

Pandas

- Stands for PanelData
- One of the data analysis library
- Used for data manipulation, analysis, and cleaning.
- Contains mainly two data structures in it
 1. Series
 2. DataFrames
- Series: is a sequence of data
- DataFrame: is a data with rows and columns

In [1]:

```
1 import pandas as pd
```

In [2]:

```
1 pd.__version__
```

Out[2]:

```
'1.0.5'
```

In [3]:

```
1 # Creating data with series:
2
3 a = pd.Series([1,2,3,4,5,6])
4 print(a)
```

```
0    1
1    2
2    3
3    4
4    5
5    6
dtype: int64
```

In [4]:

```
1 # Creating data series using numpy array
2
3 import numpy as np
4 a = np.array([1,2,3,4,5,6])
5 a
```

Out[4]:

```
array([1, 2, 3, 4, 5, 6])
```

In [5]:

```
1 pd.Series(a)
```

Out[5]:

```
0    1
1    2
2    3
3    4
4    5
5    6
dtype: int32
```

In [6]:

```
1 # Changing the index values in series:
2
3 s = pd.Series([10,11,12,13,14], index = [1,2,3,4,5])
4 print(s)
```

```
1    10
2    11
3    12
4    13
5    14
dtype: int64
```

In [8]:

```
1 s = pd.Series([2,4,6,8,10], index = np.arange(1,6))
2 print(s)
3 print(s.dtype)
```

```
1    2
2    4
3    6
4    8
5   10
dtype: int64
int64
```

In [9]:

```
1 s2 = pd.Series(['keerthi',221,9.8], index = ['str','int','float'])
2 print(s2)
```

```
str      keerthi
int         221
float        9.8
dtype: object
```

In [10]:

```
1 # Creating Data Series using dictionary
2
3 dic = {'key1': 'Keerthi', 'key2': 221, 'key3': 9.0}
4 pd.Series(dic)
```

Out[10]:

```
key1    Keerthi
key2      221
key3       9
dtype: object
```

In [12]:

```
1 # Creating data series using tuple:
2
3 t = ('a', 'b', 23, 6.8, 78)
4 pd.Series(t, np.arange(1, 6))
```

Out[12]:

```
1    a
2    b
3    23
4    6.8
5    78
dtype: object
```

In [13]:

```
1 s = pd.Series(np.random.randint(20, 40, 8))
2 print(s)
```

```
0    31
1    37
2    34
3    39
4    20
5    34
6    21
7    32
dtype: int32
```

In [14]:

```
1 s[4]
```

Out[14]:

```
20
```

In [15]:

```
1 s[2:5]
```

Out[15]:

```
2    34
3    39
4    20
dtype: int32
```

In [17]:

```
1 len(s)
```

Out[17]:

```
8
```

In [18]:

```
1 s[len(s)-1]
```

Out[18]:

```
32
```

In [19]:

```
1 s[6] = 'kits'
```

In [20]:

```
1 print(s)
```

```
0    31
1    37
2    34
3    39
4    20
5    34
6   kits
7     32
dtype: object
```

In [21]:

```
1 d = pd.DataFrame([1,2,3,4,5,6])
2 d
```

Out[21]:

```
   0
0  1
1  2
2  3
3  4
4  5
5  6
```

In [23]:

```
1 s = 'Hey I am keerthi'
2 print(len(s))
```

16

In [24]:

```
1 d = 'Hey I am keerthi'.split()
2 print(d)
3 print(len(d))
```

['Hey', 'I', 'am', 'keerthi']

4

In [25]:

```
1 d = 'Hey I am keerthi'.split('a')
2 print(d)
3 print(len(d))
```

['Hey I ', 'm keerthi']

2

In [26]:

```
1 d = 'Hey I am keerthi'.split()
2 print(pd.DataFrame(d))
3
```

```
   0
0  Hey
1    I
2   am
3 keerthi
```

In [30]:

```
1 lst = [[1,2,3],[5,6,7]]
2 pd.DataFrame(lst, columns = ('a','b','c'),index = ['list1','list2'])
```

Out[30]:

	a	b	c
list1	1	2	3
list2	5	6	7

In [36]:

```
1 a = [['Sireesha',120,'A'],['Sindhu',131,'B'],['Nandini',222,'A']]
2 data = pd.DataFrame(a, columns=('Name','Rollno','Grades'), index = [1,2,3])
3 data
```

Out[36]:

	Name	Rollno	Grades
1	Sireesha	120	A
2	Sindhu	131	B
3	Nandini	222	A

In [37]:

```
1 data.index
```

Out[37]:

```
Int64Index([1, 2, 3], dtype='int64')
```

In [38]:

```
1 data['Rollno']
```

Out[38]:

```
1    120
2    131
3    222
Name: Rollno, dtype: int64
```

In [39]:

```
1 data['Rollno'][3]
```

Out[39]:

```
222
```

In [40]:

```
1 data[-1:]
```

Out[40]:

	Name	Rollno	Grades
3	Nandini	222	A

Matplotlib

- Is a cross-platform data visualization library for making 2D plots from data

Types of plotting

- * Line plot
- * Bar plot
- * Scatter plot
- * Area plot
- * Pie chart

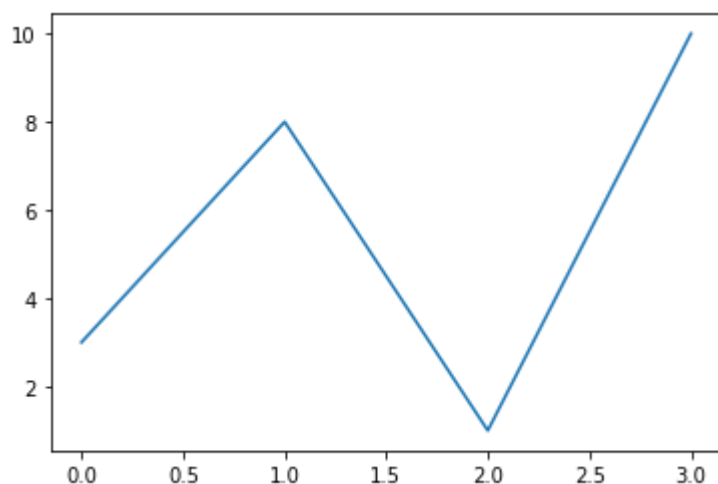
In [41]:

```
1 import matplotlib.pyplot as plt
```

Line plot

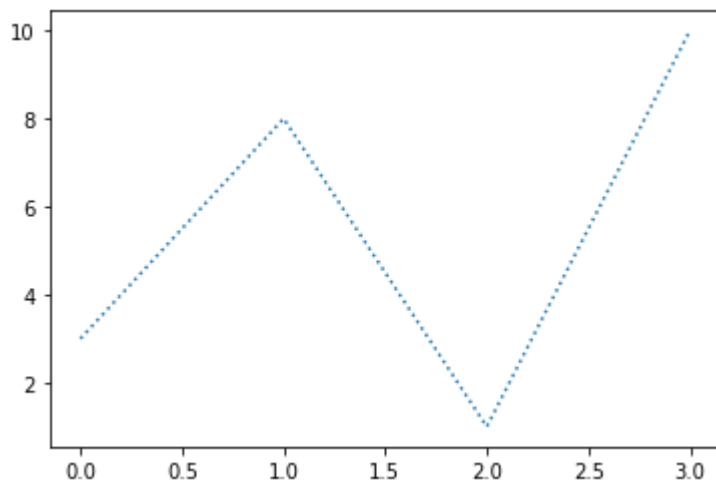
In [42]:

```
1 import numpy as np
2
3 ypoints = np.array([3,8,1,10])
4 plt.plot(ypoints)
5 plt.show()
```



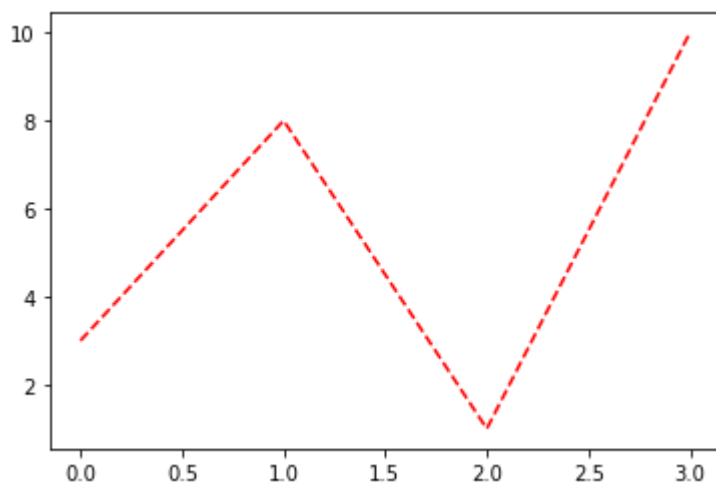
In [43]:

```
1 ypoints = np.array([3,8,1,10])
2 plt.plot(ypoints, linestyle='dotted')
3 plt.show()
```



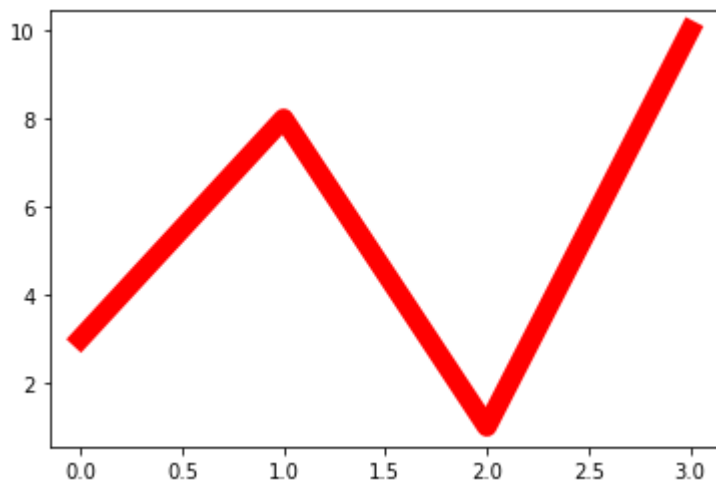
In [45]:

```
1 ypoints = np.array([3,8,1,10])
2 plt.plot(ypoints, linestyle='dashed', color='r')
3 plt.show()
```



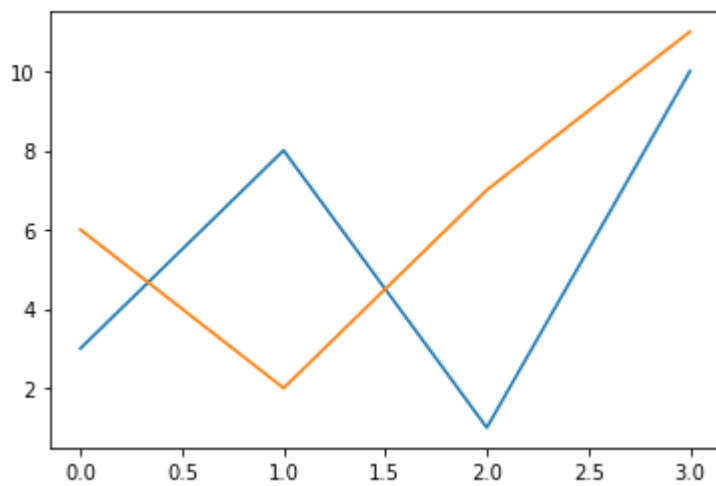
In [47]:

```
1 ypoints = np.array([3,8,1,10])
2 plt.plot(ypoints, color='r', linewidth=10)
3 plt.show()
```



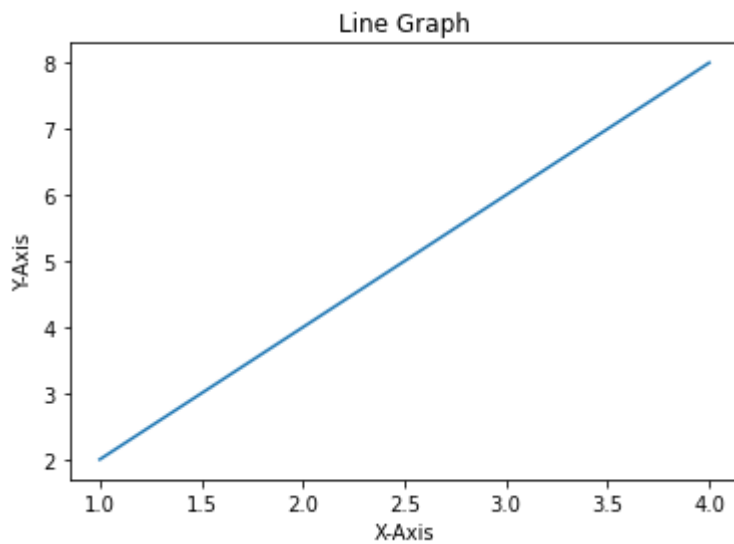
In [49]:

```
1 y1 = np.array([3,8,1,10])
2 y2 = np.array([6,2,7,11])
3
4 plt.plot(y1)
5 plt.plot(y2)
6
7 plt.show()
```



In [52]:

```
1 plt.plot([1,2,3,4],[2,4,6,8])  
2 plt.title('Line Graph')  
3 plt.xlabel('X-Axis')  
4 plt.ylabel('Y-Axis')  
5 plt.show()
```



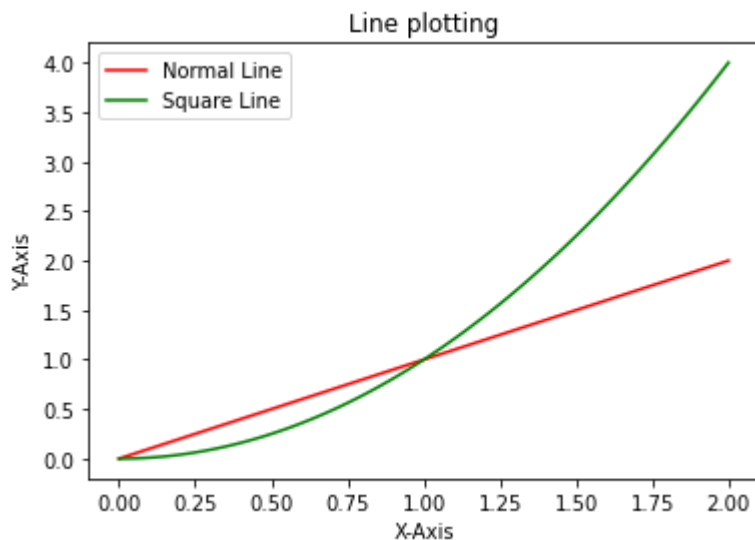
In [54]:

```
1 x = np.linspace(0,2,100)
2 print(x)
```

```
[0.          0.02020202 0.04040404 0.06060606 0.08080808 0.1010101
0.12121212 0.14141414 0.16161616 0.18181818 0.2020202 0.22222222
0.24242424 0.26262626 0.28282828 0.3030303 0.32323232 0.34343434
0.36363636 0.38383838 0.4040404 0.42424242 0.44444444 0.46464646
0.48484848 0.50505051 0.52525253 0.54545455 0.56565657 0.58585859
0.60606061 0.62626263 0.64646465 0.66666667 0.68686869 0.70707071
0.72727273 0.74747475 0.76767677 0.78787879 0.80808081 0.82828283
0.84848485 0.86868687 0.88888889 0.90909091 0.92929293 0.94949495
0.96969697 0.98989899 1.01010101 1.03030303 1.05050505 1.07070707
1.09090909 1.11111111 1.13131313 1.15151515 1.17171717 1.19191919
1.21212121 1.23232323 1.25252525 1.27272727 1.29292929 1.31313131
1.33333333 1.35353535 1.37373737 1.39393939 1.41414141 1.43434343
1.45454545 1.47474747 1.49494949 1.51515152 1.53535354 1.55555556
1.57575758 1.5959596 1.61616162 1.63636364 1.65656566 1.67676768
1.6969697 1.71717172 1.73737374 1.75757576 1.77777778 1.7979798
1.81818182 1.83838384 1.85858586 1.87878788 1.8989899 1.91919192
1.93939394 1.95959596 1.97979798 2.          ]
```

In [55]:

```
1 plt.plot(x, x, color='r', label = 'Normal Line')
2 plt.plot(x, x**2, color='g', label = 'Square Line')
3 plt.xlabel('X-Axis')
4 plt.ylabel('Y-Axis')
5 plt.title('Line plotting')
6 plt.legend()
7 plt.show()
```

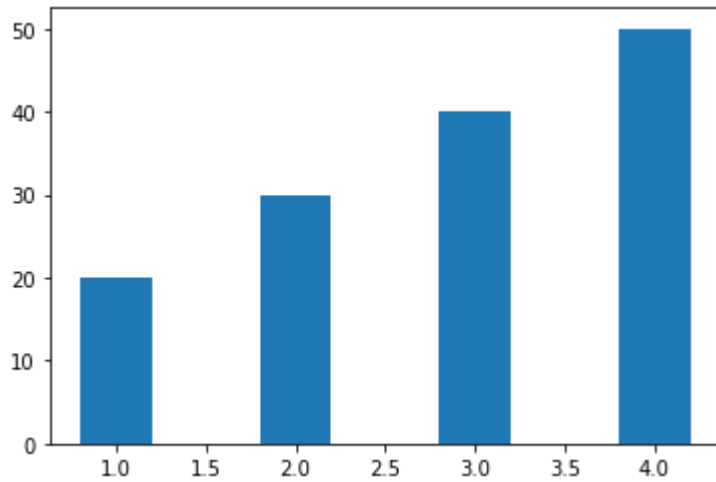


Bar Plot

- Used to create a Bar chat

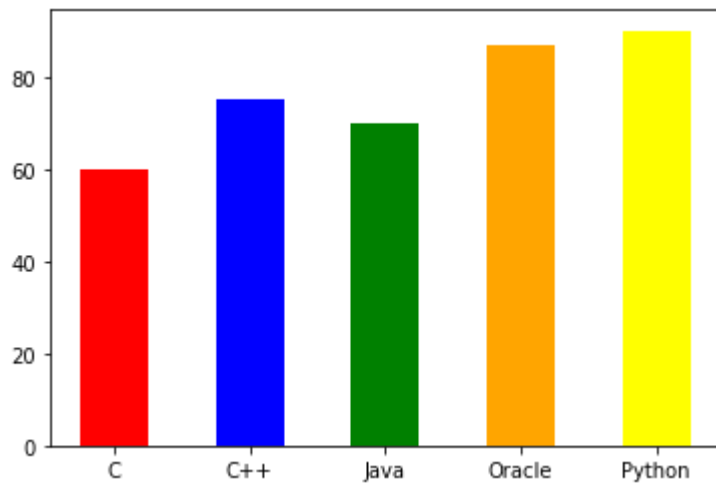
In [57]:

```
1 plt.bar([1,2,3,4],[20,30,40,50], width = 0.4)
2 plt.show()
```



In [58]:

```
1 marks = [60, 75, 70, 87, 90]
2 sbj = ['C', 'C++', 'Java', 'Oracle', 'Python']
3 plt.bar(sbj, marks, color=['red', 'blue', 'green', 'orange', 'yellow'], width=0.5)
4 plt.show()
```

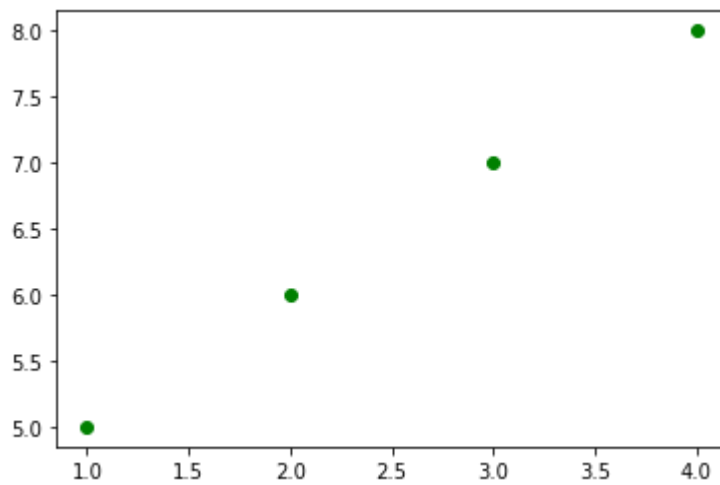


Scatter plot

- A scatter plot is a diagram where each value in the data set is represented but a dot.

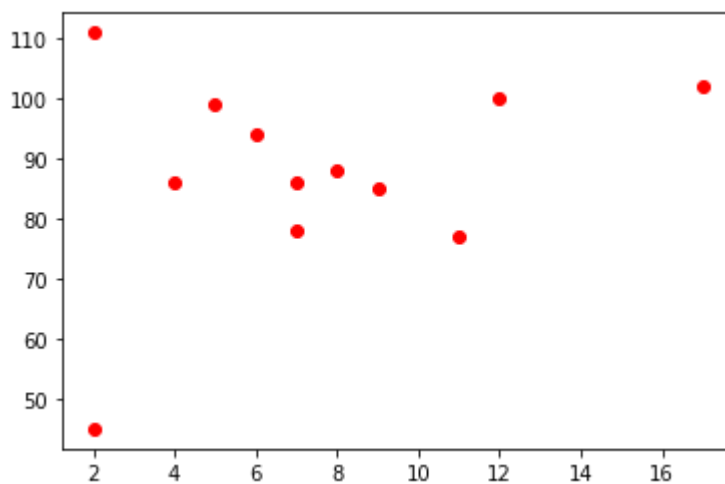
In [59]:

```
1 plt.scatter([1,2,3,4],[5,6,7,8], color='g')
2 plt.show()
```



In [60]:

```
1 x = [5,7,2,8,7,2,17,9,4,11,12,6]
2 y = [99,86,45,88,78,111,102,85,86,77,100,94]
3
4 plt.scatter(x,y, color = 'r')
5 plt.show()
```



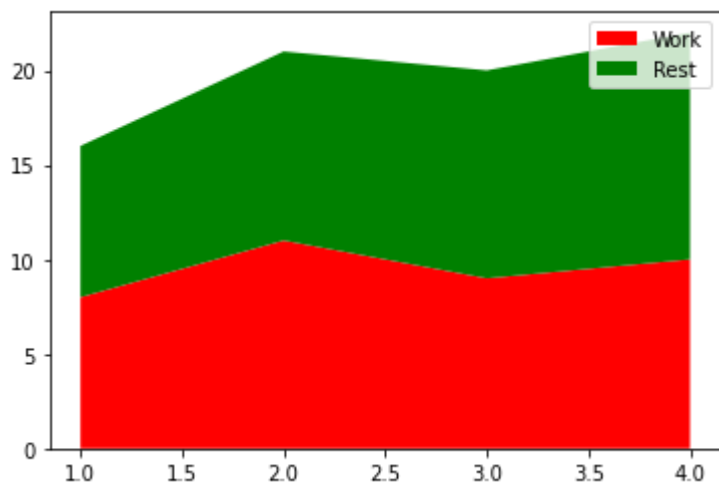
Area plot

In [61]:

```

1 days = [1,2,3,4]
2 working = [8,11,9,10]
3 sleeping = [8,10,11,12]
4 plt.stackplot(days, working, sleeping, colors=['r','g'], labels=['Work','Rest'])
5 plt.legend()
6 plt.show()

```



Box plot

In [62]:

```

1 marks = np.random.randint(20,100,101)
2 print(marks)

```

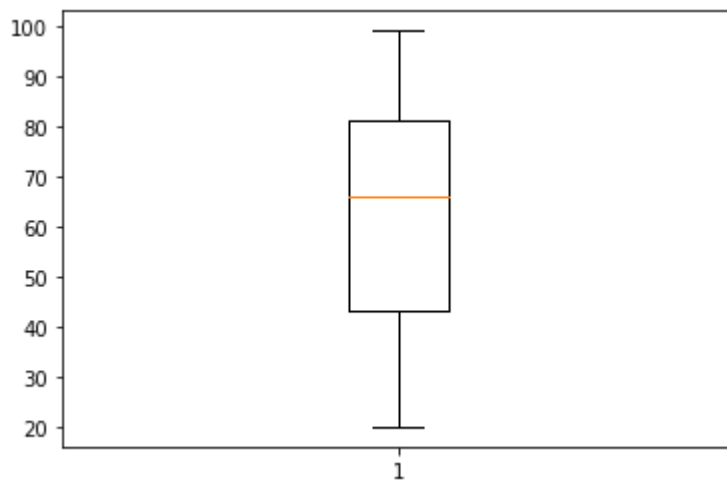
```

[60 74 27 95 43 99 81 42 47 25 20 44 66 28 59 62 34 32 26 21 21 37 64 76
 49 77 68 58 62 91 62 94 57 50 80 66 40 20 90 89 90 93 96 71 47 54 90 56
 87 75 34 63 67 94 99 26 99 70 60 81 78 45 23 34 67 28 69 32 82 75 87 95
 40 54 65 88 45 91 80 72 67 31 38 52 91 73 37 90 71 56 81 39 74 67 70 91
 84 62 32 78 78]

```

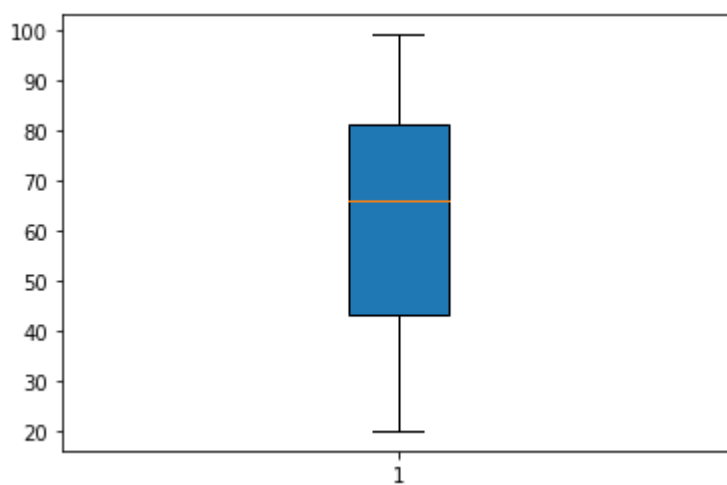
In [63]:

```
1 plt.boxplot(marks)
2 plt.show()
```



In [64]:

```
1 plt.boxplot(marks, patch_artist=True)
2 plt.show()
```

**Pie chart**

In [70]:

```
1 marks = [60,72, 50, 87, 90, 40]
2 sbj = ['C', 'C++', 'Java', 'Oracle', 'Python', 'Django']
3 plt.pie(marks, labels=sbj)
4 plt.legend()
```

Out[70]:

<matplotlib.legend.Legend at 0x2093cc0fdc0>

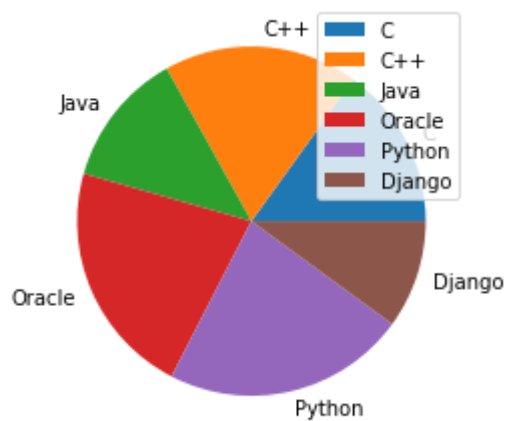
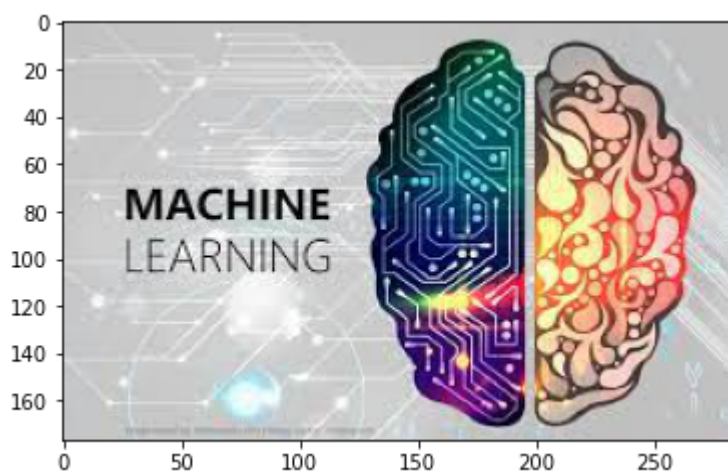


Image reading in Matplotlib

In [71]:

```
1 print(plt.imshow(plt.imread('ml.jpg')))
```

AxesImage(54,36;334.8x217.44)



In [72]:

```
1 print(dir(plt))
```

```
['Annotation', 'Arrow', 'Artist', 'AutoLocator', 'Axes', 'Button', 'Circle',
'Figure', 'FigureCanvasBase', 'FixedFormatter', 'FixedLocator', 'FormatStrFo
rmatter', 'Formatter', 'FuncFormatter', 'GridSpec', 'IndexLocator', 'Line2
D', 'LinearLocator', 'Locator', 'LogFormatter', 'LogFormatterExponent', 'Log
FormatterMathtext', 'LogLocator', 'MaxNLocator', 'MultipleLocator', 'Normali
ze', 'NullFormatter', 'NullLocator', 'Number', 'PolarAxes', 'Polygon', 'Rect
angle', 'ScalarFormatter', 'Slider', 'Subplot', 'SubplotTool', 'Text', 'Tick
Helper', 'Widget', '_INSTALL_FIG_OBSERVER', '_IP_REGISTERED', '__builtins_
__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__packag
e__', '__spec__', '_auto_draw_if_interactive', '_backend_mod', '_get_running
_interactive_framework', '_interactive_bk', '_log', '_pylab_helpers', '_set
p', '_setup_pyplot_info_docstrings', '_show', 'acorr', 'angle_spectrum', 'an
notate', 'arrow', 'autoscale', 'autumn', 'axes', 'axhline', 'axhspan', 'axi
s', 'axvline', 'axvspan', 'bar', 'barbs', 'barh', 'bone', 'box', 'boxplot',
'broken_barh', 'cbook', 'cla', 'clabel', 'clf', 'clim', 'close', 'cm', 'cohe
re', 'colorbar', 'colormaps', 'connect', 'contour', 'contourf', 'cool', 'cop
per', 'csd', 'cycler', 'dedent', 'delaxes', 'deprecated', 'disconnect', 'doc
string', 'draw', 'draw_all', 'draw_if_interactive', 'errorbar', 'eventplot',
'figaspect', 'figimage', 'figlegend', 'fignum_exists', 'figtext', 'figure',
'fill', 'fill_between', 'fill_betweenx', 'findobj', 'flag', 'functools', 'gc
a', 'gcf', 'gci', 'get', 'get_backend', 'get_cmap', 'get_current_fig_manage
r', 'get_figlabels', 'get_fignums', 'get_plot_commands', 'get_scale_docs',
'get_scale_names', 'getp', 'ginput', 'gray', 'grid', 'hexbin', 'hist', 'hist
2d', 'hlines', 'hot', 'hsv', 'importlib', 'imread', 'imsave', 'imshow', 'inf
erno', 'inspect', 'install_repl_displayhook', 'interactive', 'ioff', 'ion',
'isinteractive', 'jet', 'legend', 'locator_params', 'logging', 'loglog', 'ma
gma', 'magnitude_spectrum', 'margins', 'matplotlib', 'matshow', 'minorticks_
off', 'minorticks_on', 'mlab', 'new_figure_manager', 'nipy_spectral', 'np',
'pause', 'pcolor', 'pcolormesh', 'phase_spectrum', 'pie', 'pink', 'plasma',
'plot', 'plot_date', 'plotfile', 'plotting', 'polar', 'prism', 'psd', 'quive
r', 'quiverkey', 'rc', 'rcParams', 'rcParamsDefault', 'rcParamsOrig', 'rc_co
ntext', 'rcdefaults', 'rcsetup', 're', 'register_cmap', 'rgrids', 'savefig',
'sca', 'scatter', 'sci', 'semilogx', 'semilogy', 'set_cmap', 'set_loglevel',
'setp', 'show', 'silent_list', 'specgram', 'spring', 'spy', 'stackplot', 'st
em', 'step', 'streamplot', 'style', 'subplot', 'subplot2grid', 'subplot_too
l', 'subplots', 'subplots_adjust', 'summer', 'suptitle', 'switch_backend',
'sys', 'table', 'text', 'thetagrids', 'tick_params', 'ticklabel_format', 'ti
ght_layout', 'time', 'title', 'tricontour', 'tricontourf', 'tripcolor', 'tri
plot', 'twinx', 'twiny', 'uninstall_repl_displayhook', 'violinplot', 'viridi
s', 'vlines', 'waitforbuttonpress', 'warn_deprecated', 'winter', 'xcorr', 'x
kcd', 'xlabel', 'xlim', 'xscale', 'xticks', 'ylabel', 'ylim', 'yscale', 'yti
cks']
```

In [73]:

```
1 print(dir(pd))
```

```
['BooleanDtype', 'Categorical', 'CategoricalDtype', 'CategoricalIndex', 'DataFrame', 'DateOffset', 'DatetimeIndex', 'DatetimeTZDtype', 'ExcelFile', 'ExcelWriter', 'Float64Index', 'Grouper', 'HDFStore', 'Index', 'IndexSlice', 'Interval', 'IntervalDtype', 'IntervalIndex', 'MultiIndex', 'NA', 'NaT', 'NamedAgg', 'Period', 'PeriodDtype', 'PeriodIndex', 'RangeIndex', 'Series', 'SparseDtype', 'StringDtype', 'Timedelta', 'TimedeltaIndex', 'Timestamp', 'UInt16Dtype', 'UInt32Dtype', 'UInt64Dtype', 'UInt64Index', 'UInt8Dtype', '__builtins__', '__cached__', '__doc__', '__docformat__', '__file__', '__getattr__', '__git_version__', '__loader__', '__name__', '__package__', '__path__', '__spec__', '__version__', '_config', '_hashtable', '_is_numpy_dev', '_lib', '_libs', '_np_version_under1p14', '_np_version_under1p15', '_np_version_under1p16', '_np_version_under1p17', '_np_version_under1p18', '_testing', '_tslib', '_typing', '_version', 'api', 'array', 'arrays', 'bdate_range', 'compat', 'concat', 'core', 'crosstab', 'cut', 'date_range', 'describe_option', 'errors', 'eval', 'factorize', 'get_dummies', 'get_option', 'infer_freq', 'interval_range', 'io', 'isna', 'isnull', 'json_normalize', 'lreshape', 'melt', 'merge', 'merge_asof', 'merge_ordered', 'notna', 'notnull', 'offsets', 'option_context', 'options', 'pandas', 'period_range', 'pivot', 'pivot_table', 'plotting', 'qcut', 'read_clipboard', 'read_csv', 'read_excel', 'read_feather', 'read_fwf', 'read_gbq', 'read_hdf', 'read_html', 'read_json', 'read_orc', 'read_parquet', 'read_pickle', 'read_sas', 'read_spss', 'read_sql', 'read_sql_query', 'read_sql_table', 'read_stata', 'read_table', 'reset_option', 'set_eng_float_format', 'set_option', 'show_versions', 'test', 'testing', 'timedelta_range', 'to_datetime', 'to_numeric', 'to_pickle', 'to_timedelta', 'tseries', 'unique', 'util', 'value_counts', 'wide_to_long']
```

In []:

```
1
```