STATISTICS TESTS

1. shapiro wilk test

Tests weather a data has a guassian or normal distribution.

Assumptions

1. observation in each sample are independent and identically distributed(iid).
2. interpretation
   - the sample has a normal distribution
   - the sample has not a normal distribution

important libarary for tests

pip install scipy

In [1]:

```
# Example of the shapiro wilk test
from scipy.stats import shapiro
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
stat, p =shapiro(data1)
print ('stat=',stat)
print ('p=',p)
if p > 0.05:
    print('the data is normal')
else:
    print('the data is not normal')
```

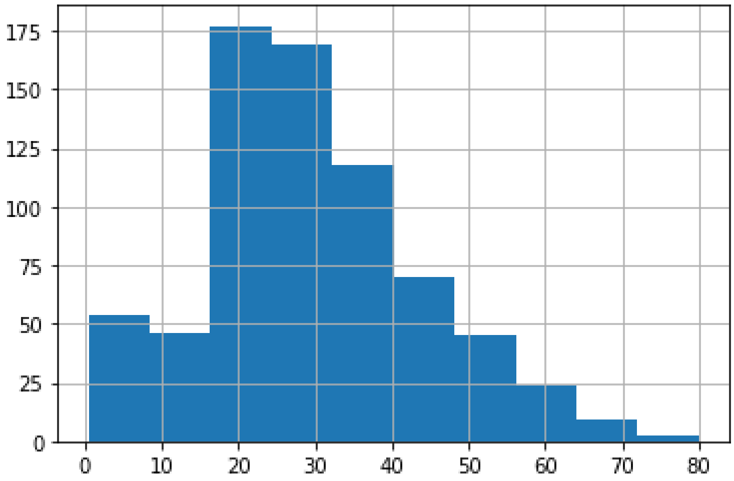
```
stat= 0.8951009511947632
p= 0.19340917468070984
the data is normal
```

In [2]:

```
#look in histogram to see normality
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
ks= sns.load_dataset('titanic')
ks['age'].hist()
```

Out[2]:

```
<AxesSubplot:>
```

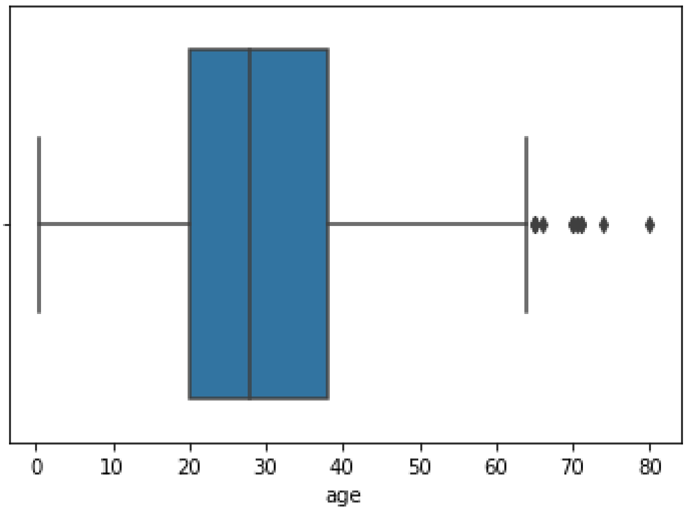


In [3]:

```
#box plot for normality view
sns.boxplot(ks['age'])
```

```
C:\Users\Azka\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid p
ositional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[3]: <AxesSubplot:xlabel='age'>

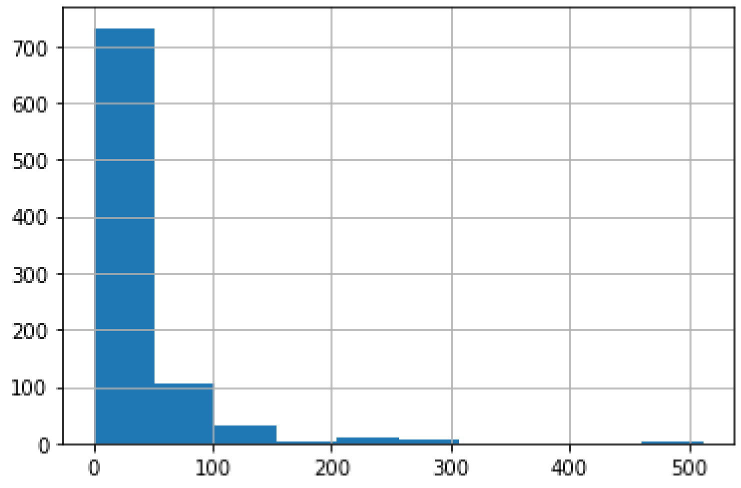


In [4]:

```
# Example of the shapiro wilk test on titanic fare
from scipy.stats import shapiro
data1 = ks['fare']
stat, p =shapiro(data1)
print ('stat=',stat)
print ('p=',p)
if p > 0.05:
    print('the data is normal')
else:
    print('the data is not normal')
ks['fare'].hist()
```

```
stat= 0.5218914747238159
p= 1.0789998175301091e-43
the data is not normal
```

Out[4]: <AxesSubplot:>



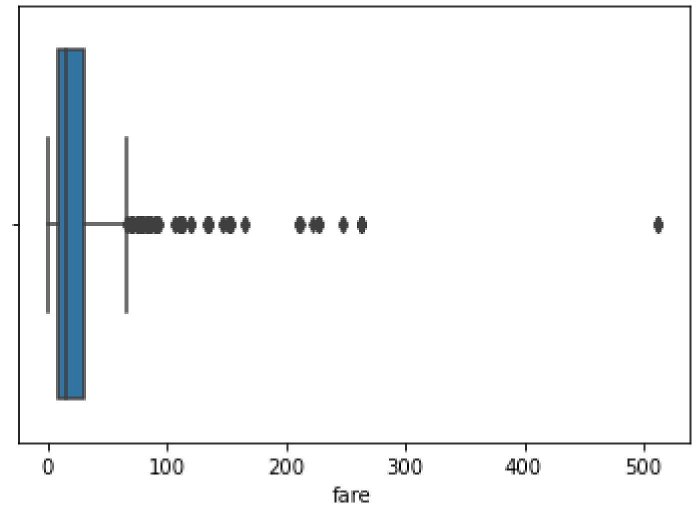
In [5]:

```
# box plot is also not normal
sns.boxplot(ks['fare'])
```

```
C:\Users\Azka\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid p
ositional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[5]:

```
<AxesSubplot:xlabel='fare'>
```



In [6]:

```
# normality test
from scipy.stats import shapiro
stat, p = shapiro(ks['age'])
print ('stat=',stat)
print ('p=',p)
if p > 0.05:
    print('the data is normal')
else:
    print('the data is not normal')
```

```
stat= nan
p= 1.0
the data is normal
```

In [7]:

```
# normality test for fare
from scipy.stats import shapiro
stat, p = shapiro(ks['fare'])
print ('stat=',stat)
```

```
print ('p=',p)
if p > 0.05:
    print('the data is normal')
else:
    print('the data is not normal')
```

```
stat= 0.5218914747238159
p= 1.0789998175301091e-43
the data is not normal
```

2. Correlation test

1. pearsons correlation coefficient
2. test weather two samples have a linear relationship
3. **Assumptions**

4. observation in each sample are independent and identically distributed(iid).

5. observation in each sampleis normally distributed
6. observation in each samplehave the same variance
7. interpretation
   - H0: the two samples are independent
   - H2: there is a dependency between two samples

In [8]:

```
# Example of the Pearson's Correlation test
from scipy.stats import pearsonr
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [0.353, 3.517, 0.125, -7.545, -0.555, -1.536, 3.350, -1.578, -3.537, -1.579]
ks.dropna()
stat, p = pearsonr(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
  print('Probably independent')
else:
    print('Probably dependent')
```

```
stat=0.688, p=0.028
Probably dependent
```

2- Spearmans Rank correlation

Test weather two samples have a monotonic relationship

Assumptions

1. Observation in each sample are independent and identically distributed (iid)
2. Observation ineach sample can be ranked
3. Interpretation
   - H0: the two samples are independent
   - H2: there is a dependency between two samples

In [9]:

```
# Example of the spearman's Correlation test
from scipy.stats import spearmanr
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
```

```
data2 = [0.353, 3.517, 0.125, -7.545, -0.555, -1.536, 3.350, -1.578, -3.537, -1.579]
ks.dropna()
stat, p = spearmanr(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
  print('Probably independent')
else:
    print('Probably dependent')
```

```
stat=0.855, p=0.002
Probably dependent
```

3. Chi-squared test

Test weather teo catagorical variables are related or independent.

Assumptions

1. Observation used in calculation of the contingency table are independent.
2. 25 or more examples in each cell of the contingency table.
3. interpretation
   - H0: the two samples are independent
   - H2: there is a dependency between two samples

In [10]:

```
#example of the Chi-squared test
from scipy.stats import chi2_contingency
table= [[10,20,30],[6,9,17]]
stat,p,dof, expected= chi2_contingency(table)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
  print('Probably independent')
else:
    print('Probably dependent')
```

```
stat=0.272, p=0.873
Probably independent
```

3- Paramatric statical hypothesis test

1- student t-test

Tests weather the means of two independent samples are significantly different. **Assumptions**

1. observation in each sample are independent and identically distributed(iid).
2. observation in each samples are normally distributed
3. observation in each samples have the same variance
4. interpretation
   - H0: the means of the samples are equal
   - H1: the means of the samples are not equal

In [11]:

```
# Example of the Student's t-test
from scipy.stats import ttest_ind
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
```

```
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169
stat, p = ttest_ind(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
  print('Probably the same distribution')
else:
    print('Probably different distribution')
```

```
stat=-0.326, p=0.748
Probably the same distribution
```

2- Paired student t-test

Tests weather the means of two paired samples are significantly different. **Assumptions**

1. observation in each sample are independent and identically distributed(iid).
2. observation in each samples are normally distributed
3. observation in each samples have the same variance
4. observation across each sample are paired
5. interpretation
   - H0: the means of the samples are equal
   - H1: the means of the samples are not equal

In [12]:

```
# Example of the Paired Student's t-test
from scipy.stats import ttest_rel
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169
stat, p = ttest_rel(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
  print('Probably the same distribution')
=else:
    print('Probably different distributions')
```

```
  File "C:\Users\Azka\AppData\Local\Temp/ipykernel_4264/1190923719.py", line 9
    =else:
    ^
SyntaxError: invalid syntax
```

4- Analysis of variance test (ANOVA)

Tests weather the means of two or mor independent samples are significantly different.
Assumptions

1. observation in each sample are independent and identically distributed(iid).
2. observation in each samples are normally distributed
3. observation in each samples have the same variance
4. interpretation
   - H0: the means of the samples are equal
   - H1: one or more of the means of the samples are not equal

In [ ]:

```
#Example of Analysis of variance test (ANOVA)
from scipy.stats import f_oneway
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
```

```
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169
data3 = [-0.28, 0.696, 0.928, -1.148, -0.213, 0.229, 0.137, 0.269, -0.870, -1.204]
stat, p = f_oneway(data1, data2, data3)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
  print('Probably the same distribution')
else:
    print('Probably different distributions')
```

Post Hoc Tests

The most common post hoc tests are:

1. Tukey’s Test (Tukey's HSD is the most preferred post-hoc test)
2. Bonferroni Procedure