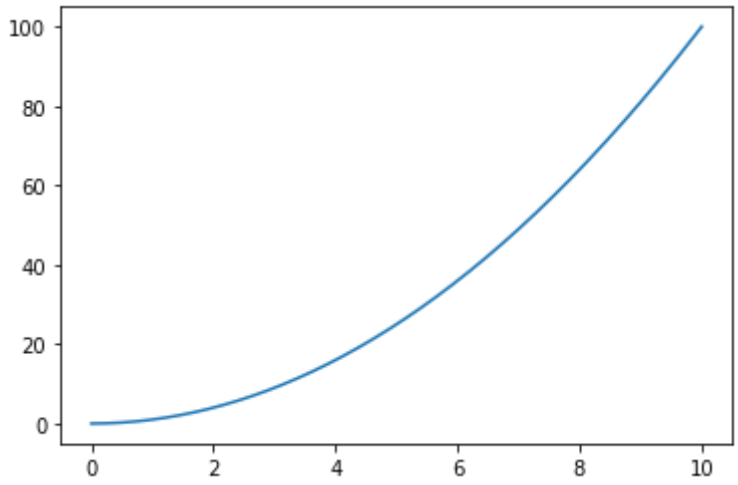
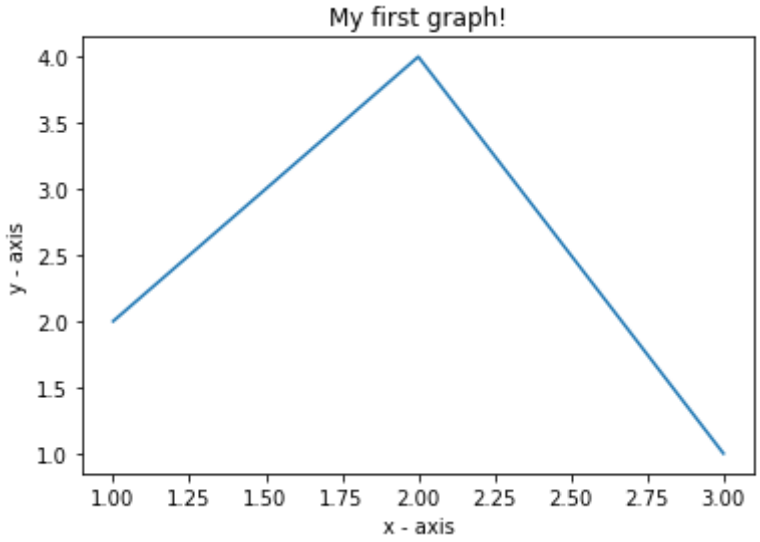
In [1]:
```
import numpy as np
import matplotlib . pyplot as plt
x = np. linspace (0 , 10, 1000)
y = np.power(x , 2)
plt . plot (x , y)
plt .show()
```

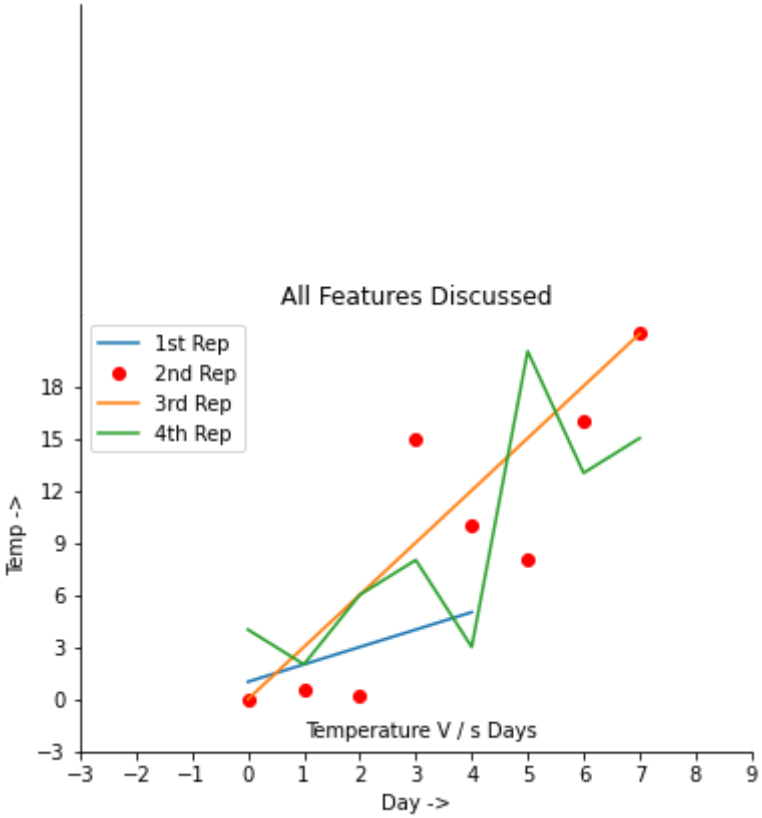


In [2]:
```
# importing the required module
import matplotlib.pyplot as plt

# x axis values
x = [1,2,3]
# corresponding y axis values
y = [2,4,1]

# plotting the points
plt.plot(x, y)

# naming the x axis
plt.xlabel('x - axis')
# naming the y axis
plt.ylabel('y - axis')

# giving a title to my graph
plt.title('My first graph!')

# function to show the plot
plt.show()
```



In [3]:

```
import matplotlib.pyplot as plt


a = [1, 2, 3, 4, 5]
b = [0, 0.6, 0.2, 15, 10, 8, 16, 21]
plt.plot(a)

# o is for circles and r is
# for red
plt.plot(b, "or")

plt.plot(list(range(0, 22, 3)))

# naming the x-axis
plt.xlabel('Day ->')

# naming the y-axis
plt.ylabel('Temp ->')

c = [4, 2, 6, 8, 3, 20, 13, 15]
plt.plot(c, label = '4th Rep')

# get current axes command
ax = plt.gca()

# get command over the individual
# boundary line of the graph body
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)

# set the range or the bounds of
# the left boundary line to fixed range
ax.spines['left'].set_bounds(-3, 40)

# set the interval by  which
# the x-axis set the marks
plt.xticks(list(range(-3, 10)))

# set the intervals by which y-axis
# set the marks
plt.yticks(list(range(-3, 20, 3)))

# legend denotes that what color
# signifies what
ax.legend(['1st Rep', '2nd Rep', '3rd Rep', '4th Rep'])

# annotate command helps to write
# ON THE GRAPH any text xy denotes
# the position on the graph
plt.annotate('Temperature V / s Days', xy = (1.01, -2.15))

# gives a title to the Graph
plt.title('All Features Discussed')

plt.show()
```



In [4]:
```
# Python Program to
# show range() basics

# printing a number
for i in range(10):
    print(i, end =" ")
print()

# using range for iteration
l = [10, 20, 30, 40]
for i in range(len(l)):
    print(l[i], end =" ")
print()

# performing sum of natural
# number
sum = 0
for i in range(1, 11):
    sum = sum + i
print("Sum of first 10 natural number :", sum)
```

```
0 1 2 3 4 5 6 7 8 9
10 20 30 40
Sum of first 10 natural number : 55
```

In [5]:
```
# Python program to
# print whole number
# using range()

# printing first 10
# whole number
for i in range(10):
    print(i, end =" ")
print()

# printing first 20
# whole number
```

```
for i in range(20):
    print(i, end = " ")
```

```
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

In [6]:
```
# Python program to
# print natural number
# using range

# printing a natural
# number upto 20
for i in range(1, 20):
    print(i, end =" ")
print()

# printing a natural
# number from 5 t0 20
for i in range(5, 20):
    print(i, end =" ")
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

In [7]:
```
# Python program to
# print all number
# divisible by 3 and 5

# using range to print number
# divisible by 3
for i in range(0, 30, 3):
    print(i, end = " ")
print()

# using range to print number
# divisible by 5
for  i in range(0, 50, 5):
     print(i, end = " ")
```

```
0 3 6 9 12 15 18 21 24 27
0 5 10 15 20 25 30 35 40 45
```

In [8]:
```
# Python program to
# increment with
# range()

# incremented by 2
for i in range(2, 25, 2):
    print(i, end =" ")
print()

# incremented by 4
for i in range(0, 30, 4):
    print(i, end =" ")
print()

# incremented by 3
for i in range(15, 25, 3):
    print(i, end =" ")
```

```
2 4 6 8 10 12 14 16 18 20 22 24
0 4 8 12 16 20 24 28
15 18 21 24
```

In [9]:
```
# Python program to
# decrement with
```

```
# range()

# incremented by -2
for i in range(25, 2, -2):
    print(i, end =" ")
print()

# incremented by -4
for i in range(30, 1, -4):
    print(i, end =" ")
print()

# incremented by -3
for i in range(25, -6, -3):
    print(i, end =" ")
```

```
25 23 21 19 17 15 13 11 9 7 5 3
30 26 22 18 14 10 6 2
25 22 19 16 13 10 7 4 1 -2 -5
```

In [11]:
```
# Python program to concatenate
# the result of two range functions


from itertools import chain


# Using chain method
print("Concatenating the result")
res = chain(range(5), range(10, 20, 2))

for i in res:
    print(i, end=" ")
```

```
Concatenating the result
0 1 2 3 4 10 12 14 16 18
```

In [12]:
```
# Python program to demonstrate
# range function


ele = range(10)[0]
print("First element:", ele)

ele = range(10)[-1]
print("\nLast element:", ele)

ele = range(10)[4]
print("\nFifth element:", ele)
```

```
First element: 0

Last element: 9

Fifth element: 4
```

In [13]:
```
# Python program to
# show range() type

# checking a type of
# range
type(range(3))
```

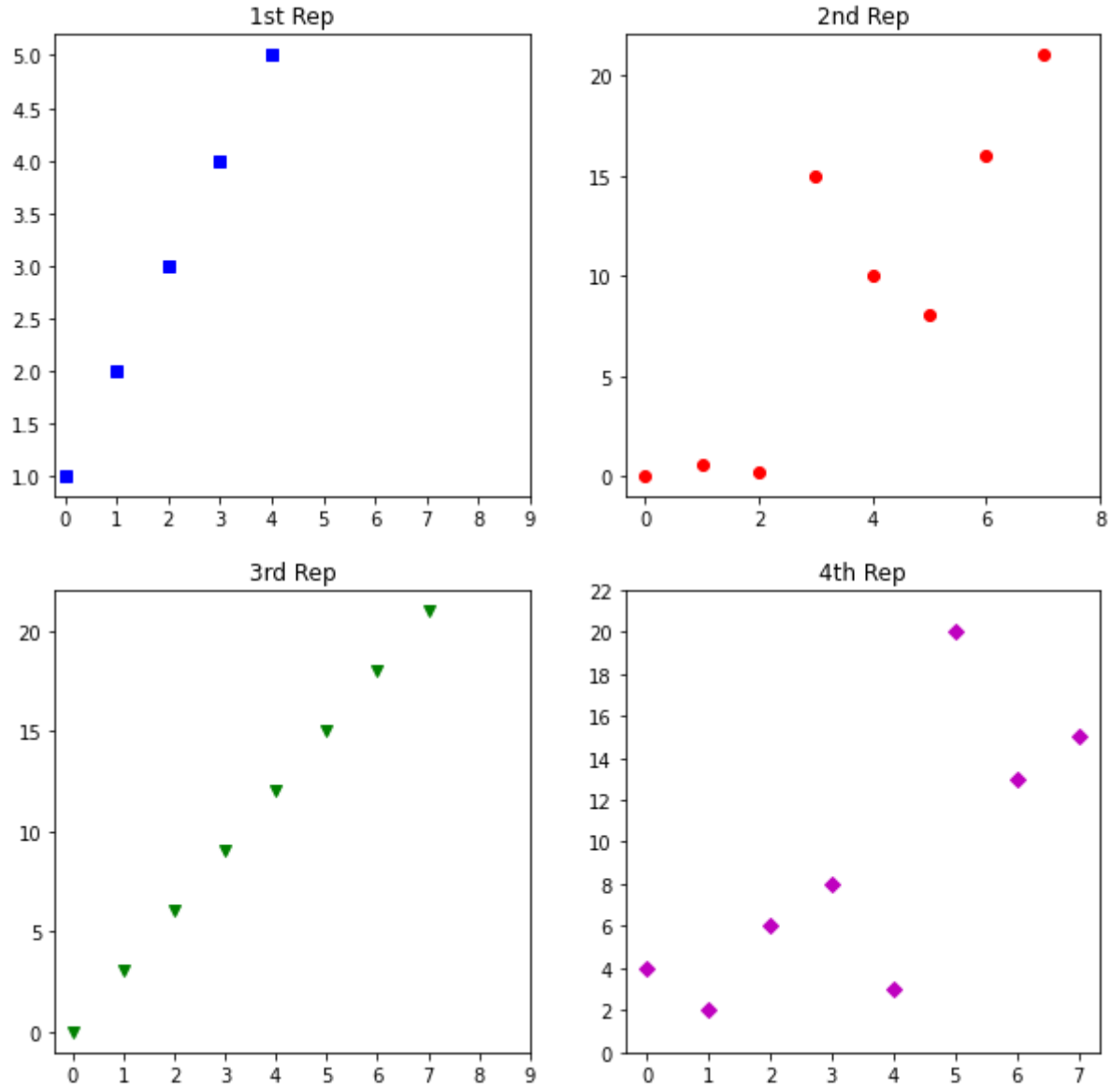
Out[13]:
```
range
```

In [14]:
```
# Python program to
```

```
# access items in a range

# accessing a items
range(3)[1]

# accessing a items
range(3)[2]
```

Out[14]: 2

In [15]:
```
import matplotlib.pyplot as plt


a = [1, 2, 3, 4, 5]
b = [0, 0.6, 0.2, 15, 10, 8, 16, 21]
c = [4, 2, 6, 8, 3, 20, 13, 15]

# use fig whenever u want the
# output in a new window also
# specify the window size you
# want ans to be displayed
fig = plt.figure(figsize =(10, 10))

# creating multiple plots in a
# single plot
sub1 = plt.subplot(2, 2, 1)
sub2 = plt.subplot(2, 2, 2)
sub3 = plt.subplot(2, 2, 3)
sub4 = plt.subplot(2, 2, 4)

sub1.plot(a, 'sb')

# sets how the display subplot
# x axis values advances by 1
# within the specified range
sub1.set_xticks(list(range(0, 10, 1)))
sub1.set_title('1st Rep')

sub2.plot(b, 'or')

# sets how the display subplot x axis
# values advances by 2 within the
# specified range
sub2.set_xticks(list(range(0, 10, 2)))
sub2.set_title('2nd Rep')

# can directly pass a list in the plot
# function instead adding the reference
sub3.plot(list(range(0, 22, 3)), 'vg')
sub3.set_xticks(list(range(0, 10, 1)))
sub3.set_title('3rd Rep')

sub4.plot(c, 'Dm')

# similarly we can set the ticks for
# the y-axis range(start(inclusive),
# end(exclusive), step)
sub4.set_yticks(list(range(0, 24, 2)))
sub4.set_title('4th Rep')

# without writing plt.show() no plot
# will be visible
plt.show()
```



In [19]:
```
import pandas as pd

import matplotlib
matplotlib.style.use('ggplot')

ts = pd.Series(np.random.randn(1000), index=pd.date_range('1/1/2000', periods=1000))

ts = ts.cumsum()

ts.plot()
```

Out[19]: <AxesSubplot:>

In [20]:
```
import pandas as pd

import matplotlib
matplotlib.style.use('ggplot')


df = pd.DataFrame(np.random.randn(1000, 4), index=ts.index, columns=list('ABCD'))
df = df.cumsum()
plt.figure(); df.plot();
```

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