

chapter 1 intro

```

1 TXT: links:
2 1: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html
3 2. https://www.geeksforgeeks.org/matplotlib-tutorial/
4
5 Videos links:
6 1: https://www.youtube.com/watch?v=axSTGczvYIE

```

In [7]:

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt

```

chapter 2: numpy concept

```

1 numpy.array(object, dtype = None, copy = True, order = None, subok =
False, ndmin = 0)

```

In [9]:

```

1 #2D array
2 #[1,2]:1 row data ; [3,4]: 2nd row data
3 a = np.array([[1, 2], [3, 4]])
4 print (a)

```

```

[[1 2]
 [3 4]]

```

In [10]:

```

1 """example 1: print nd-array
2 link: https://snakify.org/en/lessons/two_dimensional_lists_arrays/
3 """
4 a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]
5 for i in range(len(a)):
6     for j in range(len(a[i])):
7         print(a[i][j], end=' ')
8     print()

```

```

1 2 3 4
5 6
7 8 9

```

In [12]:

```

1 print(a[0])
2 print(a[0][0])
3 print(a[1])
4 print(a[2])

```

```
[1, 2, 3, 4]
1
[5, 6]
[7, 8, 9]
```

In [18]:

```

1 """example 2:
2 ndarray.size    数组元素的总个数 · 相当于 .shape 中 n*m 的值
3 """
4 array_a = np.array([[1,2,3],[4,5,6]]) #must be mp.array
5 print (array_a.shape)

```

```
(2, 3)
```

In [19]:

```

1 """example 3:array +range
2 numpy.arange(start, stop, step, dtype)
3 """
4 x = np.arange(10,20,2)
5 print (x)

```

```
[10 12 14 16 18]
```

In [21]:

```

1 a = np.arange(10)
2 b = a[2:10:3] # 从索引 2 开始到索引 7 停止 · 间隔为 2
3 print(b)

```

```
[2 5 8]
```

In [25]:

```

1 """ex 4 : calculation
2 """
3 x=np.array([-3,-2,-1,0,1,2,3])
4 y=x**2
5 z=np.square(x)
6 y2=x**2 #power of
7 print(y)
8 print(z)
9 print(y2)

```

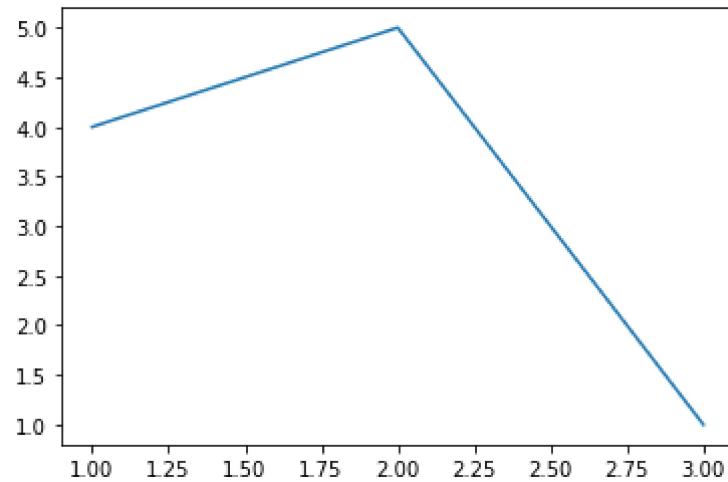
```
[-6 -4 -2  0  2  4  6]
[9 4 1 0 1 4 9]
[9 4 1 0 1 4 9]
```

chapter 3: graphs

dataframe.plot vs plt.plot They are diff

In [4]:

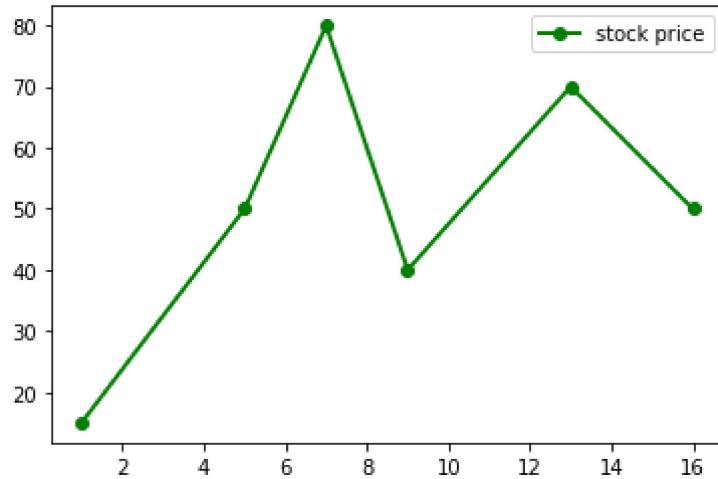
```
1 plt.plot([1,2,3],[4,5,1]) #plot(x_value,y_value)
2 #display the graph
3 plt.show()
```



In [32]:

```
1 """example 2: explain the plt.plot()
2 plt.plot(x_value,y_value)
3 if plot a csv / pivot table, it can auto defected,
4 Yet, it diff from df plot
5 df.plot (xxx finish all args )
6 plt.plot set seperately
7 至於 kind='bar', change to plt.bar=plt.bahr=plt.pie /plt.scatter
8 """
9
10 """plt.plot(x,y,color='',ls=,lw=,label=)
11 label: name of line
12 linestyle/ls: <->實線(default) ; <--> 虛線 ; <:>點線 ; <-*>虛點線
13 linewidth/lw: default 1.0
14 markszie/ms:點大小
15 marker:點類型, <o/.>
16 """
17 x=[1,5,7,9,13,16]
18 y=[15,50,80,40,70,50] #x and y must have same num of element
19 plt.plot(x,y,color='green',ls='-',marker='o',lw=2.0,label="stock price")
#cannot add args: xlabel,ylabel,title in it
21 plt.legend()
#label arg valid only use Legend() method
```

Out[32]: <matplotlib.legend.Legend at 0x1c415f9c130>

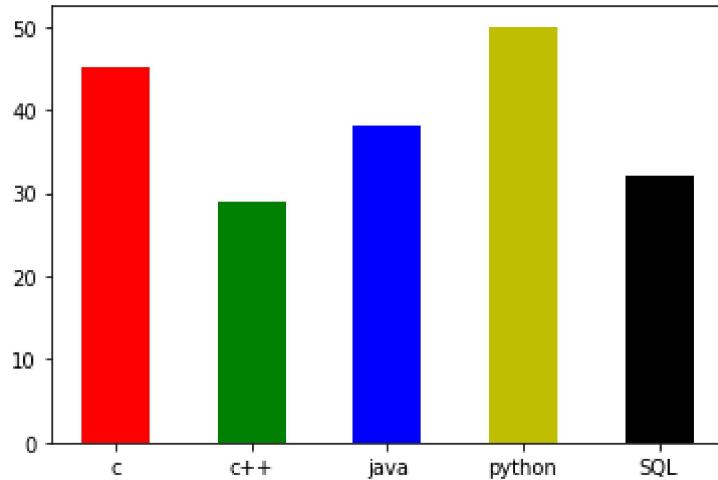


In [43]:

```

1 """example 3: fully syntax of plt.plot()
2 """
3 x=['c','c++','java','python','SQL']
4 y=[45,29,38,50,32]
5 plt.bar(x,y,width=0.5,color=['r','g','b','y','k']) #'rgbkymc'
6 plt.title='Software Poplularity'    #must below plt.bar()
7 plt.xlabel='Language';plt.ylabel='share'
8

```



chapter 4: figure()

- 1 Subplots are required when we want to show two or more plots in same figure.
- 2 it involves figure(),axes()

In [5]:

```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt

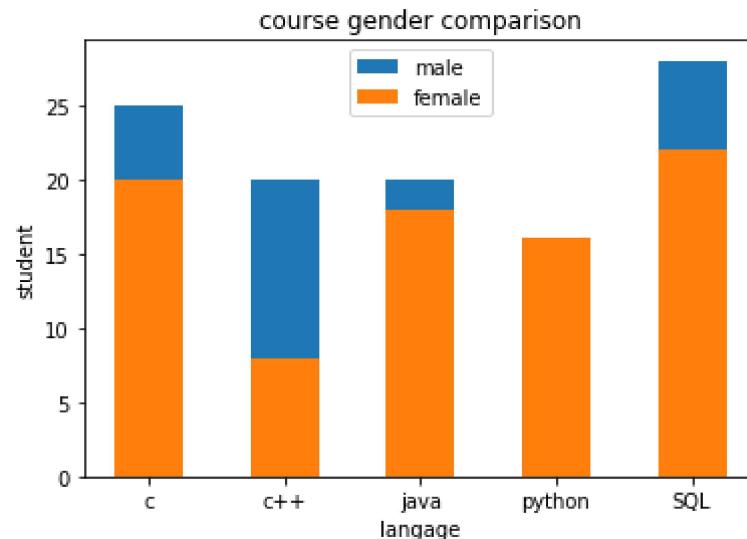
```

1 link:<https://www.geeksforgeeks.org/matplotlib-pyplot-subplots-in-python/>

In [8]:

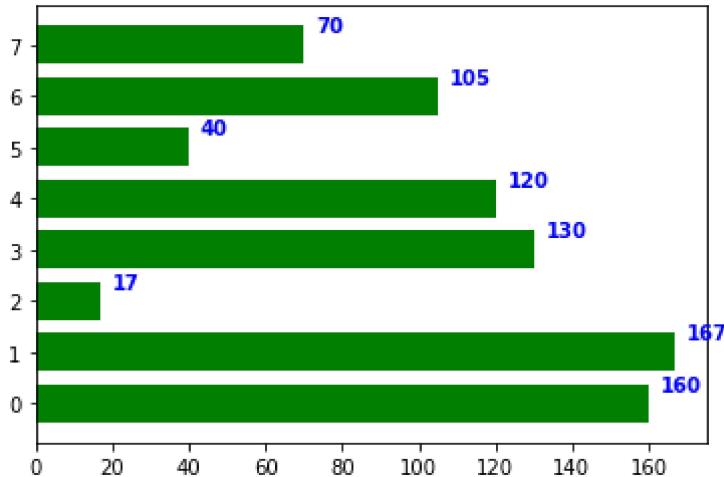
```
1 """4.1: 2 elements in a same bar
2 use bottom concept, but not use plt.figure() now
3 """
4 """ax.bar(x, height, width, bottom, align)
5 """
6 """
7
8 x=['c','c++','java','python','SQL']
9 y1=[25,20,20,16,28]
10 y2=[20,8,18,16,22]
11 plt.bar(x,y1,width=0.5,label='male')
12 plt.bar(x,y2,width=0.5,label='female')
13 plt.legend()
14 plt.title("course gender comparison")
15 plt.xlabel('langage')
16 plt.ylabel('student')
```

Out[8]: Text(0, 0.5, 'student')

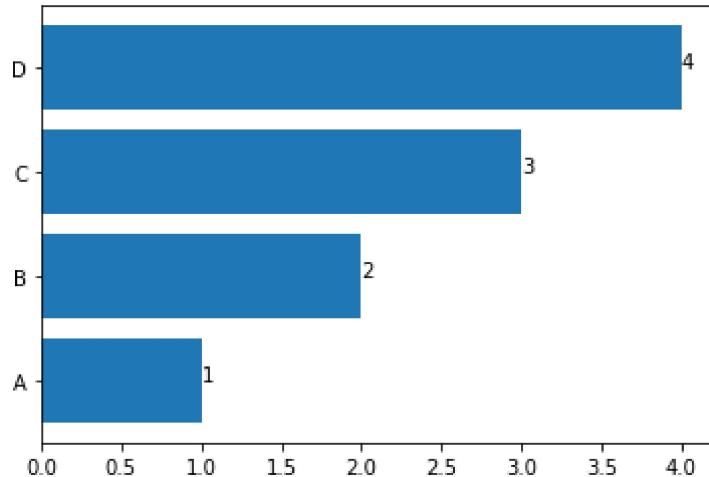


In [9]:

```
1 """example 2 show bar value
2 will explain subplot,figure,ax concept later
3 """
4 import os
5 import numpy as np
6 import matplotlib.pyplot as plt
7
8 x = [0, 1, 2, 3, 4, 5, 6, 7]
9 y = [160, 167, 17, 130, 120, 40, 105, 70]
10 fig, ax = plt.subplots()
11 width = 0.75
12 ind = np.arange(len(y))
13
14 ax.barh(ind, y, width, color = "green")
15
16 for i, v in enumerate(y):
17     ax.text(v + 3, i + .25, str(v),
18             color = 'blue', fontweight = 'bold')
19 plt.show()
```



```
In [10]: 1 import matplotlib.pyplot as plt
2 x = ["A", "B", "C", "D"]
3 y = [1, 2, 3, 4]
4 plt.barh(x, y)
5
6 for index, value in enumerate(y):
7     plt.text(value, index,
8             str(value))
9
10 plt.show()
```



4.2 figure()

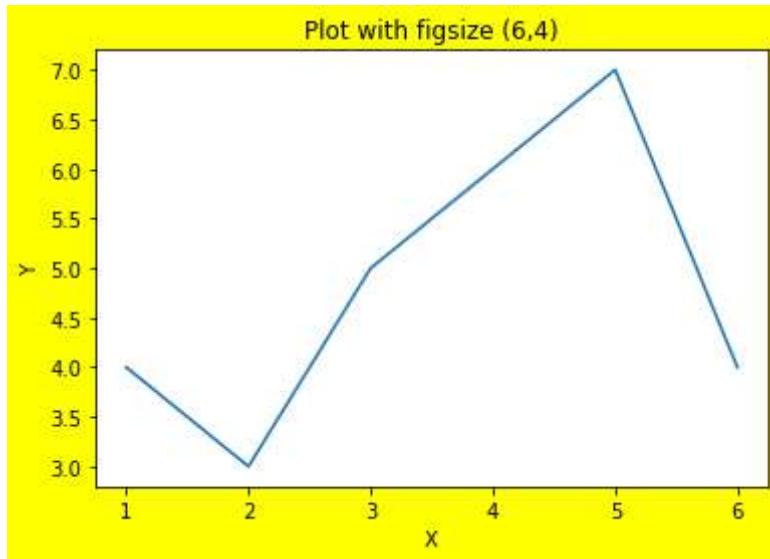
```
In [ ]: 1 """Syntax:  
2 matplotlib.pyplot.figure(num=None, figsize=None,  
3                         dpi=None, facecolor=None, edgecolor=None,  
4                         frameon=True, FigureClass=, clear=False, **kwargs)  
5 """  
6 """subplot()  
7 Returns: This method return the following values.  
8 fig : This method return the figure layout.  
9 ax : This method return the axes.Axes object or array of Axes objects.  
10 """
```

In [23]:

```

1 """example 1: simple figure()"""
2 x=[1,2,3,4,5,6]
3 y=[4,3,5,6,7,4]
4
5 plt.figure(figsize=(6,4),
6             facecolor='yellow')
7
8 plt.plot(x, y)
9 plt.xlabel("X")
10 plt.ylabel("Y")
11 #plt.xticks=(x,np.arange(1,10,step=3)) #failed,<>df.plot(),need add np.arange
12 plt.title("Plot with figsize (6,4)")
13 plt.show()

```



chapter 5 subplot() + axes()

```

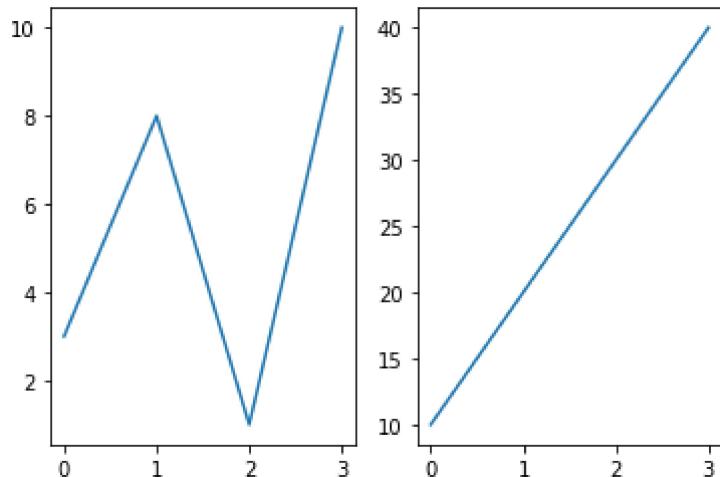
1 """
2 1=plt.figure() =containers
3 2=plt.subplot(1,2,1)=(width,height,no.of plot)
4 #graph 的高寬比例,只影響外觀,不影響答案。
5

```

```
1 plt.subplot(1, 2, 1)
2 #the figure has 1 row, 2 columns, and this plot is the first plot.
3 link:https://python-ecw.com/2020/05/26/python-matplotlib%E7%B9%AA%E5%9C%96%EF%BC%9Afigure%E5%92%8Caxes/
```

In [32]:

```
1 #2 subplot must be same arguement,else only show the later one
2 #plot 1:
3 x = np.array([0, 1, 2, 3])
4 y = np.array([3, 8, 1, 10])
5
6 plt.subplot(1, 2, 1)
7 plt.plot(x,y)
8
9 #plot 2:
10 x = np.array([0, 1, 2, 3])
11 y = np.array([10, 20, 30, 40])
12
13 plt.subplot(1, 2, 2)
14 plt.plot(x,y)
15
16 plt.show()
```



chapter 5.2 axes()

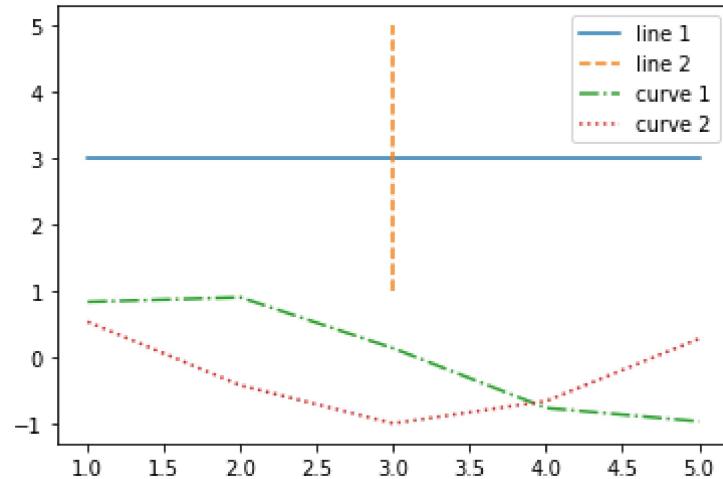
chapter 6. Plot Multiple lines

In []:

```
1 link : https://pythonguides.com/python-plot-multiple-lines/
```

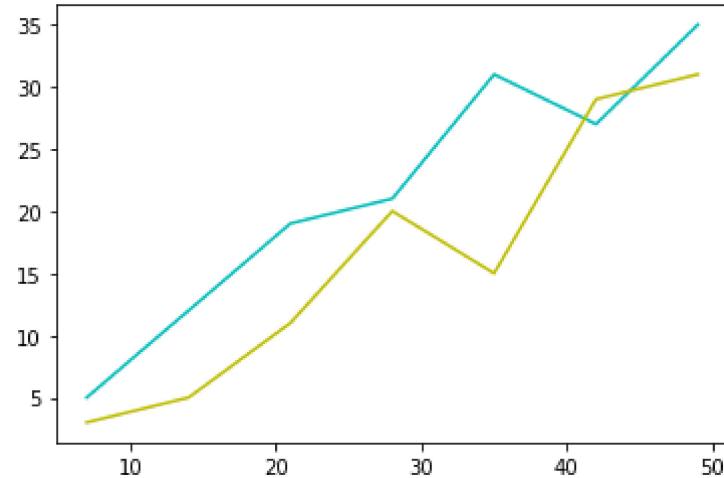
In [33]:

```
1 # based only 1 dataset to draw 4 Line
2 x = [1,2,3,4,5]
3 y = [3,3,3,3,3]
4
5 #4 plot.()=4 Lines
6 plt.plot(x, y, label = "line 1", linestyle="--")
7 plt.plot(y, x, label = "line 2", linestyle="-.")
8 plt.plot(x, np.sin(x), label = "curve 1", linestyle="-.")
9 plt.plot(x, np.cos(x), label = "curve 2", linestyle=":")
10 plt.legend()
11 plt.show()
```



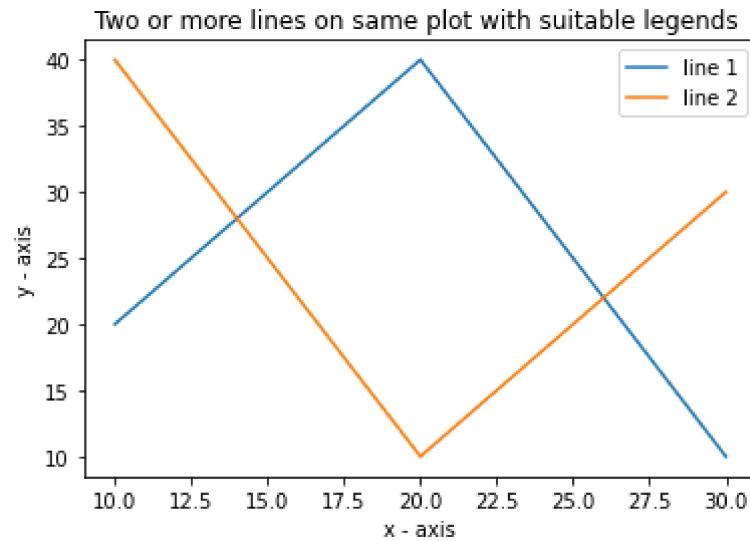
In [36]:

```
1 #2 data sets
2 x = [7, 14, 21, 28, 35, 42, 49]
3 y = [5, 12, 19, 21, 31, 27, 35]
4 z = [3, 5, 11, 20, 15, 29, 31]
5
6 # Plot a simple line chart
7 plt.plot(x, y, 'c')
8
9 # Plot another line on the same chart/graph
10 plt.plot(x, z, 'y')
11
12 plt.show()
```



In [35]:

```
1 #example 3 :
2 # Line 1 points
3 x1 = [10,20,30]
4 y1 = [20,40,10]
5 # plotting the Line 1 points
6 plt.plot(x1, y1, label = "line 1")
7 # Line 2 points
8 x2 = [10,20,30]
9 y2 = [40,10,30]
10 # plotting the Line 2 points
11 plt.plot(x2, y2, label = "line 2")
12
13 """below dataset, so commonly use"""
14 plt.xlabel('x - axis')
15 # Set the y axis Label of the current axis.
16 plt.ylabel('y - axis')
17 # Set a title of the current axes.
18 plt.title('Two or more lines on same plot with suitable legends ')
19 # show a legend on the plot
20 plt.legend()
21 # Display a figure.
22 plt.show()
```



chapter 6.2 plot multiple lines from array

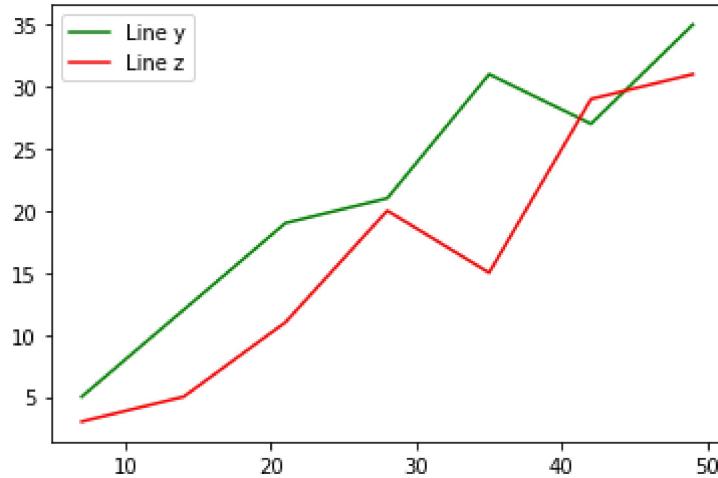
In [37]:

```
1 # Define data values in array
2 arr = np.array([[7, 5, 3], [14, 12, 5], [21, 19, 11],
3                 [28, 21, 20], [35, 31, 15], [42, 27, 29],
4                 [49, 35, 31]])
5
6 print(np.shape(arr), type(arr), arr, sep='\n')
7
```

```
(7, 3)
<class 'numpy.ndarray'>
[[ 7  5  3]
 [14 12  5]
 [21 19 11]
 [28 21 20]
 [35 31 15]
 [42 27 29]
 [49 35 31]]
```

In [38]:

```
1 plt.plot(arr[:, 0], arr[:, 1], 'g', label='Line y')
2
3 # Plot another line on the same chart/graph
4 plt.plot(arr[:, 0], arr[:, 2], 'r', label='Line z')
5
6 plt.legend()
7 plt.show()
```



chapter 6.3 multiple lines from dataframe

In [39]:

```

1 # Define data values by creating a Dataframe using a n-dimensional list
2 df = pd.DataFrame([[7, 5, 3], [14, 12, 5], [21, 19, 11],
3                   [28, 21, 20], [35, 31, 15], [42, 27, 29],
4                   [49, 35, 31]])
5
6 #add heading by using dict
7 df.rename(columns={0: 'x', 1: 'y', 2: 'z'}, inplace=True)
8
9 print(np.shape(df), type(df), df, sep='\n')
10

```

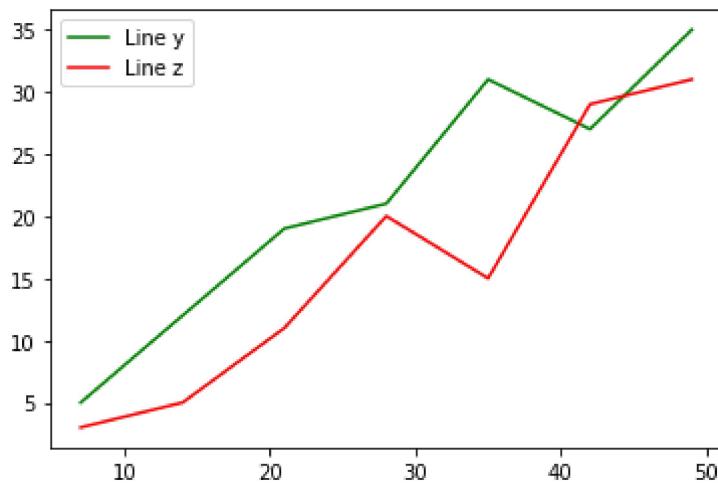
```
(7, 3)
<class 'pandas.core.frame.DataFrame'>
   x   y   z
0  7   5   3
1 14  12   5
2 21  19  11
3 28  21  20
4 35  31  15
5 42  27  29
6 49  35  31
```

In [40]:

```

1 # Plot a simple line chart
2 plt.plot(df['x'], df['y'], color='g', label='Line y')
3
4 # Plot another line on the same chart/graph
5 plt.plot(df['x'], df['z'], color='r', label='Line z')
6
7 plt.legend()
8 plt.show()

```



chapter 7. From Youtube

In [89]:

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt

```

```
1 plt.plot(x,y,color='green',ls='-',marker='o',lw=2.0,label="stock price")
```

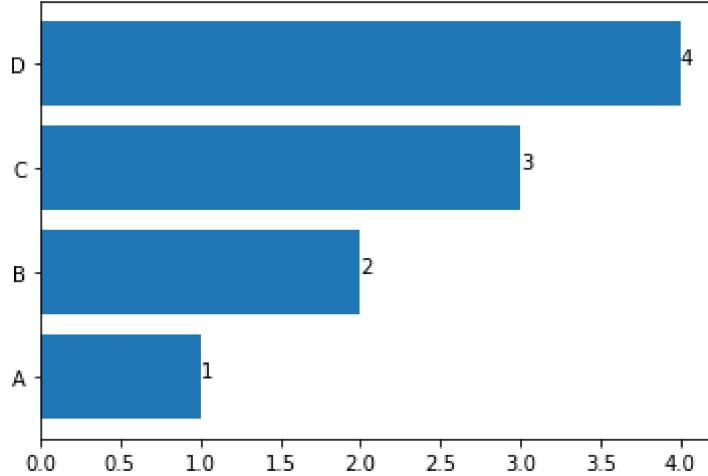
```
1 link:  
2 https://stackoverflow.com/questions/34162443/why-do-many-examples-use-fig-  
3 ax-plt-subplots-in-matplotlib-pyplot-python  
4 ~~~~~  
5 fig, ax = plt.subplots()  
6 #same as :  
7 fig = plt.figure()  
8 ax = fig.add_subplot(111)  
9 ~~~~~  
9 plt.subplots() is a function that returns a tuple containing a figure and  
axes object(s).  
10 Thus when using fig, ax = plt.subplots() you unpack this tuple into the  
variables fig and ax.  
11 簡單來將：就是 自動全部subplot(width,height,no_of_plot)  
12 same w+d,no_of_plot自動+1
```

In []:

```
1 # create a subplot with 2 rows and 1 columns  
2 fig, ax = plt.subplots(2,1)  
3  
4 Whereas, you can use plt.subplot() if you want to add the subplots separate  
5 It returns only the axis of one subplot.  
6  
7 fig = plt.figure() # create the canvas for plotting  
8 ax1 = plt.subplot(2,1,1)  
9 # (2,1,1) indicates total number of rows, columns, and figure number respect  
10 ax2 = plt.subplot(2,1,2)  
11
```

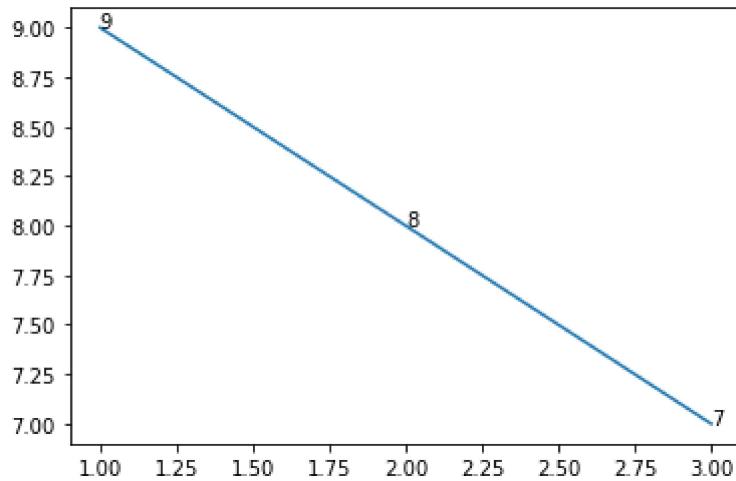
In [49]:

```
1 #show point values
2 x = ["A", "B", "C", "D"]
3 y = [1, 2, 3, 4]
4 plt.barh(x, y)
5
6 for index, value in enumerate(y):
7     plt.text(value, index, str(value))
8
```



In [55]:

```
1 #show y value
2 x=[1,2,3]
3 y=[9,8,7]
4 plt.plot(x,y)
5 for a,b in zip(x, y):
6     plt.text(a, b, str(b))
7 plt.show()
```



In [62]:

```

1 df_gas=pd.read_csv(r'C:\Users\fengs\Desktop\pd_real\mathplot\gas_prices.csv')
2 df_gas=df_gas.head(n=5)
3 df_gas

```

Out[62]:

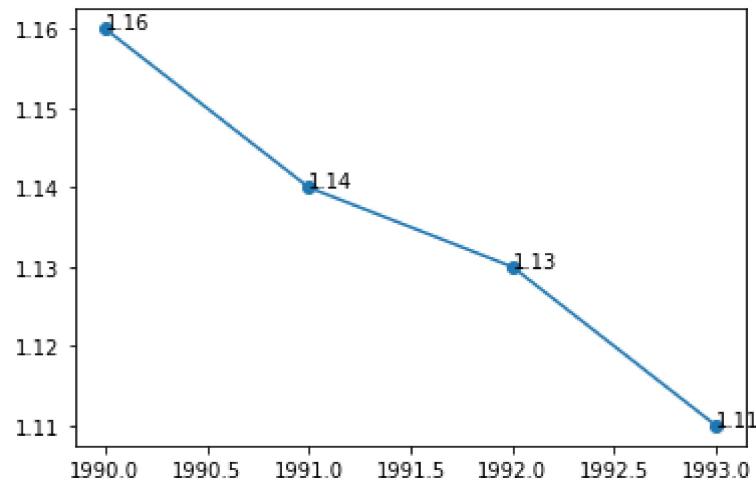
	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	UK	USA
0	1990	NaN	1.87	3.63	2.65	4.59	3.16	1.00	2.05	2.82	1.16
1	1991	1.96	1.92	3.45	2.90	4.50	3.46	1.30	2.49	3.01	1.14
2	1992	1.89	1.73	3.56	3.27	4.53	3.58	1.50	2.65	3.06	1.13
3	1993	1.73	1.57	3.41	3.07	3.68	4.16	1.56	2.88	2.84	1.11

In [64]:

```

1 #ex1: to show y value
2 x=df_gas['Year']
3 y=df_gas['USA']
4
5 plt.plot(x,y,marker='o',)
6 for a,b in zip(x,y):
7     plt.text(a,b, str(b))
8 plt.show()

```



In [108]:

```

1 #del plt
2 #the error:tuple' object is not callable ;due to the conflict of df.xticks
3
4 import matplotlib.pyplot as plt
5 import numpy as np
6 import pandas as pd

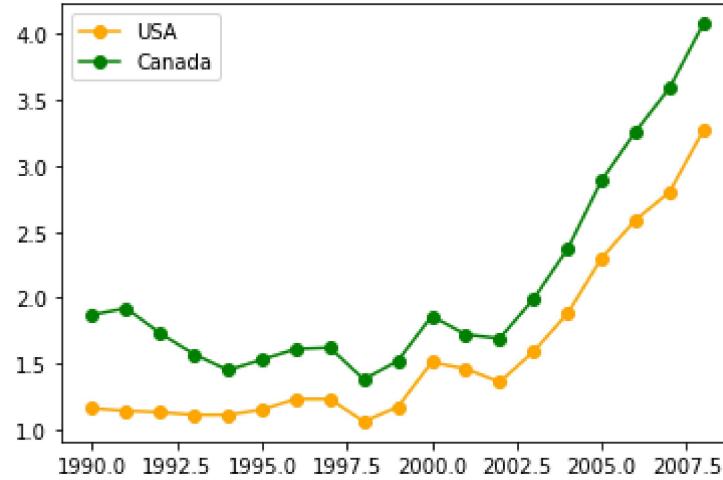
```

In [123]:

```

1 #ex2: show multiple line
2 #one line =one plot
3 x=df_gas['Year']
4 y1=df_gas['USA']
5 y2=df_gas['Canada']
6 plt.plot(x,y1,color='orange',marker='o',label='USA')
7 plt.plot(x,y2,color='green',marker='o',label='Canada')
8 #plt.xticks(df_gas.Year[::3].tolist())
9 #plt.xticks(x)
10 plt.legend()
11 plt.show()

```



In [125]:

```

1 print(df_gas['Year'][::3]) #dataframe range
2 print(type(df_gas['Year']))

```

```

0    1990
3    1993
6    1996
9    1999
12   2002
15   2005
18   2008
Name: Year, dtype: int64
<class 'pandas.core.series.Series'>

```

In [128]:

```
1 #copy from model answer
2 gas = pd.read_csv(r'C:\Users\fengs\Desktop\pd_real\mathplot\gas_prices.csv')
3
4 plt.figure(figsize=(8,5))
5
6 plt.title('Gas Prices over Time (in USD)', fontdict={'fontweight':'bold', 'f
7
8 plt.plot(gas.Year, gas.USA, 'b.-', label='United States')
9 plt.plot(gas.Year, gas.Canada, 'r.-')
10 plt.plot(gas.Year, gas['South Korea'], 'g.-')
11 plt.plot(gas.Year, gas.Australia, 'y.-')
12
13 # Another Way to plot many values!
14 # countries_to_Look_at = ['Australia', 'USA', 'Canada', 'South Korea']
15 # for country in gas:
16 #     if country in countries_to_Look_at:
17 #         plt.plot(gas.Year, gas[country], marker='.')
18
19
20
21 plt.xlabel('Year')
22 plt.ylabel('US Dollars')
23 plt.xticks(gas.Year[::3].tolist()+[2011])
24 plt.legend()
25
26 #plt.savefig('Gas_price_figure.png', dpi=300)
27
28 plt.show()
```

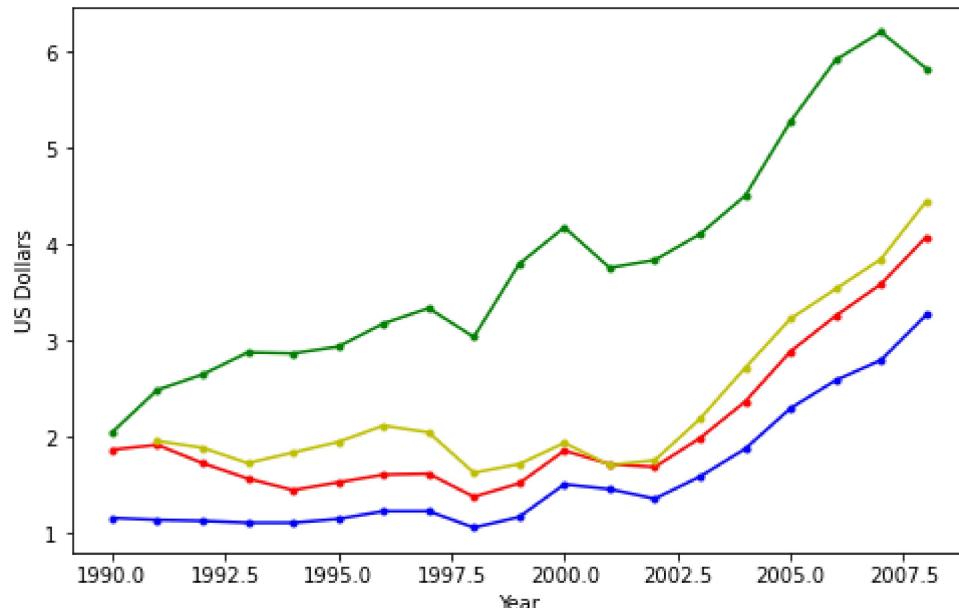
TypeError

Traceback (most recent call last)

```
~\AppData\Local\Temp\ipykernel_2900\1035432621.py in <module>
      21 plt.xlabel('Year')
      22 plt.ylabel('US Dollars')
----> 23 plt.xticks(gas.Year[::3].tolist()+[2011])
      24 plt.legend()
      25
```

TypeError: 'tuple' object is not callable

Gas Prices over Time (in USD)



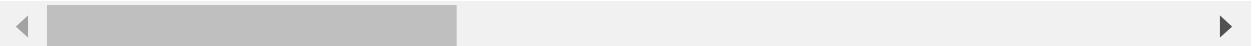
chapter 2 :fifa example

```
In [113]: 1 df_fifa=pd.read_csv(r'C:\Users\fengs\Desktop\pd_real\mathplot\fifa_data.csv'
2 df_fifa.head(4)
```

Out[113]:

	Unnamed: 0	ID	Name	Age	Photo	Nationality
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentina https://cc
1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portugal https://cc
2	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Brazil https://cc
3	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spain https://cc

4 rows × 89 columns



2.1 histogram

```
1 matplotlib.pyplot.xticks(ticks=None, labels=None, **kwargs)
```

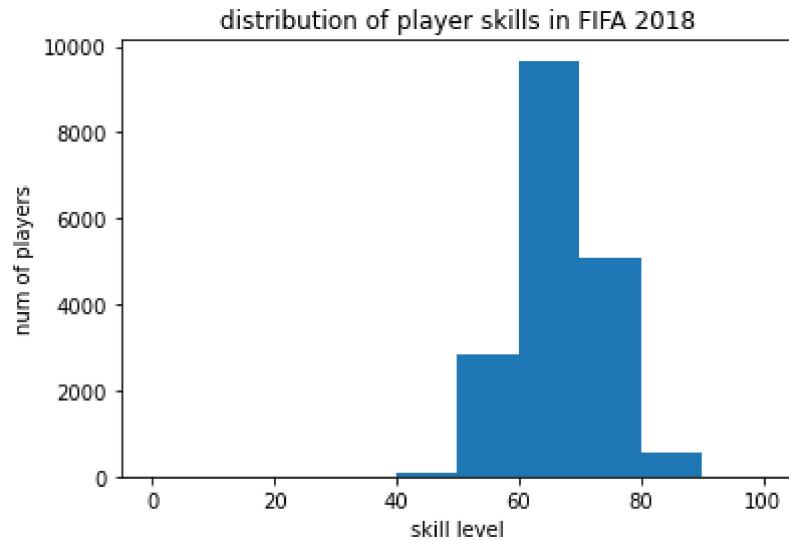
In [119]:

```
1 bins=[0,10,20,30,40,50,60,70,80,90,100]
2 plt.hist(df_fifa.Overall,bins=bins)
3
4 plt.xlabel('skill level')
5 plt.ylabel('num of players')
6 plt.title("distribution of player skills in FIFA 2018")
7 plt.xticks(bins) # AS this prompt error ,just put it at the bottum
8 plt.show()
```

TypeError

Traceback (most recent call last)

```
~\AppData\Local\Temp\ipykernel_2900/3977316486.py in <module>
      5 plt.ylabel('num of players')
      6 plt.title("distribution of player skills in FIFA 2018")
----> 7 plt.xticks(bins)
      8 plt.show()
```

TypeError: 'tuple' object is not callable

chapter 3: pie chart

In []:

1

