Assignment No : 8

In [1]:
```
import pandas as pd
import matplotlib.pyplot as pt
import numpy as np
import requests as r
```

In [2]:
```
df=pd.read_csv('iris.data')
df
```

Out[2]:

	5.1	3.5	1.4	0.2	Iris-setosa
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa
...	...	...	...	...	...
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

149 rows × 5 columns

In [3]:
```
df.columns=["SepalLength","SepalWidth","PetalLength","PetalWidth","Species"]
df
```

Out[3]:

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa
...	...	...	...	...	...
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

149 rows × 5 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 149 entries, 0 to 148
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   SepalLength  149 non-null    float64
 1   SepalWidth   149 non-null    float64
 2   PetalLength  149 non-null    float64
 3   PetalWidth   149 non-null    float64
 4   Species      149 non-null    object 
dtypes: float64(4), object(1)
memory usage: 5.9+ KB
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
SepalLength    0
SepalWidth     0
PetalLength    0
PetalWidth     0
Species        0
dtype: int64
```

In [7]:
```
df.describe()
```

Out[7]:

	SepalLength	SepalWidth	PetalLength	PetalWidth
count	149.000000	149.000000	149.000000	149.000000
mean	5.848322	3.051007	3.774497	1.205369
std	0.828594	0.433499	1.759651	0.761292
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.400000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [8]:
```
df.drop_duplicates()
```

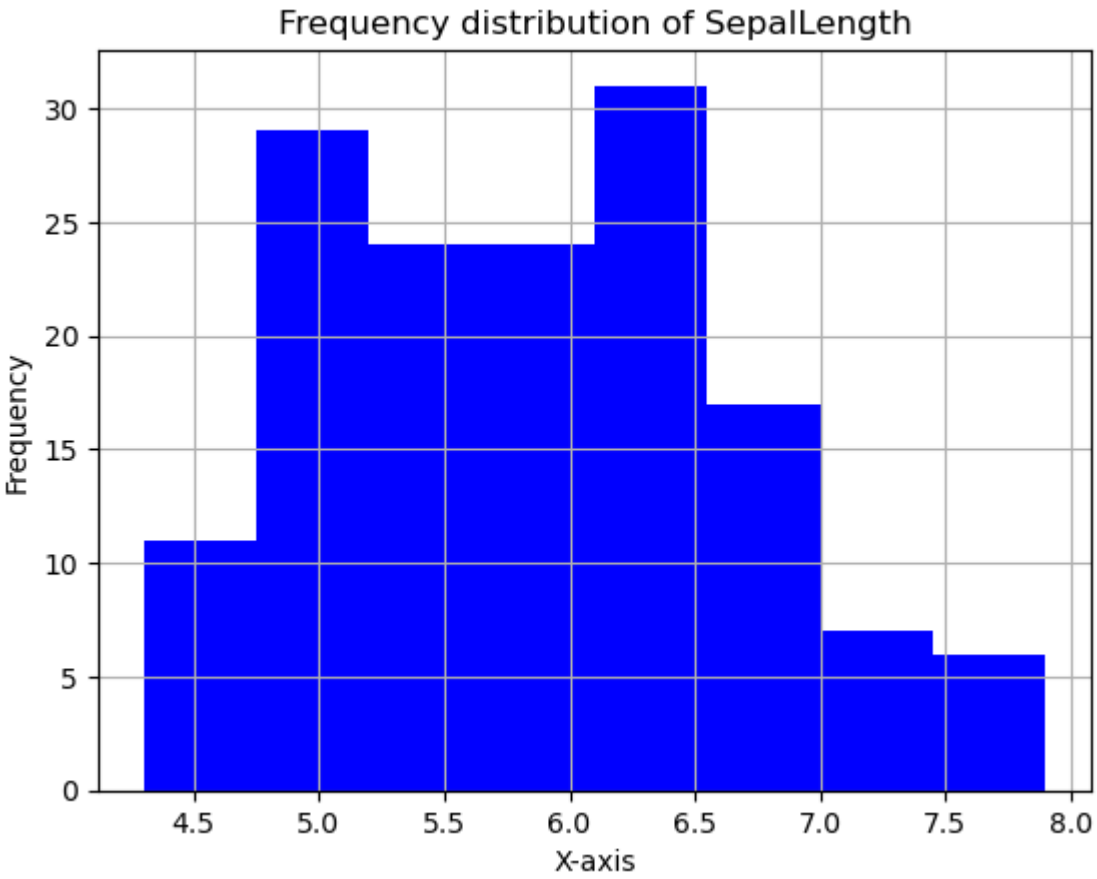
Out[8]:

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa
...	...	...	...	...	...
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

146 rows × 5 columns

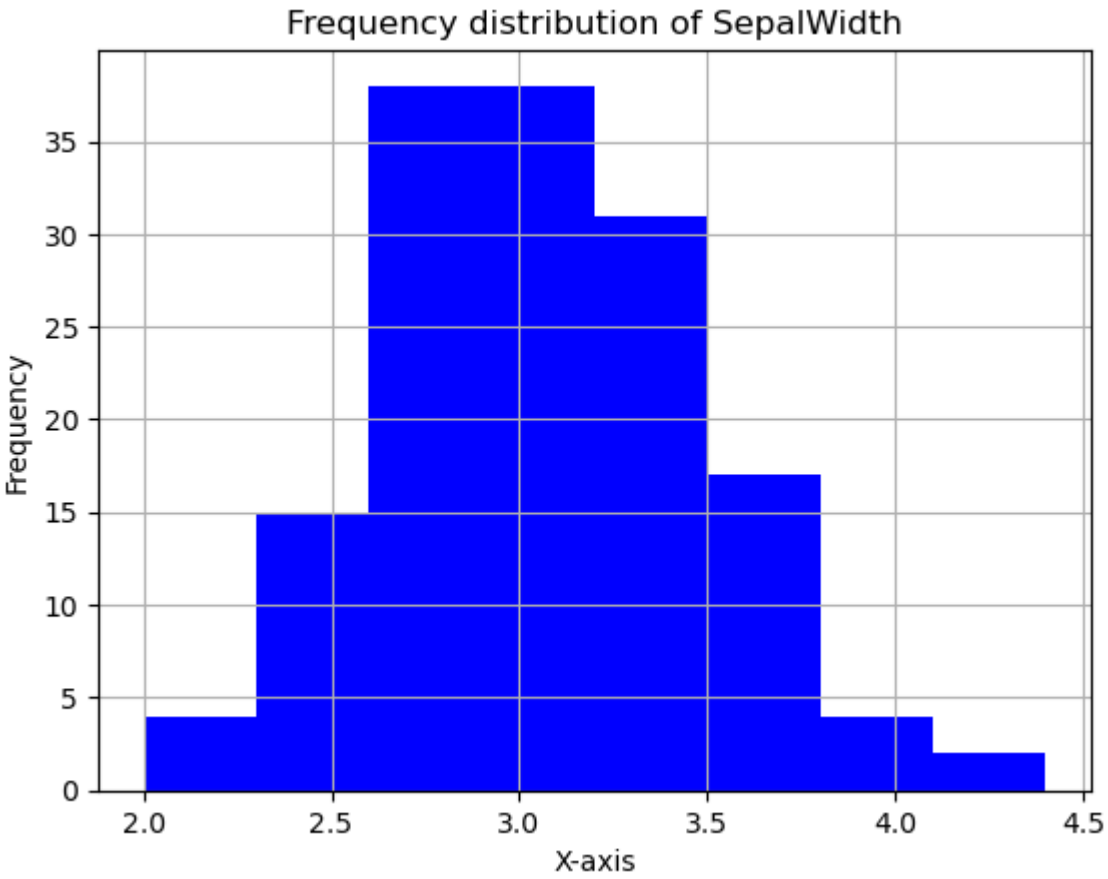
In [10]:
```
pt.title("Frequency distribution of SepalLength")
pt.xlabel("X-axis")
pt.ylabel("Frequency")
df["SepalLength"].hist(color="blue",bins=8)
```

Out[10]:
```
<AxesSubplot:title={'center':'Frequency distribution of SepalLength'}, xlabel
='X-axis', ylabel='Frequency'>
```



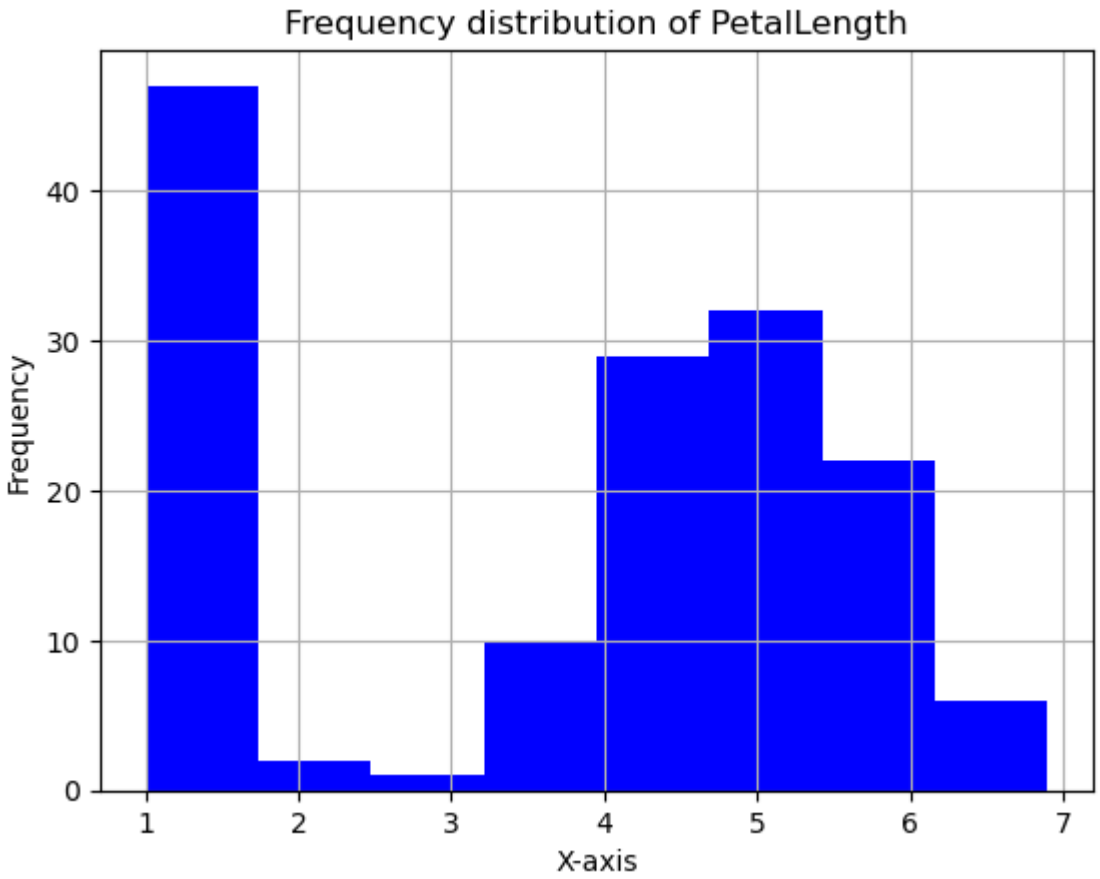
In [11]:
```
pt.title("Frequency distribution of SepalWidth")
pt.xlabel("X-axis")
pt.ylabel("Frequency")
df["SepalWidth"].hist(color="blue",bins=8)
```

Out[11]:
```
<AxesSubplot:title={'center':'Frequency distribution of SepalWidth'}, xlabel
='X-axis', ylabel='Frequency'>
```



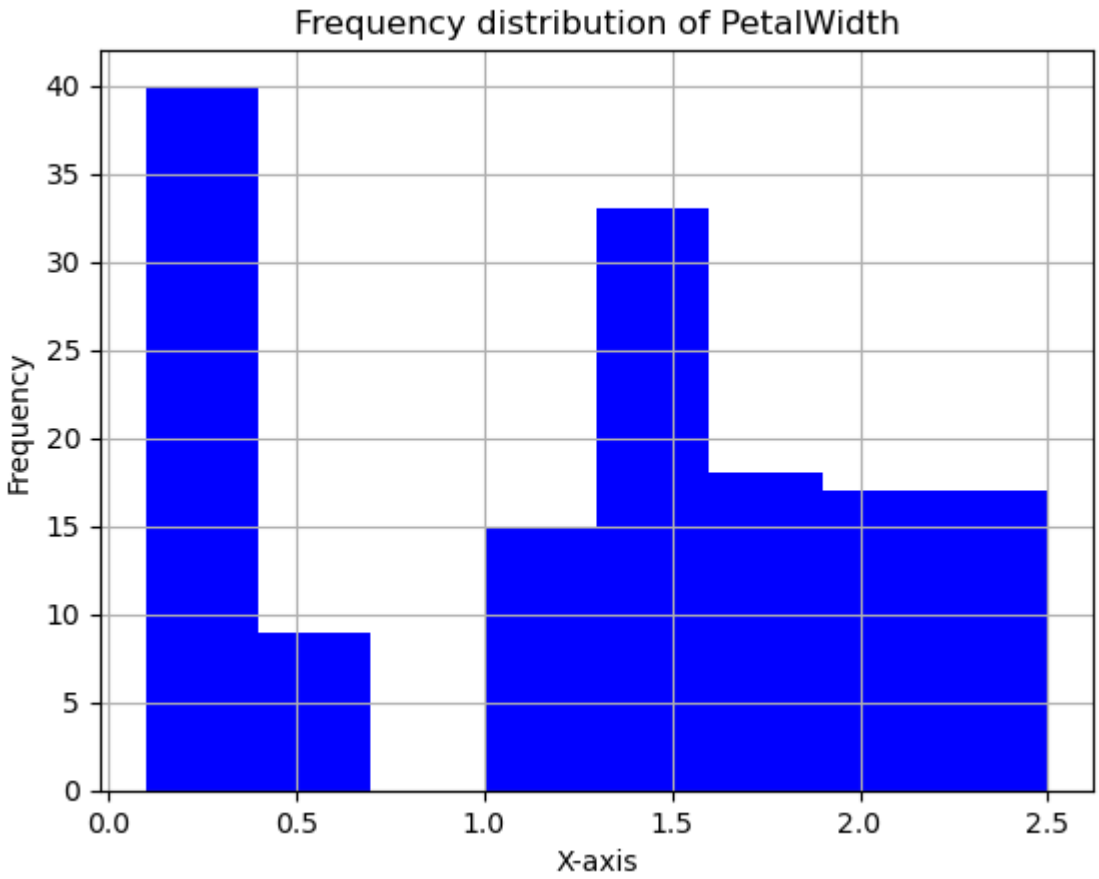
In [12]:
```
pt.title("Frequency distribution of PetalLength")
pt.xlabel("X-axis")
pt.ylabel("Frequency")
df["PetalLength"].hist(color="blue",bins=8)
```

Out[12]:
```
<AxesSubplot:title={'center':'Frequency distribution of PetalLength'}, xlabel
='X-axis', ylabel='Frequency'>
```



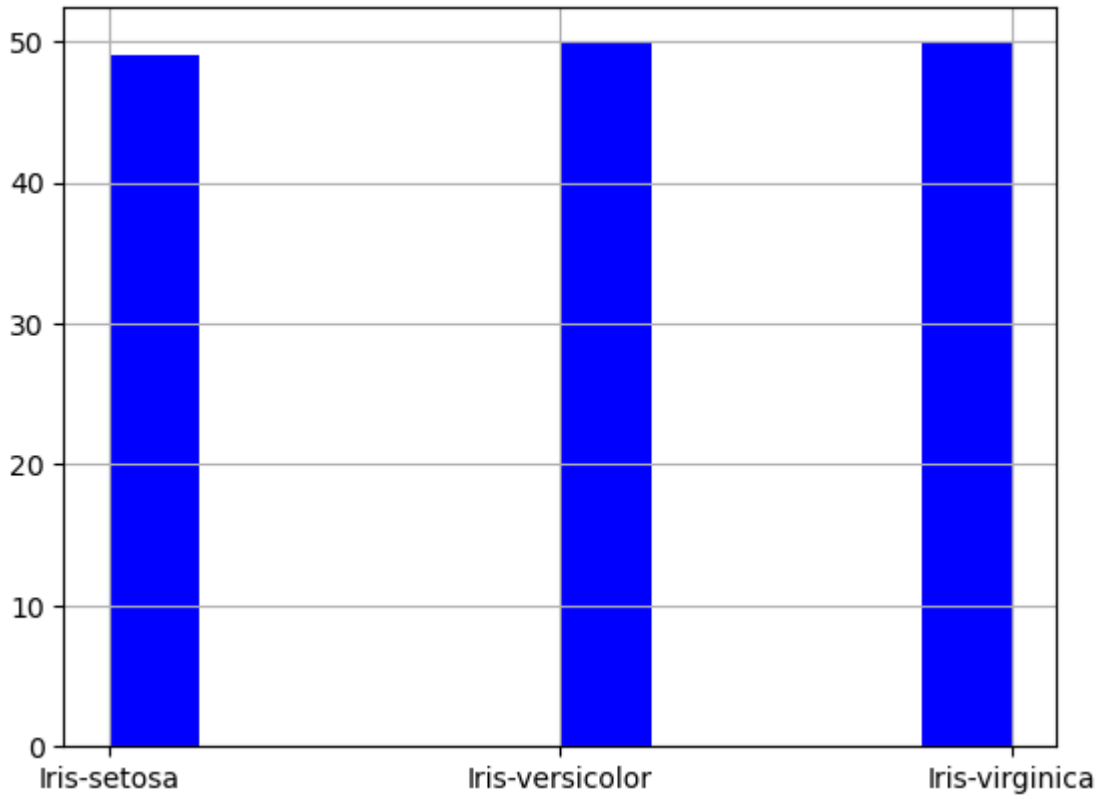
In [14]:
```
pt.title("Frequency distribution of PetalWidth")
pt.xlabel("X-axis")
pt.ylabel("Frequency")
df["PetalWidth"].hist(color="blue",bins=8)
```

Out[14]:
```
<AxesSubplot:title={'center':'Frequency distribution of PetalWidth'}, xlabel
='X-axis', ylabel='Frequency'>
```



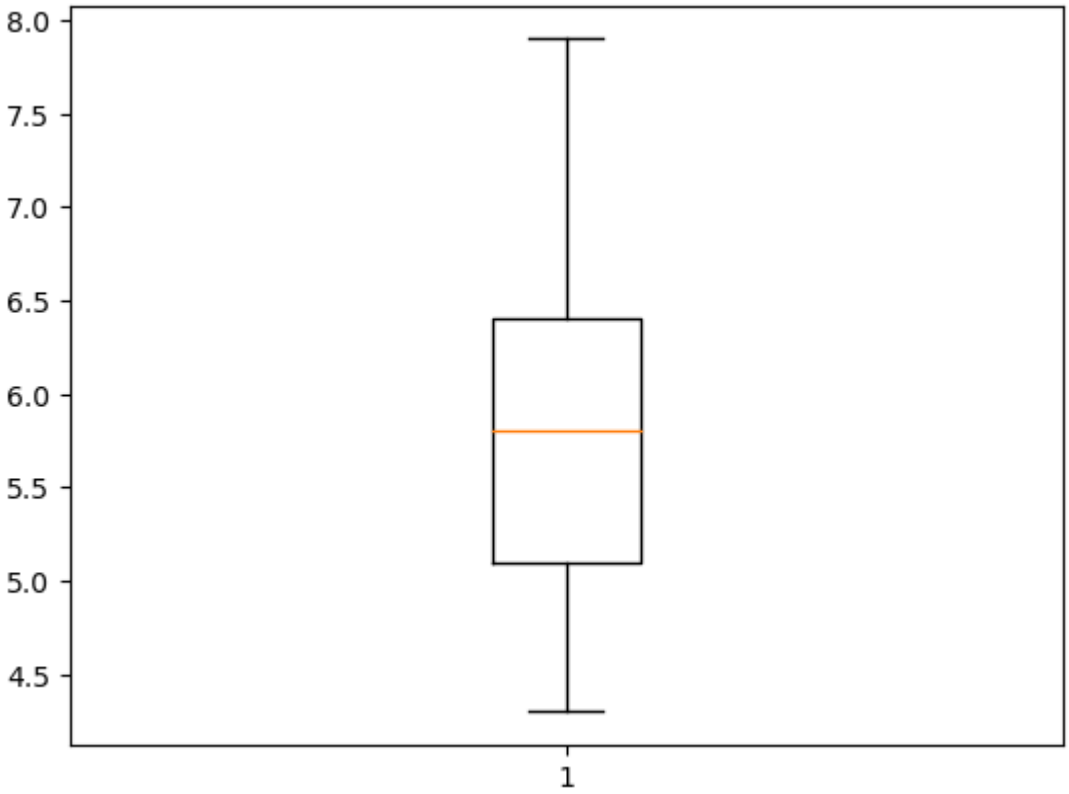
In [15]:
```
df["Species"].hist(color="blue")
```

Out[15]: <AxesSubplot:>



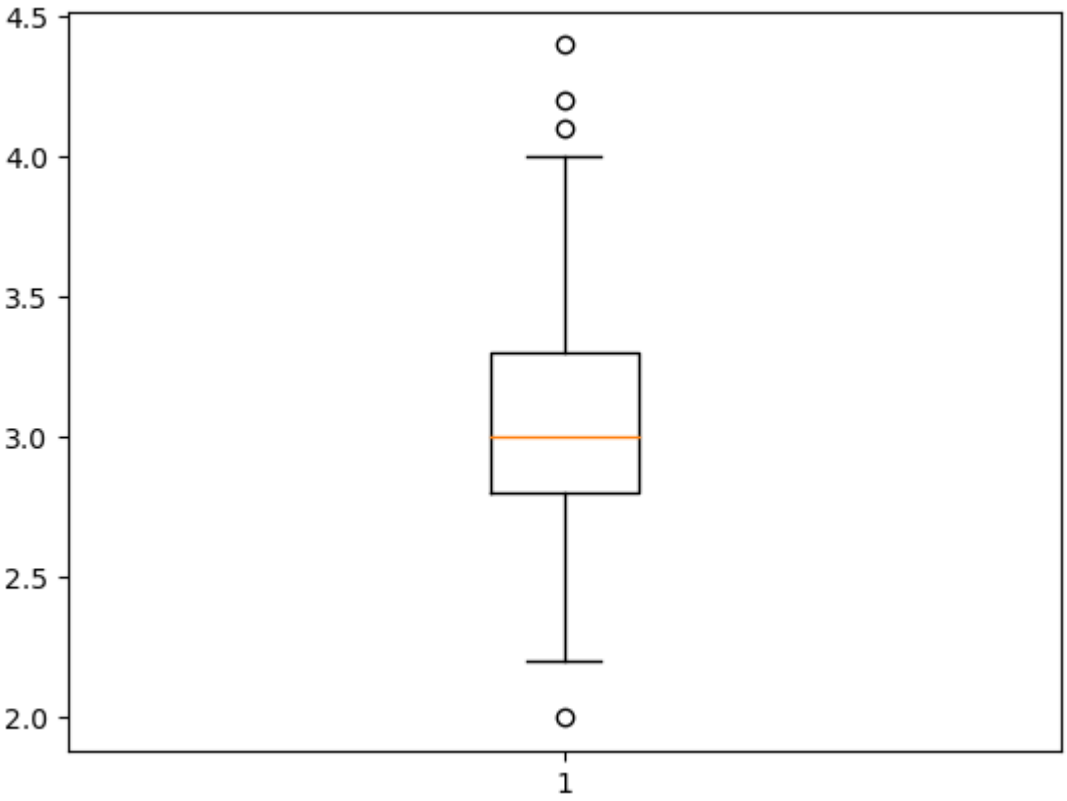
In [16]:
```
pt.boxplot(df["SepalLength"])
```

Out[16]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x250243f2cd0>,
  <matplotlib.lines.Line2D at 0x250243f2fa0>],
 'caps': [<matplotlib.lines.Line2D at 0x250244052b0>,
  <matplotlib.lines.Line2D at 0x25024405580>],
 'boxes': [<matplotlib.lines.Line2D at 0x250243f2a60>],
 'medians': [<matplotlib.lines.Line2D at 0x25024405850>],
 'fliers': [<matplotlib.lines.Line2D at 0x25024405b20>],
 'means': []}
```



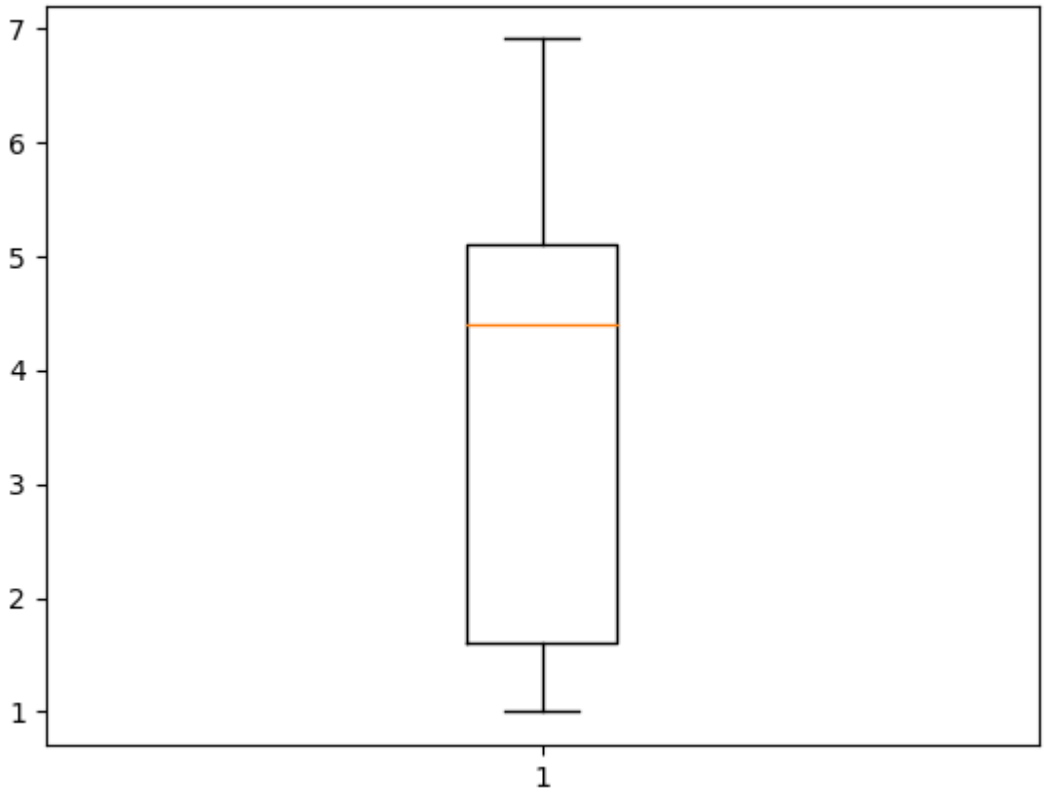
In [17]:
```
pt.boxplot(df["SepalWidth"])
```

Out[17]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x250244a69d0>,
  <matplotlib.lines.Line2D at 0x250244a6ca0>],
 'caps': [<matplotlib.lines.Line2D at 0x250244a6f70>,
  <matplotlib.lines.Line2D at 0x250244b5280>],
 'boxes': [<matplotlib.lines.Line2D at 0x250244a66d0>],
 'medians': [<matplotlib.lines.Line2D at 0x250244b5550>],
 'fliers': [<matplotlib.lines.Line2D at 0x250244b5820>],
 'means': []}
```



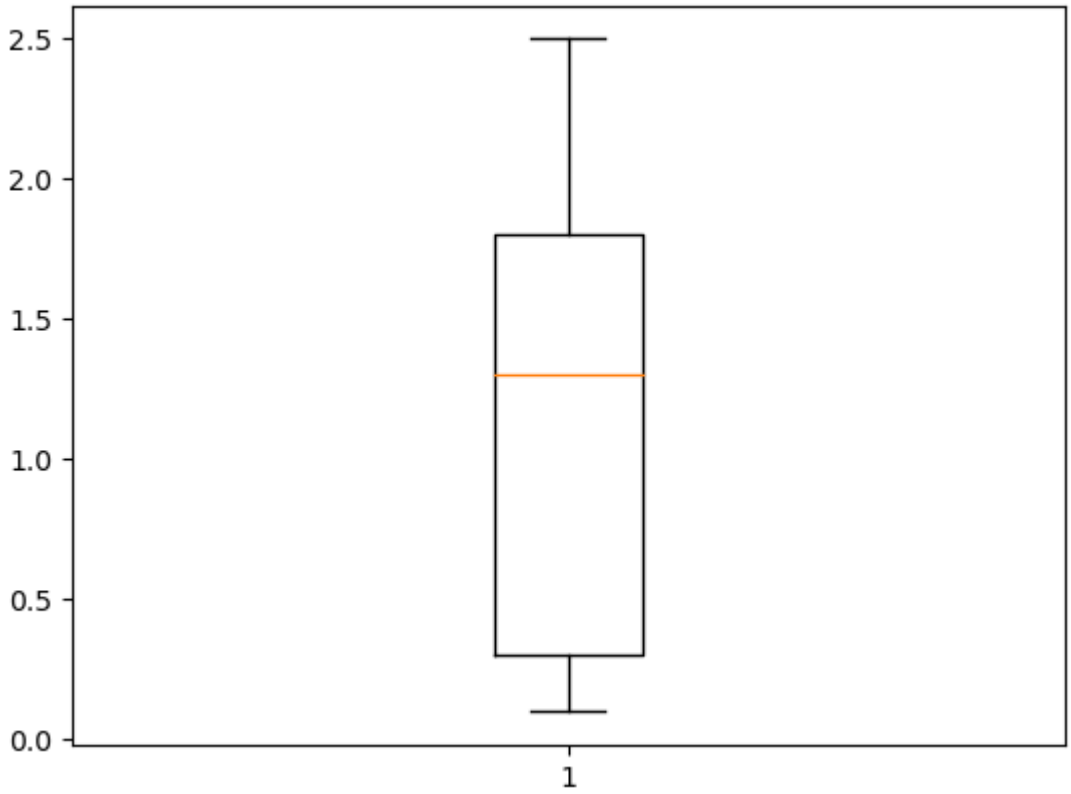
In [19]:
```
pt.boxplot(df["PetalLength"])
```

Out[19]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x2502468bac0>,
  <matplotlib.lines.Line2D at 0x2502468bd90>],
 'caps': [<matplotlib.lines.Line2D at 0x250246990a0>,
  <matplotlib.lines.Line2D at 0x25024699370>],
 'boxes': [<matplotlib.lines.Line2D at 0x2502468b7c0>],
 'medians': [<matplotlib.lines.Line2D at 0x25024699640>],
 'fliers': [<matplotlib.lines.Line2D at 0x25024699910>],
 'means': []}
```



In [20]:
```
pt.boxplot(df["PetalWidth"])
```

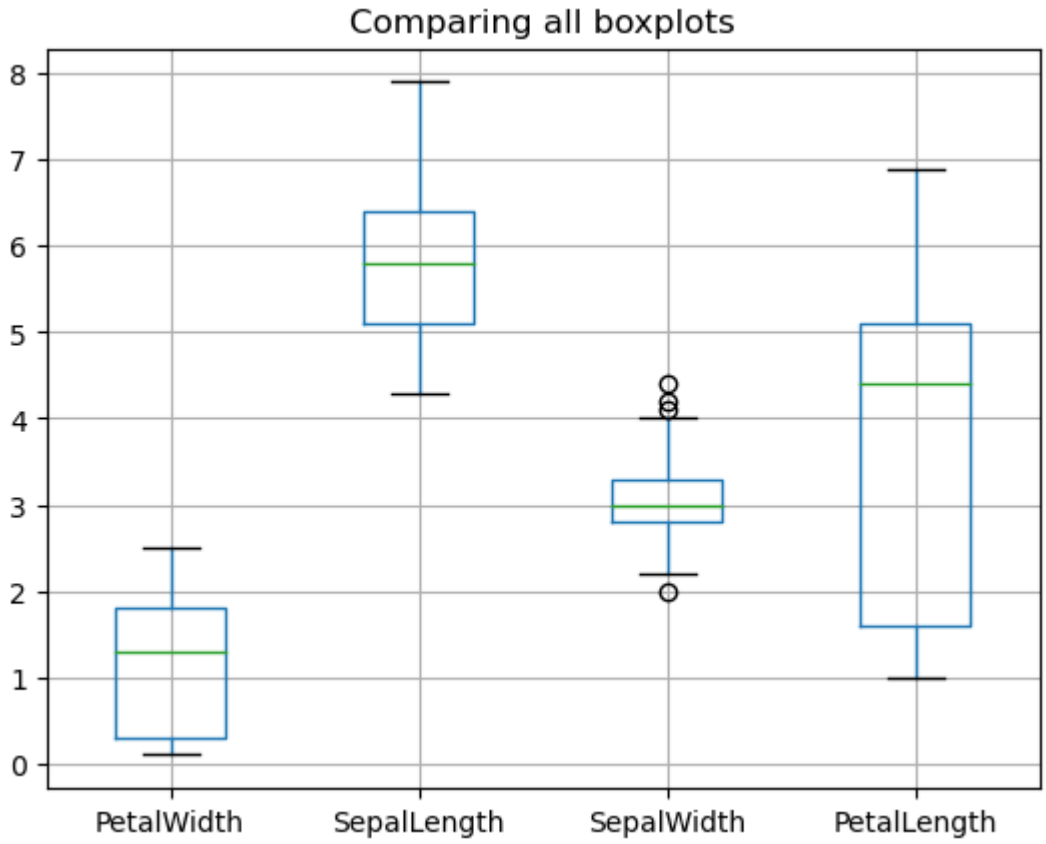
Out[20]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x2502483b100>,
  <matplotlib.lines.Line2D at 0x2502483b3d0>],
 'caps': [<matplotlib.lines.Line2D at 0x2502483b6a0>,
  <matplotlib.lines.Line2D at 0x2502483b970>],
 'boxes': [<matplotlib.lines.Line2D at 0x2502482cdf0>],
 'medians': [<matplotlib.lines.Line2D at 0x2502483bc40>],
 'fliers': [<matplotlib.lines.Line2D at 0x2502483bf10>],
 'means': []}
```



In [24]:
```
df[{"SepalLength","SepalWidth","PetalLength","PetalWidth"}].boxplot()
pt.title("Comparing all boxplots")
```

```
C:\Users\student\AppData\Local\Temp\ipykernel_4848\2131206443.py:1: FutureWar
ning: Passing a set as an indexer is deprecated and will raise in a future ve
rsion. Use a list instead.
  df[{"SepalLength","SepalWidth","PetalLength","PetalWidth"}].boxplot()
```

Out[24]:
```
Text(0.5, 1.0, 'Comparing all boxplots')
```



In [ ]: