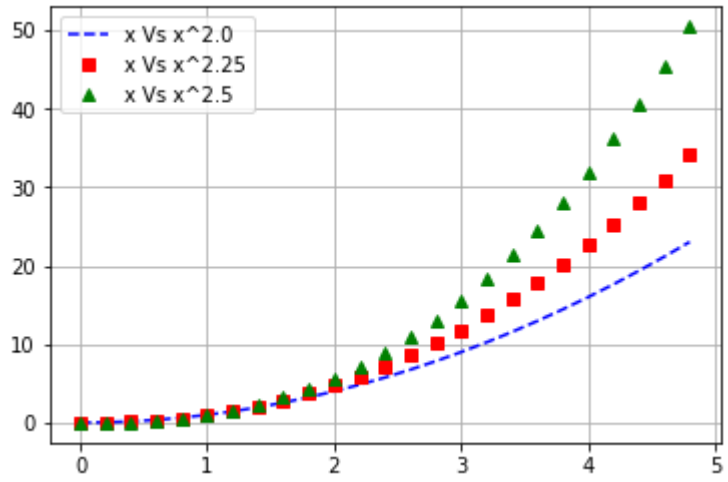


In [4]:
```
#plot data without joining them in a line

x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]

plt.plot(x, y, 'ro') #

plt.title('f(x) = x^2')
plt.ylabel('Numbers')
plt.xlabel('Squeres')

plt.grid()
plt.show()
```

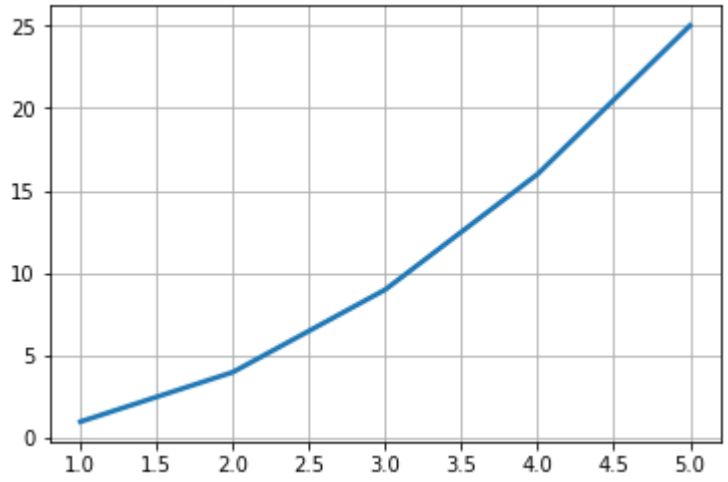


In [5]:
```
import numpy as np
```

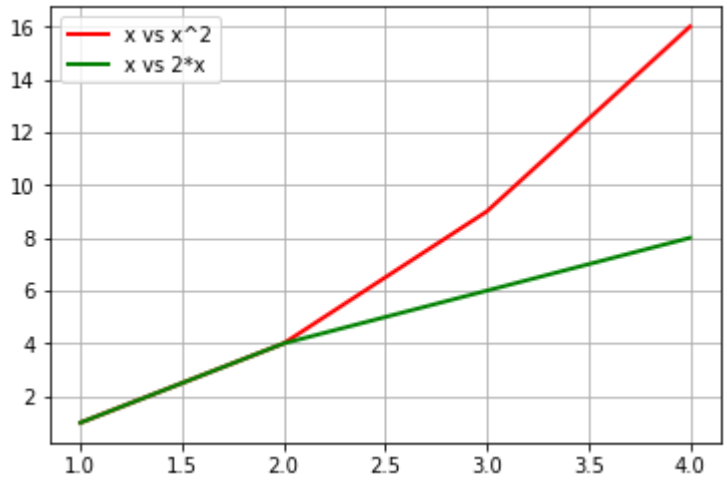
In [6]:
```
t = np.arange(0.0, 5.0, 0.2)

plt.plot(t, t**2.0, 'b--', label='x Vs x^2.0') #here, b-- means blue and dash
 line
plt.plot(t, t**2.25, 'rs', label='x Vs x^2.25') #here, rs means red and sloid
 boxes
plt.plot(t, t**2.5, 'g^', label='x Vs x^2.5') #here, g^ means green and triang
le

plt.grid()
plt.legend() #to add label for each type of lines
plt.show()
```



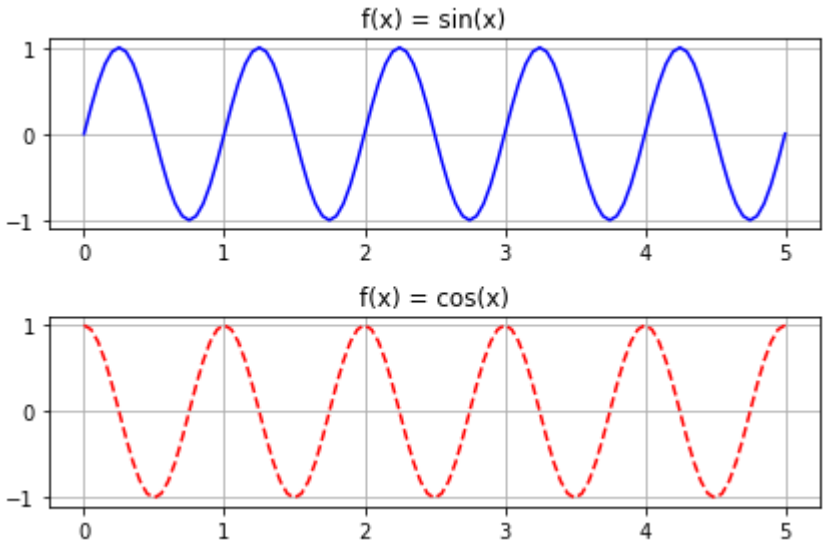
In [7]:
```
x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]

plt.plot(x, y, linewidth=2.5) #Here, linewidth is the value of width

plt.grid()
plt.show()
```



In [8]:
```
x1 = [1, 2, 3, 4]
y1 = [1, 4, 9, 16]

x2 = x1
y2 = [1, 4, 6, 8]

lines = plt.plot(x1, y1, x2, y2)

#use keywords args
plt.setp(lines[0], color='r', linewidth=2, label='x vs x^2')

#use key-value pair
plt.setp(lines[1], 'color', 'g', 'linewidth', 2.0, 'label', 'x vs 2*x')

plt.grid()
plt.legend()
plt.show()
```



In [9]:
```
#Multiple plots/figure

values = np.arange(0.0, 5.05, 0.05)

plt.figure(1)

sbplt1 = plt.subplot(211)
plt.grid()
sbplt1.title.set_text('f(x) = sin(x)')
plt.plot(values, np.sin(2*np.pi*values), 'b-')

sbplt2 = plt.subplot(212)
plt.grid()
sbplt2.title.set_text('f(x) = cos(x)')
plt.plot(values, np.cos(2*np.pi*values), 'r--')

plt.tight_layout()
plt.show()
```



In [10]:
```
#works on different figure

plt.figure(1)

f1 = plt.subplot(311)
plt.plot([1, 2, 3])

f2 = plt.subplot(312)
plt.plot([4, 5, 6])

plt.figure(2)
plt.plot([-1, -2, 0, 4])

f1.title.set_text('Just a title')

plt.figure(1) #here, we can come to figure 1 again after working on figure 2
plt.subplot(3, 1, 3)
plt.plot([3, 5, 6, 9])

plt.tight_layout()
plt.show()
```