Plotting Points

FOSSEE

1 Plotting Points with Lists

```
In []: x = [0, 1, 2, 3] # Creating a list
In []: y = [7, 11, 15, 19]
In []: plot(x, y)
In []: clf()
In []: plot(x, y, 'o')  # Plotting Circles
```

1.1 Line style/marker

The following format string characters are accepted to control the line style **or** marker:

character	description
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker

```
'*'                 star marker
'h'                 hexagon1 marker
'H'                 hexagon2 marker
'+'                 plus marker
'x'                 x marker
'D'                 diamond marker
'd'                 thin_diamond marker
'|'                 vline marker
'_'                 hline marker
================    ==============================
```

1.2 Marker combinations

```
In []: plot(x, y, 'ro')
```
This plots figure with red colored filled circles.
Similarly other combination of colors and marker can be used.

2 Lists

Initializing

```
In []: mtlist = []       # Empty List
In []: lst = [ 1, 2, 3, 4, 5]
```

Slicing

```
In []: lst[1:3]          # A slice.
Out[]: [2, 3]

In []: lst[1:-1]
Out[]: [2, 3, 4]
```

2.1 Appending to lists

```
In []: a = [ 6, 7, 8, 9]
In []: b = lst + a
In []: b
Out[]: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In []: lst.append(6)
In []: lst
Out[]: [ 1, 2, 3, 4, 5, 6]
```

2.2 Iterating over a List

```
In []: for element in b:  # Iterating over the list, element-wise
 ....:     print element       # Print each element
 ....:
```

3 Strings

3.1 Splitting Strings

```
In []: greet = ``hello world''
In []: print greet.split()
Out[]: ['hello', 'world']
In []: greet = ``hello, world''
In []: print greet.split(',')
Out[]: ['hello', ' world'] # Note the white space before 'world'
```

A string can be split based on the delimiter specified within quotes. A combination of more than one delimiter can also be used.

```
In []: greet.split(', ')
Out[]: ['hello', 'world']
```

Note the white space is not there anymore.

4 Plotting from Files

4.1 Opening files

```
In []: f = open('datafile.txt')
```
By default opens in read mode.
If file does not exist then it throws an exception
```
In []: f = open('datafile.txt','r')
```
Specifying the read mode
```
In []: f = open('datafile.txt', 'w')
```
Opens the file in write mode.
If the file already exists, then it deletes all the previous content and opens.

4.2 Reading from files

Just like lists files are iterable as well.

```
In []: for line in f:
  ...:     print line
  ...:
  ...:
```

4.3 Plotting

```
l = []
t = []
for line in open('pendulum.txt'):
    point = line.split()
    l.append(float(point[0]))
    t.append(float(point[1]))
tsq = []
for time in t:
    tsq.append(time*time)
plot(l, tsq, '.')
```