1. Create a list named my_list in python with the following data points

In [9]:
```
my_list = [45.4, 44.2, 36.8, 35.1, 39.0, 60.0, 47.4, 41.1, 45.8, 35.6]
print(my_list)
```

```
[45.4, 44.2, 36.8, 35.1, 39.0, 60.0, 47.4, 41.1, 45.8, 35.6]
```

In [6]:
```
# a.    Print the 5th element in the list
my_list[4]
```

Out[6]:
```
39.0
```

In [11]:
```
# b.    Append 55.2 to my_list
my_list.append(55.2)
print(my_list)
```

```
[45.4, 44.2, 36.8, 35.1, 39.0, 47.4, 41.1, 45.8, 35.6, 55.2]
```

In [12]:
```
# c.    Remove the 6th element in the list
my_list.remove(my_list[5])
print(my_list)
```

```
[45.4, 44.2, 36.8, 35.1, 39.0, 41.1, 45.8, 35.6, 55.2]
```

In [15]:
```
# d.    Iterate over the list to print data points greater than 45
for i in range(0,len(my_list)):
    #print(i)
    if my_list[i] > 45:
        print(my_list[i])
```

```
45.4
45.8
55.2
```

2. Introduction to numpy –

In [17]:
```
# a.    Import the numpy library using the following command – import numpy
import numpy
```

In [19]:
```
# b.    Declare numpy array with the same data points as in my_list using numpy.a
my_list = numpy.array(my_list)
```

In [21]:
```
# c.    Compute the mean and standard deviation using numpy.mean() and numpy.std(
print(numpy.mean(my_list))
print(numpy.std(my_list))
```

```
42.0222222222
6.05984557518
```

In [23]:
```
# d.    Use logical referencing to get only those values that are less than 45
my_list[my_list<45]
```

Out[23]:
```
array([ 44.2,  36.8,  35.1,  39. ,  41.1,  35.6])
```

In [24]:
```
# e.    Compute the max and min of the array using numpy.max() and numpy.min()
print(numpy.max(my_list))
print(numpy.min(my_list))
```

```
55.2
35.1
```

3. Introduction to pandas –

In [26]:
```
# a.    Import the pandas library – import pandas
import pandas
```

In [33]:
```
# b.    Read the IRIS dataset into iris using pandas.read_csv(). Data file –
iris = pandas.read_csv("Iris.csv")
```

In [34]:
```
# c.    Using iris.head(), display the head of the dataset
iris.head()
```

Out[34]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [43]:
```
# d.    Use DataFrame.drop() to drop the id column
iris2 = iris.drop('Id', axis=1)
```

In [44]:
```
# e.    Subset dataframe to create a new data frame that includes only the measur
iris2 = iris2[iris2.Species == 'Iris-setosa']
```

In [45]:
```
# f.    Use DataFrame.describe() to get the summary statistics
iris2.describe()
```

Out[45]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.000000	50.000000	50.00000
mean	5.00600	3.418000	1.464000	0.24400
std	0.35249	0.381024	0.173511	0.10721
min	4.30000	2.300000	1.000000	0.10000
25%	4.80000	3.125000	1.400000	0.20000
50%	5.00000	3.400000	1.500000	0.20000
75%	5.20000	3.675000	1.575000	0.30000
max	5.80000	4.400000	1.900000	0.60000

In [47]:
```
# g.    Use DataFrame.groupby() to create grouped data frames by Species and comp
iris3 = iris.groupby('Species')
iris3.describe()
```

Out[47]:

		Id	PetalLengthCm	PetalWidthCm	SepalLengthCm	SepalWidth
Species						
Iris-setosa	**count**	50.00000	50.000000	50.000000	50.000000	50.000000
	mean	25.50000	1.464000	0.244000	5.006000	3.418000
	std	14.57738	0.173511	0.107210	0.352490	0.381024
	min	1.00000	1.000000	0.100000	4.300000	2.300000
	25%	13.25000	1.400000	0.200000	4.800000	3.125000
	50%	25.50000	1.500000	0.200000	5.000000	3.400000
	75%	37.75000	1.575000	0.300000	5.200000	3.675000
	max	50.00000	1.900000	0.600000	5.800000	4.400000
Iris-versicolor	**count**	50.00000	50.000000	50.000000	50.000000	50.000000
	mean	75.50000	4.260000	1.326000	5.936000	2.770000
	std	14.57738	0.469911	0.197753	0.516171	0.313798
	min	51.00000	3.000000	1.000000	4.900000	2.000000
	25%	63.25000	4.000000	1.200000	5.600000	2.525000
	50%	75.50000	4.350000	1.300000	5.900000	2.800000
	75%	87.75000	4.600000	1.500000	6.300000	3.000000
	max	100.00000	5.100000	1.800000	7.000000	3.400000
Iris-virginica	**count**	50.00000	50.000000	50.000000	50.000000	50.000000
	mean	125.50000	5.552000	2.026000	6.588000	2.974000
	std	14.57738	0.551895	0.274650	0.635880	0.322497
	min	101.00000	4.500000	1.400000	4.900000	2.200000
	25%	113.25000	5.100000	1.800000	6.225000	2.800000
	50%	125.50000	5.550000	2.000000	6.500000	3.000000
	75%	137.75000	5.875000	2.300000	6.900000	3.175000
	max	150.00000	6.900000	2.500000	7.900000	3.800000

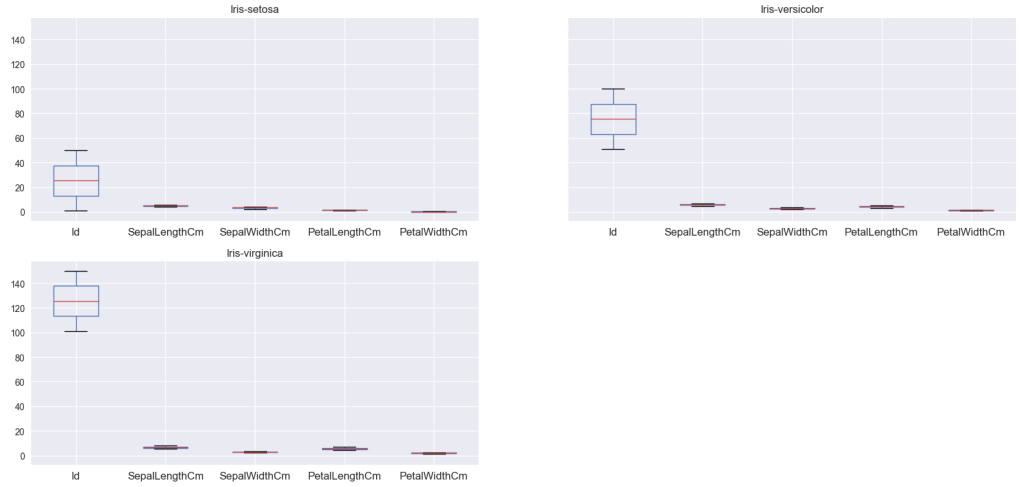
In [74]:
```
# h.    Use DataFrame.boxplot() to plot boxplots by Species

iris3.boxplot()
#import matplotlib.pyplot as plt
#plt.boxplot(iris2.SepalLengthCm)
```

Out[74]:
```
{'boxes': [<matplotlib.lines.Line2D at 0x29803d662e8>],
 'caps': [<matplotlib.lines.Line2D at 0x29803ee4fd0>,
  <matplotlib.lines.Line2D at 0x29803f2efd0>],
 'fliers': [<matplotlib.lines.Line2D at 0x2980283f400>],
 'means': [],
 'medians': [<matplotlib.lines.Line2D at 0x298042dd5c0>],
 'whiskers': [<matplotlib.lines.Line2D at 0x2980281e7b8>,
  <matplotlib.lines.Line2D at 0x29803f05d30>]}
```

In [76]:
```
from IPython.display import Image
Image("iris3Boxplot.png")
```

Out[76]:



In [73]:
```
# i.    Plot a scatter matrix plot using the seaborn library. Use the following t
import seaborn as sns
sns.pairplot(iris,hue='Species')
```

Out[73]:
```
<seaborn.axisgrid.PairGrid at 0x29800be2b38>
```

In [75]:
```
from IPython.display import Image
Image("scatterMatrix.png")
```

Out[75]: